第三届

“先进计算与内生安全”

学术会议

THE 3TH CONFERENCE ON ADVANCED COMPUTING AND ENDOGENOUS SAFETY & SECURITY

论文集

江苏·南京

2020年10月

目　录

一种新型基于硬件的软件定义流量发生器

董春雷[1]，沈剑良[1]，谭力波[2]，王盼[2]，陈艇[1]

[1]国家数字交换系统工程技术研究中心 郑州 450002；

[2]天津市滨海新区信息技术创新中心；天津 300450

摘　要：流量发生器在各种网络环境中应用广泛，是协议控制器芯片、路由器、交换机以及网络本身测试与分析过程中的重要工具。针对现有流量发生器种类较少、配置灵活度不高、无法实现数据流量大小精确可控以及不便实现多种数据流量混合的问题，本文提出一种新的基于硬件的软件定义流量发生器实现架构，在可灵活配置输出数据包的协议类型、数量、长度、间隔等的基础上，实现输出数据流量大小的精确可控及多种数据流量的灵活混合。

关键词：流量发生器、基于硬件、软件定义、多流混合、流量大小精确可控

A New Implementation of Software Defined Traffic Generator Based on Hardware

DONG Chunlei[1], SHEN Jianliang[1], TAN Libo, WANG Pan, CHEN Ting[1]

1.National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450000, ChinaInformation Technology Innovation Center of Tianjin BINHai New Area

Abstract: Traffic generator is widely used in various network environments. It is an important tool in the testing and analysis of protocol controller chips, routers, switches, and the network itself. To address the problem that not much type , less flexible in configuration and multi-flow condition, not available of controlling the flow imprecisely of exiting traffic generator , in this article we describe in detail a new software defined traffic generator based on hardware , it realizes the configuration of packet protocol type, packet quantity, packet length, packet interval through software, Such a structure has the character of Realize the flexible mixing of multiple flows and precise control of the flow size.

Key words: traffic generator; hardware based; software defined; mixing of multiple flows; precise control of the flow size

1　引言

流量发生器是一种可以产生指定参数（数据流协议类型、数据流大小、数据流包长度等）数据流的工具，主要功能是在仿真验证、FPGA（Field-Programmable Gate Array）原型验证以及系统实测过程中模拟真实的流量激励源，达到测试待测应用或待测设备的目的。流量发生器在路由器、交换机、协议控制器等产品/设备的验证、测试以及新协议的开发中，起着至关重要的作用。

流量发生器能够模拟真实网络产生的数据，在待测目标实体暂不具备的开发阶段以及待测目标实体具备的实测阶段，为待测目标的功能验证以及性能评估提供一种方便、有效且必要的手段。RFC2544 包含的包延迟、背靠背帧、丢包率、吞吐量等指标测试[1]，借助于流量发生器，都可以实现。另外，借助流量发生器可以模拟网络中的恶意攻击行为从而对安全防御系统进行测试，还

基金项目：国家核高基重大专项基金资助项目（No.2017ZX01030301）；国家自然科学基金资助项目（No.61572520）；

Foundation item: National Core Electronic Devices, High-end Generic Chips and Basic Software Major Projects (No.2017ZX01030301), The National Natural Science Foundation of China (No.61572520)

可以产生特殊流量用来进行一些专有系统指标的测试。流量发生器的使用可以简化验证、测试环境，避免将待测对象放入实际运行的环境中去，可有效降低验证、测试的成本与风险，大大缩短验证、测试时间。

目前，国内外的流量发生器在实现方式上基于软件的居多[2]，由于软件指令以串行方式执行，且CPU存在中断和任务调度等不确定性，软件发包难以实现全线速，对短帧的处理尤其会受到严重限制。即使采用实时系统，受限于CPU的最高工作频率，其发送速度也将受限。Ixia等公司推出有针对以太网流量发生器的硬件解决方案[3]，可以提供更专业更全面的用于仿真、测试的功能，但非常昂贵，且仅支持以太网协议，也有国外公司推出FC（Fiber Channel）协议测试仪，价格昂贵且输出数据包的优先级、间隔等控制不灵活。

针对以上问题，国内外学者不断展开研究，一些基于硬件的流量发生器相继面世。经研究总结，现有的流量发生器存在以下两点不足：

1. 不够灵活。大多数流量发生器只能实现一种特定协议类型数据流的输出，配置的灵活度有限，无法用于协议自定义或者协议类型多变系统的验证、测试，无法支持高层自定义协议[4]；

2. 不便实现多条数据流量的灵活混合，数据流量大小无法精确控制。有个别流量发生器通过采用依据寄存器配置对包内容进行填充的机制在一定程度上克服了灵活性局限，但还不能方便实现多条数据流量的灵活混合以及无法实现数据流量大小的精确可控。

针对现有流量发生器的上述不足，本文提出了一种新型基于硬件的软件定义流量发生器。本流量发生器通过采用已有的由硬件根据寄存器配置对包内容进行填充的机制，打破了流量发生器的灵活性局限，在此基础上通过架构层面的创新，实现了数据流量的方便、灵活混合以及数据流量大小的精确可控。表1列出了本文所述流量发生器与现有流量发生器的功能性能对比情况。

表1　与现有流量发生器的功能性能对比

文献	实现方式	发包速度	支持协议类型	流量大小是否精确可控	多流混合方便程度
1	基于软件	慢	以太网	否	低
2	基于硬件	快	以太网	否	低
3	基于硬件	快	以太网	否	低
4	基于硬件	快	软件定义	否	低
本文	基于硬件	快	软件定义	是	高

2　流量发生器数据流量的精确控制

2.1　数据流量的概念

数据流量是指流量发生器单位时间内输出的数据量，单位为比特/秒（bits/s）。

假如某流量发生器支持的最大数据流量（即满流量）为100Gbits/s（数据包之间无间隔，即背靠背输出），由于测试需要，要求产生满流量的10%的数据流量，则输出的数据流量应为10Gbits/s。

2.2　如何控制数据流量

我们假设包间隔为I（Interval），包长为L（Length），输出数据流量为满流量的P%（Percent，取值范围为1-100），则三者之间满足如下关系：

L/（L+I）= P/100 （1）

由公式（1）可推出：

I=（100/P － 1）* L （2）

由公式（2）可知，产生P%的流量时，包间隔I与包长L之间要满足等式（2）的关系。

通过观察等式（2），我们发现，当P在1-100之间取值时，绝大多数情况下I与L之间并不是精确的整数倍关系，比如，P=3时，I=（100/3 - 1）* L，I约等于32*L；P=26时，I=（100/26 - 1）* L，I约等于3*L；P=28时，I=（100/28 - 1）* L，I约等于3*L。考虑到包间隔I只能取整数，我们产生的流量大多数情况下不精确等于我们的设定值P%，会有不小的误差。

另外，等式（2）中既包含除法又包含乘法，导致等式（2）在硬件逻辑中实现时面积、功耗较大。

2.3　如何实现数据流量的精确控制

经过2.2小节的分析我们知道，用硬件逻辑直接实现等式（2）无法达到精确控制流量的目的。这使得我们想到可能需要从架构层面进行考虑，来实现数据流量的精确控制。

仔细研究等式（2），总结发现：

P=1时：I=99L；

P=2时：I=49L；

P=5时：I=19L；

P=10时：I=9L；

P=20时：I=4L；

P=50时：I= L；

P=100时：I=0；

即，当且仅当P=1、2、5、10、20、50、100时，I与L之间为精确的倍数关系，此时的流量大小精确等于目标流量大小。而且以上倍数关系很容在硬件逻辑中实现，比如P=1时，I=99L=（64+32+2+1）L，其中64L、32 L、2 L都可以通过左移L方便的实现。

1、2、5、10、20、50、100，这几个数字跟我们的人民币面值一样，用这几个数值可以构成1-100之间的任意值。受此启发，我们只需六条数据流，其中每条数据流的流量都可以配置成1%、2%、5%、10%、20%、50%、100%这几种流量中的任一种，通过这六条数据流量的组合汇聚，我们可以精确实现1-100%之间的任意大小数据流量。比如，我们要实现88%的数据流量，这时我们可以配置6条数据流的流量大小分别为1%、2%、5%、10%、20%、50%，然后让它们汇聚输出，这样我们最终得到的数据流就是精确的88%。同理，要实现100%的数据流量时，只需启用一条数据流，并配置其流量为100%即可，当然，也可以同时启用多条数据流，各流的总和等于100%，比如同时启用两条数据流，各数据流的流量大小都等于50%，再比如，同时启用5条数据流，各数据流的流量大小都等于20%。

上述新型流量发生器的原理图如图1所示。

图1中，Pkt_gen_0~5为基于硬件的软件定义包生成模块（前期已由我申请过专利并发表过相关论文）。各包生成模块通过配置接口既可以实现对包长、包间隔、包数量的软件定义，还可以实现对生成数据包的协议类型以及包中关键字段（用户比较关心的字段）的软件定义，各包生成模块的流量可软件定义为满流量的1%、2%、5%、10%、20%、50%、100%中的任一值。包生成模块的起停也可通过软件进行控制。

各包生成模块的控制相互独立，各条数据流的特性可定义的互不相同，在需要多种数据流混合输出的场景下，对各包生成模块分别按需进行配置，即可得到多种数据流混合的数据流输出，相比传统获得多种数据流混合的实现方式，本流量发生器实现多流混合输出的方式更加方便快捷。图2所示为实现不同数据流混合的传统方式，该方式需要使用流量发生器的多个数据流端口，且需要一个交换设备用于对多条数据流进行汇聚转发，除了需要使用的资源（流量发生器的多个数据流端口、多根光纤、交换设备）较多外，操作也比较复杂（需要控制多条数据流的路由查表字段并配置交换设备路由转发表）。

每一个软件定义包生成模块后面都对应的有一个数据缓存Buffer，在后级调度模块调度输出其它数据流的流量时，该buffer用于缓存本数据流的数据。

RR（Round-Robin）轮询调度模块用于对各条数据流的数据进行轮询调度，对后级反压信号进行处理，实现多流数据的无缝衔接，在数据缓存与后级模块之间起到桥梁枢纽作用。

2.4 总体工作流程

首先，根据拟生成的目标流量大小，确定六条数据流各自的流量大小（0%、1%、2%、5%、10%、20%、50%、100%中的任一值），即确定数据流的组合情况；然后，通过软件配置各数据流的包生成模块，完成流量大小、流量协议类型、包长、包间隔、包数量、包内容等的设定；最后，通过主机发送开始命令到各数据流包生成模块，各包生成模块即按照配置开始产生数据流，各数据流经汇聚后得到目标流量。

3 流量发生器的硬件实现

本流量发生器采用硬件描述语言Verilog开发实现。如图1所示，本流量发生器按功能可划分为包生成模块、数据包缓存Buffer模块及RR轮询调度模块。

3.1 包生成模块

如图4所示，包生成模块由配置模块、发包控制模块、payload产生模块、包头产生模块、整包封装模块共5大模块组成。

3.1.1 配置模块

配置模块用来实现对包生成模块控制寄存器的配置。如图5所示，来自配置软件的命令经CPU处理后，送至I2C模块， I2C模块将配置信息转换为AXI_Lite格式并输入到控制寄存器模块完成对包生成模块控制寄存器的配置，配置后的控制寄

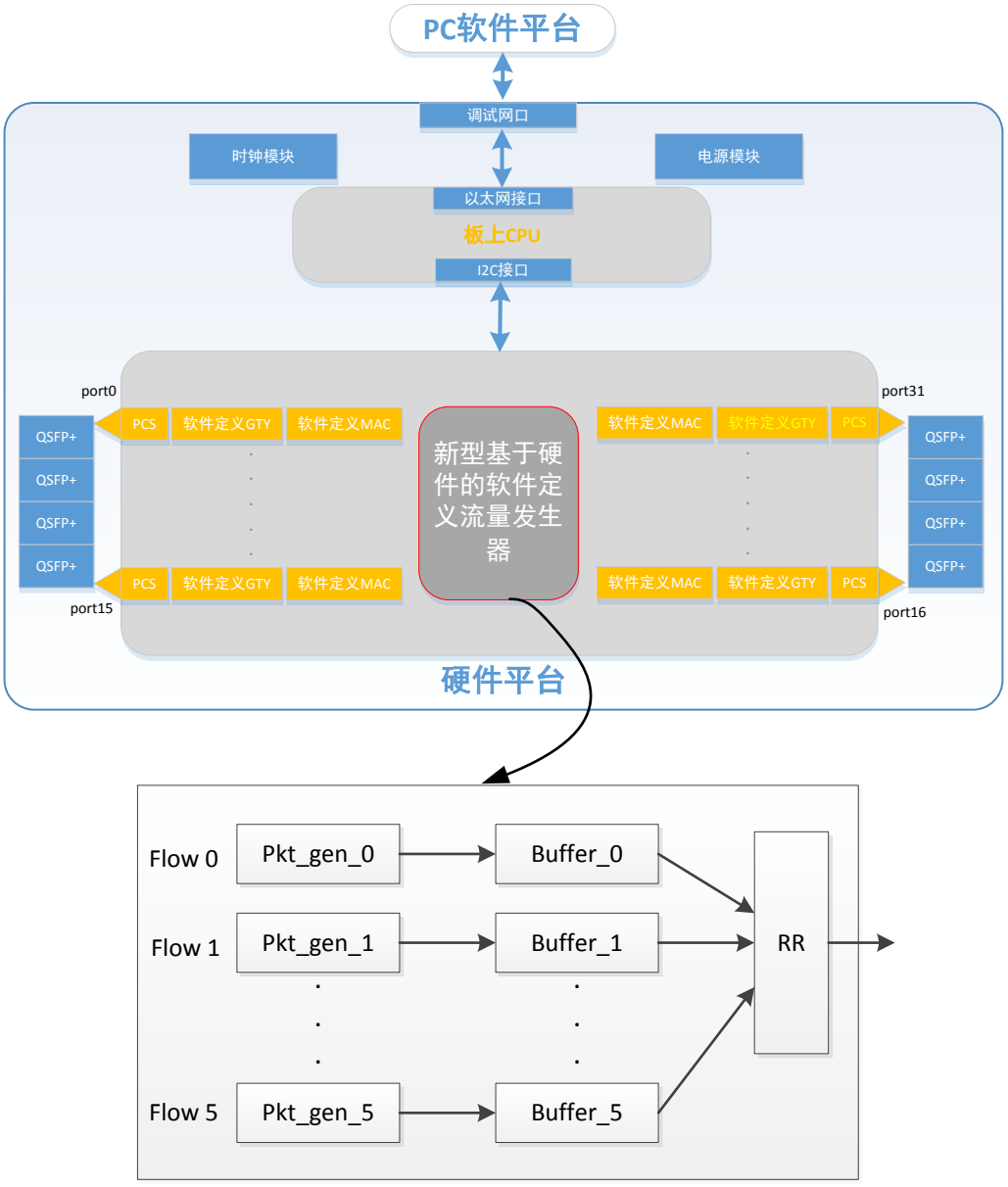


图1　新型流量发生器结构框图

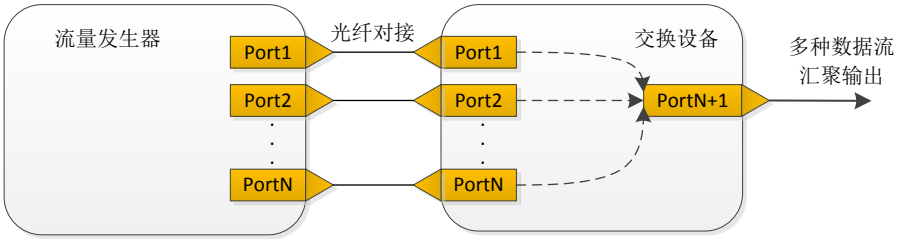


图2　实现多种数据流混合输出的传统方式

存器向后级模块提供包间隔、包数量、包长、包头内容以及发包启停等相关控制信号。

3.1.2　发包控制模块

发包控制模块接收来自配置模块的软件定义配置信号，对其进行逻辑处理后产生包间隔、包数量、包长及发包启停等相关内部控制信号，这些内部控制信号相互配合，以一定的时序作用于后级组包模块及payload产生模块，实现对整个组

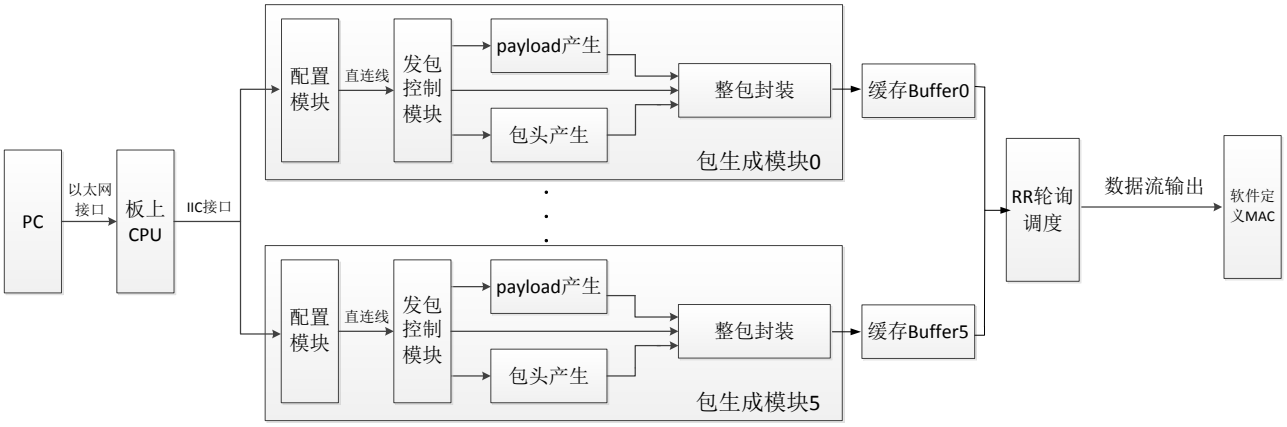


图3　总体工作流程示意图

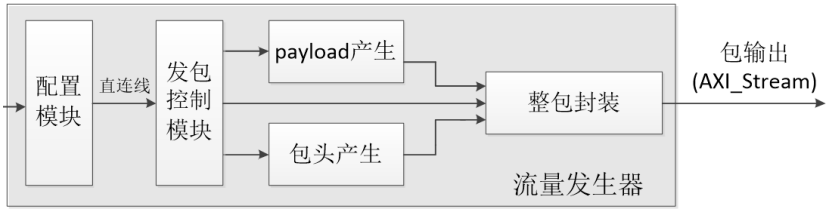


图4　包生成模块结构示意图

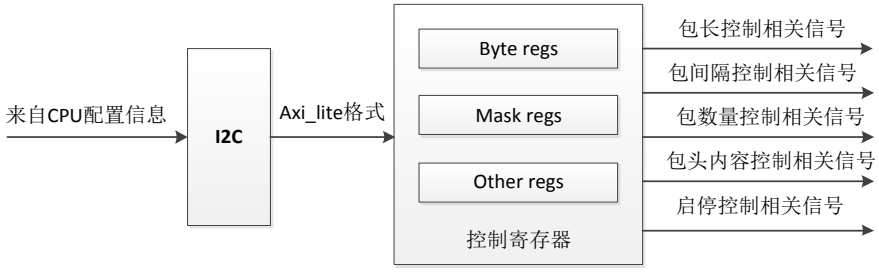


图5　配置模块框图

包过程的控制。

3.1.3　包头产生模块

包头产生模块在来自发包控制模块的包头内容控制信号作用下，生成特定内容的包头（包头的各个字段可取值为随机值，也可取值为固定值）。图6为其原理示意图。在该示意图中，支持的包头大小最大为50字节，需要说明的是，本流量发生器支持的包头大小不限于50字节，可根据需要进行扩展。

包头生成基本思路：每种协议类型的包中都有包头，包头由各种各样的字段组成，针对包头中所包含的各个字段，我们通过控制寄存器来约束其为随机值或特定值。

对包头中包含的字段进行控制的最直接的方法是使用寄存器逐比特指定包头中每一比特位的值，但在包头包含比特位较多的情况下，这种控制方法较繁琐。本流量发生器中采用的方法是：将包头按字节（Byte）进行划分，然后为每一字节分配一个8bit的掩码mask，包头字节中的比特位与掩码mask中的比特位一一对应，与掩码mask中为1的比特位相对应的字节比特位配置为固定值，与掩码mask中为0的比特位相对应的字节比特位配置为随机值。

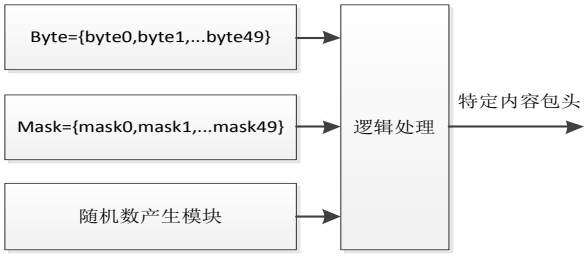


图6　包头产生模块示意图

如图7所示，Byte1用于指定包头第1个字节的值，与其对应的掩码mask为mask1， Byte2用于指定包头第2个字节的值，与其对应的掩码mask为mask2。类似的，Byte50用于指定包头第50个字节的值，与其对应的掩码mask为mask50。

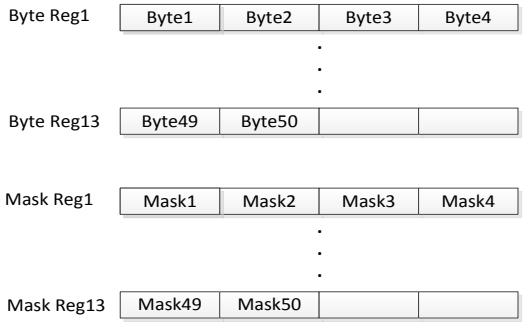


图7　包头配置寄存器

3.1.4　payload产生模块

接收到来自发包控制模块的payload长度控制信号后，payload产生模块生成指定payload长度（payload长度以字节为单位）的AXI_Stream格式的数据。其中，数据内容根据控制信号的不同，可为随机值、固定值，或按一定规律变化的数值，比如第一拍数据为64’h0001_0001_0001_0001，第二拍数据为64’h0002_0002_0002_0002。

3.1.5　整包封装模块

整包封装模块接收来自包头产生模块的包头以及来自payload产生模块的负载payload，在指定的包头结束位置处实现包头与负载payload的拼接，其中，包头结束位置由配置寄存器指定，具体数值可软件定义。

拼接完成后的包再根据需要经CRC计算与添加后即得到最终要生成的包，最终生成的包以AXI_Stream格式输出。需要注意的是，与data信号一起输出的valid、keep、last等信号需要进行时序调整，以与data对齐输出。

3.2　数据缓存Buffer模块

如前所述，每一个软件定义包生成模块后面都有一个相应的数据缓存Buffer，其作用是：在前级包生成模块产生的数据流暂时没有被RR轮询调度选中时，接收并保存包生成模块产生的数据流，使得包生成模块的数据流产生过程持续进行。数据缓存模块的缓存大小设计原则是：在后级没有反压的情况下，该缓存不应该出现反压阻塞前级包生成模块的情况，即要保证包生成模块能够按照配置的流量大小持续产生数据流。

3.3　RR轮询调度模块

RR轮询调度模块用于对各条数据流进行轮询调度，该模块主要有两个作用。

1、实现多条数据流的无缝衔接。无缝衔接，意即当前数据流的一个数据包传输完毕后，下一拍立即选中并输出下一条数据流的数据包，不同数据流之间切换时没有时间间隔。

2、对后级反压信号进行处理后向前级模块传递。当后级反压信号无效时，轮流调度输出各条数据流的数据；当后级反压信号由无效变为有效时，调度输出未传输完毕的数据包后（整包传输功能），阻塞所有数据流的输出；当后级反压信号由有效变为无效时，根据后级反压信号由无效变为有效时的调度结果，调度输出旧的调度结果选中的数据流的下一条数据流的数据（后级反压信号由无效变为有效，相当于暂停了整个流量发生器的数据包输出进程，后级反压信号由有效变为无效后，整个流量发生器的数据包输出进程从暂停处恢复并继续进行）。

RR轮询调度模块在各数据流数据缓存Buffer与后级待测模块/设备之间起着承前启后的桥梁枢纽作用。

4　功能验证

基于设计进度，整个验证过程大致分为设计初期的仿真验证及设计后期的FPGA原型验证。仿真验证易于问题定位，适用于设计开发的前期，可在代码设计完成后发现其存在的绝大部分bug，仿真验证的不足之处是速度慢，不便进行大量发包验证。通过仿真验证完成绝大部分问题的定位解决后，适合展开FPGA原型验证，FPGA原型验证更贴近真实使用场景，方便进行拷机测试，可以发现一些仿真验证无法发现的深层次问题。仿真验证与FPGA原型验证，二者相辅相成。

4.1　仿真验证

基于UVM验证平台，对开发的逻辑进行了仿真验证，验证了流量发生器各功能点及逻辑时序的正确性。仿真时通过流量发生器寄存器访问接口访问各包生成模块中的配置寄存器，配置产生各种流量大小的数据流量，涉及到各种数据流组

合情况。

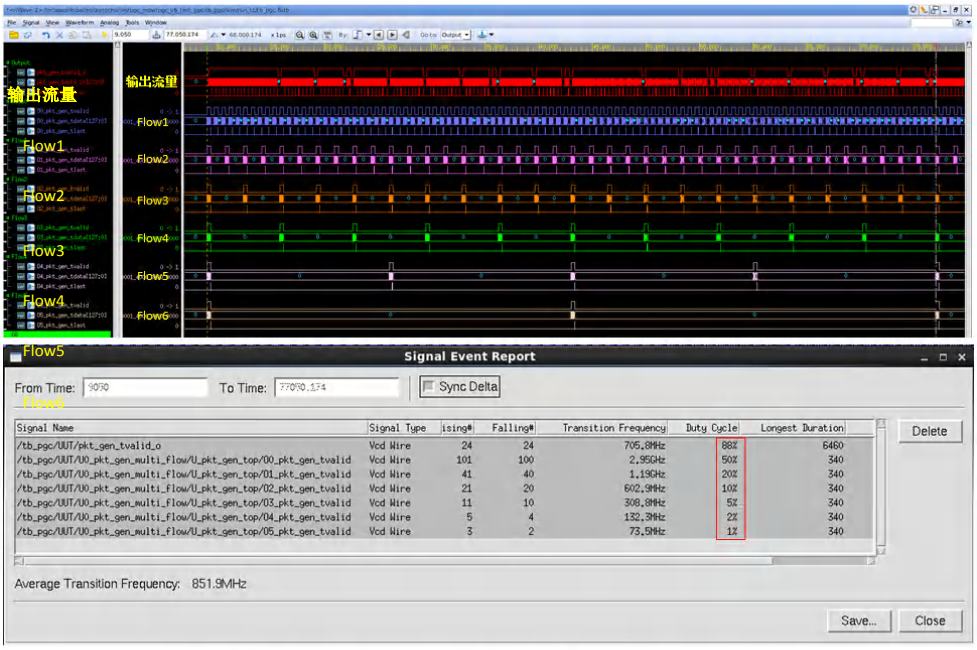


图 8　配置输出满流量的 88%的数据流量时的仿真波形及统计结果

以产生满流量的 88% 的数据流量为例，配置协议类型为 RapidIO，数据包负载大小为 64 字节，ftype 为 5，Ttype 为 5，持续发包，流量发生器内部各包生成模块的流量大小分别配置为满流量的 50%、20%、10%、5%、2%、1%。图 8 所示为上述配置下流量发生器的最终输出波形及内部各包生成模块的输出波形。由图 8 可知，流量发生器输出的总流量大小、内部各数据流大小、包协议类型、包内容、包负载长度及包数量等均符合配置要求。与上述过程类似，还对其它流量大小、其它协议类型、包头字段控制、包负载内容控制、包长控制、包间隔控制等进行了仿真验证，另外，还进行了包类型随机、包内容随机、包长随机等大量随机验证，均与设计预期相符。

4.2　FPGA 原型验证

通过赛灵思集成设计环境 vivado 设计套件对流量发生器逻辑进行综合实现后生成二进制 bit 文件，通过 JTAG 下载至 FPGA 原型验证平台。

通过主机配置协议类型为 RapidIO，数据包负载大小为 64 字节，ftype 为 5，Ttype 为 5，持续发包，流量发生器内部各包生成模块的流量大小分别为满流量的 50%、20%、10%、5%、2%、1%。启动发包后，通过 ILA（vivado 在线逻辑分析仪）抓取波形信号与图 8 仿真波形进行对比，波形一致，流量发生器逻辑时序与逻辑功能在 FPGA 上得到了验证。

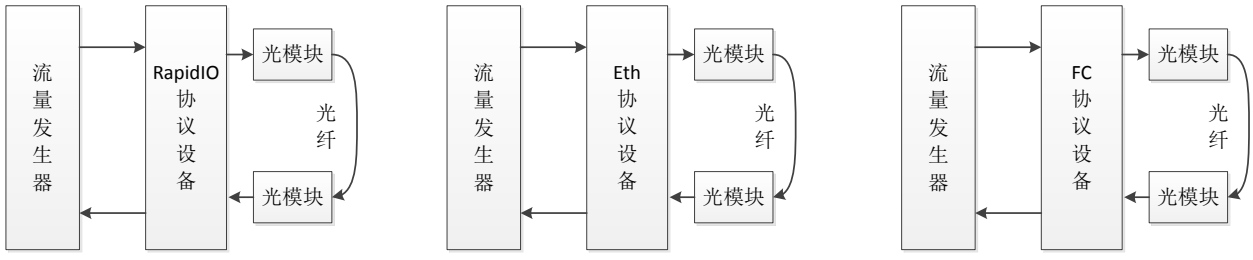


图 9　协议一致性验证

通过将生成的数据流量与对应的协议控制器设备对接，可以检验生成数据流量的协议一致性。

将流量发生器与商用 RapidIO 设备以如图 9 所示的方式连接，配置流量发生器输出 RapidIO 协议

数据包到商用RapidIO设备，同时，商用RapidIO设备将收到的包环回至流量发生器。配置流量发生器满流量持续发送负载长度为4字节的整数倍、负载长度在64~256字节范围内随机、Ttype为5、Ftype为5的RapidIO协议数据包，20秒后停止发包，通过读取流量发生器收发包数量统计结果寄存器，发现收发包数量均为53628526个，无丢包。与上述过程类似，遍历各种流量大小、各种负载长度、各种数据包字段内容，流量发生器收发包数量均相等，且RapidIO协议控制器不报错。说明流量发生器生成的RapidIO协议数据包满足RapidIO协议要求。

同理，配置流量发生器满流量持续发送负载长度为2字节的整数倍、负载长度在46~1500 Bytes范围内随机的以太网数据包，20秒后停止发包，然后通过读取流量发生器收发包数量统计结果寄存器，发现收发包数量均为12097478个，无丢包，遍历各种流量大小、各种数据包字段内容，均无丢包，无错误发生。说明流量发生器生成的以太网数据包满足以太网协议要求。

同样的，配置流量发生器满流量持续发送负载长度为2字节的整数倍、负载长度在0~2112字节范围内随机的FC（Fiber Channel）协议数据包，20秒后停止发包，然后通过读取流量发生器收发包数量统计结果寄存器，发现收发包数量均为17026358个，无丢包，遍历各种流量大小、各种数据包字段内容，均无丢包，无错误发生。说明流量发生器生成的数据包满足FC协议要求。

本流量发生器与商用RapidIO交换芯片CPS-1848（IDT）、商用FC测试仪、商用以太网测试仪及以太网交换芯片都曾进行对接拷机测试，拷机时常均超过60个小时，拷机期间均未发生错误，进一步说明本流量发生器产生的RapidIO协议数据包、FC协议数据包以及以太网协议数据包满足协议一致性要求。

5 结束语

本文提出的流量发生器能够实现对包数量、包间隔、包长、包头内容、包负载内容及包负载大小进行软件定义，能够对数据流的协议类型进行软件定义，还能够对数据流量的大小进行精确的软件定义，而且从架构上天然地方便实现多种数据流的灵活混合。在自研的一款RapidIO交换芯片以及自研的一款支持多种异构协议互连的交换芯片的仿真验证与FPGA原型验证工作中，本流量发生器得到了充分应用，在RapidIO协议测试仪不具备的情况下，为项目的顺利推进提供了有力支撑。另外，在自研的一款RapidIO控制器芯片中，该流量发生器以IP核的形式得到了集成。本流量发生器全自主正向开发，填补了国内在RapidIO协议测试仪等方面的空白，在当前美国商务禁令和国际形势下，具有重要意义。

参考文献:

[1] 韩青．分组流量发生器的设计与实现[D]. 西安电子科技大学，2018.

[2] 姚铭, 应福军．基于 NetFpga 的数据包发生器的实现[J]. 计算机应用于软件，2013，30(1)：242-244.

[3] 任志恒, 武杰, 孔阳, 等．基于 FPGA 的高速以太网流量发生器[J]. 核电子学与探测技术，2011，32(6)：709-713

[4] 基于FPGA的软件定义流量发生器[J].
吕平,董春雷,刘冬培,张文建,汪欣．通信学报．2018(S2)

[5] S. Bradner&J. Macqaid，1999. Request For Comments 2544: Benchmarking Methodology for Network Interconnect Devices[EB/OL].[2011-4-29]http://www. ietf. org/rfc/rfc2544. txt
.

[6] IEEE, 2010, IEEE Standard 802. 3 Part3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications[S]．IEEE802. 3az-2010.

[7] 苗玉良．协议自定义的网络流量发生系统的设计与实现[D]. 重庆邮电大学，2017.

异构嵌入式设备安全启动机制研究综述

李荣泰[1]，常瑞[2]，苗新亮[1]，董卫宇[1]

[1]数学工程与先进计算国家重点实验室（战略支援部队信息工程大学），郑州 450001；

[2]浙江大学计算机科学与技术学院，杭州 310027

摘　要：随着物联网时代到来，各类异构嵌入式设备的安全性问题愈加值得关注。安全启动机制是确保这些设备启动阶段安全性的重要措施，可以防止攻击者对固件和系统映像篡改后进行非法启动，确保启动系统的可信。本文对启动阶段存在的各类安全威胁进行了分析，并对不同架构上的安全启动机制进行了详细阐述，提出了一种通用的异构嵌入式设备安全启动机制框架，综述了基于此框架的硬件辅助、信任根增强、形式化验证等安全增强改进方法，最后结合安全启动机制从PC端应用到云端的现状，提出了异构嵌入式设备安全启动机制未来可能的研究方向。

关键词：异构设备、安全启动、可信根、形式化验证

A Survey on Secure Boot Mechanism of Heterogeneous Embedded Devices

Li Rongtai[1], Chang Rui[2], Miao Xinliang[1], Weiyu and Dong[1]

1.State Key Laboratory of Mathematical Engineering and Advanced Computing (Strategic Support Force Information Engineering University), Zhengzhou 450001;

2.College of Computer Science and Technology, Zhejiang University, Hangzhou 310027

Abstract: With the incoming Era of Internet of Everything (IoT), the security issues of various heterogeneous embedded devices have attracted more and more attention. The secure boot mechanism is a significant measure to ensure the security of these devices during bootstrap phase. It can prevent attackers from tampering with the firmware and system images to boot system illegally, and ensure the credibility of the booting system. In this paper, we analyze the various security threats in the bootstrap phase, and elaborate on the secure boot mechanism on different architectures. Then we propose a general framework for the secure boot mechanism of heterogeneous embedded devices. Furthermore, the security enhancement method such as hardware-assisted, root-of-trust enhancement, and formal verification based on the framework are researched. Finally, the future research directions are pointed out in terms of the application of the secure boot mechanism being migrated from PC to the cloud platform.

Key words: heterogeneous devices; secure boot; root of trust; formal verification

1　引言

随着物联网（Internet of Things，IoT）技术的发展，越来越多的智能设备，如可穿戴设备、智能台灯、扫地机器人以及智能家电等通过网络进行互联，由此打破设备之间的隔离，进而给用户

提供了丰富便利的服务。然而，万物互联不仅给用户提供了便携服务，同时也使得原孤立设备面临更大的攻击面，遭受来自于互联网的安全威胁和攻击［1］。攻击者利用网络远程访问这些智能设备来窃取用户数据［2］，甚至篡改设备固件［3］和系统映像，进行拒绝服务攻击（Deny of Service，DoS）或植入后门［4］以持久性监控用户行为，对用户的安全隐私造成了极大威胁。为保证这些智能终端设备的安全性，一个重要的措施就是在系统引导过程中验证系统和软件的完整性、真实性、合法性，从而保证设备启动阶段的安全。

安全启动机制［5］是确保计算机系统安全性的基础。安全启动过程就是信任链的构建过程，以信任根（Root of Trust，RoT）为基础，逐步对启动过程中每个阶段的数据和资源进行验证。通过一级度量一级，一级验证一级，一级信任一级，直到整个信任链构建完成，则系统也成功安全启动。启动过程中当前阶段都会对下一阶段即将运行的代码进行认证，若认证成功，则将控制权转移给下一阶段代码；若任意阶段认证失败，则系统启动过程中止。通过安全启动机制，用户可以确保没有攻击者篡改过系统和其它软件，进而启动预期系统。然而目前仍存在众多的安全启动威胁［6］-［18］，因此许多原始设备制造商（Origin Equipment Manufacturer，OEM），芯片设计商以及开源平台都针对自己的架构提出了相应的安全启动方案［37］-［41］。工业界，学术界的研究者们也纷纷对安全启动机制进行了深入研究，具体包括有：RISC-V 处理器安全启动和远程认证［22］，虚拟机可信启动［23］，UEFI 安全启动［24］，电子控制单元（Electronic Control Units，ECU）安全启动［25］，FPGA 设备安全启动［26］-［33］等等。异构嵌入式设备都为加强启动安全性提出了不同的安全启动机制，但这些启动机制存在不足和漏洞，仍有可以改进的部分。因此，本文主要针对异构嵌入式设备安全启动机制进行研究。

本文的主要贡献有以下几个方面：

1. 分析了嵌入式设备在系统启动阶段面临的安全威胁，包括系统启动阶段可能存在的漏洞和攻击行为；

2. 重点综述了不同架构上的安全启动方案及能达到的安全效果，分析了相关研究方案，提出了一个通用的安全启动框架，并给出了基于此框架的硬件辅助、信任根增强、形式化验证等安全增强改进方法；

3. 结合安全启动机制从PC端应用到云端的现状，分析了安全启动机制在软硬件方面可能的研究方向。

2　安全启动威胁

安全启动威胁来源广泛。从硬件层面来看，根据设备是否具有安全硬件可以分为针对没有安全硬件设备的攻击以及针对具有安全硬件设备的攻击。由于无安全硬件设备缺乏对启动各阶段软件的硬件保护和检查，此类设备易遭受到启动攻击，攻击者可以轻易地修改系统和其它软件。对于具有安全硬件的设备，针对无安全硬件设备的攻击行为不再适用，但安全硬件可能存在漏洞，因此攻击者可以利用安全硬件自身的漏洞来攻击启动过程，同样对用户安全造成威胁。从软件层面上考虑安全启动威胁，可将其视作启动各阶段的代码漏洞。一旦代码具体实现中存在可被利用的漏洞，攻击者便能通过此类漏洞绕过启动流验证，软件漏洞同时也是最为常见，影响范围最广的漏洞。另一类安全启动威胁来源是启动机制自身存在问题甚至系统缺乏安全启动机制，攻击者可以利用启动机制缺陷来绕过启动检查，或者在缺乏安全启动机制设备上直接修改设备程序。

2.1　硬件漏洞

现场可编程门阵列（Field Programmable Gate Arrays，FPGA）由于其高效性和灵活性，已广泛应用于嵌入式设备，航空航天，大数据以及云计算等关键领域。但FPGA硬件本身的安全性问题不容小觑，Jacob等人［6］利用了FPGA与CPU在同一内存总线互联的硬件特性，且代码认证过程中存在中断导致控制权转移的原语，采用恶意硬件打破了基于FPGA系统的安全启动。Ender等人［7］发现了Xilinx 7系列FPGA上不可修补的硬件漏洞StarBleed，攻击者可以利用配置寄存器WBSTAR破解Bitstream的加密和鉴权，甚至修改bitstream来绕过安全启动。除广泛运用的FPGA硬件以外，其他硬件诸如雷电接口或Boot ROM代码等

也存在漏洞。Trammell［8］发现了存在苹果Mac计算机上的雷电接口漏洞，攻击者仅须短暂物理接触电脑，便可以将恶意固件写入Boot ROM。Boot ROM代码作为硬件信任根，是整个安全启动过程的基础，其安全性十分重要。checkm8［9］作为无法修补的硬件漏洞，严重影响了数百万苹果设备的安全性，攻击者可以利用该漏洞绕过安全启动并对设备进行任意升级或降级操作。同样的仍存在CVE-2017-12223［10］，CVE-2018-15370［11］，CVE-2018-6240［12］等Boot ROM代码漏洞，这些漏洞可被用于绕过安全启动检查，引导启动未签名的系统以及实现任意物理地址写入。

2.2 Bootloader软件漏洞

设备启动过程最终需要落实到具体代码执行，若代码存在可被攻击者利用的漏洞，可能会导致安全启动机制被绕过，因此对于引导程序代码的安全性分析尤为重要。Bootloader程序作为Boot ROM代码和OS之间的桥梁，运行在各类设备上，如智能手机，平板电脑，以及各类PC主机，其安全性与人们息息相关，因此众多研究者对不同设备的Bootloader进行了分析。Redini等人［13］针对移动设备的Bootloader安全性问题，使用静态分析和动态符号执行结合开发了多标签污点分析工具BootStomp，发现了4个Bootloader中存在的6个安全漏洞，攻击者可以通过这些漏洞执行任意代码破坏整个信任链。Hay［14］发现了数个OEM Bootloader中fastboot接口存在的漏洞，包括安全启动和设备锁绕过，恶意充电器、耳机攻击，内存转储数据泄露和隐藏功能启用。Eclypsium的研究人员披露了Gurb2 Bootloader中的漏洞（BootHole，CVE-2020-10713［15］），即使设备开启了安全启动，攻击者可以利用该漏洞在启动过程中执行任意代码，打破了安全启动机制并安装持续性的Bootkit来控制设备。U-Boot作为广泛使用的Bootloader程序，其安全性问题也十分重要，目前已发现CVE-2018-1000205［16］，CVE-2018-3968［17］，CVE-2020-10648［18］等漏洞可以绕过U-Boot自身的验证启动。

2.3 启动机制缺陷

启动机制设计若存在错误，则安全启动过程就可能被绕过。Husain等人［21］发现了Chromium OS上Verified Boot过程中存在的漏洞，通过恶意的rootfs替换原rootfs便可以绕过安全启动。UEFI作为取代传统BIOS的新一代操作系统和固件之间的接口，一经提出就广泛运用于现代计算机的启动过程中。但是由于其本身规范繁杂，也存在不少的漏洞，因此，对UEFI规范的安全性分析必不可少。Bashun等人［19］对UEFI的安全问题做了详细分析，描述了其安全问题来源并对其分类。Pankov等人［20］针对基于Intel计算机的UEFI BIOS和Intel管理引擎（Management Engine，ME）软件的漏洞，进行了相关分析。

3 异构平台安全启动方案

针对当前存在的启动威胁，异构平台也推出了不同的安全启动方案来增强设备启动阶段安全，如Boot Guard［37］，Validated Boot［38］，Trusted Boot［39］，Verified Boot［40］。这些方案要么采用新增硬件方式来加强安全启动，要么对安全启动规范进行改进，以适应不同的服务。本节首先分别介绍了异构平台所提出的启动方案，然后对其进行安全启动效果分析。

3.1 Intel平台安全启动

Intel针对安全启动推出了自己的安全启动方案，使用Boot Guard技术保护Intel平台上操作系统或虚拟机的安全性。Boot Guard在安全启动过程中引入了名为认证代码模块（Authenticated Code Module，ACM）的黑盒，同时将BIOS分成初始引导块（Initial Boot Block，IBB）和厂商引导块（OEM Boot Block，OBB）两部分。启动流程图见图1，下面对具体的安全启动流程进行说明：

1. 设备上电后，CPU执行硬编码在CPU内部Boot ROM中的微代码。

2. 微代码对ACM进行验证，验证通过后，ACM得到执行并验证IBB，验证通过后控制权转移到IBB。

3. IBB模块负责UEFI中SEC/PEI部分，该模块执行后对OBB进行验证，验证通过后控制权转移到OBB。

4. OBB模块负责UEFI中驱动执行环境（Driver Execution Environment，DXE）部分，OBB执行完毕后，整个UEFI环境准备完毕，接下来进入UEFI安全启动过程。

5. UEFI安全启动执行，进行一系列准备活动后，对操作系统引导程序进行验证，验证通过后控制权转移到OS bootloader。

6. Bootloader程序对系统映像进行验证，验证通过后将启动操作系统。

其中若任意阶段验证失败，则系统启动过程中止。整个安全启动的信任根为CPU硬件，CPU会对微代码进行验证，由于CPU无法被伪造和篡改，因此整个安全启动过程是可信的。

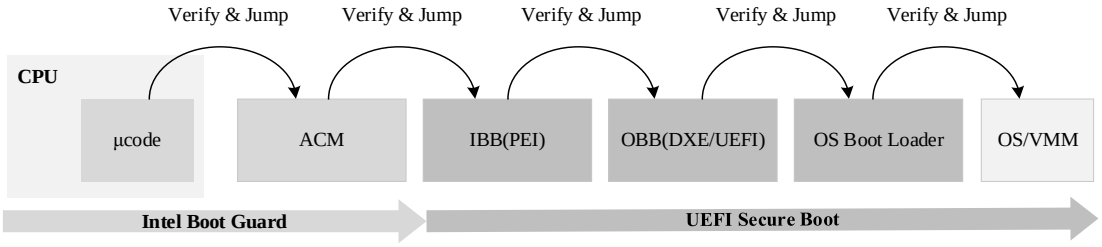


图 1　Intel Boot Guard技术安全启动流程

Boot Guard技术不仅支持安全启动，同时也提供测量启动（Measured Boot）选项。测量启动对启动过程中的每个组件进行度量，将最终度量值存储到可信平台模块（Trusted Platform Module，TPM）的平台配置寄存器（Platform Configuration Register，PCR）中，用户可以根据度量值判断系统状态并自行决定下一步操作。

3.2　AMD平台安全启动

AMD平台也推出了硬件验证启动（Hardware Validated Boot，HVB）方案，通过添加平台安全处理器（Platform Security Processor，PSP）硬件来加强启动过程的安全性。其中PSP硬件包含专用的32位微控制器，隔离的片内ROM & SRAM，密码协处理器，系统内存和资源接口。PSP内Boot ROM作为信任根，引导系统进行安全启动。图2描述了其启动流程，具体的安全启动流程如下：

1. 系统上电后，PSP初始化，片内Boot ROM代码开始执行。

2. Boot ROM代码对片外引导程序验证，验证通过后将控制权转交给引导程序。

3. 引导程序执行验证PSP外部代码，首先验证第一区块的BIOS代码，验证通过后控制权传递给BIOS，此后由BIOS对后续程序进行验证。

4. BIOS逐步验证UEFI驱动，UEFI应用和系统bootloader，最后将控制权转移给OS bootloader。

5. Bootloader程序执行，加载系统映像，系统启动完成后，整个安全启动流程结束。

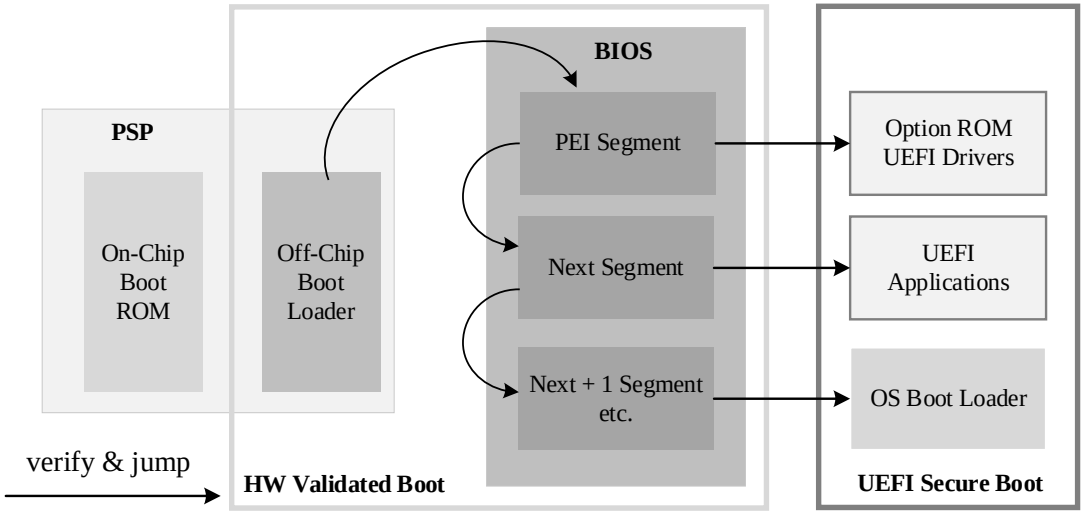


图 2　AMD Hardware Validation Boot安全启动流程

3.3 ARM平台安全启动

ARM针对安全启动提出了自己的平台安全架构（Platform Security Architecture，PSA），PSA定义了安全架构和Trusted Boot。在安全架构方面，如果硬件支持软件隔离，则PSA上的软件切分为两个安全域：安全处理环境（Secure Processing Environment，SPE）和非安全处理环境（Non-Secure Processing Environment，NSPE）。SPE由PSA RoT和可信服务（Trusted Services）组成，可信服务用于为非安全应用提供特定的安全服务。NSPE提供安全性要求不高的一般功能。在具有TrustZone安全扩展的ARM处理器上，SPE可视为安全世界，NSPE可视为普通世界。Trusted Boot启动流如图3所示，下面对Trusted Boot过程进行说明：

1. 系统上电或重置后，嵌入在Boot ROM的引导代码作为硬件信任根开始运行。

2. 引导代码检查Trusted Boot软件的真实性，检查通过后，控制权转移到Trusted Boot软件。

3. Trusted Boot软件验证SPE软件和NSPE软件，验证通过后，会相继运行SPE和NSPE软件。

4. SPE初始化系统和可信服务的信任根，启动安全世界系统。

5. NSPE初始化非安全相关外围设备，启动普通世界内核，当普通世界内核启动成功后，系统启动完成。

在Trusted Boot过程中，Boot ROM内代码作为信任根不可更改。Trusted Boot软件是可选的第二级Bootloader，仅在多阶段启动过程中才可运行，单阶段启动过程中仅由Boot ROM代码验证SPE和NSPE软件。

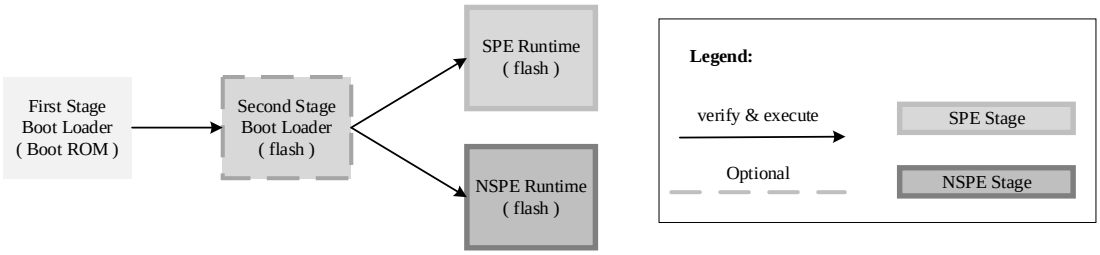


图3 ARM Trusted Boot安全启动流程

3.4 Android平台安全启动

Google自从Android 4.4时，就针对Android开放源代码项目（Android Open-Source Project，AOSP）提出了Verified Boot技术。Verified Boot不仅建立了从硬件保护信任根到Bootloader，再到Boot分区和其他验证分区的完整信任链，同时描述了Android Bootloader应当实现的其他高级功能。Android设备可以分为两种状态，LOCKED和UNLOCKED，通过Fastboot命令进行状态切换。处于LOCKED状态的设备仅可以加载被信任根（包括OEM信任根和用户自定义信任根）签名过的OS，而处于UNLOCKED状态的设备可以对分区进行修改，刷写未签名过系统映像并启动。Verified Boot启动流程如图4所示：

Verified Boot技术也在不断发展，从Android 7.0开始，系统开始强制执行启动时验证，使得篡改过的设备无法启动。 在Android 8.0以及更高的版本上，该技术更名为安卓启动时验证（Android Verified Boot，AVB），提供了不同分区更新，对于签名分区通用脚注格式以及防回滚保护功能。Weiss对AVB机制及其作为安全特征的优缺点进行了分析，有助于增进对Android设备安全性的了解［42］。

3.5 基于RISC-V平台启动

RISC-V作为开源精简指令集，一经提出就受到广大研究者的关注。该架构上的启动方案与前文提到的方案中有诸多相同之处，其启动流程见图5，具体的启动流程如下：系统重置后，在boot ROM内的零级引导程序（Zero Stage Bootloader，ZSBL）作为信任根开始运行，加载第一级引导程序（First Stage Bootloader，FSBL）并对其验证，验证通过后，控制权转移到FSBL，FSBL对RISC-V规范提出的OpenSBI进行验证并将控制权转移到OpenSBI，OpenSBI进行部分初始化操作后，对第二级引导程序（Second Stage Bootloader，SSBL）验证并将控制权转移，最后由SSBL引导操作系统内核启动。对于部分裸金属设备而言，可以在FSBL阶段直接加载OS启动，也可以跳过OpenSBI阶

段，直接加载U-Boot。

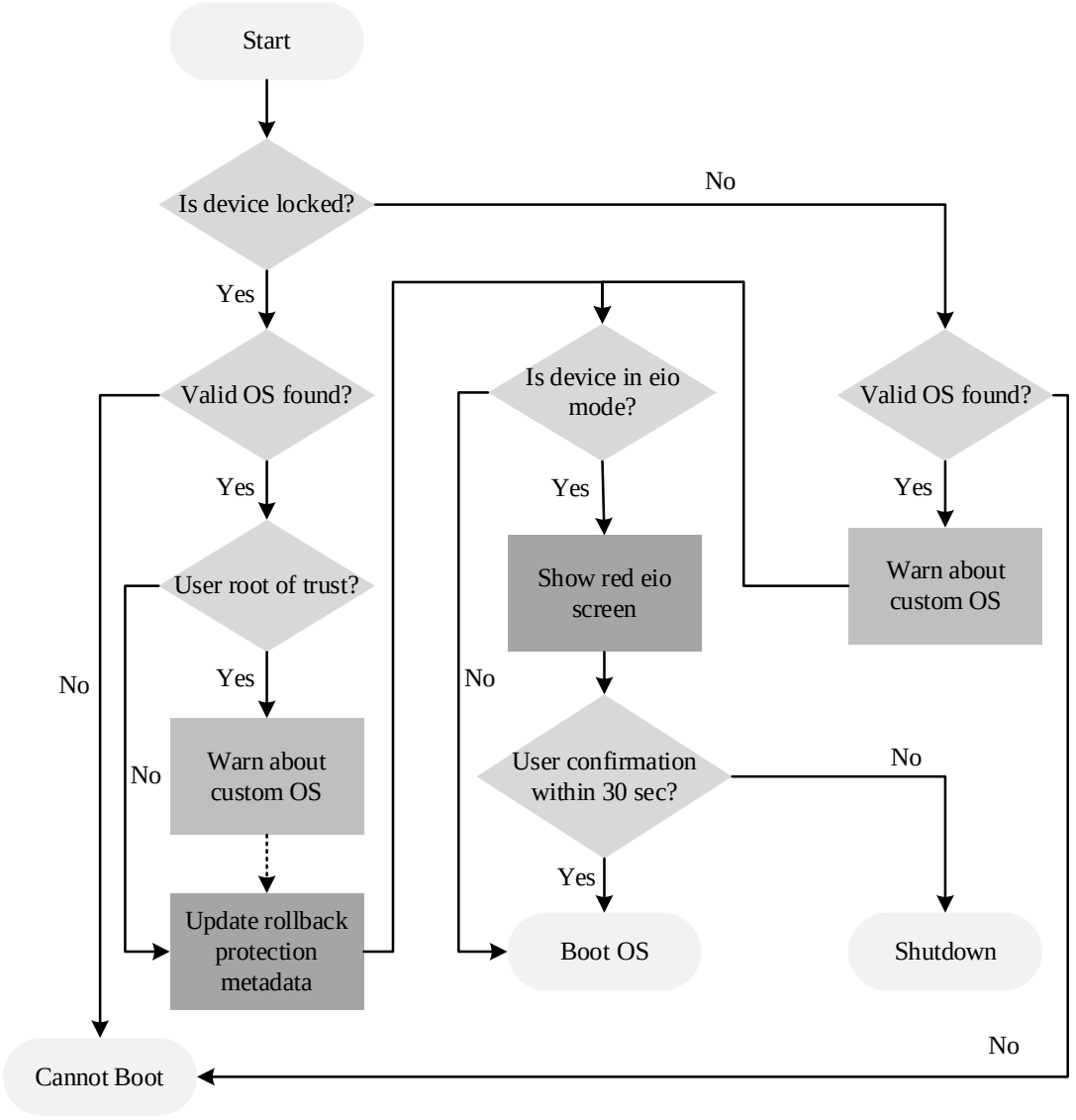


图 4　Google Verified Boot安全启动流程

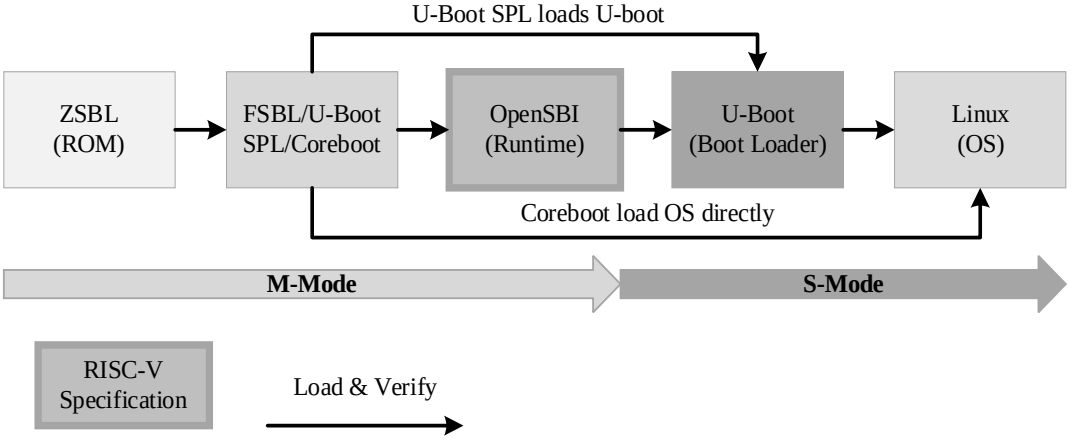


图 5　RISC-V平台系统启动流程

4 通用安全启动框架

4.1 异构平台安全启动机制分析

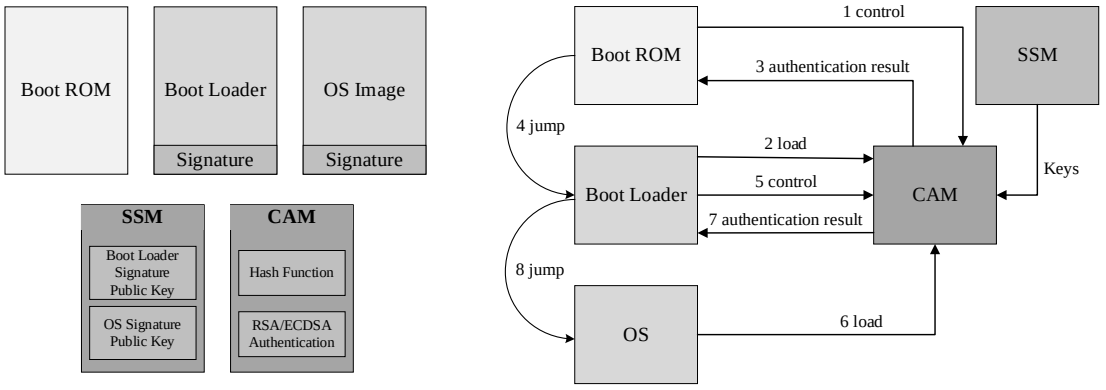
综上所述，不同平台架构针对系统启动阶段采取了不同的安全启动机制。例如，新增硬件方式更改了信任根，将信任根降到更低层次或者与原执行环境进行隔离，减少了攻击面进而加强了启动阶段安全性；附加服务的方式在安全启动的基础上，增加了对不同执行环境安全性保证或者提供给用户更多功能。表 1 比较分析了异构平台的安全启动机制。其中 Intel 和 AMD 平台采用新增不同硬件的方式来加强安全启动过程，Intel 平台将 Boot Guard 技术嵌入在 CPU 内部，其安全性保证由 Intel 提供；而 AMD 平台采取添加 ARM Cortex-A5 芯片用作 PSP 硬件，但目前已经发现了不少 PSP 内部漏洞，使得 HVB 启动方案的安全性大大降低；ARM 和 Google 对启动规范进行了改进，ARM 平台提出的 Trusted Boot 不仅确保了 NSPE 系统的合法性，同时确保了 SPE 系统的安全性，给用户提供了两个执行环境的安全性；Google 推出的 Verified Boot 技术对 Bootloader 程序要求新增了许多其它功能，如设备解锁，提供用户刷入自定义 OS，提醒用户当前设备状态等。总体来说，Intel 平台启动过程安全性依赖于自身；而 AMD 平台启动安全依赖于 ARM 处理器（PSP）的安全；ARM 平台安全启动新增了对可信执行环境（Trusted Execution Environment，TEE）的安全保护；Google 在安全性和用户需求两者之间进行平衡，将安全需求最终交由用户来决定；RISC-V 平台目前并没有针对安全启动提出相应的规范，这是可待改进的地方。

表 1 异构平台安全启动方案总结

平台	技术名称	具体方案	安全增强效果
Intel	Boot Guard	新增硬件	兼顾 Secure Boot 和 Measured Boot
AMD	Validated Boot	新增硬件	安全处理器与应用处理器隔离
ARM	Trusted Boot	附加服务	兼顾 NSPE 系统安全和 SPE 系统安全
Google	Verified Boot	附加服务	兼顾 Secure Boot 和用户需求

4.2 通用安全启动框架

根据上述不同硬件平台架构的安全启动机制，本文提出了一种通用安全启动框架。系统可以分为安全启动组件和安全启动流程两个部分。安全启动组件由各级软件和安全硬件组成，其中软件通常包括嵌入在 Boot ROM 中不可更改的 Boot Code，用于引导系统启动的 Boot Loader，以及操作系统映像；而硬件由安全存储模块（Secure Storage Module，SSM）和代码验证模块（Code Authentication Module，CAM）组成，安全存储模块可以防止恶意攻击者的篡改，用于存储对各级软件验证的密钥，代码验证模块实现对各级软件的签名验证。



(a) Secure Boot Components　　(b) Secure Boot Flow

图 6 安全启动框架

如图 6 所示，描述了整个安全启动框架。在图 6（a）中，软件部分由 Boot ROM，Boot Loader 和 OS 映像组成，Boot ROM 内的代码作为信任根不可更改，无需对其进行验证，而 Boot Loader 和 OS 映像尾部附加了用于验证其合法性的签名；在硬件部分中，安全存储模块存储了用于验证 Boot Loader 和 OS 签名的公钥，代码验证模块包含了 Hash 功能和签名认证功能。图 6（b）描述了通用

的安全启动流程：

1. 系统上电后，Boot ROM 内的代码开始执行，控制代码验证模块读取 Boot Loader，Boot Loader 签名和安全存储模块中的 Boot Loader 签名公钥。

2. 代码验证模块通过 Hash 功能对 Boot Loader 进行哈希，得到哈希值 V_1，使用 Boot Loader 签名公钥对 Boot Loader 签名进行解密，得到解密值 V_2，比较 V_1 和 V_2 的值是否相等，并将结果返回给 Boot ROM 程序。

3. Boot ROM 程序判断 Boot Loader 验证是否通过，若通过则跳转到 Boot Loader 程序，否则启动终止。

4. Boot Loader 程序开始执行，初始化系统启动的硬件资源后，继续控制代码验证模块读取 OS 映像，OS 签名和 OS 签名公钥。

5. 代码验证模块通过 Hash 功能对 OS 映像进行哈希，得到哈希值 V_3，使用 OS 签名公钥对 OS 签名进行解密，得到解密值 V_4，比较 V_3 和 V_4 的值是否相等，并将结果返回给 Boot Loader 程序。

6. Boot Loader 程序判断 OS 映像验证是否通过，若通过则将控制权转移到 OS，启动系统，否则启动终止。

若系统启动阶段存在多级 Boot Loader 或其他待验证组件，则安全存储模块也会添加对应的签名公钥，启动流中也会添加相应的验证过程。

5 安全启动改进技术

5.1 硬件辅助安全启动

FPGA 由于其可编程的硬件特性，适用于多数对于灵活性和运行速度要求较高的系统，因此存在很多研究基于 FPGA 的安全启动工作。Devic 等人［30］提出了基于 FPGA 的 Linux 安全启动机制，能够保护从 Bitstream 到内核引导启动链的完整性，防御攻击者对系统映像的篡改和重放攻击。Pocklassery 等人［31］提出了基于 FPGA 的自我认证安全启动机制，该机制采用了自我认证的 PUF 架构，能够对引导期间加载到 FPGA 可编程逻辑区中的未加密 Bitstream 进行自我认证。Streit 等人［32］通过在 FPGA 可编程逻辑区中实现可信内存接口单元（TMIU）对非易失性存储器（Non-Volatile Memory，NVM）和可编程片上系统（System of Chip，SoC）进行身份验证，实现了在可编程 SoC 架构上从 NVM 加载系统的安全启动方案。Hajyahya 等人［33］针对 RISC-V 架构以处理器作为可信计算基过大，可能会遭受类似于 Meltdown 或 Spectre 侧信道攻击的问题，通过添加硬件实现的代码认证单元（Code Authentication Unit，CAU）和密钥管理单元（Key Management Unit，KMU），加强了基于 RISC-V SoC 的安全启动过程。Khalid 等人［43］在双核 LEON3 系统上，将处理器分为安全处理器和应用处理器，通过安全处理器度量普通 Boot Loader 和系统 Kernel，实现了嵌入式设备的可信引导过程，但该方案增加了 25% 的启动时间开销。González［44］提出了一个用于解决安全启动和可信启动双方缺点的新架构，在 ARM 架构上采用 TrustZone 作为 TEE，安全元件（Secure Element，SE）作为防篡改单元，实现了对启动系统的一阶段验证和软件对系统的二阶段验证。

5.2 信任根增强

Zhao 等人［34］针对现有 ARM TrustZone 架构的信任根可能被破坏和灵活性不够的问题，基于片上 SRAM 物理不可克隆功能实现了 TEE 信任根，用于防御针对整个富 OS 的软件攻击。Huang 等人［35］针对部分计算平台缺乏可信硬件的问题，引入一个安全硬件 USB Key 来模拟 TPM 的基本功能，并提出一个基于 USB Key 完整性验证模型用于实现 OS 启动过程的逆向完整性验证，最终实现了在没有 TPM 的终端系统中的可信启动。Jiang 等人［36］提出了基于 ARM TrustZone 的安全启动模式，使用 eFuse 存储各级启动代码的证书，用于确保嵌入式设备的隔离执行环境和应用程序的完整性和真实性，并保证了其信任链相对于 TCG 提出的信任链更为可信。

5.3 形式化验证

形式化方法是根据某个或某些规范或属性，采用严格的数学方法证明软硬件的正确性，其验证方法分为定理证明和模型检测。对安全启动进行形式化验证可有效证明其安全性。AWS 团队［45］为确保其云数据中心的启动安全，采用 C 有界模型检查器（C Bound Model Checking，CBMC）对启动阶段的引导代码进行符号模型检查，形式化验证了其代码内存安全，为其数据中心安全性建立坚实基础。Straznickas［46］设计了一个

Bootloader程序，并采用Coq工具以低级RISC-V语义描述了其规范，将Bootloader的功能正确性描述为一个定理，最后证明了数个必要引理以证明定理的正确性。Muduli等人［47］提出了一种对经过身份验证的固件加载程序端到端安全性的验证方法。Huang等人［48］使用指令集抽象（Instruction-level Abstraction，ILA）对硬件和固件进行了协同验证。

表2对现有安全启动增强方案进行了总结。

表 2　安全启动增强方案

改进方案	名称	具体方案
硬件辅助安全启动	Florian	启动流保护
	SASB	bitstream自我认证
	Franz-Josef	验证SoC和NVM
	RISC-V	专用硬件验证代码
	Khalid	安全核心验证普通世界
	González	Secure Boot和Trusted Boot相结合
信任根增强	Zhao	SRAM-Based PUF
	Huang	外置TPM
	Jiang	信任链改进
形式化验证	Cook	模型检测
	Straznickas	定理证明
	Muduli	安全性验证
	Huang	固件硬件协同验证

6　未来研究趋势

安全启动机制应用十分广泛，从传统PC到嵌入式设备以及云计算领域，都需要确保计算机系统启动过程的安全性。下面就针对目前安全启动机制存在的缺陷，如何加强安全启动讨论潜在的研究方向。

6.1　借助于形式化方法的安全启动

6.1.1　*启动代码形式化验证*

启动过程最终需要落实到具体代码的执行，代码的安全性需要得到保证。但现有的漏洞检测技术只能发现未知漏洞，无法保证代码不存在漏洞。形式化方法可以从数学理论上证明系统符合用户定义的安全，信任根代码作为系统引导的基础，使用形式化技术对信任根代码以及启动各阶段代码进行分析可以确保启动过程的正确性和安全性，这是未来值得研究的一个方向。

6.1.2　*启动流形式化分析*

部分安全启动机制不够完善，面临攻击者篡改启动流的威胁。因此对启动过程进行建模，考虑启动过程中可能的攻击行为，并对启动流逻辑进行形式化分析，确保启动流不存在TOCTOU等逻辑错误，保证启动流的完整性，也是一个值得研究的方向。

6.2　使用安全语言的引导代码

现有的大多数引导代码是使用C/C++所编写的，虽然灵活但不安全，易导致内存安全漏洞，开发者需要足够的安全知识才能编写出安全性较高的程序。采用安全语言编写引导代码可以避免此类缺点，使得开发者仅需关注于代码实现的逻辑，无需操心未定义行为和内存安全，减少了开发者工作量。因此使用安全语言编写引导代码来加强启动安全是一个可能的研究方向。

6.3　选择开源硬件

虽然安全硬件可以有效加强启动过程安全性，但硬件自身安全性值得考虑。针对闭源硬件自身存在安全隐患的问题，可以采用基于RISC-V等开源指令集设计专用的安全硬件，并采用形式化技术对硬件逻辑进行验证来提升硬件本身安全性的方法。因此使用开源硬件加强安全启动是一个潜在的研究方向。

7　总结

安全启动机制作为保证嵌入式系统基础安全性的重要措施，被广泛运用于各类异构嵌入式设备，诸多研究者们对此进行了深入研究。本文对安全启动机制进行了全面综合的分析综述，阐述了启动阶段面临的安全威胁、已有的安全启动方案等，分析比较了不同处理器平台的安全启动机制，提出了一种适用于异构平台的通用安全启动框架，并论述了已有的针对安全启动机制的改进措施，最后分析了未来可能的研究趋势。

参考文献:

[1] Abdul-Ghani H A, Konstantas D, Mahyoub M. A comprehensive IoT attacks survey based on a building-blocked reference model［J］. IJACSA) International Journal of Advanced Computer Science and Applications，2018，9(3)：355-373.

[2] Acar A, Fereidooni H, Abera T, et al. Peek-a-Boo: I see your smart home activities, even encrypted!［C］//Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks. 2020：207-218.

[3] CVE-2019-1649［EB/OL］. https://cve. mitre. org/cgi-bin/cvename. cgi?name=CVE-2019-1649
.
[4] CVE-2019-6260［EB/OL］. https://cve. mitre. org/cgi-bin/cvename. cgi?name=CVE-2019-6260
.
[5] Arbaugh W A, Farber D J, Smith J M, et al. A secure and reliable bootstrap architecture［C］. ieee symposium on security and privacy，1997：65-71.
[6] Jacob N, Heyszl J, Zankl A, et al. How to Break Secure Boot on {FPGA} SoCs Through Malicious Hardware［C］. cryptographic hardware and embedded systems，2017：425-442.
[7] Ender M, Moradi A, Paar C. The Unpatchable Silicon: A Full Break of the Bitstream Encryption of Xilinx 7-Series FPGAs［C］//29th {USENIX} Security Symposium ({USENIX} Security 20). 2020.
[8] Hudson T, Rudolph L. Thunderstrike: EFI firmware bootkits for Apple MacBooks［C］// Proceedings of the 8th ACM International Systems and Storage Conference. 2015：1-10.
[9] Checkm8 | Activation Lock Bypass Software［EB/OL］. https://checkm8. info
.
[10] CVE-2017-12223［
EB/OL］. http://cve. mitre. org/cgi-bin/cvename. cgi? name=CVE-2017-12223
.
[11] CVE-2018-15370［
EB/OL］. http://cve. mitre. org/cgi-bin/cvename. cgi? name=CVE-2018-15370
.
[12] CVE-2018-6240［EB/OL］. http://cve. mitre. org/cgi-bin/cvename. cgi?name=CVE-2018-6240
.
[13] Redini N, Machiry A, Das D, et al. BootStomp: On the Security of Bootloaders in Mobile Devices［C］. usenix security symposium，2017：781-798.
[14] Hay R. fastboot oem vuln: Android bootloader vulnerabilities in vendor customizations［C］// 11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17). 2017.
[15] CVE-2020-10713［
EB/OL］. http://cve. mitre. org/cgi-bin/cvename. cgi? name=CVE-2020-10713
.
[16] CVE-2018-1000205［
EB/OL］. http://cve. mitre. org/cgi-bin/cvename. cgi? name=CVE-2018-1000205
.
[17] CVE-2018-3968［EB/OL］. http://cve. mitre. org/cgi-bin/cvename. cgi?name=CVE-2018-3968
.
[18] CVE-2020-10648［
EB/OL］. http://cve. mitre. org/cgi-bin/cvename. cgi? name=CVE-2020-10648
.
[19] Bashun V, Sergeev A, Minchenkov V, et al. Too young to be secure: Analysis of UEFI threats and vulnerabilities［C］//14th Conference of Open Innovation Association FRUCT. IEEE，2013：16-24.
[20] Pankov I D, Konoplev A S, Chernov A Y, et al. Analysis of the Security of UEFI BIOS Embedded Software in Modern Intel-Based Computers［J］. Automatic Control and Computer Sciences，2019，53（8）：865-869.
[21] Iftekhar Husain M, Mandvekar L, Qiao C, et al. How to Bypass Verified Boot Security in Chromium OS［J］. arXiv，2012：arXiv: 1202. 5282.
[22] Lebedev I, Hogan K, Devadas S, et al. Invited Paper: Secure Boot and Remote Attestation in the Sanctum Processor［J］. ieee computer security foundations symposium，2018：46-60.
[23] Sherwood D, Walker J W, Walton T. Trusted boot of a virtual machine：U. S. Patent 9，721，103［P］. 2017-8-1.
[24] Wilkins R, Richardson B. UEFI secure boot in modern computer security solutions［C］//UEFI Forum. 2013.
[25] Sanwald S, Kaneti L, Stöttinger M, et al. Secure Boot Revisited［J］. 2019.
[26] Siddiqui A S, Shirley G, Bendre S, et al. Secure Design Flow of FPGA Based RISC-V Implementation［C］//2019 IEEE 4th International Verification and Security Workshop (IVSW). IEEE，2019：37-42.
[27] Liu Y, Briones J, Zhou R, et al. Study of secure boot with a FPGA-based IoT device［C］//2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS). IEEE，2017：1053-1056.
[28] Rouget P, Badrignans B, Benoit P, et al. SecBoot—lightweight secure boot mechanism for Linux-based embedded systems on FPGAs［C］// 2017 12th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC). IEEE，2017：1-5.
[29] Siddiqui A S, Nicholas G S, Joseph S R, et al. Multilayer camouflaged secure boot for SoCs［C］//2019 20th International Workshop on Microprocessor/SoC Test，Security and Verification（MTV）. IEEE，2019：56-61.
[30] Devic F, Torres L, Badrignans B, et al. Securing Boot of an Embedded Linux on FPGA［C］. ieee international symposium on parallel & distributed processing，workshops and phd forum，2011：189-195.
[31] Pocklassery G, Che W, Saqib F, et al. Self-authenticating secure boot for FPGAs［C］. hardware oriented security and trust，2018：221-226.
[32] Streit F J , Fritz F , Becher A , et al. Secure Boot from Non-Volatile Memory for Programmable SoC Architectures［J］. 2020.
[33] Hajyahya J, Wong M M, Pudi V, et al. Lightweight Secure-Boot Architecture for RISC-V System-on-Chip［C］. international symposium on quality electronic design，2019：216-223.
[34] Zhao S, Zhang Q, Hu G, et al. Providing Root of Trust for ARM TrustZone using On-Chip SRAM［C］. workshop on trustworthy embedded devices，2014：25-36.

[35] Huang, Chenlin , et al. "Research on Linux Trusted Boot Method Based on Reverse Integrity Verification. "entific Programming 2016. Pt. 1(2016):1-12.

[36] Jiang H, Chang R, Ren L, et al. Implementing a ARM-Based Secure Boot Scheme for the Isolated Execution Environment［C］. computational intelligence and security，2017：336-340.

[37] Secure the Network Infrastructure – Secure Boot Methodologies［EB/OL］. https://builders. intel. com/docs/networkbuilders/secure-the-network-infrastructure-secure-boot-methodologies. pdf
.

[38] Lai R. Amd security and server innovation［J］. UEFI PlugFest-March，2013：18-22.

[39] Arm Platform Security Architecture Trusted Boot and Firmware Update ［EB/OL］. https:// developer. arm. com/-/media/Arm% 20Developer% 20Community/PDF/PSA/DEN0072-PSA_TBFU_1. 1-BETA0. pdf
.

[40] Verified Boot | Android Open Source Project［EB/OL］. https://source. android. google. cn/security/verifiedboot
.

[41] Sanders L. Secure boot of zynq-7000 all programmable SoC［J］. Application note XAPP1175 (v1. 0)，Xilinx，2013.

[42] Weiss B. An Investigative Study on Android Verified Boot Process［J］. 2019.

[43] Khalid O, Rolfes C, Ibing A. On implementing trusted boot for embedded systems［C］//2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST). IEEE，2013：75-80.

[44] González J, Hölzl M, Riedl P, et al. A practical hardware-assisted approach to customize trusted boot for mobile devices［C］// International Conference on Information Security. Springer，Cham，2014：542-554.

[45] Cook B, Khazem K, Kroening D, et al. Model Checking Boot Code from AWS Data Centers［C］. computer aided verification，2018：467-486.

[46] Straznickas Z. Towards a Verified First-Stage Bootloader in Coq［D］. Massachusetts Institute of Technology，2020.

[47] Muduli S K, Subramanyan P, Ray S, et al. Verification of Authenticated Firmware Loaders［C］. formal methods in computer-aided design，2019：110-119.

[48] Huang B Y, Ray S, Gupta A, et al. Formal security verification of concurrent firmware in SoCs using instruction-level abstraction for hardware［C］//Proceedings of the 55th Annual Design Automation Conference. 2018：1-6.

[作者简介]

李荣泰 （1998-），男，硕士研究生。主要研究领域为嵌入式系统安全，形式化方法。

常瑞 （1981-），女，博士，浙江大学副教授。主要研究方向为计算机体系架构，嵌入式系统安全，形式化方法等。

苗新亮 （1994-），男，博士研究生。主要研究领域为计算机体系架构，形式化安全。

董卫宇（1976-），男，博士，副教授，主要研究方向为系统虚拟化，体系结构。

基于FPGA的光纤通道控制器IP设计与实现

陈艇，沈剑良，吕平，信息工程大学，郑州河南

摘　要：光纤通道（Fibre Channel，FC）协议是一种高速、高可靠、可远距离传输的通信协议，它在高性能存储、新一代航空电子系统中被广泛使用，FC控制器IP是FC交换芯片和FC端点设备的关键模块，本文在研究和分析FC协议规范的基础上，提出了一种低延迟、最高支持8.5G速率的FC协议控制IP的架构，并通过FPGA进行了设计与实现，实验结果表明，相比于传统的FC控制器，该控制器具有更高的传输速率和更低的延迟。

关键词：光纤通道、FC控制器、FPGA

Design and Implementation of Fibre Channel controller based on FPGA

——(chenting, shenjianliang, lvping)

Abstract: Fibre channel (FC) protocol is a high-speed, high reliability and long-distance transmission communication protocol. It is widely used in high-performance storage network and new generation avionics system. FC controller IP is the key module of FC switching chip and FC endpoint equipment. Based on the research and analysis of FC protocol specification, this paper proposes a FC protocol controller IP architecture with low delay and support max of 8. 5g rate. The experimental results show that compared with the traditional FC controller, the proposed FC controller has higher transmission rate and lower delay.

Key words: Fibre Channel; FC Controller; FPGA

1　引言

FC协议是有美国标准化委员会（ANSI）于1988年提出的串行高速传输总线，具备高带宽、高可靠、高效率、抗干扰性强等优点。FC网络能够提供稳定可靠的数据传输，支持点到点、仲裁环和交换网（Fabric）三种不同的拓扑，其中Fabric模式下最多支持1千6百多万个设备互连，因此可根据不同的应用场景简单方便地构建大型的通信网络[1]。由于FC网络的可靠性，FC协议也广泛应用于航空电子系统上，由此产生了FC-AE[2]和FC-AV[3]等专用的协议。

FC协议主要分成五层结构，如下图所示，各层的功能如下：

FC-0：定义了连接物理介质的界面、电缆等电气特性和参数；

FC-1：定义了物理子层编码或解码规则，在8.5G及以下速率采用了8B/10B编解码规则；

FC-2：定义了链路控制和数据传输的规则，是光纤通道的核心，主要规定了原语信号、原语序列、帧格式、流量控制和服务质量等；

FC-3：定义了一些常用服务，如数据加密和压缩。

FC-4：定义了光纤通道和上层应用之间的接口，也称为协议映射层，上层协议包括串行SCSI协议，FC-AE-ASM协议等。

光纤通道协议控制器的主要实现FC-2层的功能，它控制FC端口之间的链路建立、数据帧的传输、流量控制等操作，是FC网络设备的关键模块。

目前高性能FC交换和端点设备基本上被国外的垄断，主要用于商用的高性能存储系统当中，而在国内，浙江大学、电子科技大学、中航工业631所等高等院校和研究所也从90年代开始研究

基金项目：国家科技重大专项：（No. 2016ZX01012101）。

图1　FC协议层次结构

FC控制器的设计和应用，而他们的设计主要是面向航空电子系统的FC-AE协议产品，速率较低。本文从研究和分析协议出发，正向开展一款支持8.5G、4.25G、2.125G、1.0625G四种速率的高速FC控制器的设计与实现，为后续更高速率的FC协议控制器提供支撑。

2　FC控制器架构与设计

根据FC协议[4]的规定，FC-2层又分为FC-2M层（FC-2 Multiplexer sublevel）、FC-2P层（FC-2 Physical sublevel）、FC-2V层（FC-2 Virtual sublevel），其中FC控制器主要完成FC-2P层的功能。FC-2P主要包括端口状态机、速率自适应、缓存到缓存（Buffer-to-Buffer）的信用量控制、帧的传输与接收等[5] [6]，其中速率自适应是一个可选的功能，本文不具体展开介绍。如下图所示FC控制器的整体架构，FC-0 & FC-1可由FPGA的GTY实现，而FC-2P由接收通路和发送通路组成，其中接收通路包括传输字同步、控制符和数据分离、解扰、CRC校验、接收报文缓存等模块；发送端主要包括发送报文缓存、CRC计算、加扰、控制符生成、以及控制符和数据合并模块等。接收通路和发送通路由端口状态机和信用量控制模块相连，实现数据的正确收发。

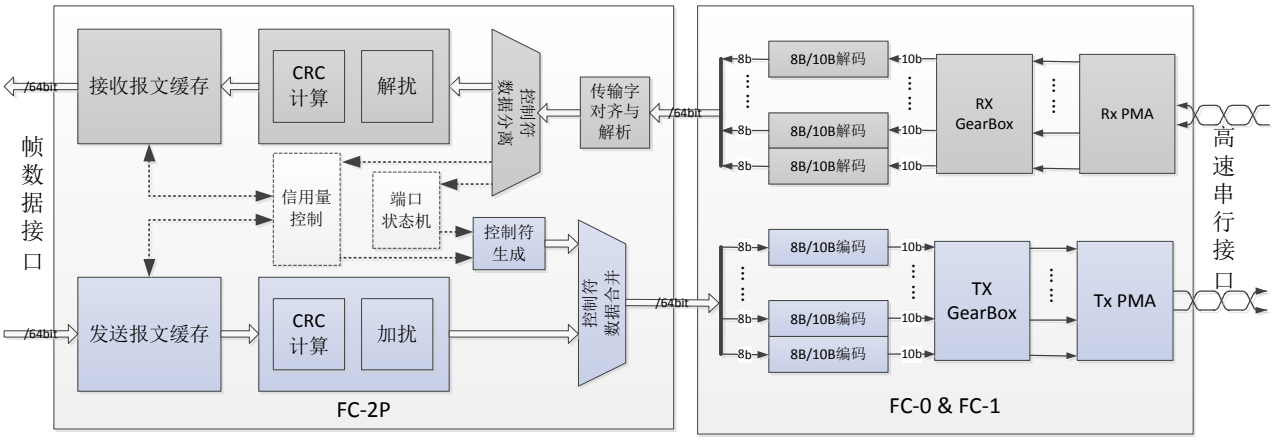


图2　FC控制器总体架构

2.1　数据通路和时钟设计考虑

本文设计的FC控制器最高支持8.5G速率，考虑到FC控制器在FPGA上实现难度，本文FC控制器采用64bit位宽数据通路，那么FC控制器主频为106.25MHz。数据通路上，本文对FC控制器的时序和功能做了精细的划分，比如传输字对齐与解析模块2个时钟周期，控制符与数据分离1个时钟周期，解扰和CRC计算总共4个时钟周期，接收报文缓存在cut-through模式下写入和读出最小只需3个时钟周期，因此接收通路10个时钟周期；而同理，发送通路8个时钟周期，本文FC控制器收发通路环回最小延迟为18个时钟周期。

2.2　端口状态机

FC协议中规定了FC端口状态机的状态数量及其之间的转换条件，如图3所示，状态机的状态总共分成4类，包括建链（Active），链路恢复（Link recovery）、链路失败（Link Failure）和离线（Offline），4类状态又由9个子状态组成。图3表格中第三行表示在每一个子状态下链路发送的原语序列，第一列表示链路上接收的原语序列或者出现的事件，而表格的中间表示状态机下一个状态。从图中可以看出，FC协议对端口状态机做了

	Active	Link Recovery			Link Failure		Offline		
Current State	**AC**	**LR1**	**LR2**	**LR3**	**LF1**	**LF2**	**OL1**	**OL2**	**OL3**
Primitive Sequence transmitted while in state	Fill Word	LR	LRR	Idle	OLS	NOS	OLS	LR	NOS
Input Event:	**Next State:**								
L >> LR	LR2	LR2	LR2	LR2	LR2	LF2	LR2	LR2	LF2
L >> LRR	LR3	LR3	LR3	LR3	LF1	LF2	OL1	LR3	LF2
L >> Idles	AC	LR1	AC	AC	LF1	LF2	OL1	OL2	OL3
L >> OLS	OL2	OL2	OL2	OL2	OL2	OL2	OL2	OL2	OL2
L > > NOS	LF1	LF1	LF1	LF1	LF1	LF1	LF1	LF1	LF1
Loss-of-Signal	LF2	LF2	LF2	LF2	LF2	LF2	OL3	OL3	OL3
Loss of Sync >(R_T_TOV)	LF2	LF2	LF2	LF2	LF2	LF2	OL3	OL3	OL3
Event time-out (R_T_TOV)	N/A	LF2	LF2	LF2	LF2	N/A	OL3	OL3	N/A
Link time-out (E_D_TOV)	LR1	LR1	LR1	LR1	LR1	LR1	LR1	LR1	LR1

图3　FC端口状态机状态图[4]

严格规定，设计中只需确保按其执行即可。

端口只有进入Active状态下才能正常的收发数据帧，而在收发数据帧的同时，需要经常发送维护链路的原始信号，比如信用量控制的R_RDY、BB_SCr、BB_SCs、IDLE等，因此可将端口状态机分成两个主状态：建链状态和非建链状态，在非建链状态下，完全遵循图3的状态进行转换，而在建链状态下，除了IDLE序列之外，其它原始信号的传输优先级大于数据帧，但是一旦数据帧在传输过程中不能插入其它原始信号或者IDLE，因此在空闲的状态下，原始信号和数据帧的优先级如下： BB_SCr>BB_SCs> R_RDY>数据帧传输字>IDLE。

2.3　流量控制和恢复

FC协议支持缓存到缓存（buffer-to-buffer）的流控和端到端的流控，其中缓存到缓存的流控负责两个互连端口之间的流量控制，确保数据帧在相邻两个端口之间的可靠传输，缓存到缓存的流控由控制器实现，而端到端的流控一般由端口协议栈实现。FC协议采用基于buffer-to-buffer信用量（BB_Credit）的方式进行流量控制[7]，两个相互连接的端口在初始化时默认信用量为1，即发送一个报文，需要等待对方端口确认之后才能发送下一个。相互连接的端口信用量通过协商确认[8]，如果对方的信用量为N，那么本地端口可在不用等待到对方的确认连续发送N个数据帧，当对方端口的接收报文缓存释放一个报文的存储空间，将发送一个报文确认信号R_RDY信号给本地端口，本地端口收到R_RDY信号之后可再发送一个数据帧。如图4所示，FC控制器中设置一个了BB_Credit_CNT变量，表示已发送待确认的数据帧个数，当端口每发送一个数据帧，BB_Credit_CNT加1，而端口收到一个R_RDY时，BB_Credit_CNT减1，当BB_Credit_CNT等于BB_Credit时，那么将停止发送数据帧。

此外，FC协议支持信用量恢复机制，FC控制器在发送一定数量（$2^{BB\_SC\_N}$，其中BB_SC_N=log（BB_Credit））的数据帧和R_RDY之后发送BB_SCs和SS_SCr原语信号进行确认，而端口也会对接收的数据帧和R_RDY进行统计，当接收到BB_SCs原语信号时，如果本地接收数据帧统计小于$2^{BB_SC_N}$，表示丢失了部分帧，那么通过发送一定数量的R_DY进行补偿丢失的信用量；当接收到SS_SCr原语信号时，如果本地接收R_RDY数量小于$2^{BB_SC_N}$，表示丢失了部分R_RDY，此时通过减少BB_Credit_CNT的值进行补偿。

3　实验与结果

本文通过Xilinx公司的FPGA对FC控制器进行实现和验证，其中FC-1和FC-0层的功能由FPGA的GTY实现，并且通过与JDSU的FC测试仪进行对接测试，如下图所示，通过FC测试仪在不同频点下发送各种测试激励，测试FC控制器协议兼容性和功能性能。表1显示了本文FC控制器与

Xilinx 公司的 FC IP 核主要功能性能对比，本文的 FC 控制器数据位宽是 Xilinx 公司 IP 的两倍，最高支持 8.5G，而 Xilinx 公的 IP 最高支持 4.25G，因此控制器的主频相同，这样有利于功耗和延迟的控制，但是总资源大概增加了 50%。本文 FC 控制器在设计上对功能和流水线进行了精细划分，其收发通路最小延迟只有 18 个时钟周期，而 Xilinx 公司的 IP 需要 64 个时钟周期。

图 4 FC 控制器测试示意图

实验结果表明，本文设计的 FC 控制器相比与 Xilinx 的 IP 核，在主频相同的情况下，最高速率提升了一倍，延迟只有它的 40%，而总资源消耗增加不到一倍。

4　结束语

本文通过对 FC 协议的研究和分析，提出了一种低延迟的 FC 控制器架构，通过对数据主通路功能和流水线进行了精细的划分，并通过 FPGA 进行了实现和验证，符合协议的兼容性，并且部分性能优于同类产品，本文的 FC 协议控制器可用于 FC 端点设备和 FC 交换设备，并且其低延迟的架构对其它协议的控制器设计具有一定的指导意义。

表 1　功性能比较

功性能项目	本文 FC 控制器	Xilinx FC IP 核[9]
主频时钟（MHz）	106.25	106.25
数据位宽	64bit	32bit
支持速率	1.0625G/2.125G/4.5G/8.5G	1.0625G/2.125G/4.5G
最大信用量	32	NA
控制器传输延迟（不包括 FC-1 和 FC-2 层）	18cycle	45cycle
资源消耗 Slices/LUT/FFs	1855/3601/4524	1325/2542/1890

参考文献:

[1] 田泽, 徐文龙, 许恒, et al. FC光纤通道技术研究综述[J]. 电子技术应用，2016，42(009)：143-146.

[2] INCITSANSI. Fibre Channel-Avionics Environment. (FC-AE) . Rev 2. 6，2002-2-7

[3] INCITSANSI. Fibre Channel-Avionics Environment. Upper Layer Protocal MIL-STD-1553B Notice2（FC-AE-1553）. Rev 0. 95，2006-12-4

[4] INCITSANSI. Fibre Channel Framing and Signaling (FC-FS-4). Rev 1. 80，2005-6-6

[5] 刘鑫,陆文娟,光纤通道在航空电子环境的应用及关键技术研究．系统与应用，2006，30(6)：55-58

[6] 张延年．光纤通道协议处理模块的设计与验证[D]. 西安电子科技大学，2016. 75.

[7] LIN Qiang, XIONG, Hua-gang, ZHANG, Qi-shan. Credit Determination of Fibre Channel in Avionics Environment[J]. 中国航空学报(英文版)，2007，(3).

[8] 乔家庆,冯收,凤雷,等．光纤交换网络中设备登录服务的研究与实现[J]. 测试技术学报，2012，(6）. doi: 10. 3969/j. issn. 1671-7449. 2012. 06. 004.

[9] Xilinx, LogiCORE IP Fibre Channel v3. 5, User Guide Guide DS270, April 19，2010

[作者简介]

陈艇（1984-），男，博士，助理研究员，主要研究方向：大规模集成电路设计、微处理器体系结构设计及高速数字信号处理。

沈剑良（1982-）男，博士，副研究员，主要研究方向：新一代网络信息系统架构设计、大规模集成电路设计。

吕平（1977-）女，博士，副研究员，主要研究方向：新型体系结构、新一代信息通信技术。。

拟态路由器控制面限流机制设计

张进[1]，朱绪全[1]，杨盾[1]，江逸茗[2]
[1]网络通信与安全紫金山实验室；
[2]中国人民解放军战略支援部队信息工程大学

摘　要：路由器控制面限流机制的作用是保证上送至控制面的流量不超出控制面的处理能力，并防止少数大流量的数据流挤占上报带宽。本文给出了一种安全、易实现的拟态路由器控制面限流机制的设计方法。该方法的特点是将粗粒度的QoS机制和细粒度的异常流检测机制相结合，一方面，能够保证整体上送流量不超出控制面的处理能力，同时，也能检测并阻止针对控制面的DoS攻击流。采用真实网络流量，通过仿真实验，证明了该方法的有效性。

关键词：网络安全、路由器、报文抽样

Designing Control Plane Rate-Limiting Mechanism for Mimic Router

Zhang Jin[1*], zhu Xuquan[1], Yang Dun[1], Jiang Yiming[2]
1.Purple Mountain Laboratory;
2.Information Engineering University

Abstract: 路由器控制面限流机制的作用是保证上送至控制面的流量不超出控制面的处理能力,并防止少数大流量的数据流挤占上报带宽。本文给出了一种安全、易实现的拟态路由器控制面限流机制的设计方法。该方法的特点是将粗粒度的QoS机制和细粒度的异常流检测机制相结合,一方面,能够保证整体上送流量不超出控制面的处理能力,同时,也能检测并阻止针对控制面的DoS攻击流。采用真实网络流量,通过仿真实验,证明了该方法的有效性。Control plane rate-limiting of a router is to ensure that the control plane will not being overwhelmed by the traffic sent to it, and that the reporting bandwidth will not being monopolized by some heavy-hitter flows. We present the design of a secure and feasible control plane rate-limiting mechanism for mimic router. The new method combines coarse grained QoS with fine grained anomaly flow detection so as to guarantee the whole traffic sent to control plane not exceeding its processing capability, and DoS attack targeted to the control plane being detected and eliminated. Simulation results with real world network traffic show the effectiveness of the proposed method.

Key words: Network Security; Router; Packet Sampling

1　引言

从系统结构上看，路由器可以自下而上划分为三个功能层面：数据面（Data Plane，DP）、控制面（Control Plane，CP）和管理面（Management Plane，MP）。数据面的基本功能是根据转发表，进行快速的报文转发，并将需要控制面和管理面处理的报文上送。控制面的主要功能是运行

各类路由协议，计算生成供数据面使用的转发表。此外，在具体实现过程中，某些处理流程复杂的转发报文，例如带选项的IP报文，也会上送控制面进行处理，也可以将此类处理通道称为数据面的慢通道（Slow Path）。管理面的基本功能是向用户提供配置协议和配置接口，允许用户对路由器进行管理控制。

在具体实现时，传统数据面通常采用专用集成电路（ASIC）芯片或者网络处理器（NP）实现线速报文转发。近年来，随着网络功能虚拟化的发展，数据面也可以基于通用高性能多核处理器和专门优化的软件实现。控制面（包括数据面的慢通道）和管理面通常采用通用处理器和系统软件实现，不具备（正常情况也不需要）线速的报文处理能力。由于数据面和管理面的功能对于路由器而言极为重要，但处理能力有限，因此容易成为拒绝服务攻击的目标。因此，必须对从数据面上送的报文进行速率限制，保障路由器的正常运行。

本文给出了一种拟态路由器控制面流量限制方法。事实上，本方法保护的对象是不仅包括控制面，还包括数据面慢通道以及管理面，本文统称其为“控制面”。该方法通过对上报拟态路由器控制面的报文进行分类和流量限制，以实现以下几点目标：

1）保证总的上报流量不超出控制面的处理能力；

2）“重要”的报文能够优先上送；

3）能够检测并限制拒绝服务攻击。

本文提出的限流方法的基本策略是将粗粒度和细粒度的流量控制策略相结合，能够实现上述三点限流目标，且简单易实现。本文首先描述了限流机制的设计方法，其次通过仿真，对该方法的有效性和实现代价进行了分析，仿真分析的结果表明了该方法的有效性。

2　相关研究

RFC 6192给出了一种保护路由器控制面的方法框架［1］，其基本原理是对报文进行分类，并为不同类别的报文定义不同的控制策略，如上送、丢弃或限速（rate-limit）。具体而言，RFC 6192给出了一个流量类别白名单，该白名单明确定义了若干类流量的处理规则；对于未在白名单上的报文类别，则进行总体流量限速。当前商用路由器的CP限流策略基本都基于RFC 6192方法框架。例如，CISCO［2］和Mellanox［3］的控制面策略（Control Plane Guide， CoPP），通过定义一组ACL规则，同时结合QoS机制，对特定类型的报文进行丢弃或者限速。再如6wind的控制面限流机制［4］，本身就是数据面快速处理通道（Fast Path）QoS机制的一部分，其实现原理是首先根据预先定义的流量阈值，对报文进行三色标记［5］，依次是绿、黄、红，如图1所示；同时，依据报文协议类别，将报文划分为低、中、高三种优先级。高优先级类别的所有报文都上送CP；中优先级类别只有绿色和黄色报文上送CP，红色报文丢弃；低优先级类别只有绿色报文上送CP，其他丢弃。

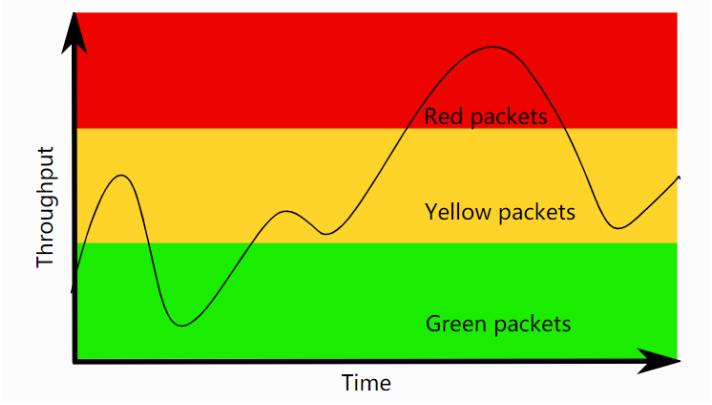


图1　根据流量大小对报文进行三色标记

上述几种方法的实际效果取决于流量控制的粒度。粗粒度的流量控制策略无法获得完善的限制效果，但是精细粒度的流量控制策略实现代价高、开销大。例如，若仅仅对上送CP的所有OSPF协议报文进行整体限速，则在某台设备向路由器恶意发送大量的OSPF报文的情况下，恶意发送的报文会挤占正常OSPF报文的上报带宽，导致路由器无法和邻居进行正常的OSPF协议报文交互。若根据源IP区分数据流，对每条流的OSPF报文进行限速，则需要维护每条数据流的状态，同时在上报报文处理流程中，需要增加逐报文的状态查找，实现代价较高。特别是在高速骨干网，或者设备面临DDoS攻击时，由于并发数据流数目多，维护每条数据流的状态将极大的消耗设备的计算资源。

3　总体方案

本文给出了一种简单、高效的路由器控制面限流机制的设计方法。其基本的设计思想是将粗粒度和细粒度的流量控制机制相结合，一方面，对上报CP的报文进行分类限速；另一方面，对各类报文进行异常检测，并对异常流实施限速。该机制的原理框图如图2所示。

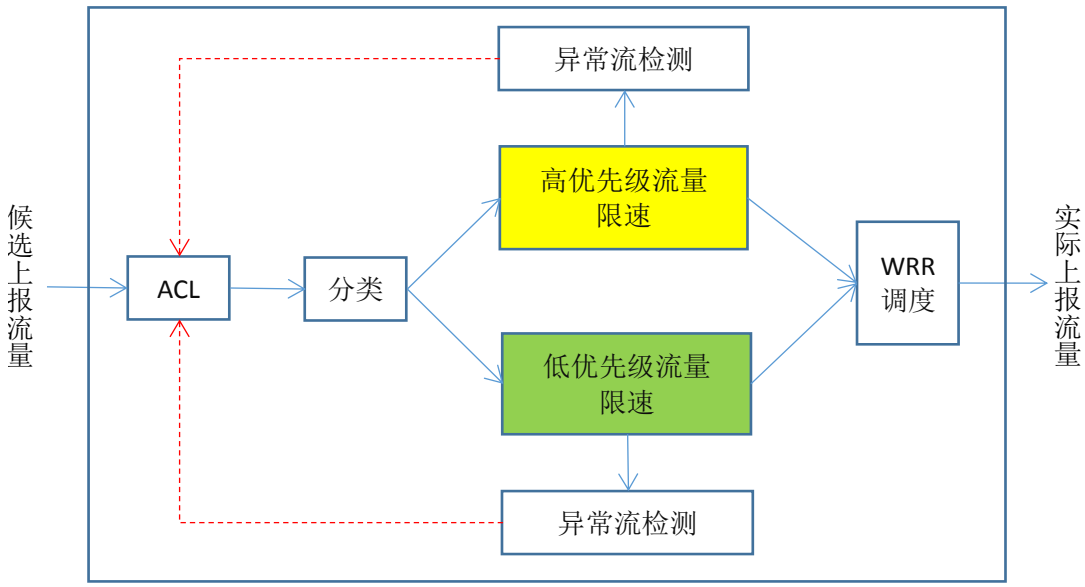


图2　CP限流机制原理框图

需要上报CP的报文首先通过ACL模块进行检测，根据规则匹配的结果，确定报文的处理动作：丢弃或者通过。ACL模块从逻辑上看是一个Match-Action列表，并能够支持对特定数据流的限速。通过ACL检测的报文，被分为高、低两种优先级，具体的分类方法由用户根据实际使用需求进行定义，例如，将路由协议报文、目的IP地址是本路由器的报文归入高优先级，其他报文归入低优先级。分类后的报文送入限速模块，分别进行限速，高、低优先级两类流量的速率上限可以独立定义。调度模块根据两类流量的比例关系进行加权轮转（Weighted Round Robin， WRR）调度，将总的上报带宽在两类流量间进行分配。限速模块的入口流量同时送异常流检测模块，该模块的功能是检测上报数据流中流量较大的流，并将异常流的流标识反馈给ACL模块，由ACL模块对异常流进行限速。数据流根据<源IP，源端口>二元组进行区分，即

$f = <src\_ip,\ src\_port>$

上述CP限速机制不依赖于QoS机制的支持，实现简单；同时，该机制也能够检测并阻止针对CP的DoS攻击，能够实现本文开篇提到的所有三给限流目标。总体而言，上述CP限流机制比现有的基于RFC 6192的方法框架更易于实现，更安全。

4 异常流检测

异常流检测模块的基本功能是识别出流量较大的数据流（也称为Heavy-Hitter［6］［7］［8］），并将其流标识二元组上报给ACL模块。已有的研究工作给出了多种Heavy-Hitter识别算法，例如Sample and Hold［9］，Count-min Sketch［10］，HashPipe［6］，Gated Sketch［8］。上述4种方法虽然具有较高的识别精度，但是，由于需要线速的逐包处理，有些还需要硬件的可重配置匹配表（Reconfigurable Match Table，RMT）表的支持［6］［8］，实现难度较大，代价较高。为了减小实现难度，降低实现代价，本方案采用抽样的方式进行Heavy-Hitter识别。抽样识别的基本方法是仅仅处理被抽取到的报文，忽略未被抽取的报文。当前成熟的流量测量技术如Netflow［11］和sFlow［12］，均采用抽样方法进行数据流的测量统计。

抽样检测的基本原理是对于上报CP的所有报文，每N个选取一个，提取其数据流标识，假设为f。接着，以f为关键字，查找流状态表T，若命中，则更新对应的表项$T[f]$；否则，在T中为数据流f创建新的表项。数据流表T的各个表项由下列四元组组成

T［f］=<f，t_create，t_update，c>

其中，f为该数据流的流标识，同时也是数据流表T的查表关键字；*t_create*为该表项的创建时间；*t_update*为该表项的最近一次更新时间；c为数据流f共被抽取到的报文数。通过估算f的流量，并和门限值进行比较，可以确定是否需要对f进行流量限制。假设流量门限值为H，抽样间隔为N，当前时刻为*t_current*，则如果$cN/$（*t_current* - *t_create*）$>H$，则生成一条ACL规则$<f, H>$，对f的流量进行限制。被抽取到的报文P的处理流程如图3所示：

```
1.      从 P 中提取其流标识 f, f 为 P 的 IP 首部<源 IP 地址, 源端口>二元组;
2.  if (T[f] == NULL)
3.      在 T 中新建表项 T[f] ← <f, t_create, t_update, c>;
4.      t_create ← t_current;
5.      t_update ← t_current;
6.      c ← 1;
7.  else
8.      t_update ← t_current;
9.      c ← c + 1;
10.     if cN / (t_current - t_create) > H then    //H 为流量门限, N 为抽样间隔
11.         新增一条 ACL 规则<f, H>;
12.     end if;
13. end if;
```

图3 被抽取到报文的处理流程

为了避免已经结束的数据流长期占用流状态表的空间，需要周期性的对流状态表进行扫描。若发现某个表项的更新时间距离当前时间已经超出了超时门限，则将该表项删除；同时，查询ACL规则表，若存在和被删除表项相对应的规则，也将此规则删除。数据流的超时门限可以参考TCP连接的超时门限进行设置，在Linux系统中，该门限值默认为127秒，并可以根据需要进行修改。

数据流状态表T采用哈希表实现。传统的哈希表采用链表或开放定址等方式处理冲突，当表负载率较高时，其最坏条件下的查找时间不确定。相比较而言，布谷鸟哈希表（Cuckoo Hashing）［13］在最坏条件下的查找时间确定，当前也被DPDK所使用。具体实现时，可以直接采用DPDK的布谷鸟哈希表函数库［14］来实现数据流状态

表T。

5 流量限速

流量限速采用令牌桶实现，其基本原理如图4所示。以恒定的速率c向令牌桶中放入令牌，令牌桶的最大容量和缓存队列的容量相等，假设为B，当桶满之后，再生成的令牌则被丢弃。缓存队列若有报文需要输出时，首先查询令牌桶中的令牌数量。若剩余令牌量大于待输出报文的长度，假设为L，则将此报文输出，同时将桶中令牌数量减小L；否则，不输出报文，继续等待。初始时刻，桶中令牌数量设置为B。通过设置不同的c和B的参数取值，则能够实现对流量平均速率和突发速率的限制。

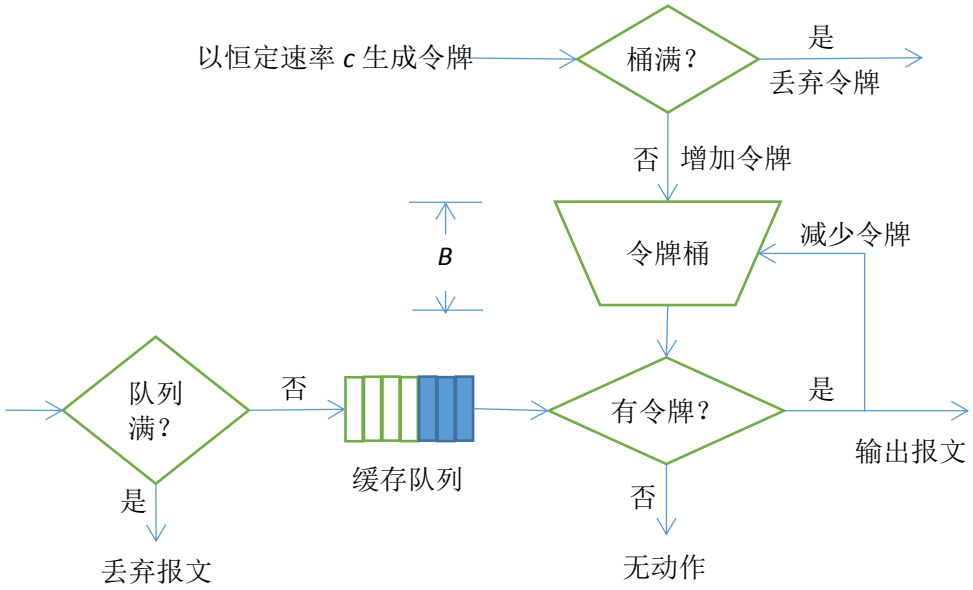


图4 采用令牌桶进行流量限速的原理示意

6 仿真分析

选择合理的报文抽样概率，对限流机制的限流效果和实现代价有着较大影响。直观看来，抽样概率越大，则识别的准确率越高。然而，较大的抽样概率一方面会增大CPU的处理负荷，另一方面也会导致较多流量没有超出门限的正常流进入数据流状态表，导致流表膨胀，提高实现代价。本节采用的仿真的方法，分析了抽样概率和异常流识别的准确性、所需消耗的流表空间之间的关系。

衡量异常流识别准确性的指标是误报率（False Positive Rate，FPR）和漏报率（False Negative Rate，FNR）。假设异常流的流量阈值为H，也即，若数据流f的流量大于H，则f是异常流，否则f是正常流。FPR和FNR的定义如下：

$$FPR = \frac{\text{估算流量大于H,且实际流量不大于H的数据流数目}}{\text{实际流量不大于H的数据流数目}}$$

$$FNR = \frac{\text{估算流量不大于H,且实际流量大于H的数据流数目}}{\text{实际流量大于H的数据流数目}}$$

从实际影响看，误报会导致数据流表和ACL规则表膨胀，增大实现代价，但是不会影响限流效果。反之，漏报则会影响限流效果，导致图2中高优先级或（和）低优先级的流量无法抵御恶意用户的拒绝服务攻击，影响正常的协议报文交互，从而对路由器的整体功能产生负面影响。在实际使用中，正确的做法是允许相对较高的FPR，保证较低的FNR。

考虑到部分被抽样到进入流表的、且实际流量小于H的数据流，其流量估算值也并不超出H。因此，FPR并不能准确的反应实际所需的数据流表空间。定义数据流抽样率（Flow Sample Rate，

FSR）为

$$FSR = \frac{被抽样到的数据流数目}{总的数据流数目}$$

对于任意数据流，只要有一个属于该数据流的报文被抽样到，则称该数据流被抽样到。数据流表空间的需求可以通过FSR来衡量。

实验所用的数据来源于NLANR公布的在OC-192和OC-48骨干网链路上采集的流量数据［15］。本文截取了Abilene-III数据集中20040601-193121-0数据文件的第1分钟的流量数据（下文称为Trace1）以及Abilene-I数据集中IPLS-KSCY-20020814-090000数据文件的第1分钟的流量数据（下文称为Trace2）作为实验数据源。Trace1和Trace2的详细信息如表1所示。

表1　实验所采用的骨干网流量数据的详细信息

名称	链路速率	持续时间	总报文数	总流数	最大流流量（报文）
Trace 1	OC−192	60s	6,410,992	172,004	459,668
Trace 2	OC−48	60s	3,541,483	45,807	112,008

图5给出了实验所采用的流量数据中，各条数据流的流量分布情况，各数据流按照流量值降序排序。注意图5为对数坐标。从图5可见，绝大部分数据流的流量较小。事实上，现有的研究表明，网络数据流的流量服从帕累托分布［9］，其直观表现是大部分数据流的流量较小，少部分的Heavy-hitter数据流的流量占据了链路流量的绝大部分比例。图5所示的结果和现有发现一致。

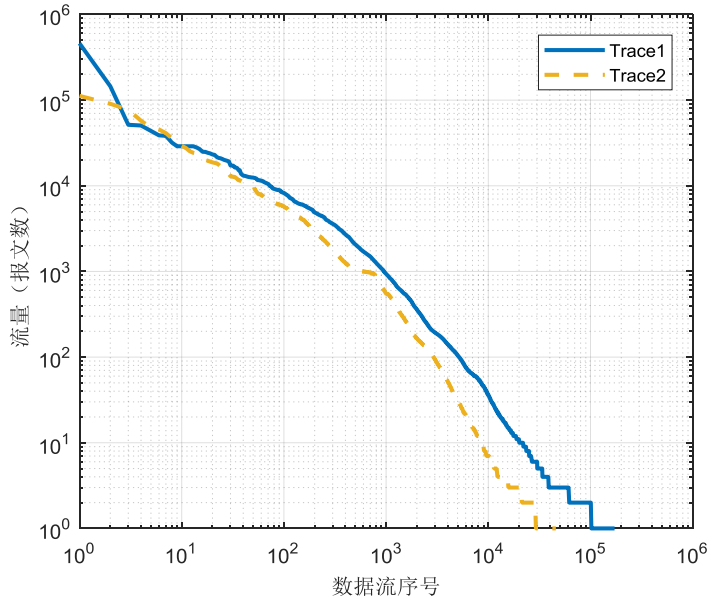


图5　实验数据的流长分布

表2和表3分别给出了采用Trace1和Trace2进行仿真实验时，不同的异常流门限（TH）和抽样概率（SR）下的漏报概率（FNR）和数据流抽样率（FSR）。表2和表3中的TH值的单位为报文每秒（Packet Per Second，PPS），例如，TH为50时，假设平均报文长度为512字节，则对应的数据流速率为200Kbps。实验结果表明，不同的异常流门限下，报文抽样概率取0.05~0.01时，可以在漏报率和流抽样率之间取得较好的折衷。此时，对应的漏报率约为5%~10%，所需的数据流表空间约为总数据流数目的10%~20%。需要指出的是，这里的漏报率是统计平均值。对于单条数据流而言，漏报率和其流量大小负相关。在实际使用中，可以将异常流门限设置的较为保守。这样，即使存在一定的漏报率，如果某条数据流的流量明显大于异常流门限，则该数据流仍有较高的概率被检测到。

7　结论

路由器控制面限流机制对于保障路由器的安全稳定运行有着重要的意义。本文给出了一种安全有效且简单易实现的控制面限流机制。该方法

表2　采用Trace1的仿真实验结果

TH (PPS)	SR	FNR	FSR
50	0.1	0.0279	0.2700
50	0.05	0.0371	0.1792
50	0.01	0.0816	0.0677
50	0.005	0.1171	0.0437
50	0.001	0.2620	0.0151
100	0.1	0.0438	0.2705
100	0.05	0.0546	0.1796
100	0.01	0.0931	0.0676
100	0.005	0.1141	0.0438
100	0.001	0.2154	0.0152
200	0.1	0.0333	0.2696
200	0.05	0.0389	0.1794
200	0.01	0.0917	0.0678
200	0.005	0.1139	0.0437
200	0.001	0.1870	0.0151
400	0.1	0.0222	0.2701
400	0.05	0.0389	0.1794
400	0.01	0.0750	0.0676
400	0.005	0.0700	0.0437
400	0.001	0.1744	0.0151

表3　采用Trace2的仿真实验结果

TH (PPS)	SR	FNR	FSR
50	0.1	0.0196	0.3322
50	0.05	0.0243	0.2372
50	0.01	0.0550	0.1074
50	0.005	0.0876	0.0747
50	0.001	0.2219	0.0297
100	0.1	0.0424	0.3320
100	0.05	0.0538	0.2368
100	0.01	0.0929	0.1072
100	0.005	0.1159	0.0750
100	0.001	0.2157	0.0296
200	0.1	0.0271	0.3323
200	0.05	0.0314	0.2373
200	0.01	0.0586	0.1074
200	0.005	0.0771	0.0746
200	0.001	0.1543	0.0295
400	0.1	0.0231	0.3325
400	0.05	0.0231	0.2372
400	0.01	0.0462	0.1074
400	0.005	0.0600	0.0748
400	0.001	0.1338	0.0296

的特点是将粗粒度的QoS机制和细粒度的异常流检测机制相结合，一方面，能够保证整体上送流量不超出控制面的处理能力，同时，也能检测并阻止针对控制面的DoS攻击流。仿真实验证明了该方法的有效性。下一步工作是将该机制进行实际实现并测试。

参考文献:

[1] RFC 6192: Protecting the Router Control Plane.

[2] CISCO. Cisco Control Plane Policing Feature Guide. https://www.cisco. com/c/dam/en/us/td/docs/switches/lan/catalyst6500/ios/15-4SY/cisco-copp-feature-guide. pdf

[3] Mellanox. Control Plane Policing. https://docs. mellanox. com/display/Onyxv390608/Control+Plane+Policing

[4] 6wind. Fast Path QoS - Exception Rate Limitation Overview. https://doc. 6wind. com/6windgate-5/latest/mds/fast-path/qos-erl/overview. html

[5] RFC 4115. A Differentiated Service Two-Rate, Three-Color Marker with Efficient Handling of in-Profile Traffic.

[6] SivaramanVibhaalakshmi, NarayanaSrinivas, RottenstreichOri, MuthukrishnanS. , and RexfordJennifer. 2017. Heavy-Hitter Detection Entirely in the Data Plane. In Proceedings of the Symposium on SDN Research (SOSR '17). Association for Computing Machinery, New York, NY，USA，164 – 176. DOI:https://doi. org/10. 1145/3050220. 3063772

[7] HarrisonRob, Qizhe Cai, GuptaArpit, and RexfordJennifer. 2018. Network-Wide Heavy Hitter Detection with Commodity Switches. In Proceedings of the Symposium on SDN Research (SOSR '18). Association for Computing Machinery, New York, NY, USA, Article 8，1 – 7. DOI:https://doi. org/10. 1145/3185467. 3185476

[8] TurkovicBelma, OostenbrinkJorik, KuipersFernando. Detecting Heavy Hitters in the Data-plane. https://arxiv. org/abs/1902. 06993

[9] EstanC. and VargheseG. . New directions in trac measurement and accounting. ACM Trans. Computer Systems，21(3)，2003.

[10] CormodeG. and MuthukrishnanS. . An improved data stream summary: The count-min sketch and its applications. Journal of Algorithms，55(1)：58 – 75, 2005.

[11] NetworksCisco. Netow. http://www. cisco. com/c/en/us/products/ios-nx-os-software/ios-netow/index. html.

[12] SFlow. http://sow. org/.

[13] Pagh R, Rodler F F. Cuckoo hashing[J]. Journal of Algorithms，2004，51(2)：122-144.

[14] DPDK Hash Library. https://doc. dpdk. org/guides/prog_guide/hash_lib. html

[15] NLANR. Abilene-I data set. http://pma. nlanr. net/Traces/long/ipls1. html

拟态构造蜜罐研究

胡先君[1]，王涵[1]，卜佑军[2]
[1]网络通信与安全紫金山实验室；
[2]中国人民解放军战略支援部队信息工程大学

摘　要：当前主动防御是扭转网络空间攻防不对称现状的关键技术，其中蜜罐作为主动防御技术的一种，部署在内网中吸引攻击者的攻击，消耗攻击者的精力，发现攻击者的攻击手段和攻击意图。但同时蜜罐自身如果存在安全漏洞，那么蜜罐就成为攻击者攻击内网的入口，对内网安全产生危害。本文针对蜜罐系统中的虚拟机逃逸场景，基于拟态防御思想，提出具有内生安全的拟态构造蜜罐，以防御虚拟机逃逸攻击的发生。拟态构造蜜罐通过底层虚拟化平台的异构来抵御虚拟化平台的逃逸漏洞，并且此异构对攻击者无感，利于完整收集攻击者的攻击行为。最后，通过分析讨论与其他解决方案的差异来说明本方案中提出的拟态构造蜜罐的优势。
关键词：蜜罐、拟态防御、异构执行体

Study on Mimic Structure Honeypot

Hu Xianjun[1], Wang Han[1], Bu Youjun[2]
1.Purple Mountain Laboratories;
2.Information Engineering University

Abstract: Nowadays, the active defense is vital to reverse the current attack-defense asymmetry in cyberspace, and the honeypot, as a kind of active defense technology, is deployed in the intranet to attract the attacker's attack, consume the attacker's energy, and find out the attacker's means and intention. Meanwhile, if there is a vulnerability in the honeypot, then the honeypot will become the entrance of the attacker to attack the internal network, which will harm the security of the internal network. Aiming at the scene of virtual machine escaping in the honeypot system, in this paper, we proposed an endogenous mimic structure honeypot based on the idea of mimic defense to defend it. The mimic structure honeypot can resist the escape vulnerability of the virtualization platform through the heterogeneity of the underlying virtualization platform, and this heterogeneity is not perceived by the attacker, which is conducive to collect the attacker's behavior. Finally, through analyzing and discussing the differences with other solutions, we described the advantages of the proposed mimic structure honeypot.
Key words: honeypot; mimic defense; heterogeneous executor

1　引言

当前互联网中的网络安全威胁问题层出不穷，在这种攻防环境下，防守方任何一个微小的疏忽或漏洞，都可能成为攻击者攻破一个大型系统的关键。因此防守方必须做到万无一失才能确保系统安全稳定运行。但网络中存在着大量的未知漏洞、未知威胁，防守方对此知之甚少，而系统却

完全暴露在攻击者面前。在这种攻防博弈严重不对等的情况下，需要改变传统的被动防御为主动防御，在攻击发生前，提高攻击难度，在攻击发生时，主动隔离、引导攻击流量并进行跟踪分析；在攻击发生后，能够溯源和还原攻击过程。

蜜罐作为一种主动防御手段，其作为一种网络安全资源，其价值在于被探测、攻击和攻陷。在真实的网络环境中部署蜜罐系统，迷惑干扰攻击者的探测，隐藏真实的网络资产，对攻击行为进行捕获分析，掌握攻击手段和攻击意图，达到扭转攻防博弈不对称的局面。蜜罐技术的使用最初是在Clifford Stoll的小说《The Cukoo’s Egg》[1]中出现，用于追踪商业间谍。之后Fred Cohen在文献[2]中介绍利用欺骗方式来保护信息是可行的，分析基于欺骗技术背后的理论问题，并发布了采用欺骗技术的防御工具DTK（Deception Toolkit）[3]，DTK采用软件模拟，对扫描探测攻击有较好效果，并且不会被攻击者真正攻陷，但此工具易于被有经验的攻击者识别。1999年，Lance Spitzne等发起成立了非盈利性的研究组织“蜜网项目组（The Honeynet Project）”[4]，该组织致力于网络安全研究，通过蜜网技术改善早期蜜罐技术的缺陷，构建高度可控真实的诱骗环境。随后在2003年，Lance Spitzne提出了蜜场（honeyfarm）[5]和蜜标（honeytoken）[6]的概念，蜜场是为了解决在一个分布式环境中部署蜜网的困难，通过流量重定向将非法流量牵引到一个集中部署蜜罐的网络中，而蜜标是一种引诱攻击者使用的虚假信息资源。2005年，Niel Provos提出了一种虚拟蜜罐框架Honeyd[7]，可以在不同的网络层仿真不同的计算机系统，用于解决在不同的操作系统和硬件上部署物理蜜罐需要大量时间和经验的困难。在同一年，蜜网项目组发布了第三代蜜网，并开发除了kanga分布式蜜网。2006年，诸葛建伟等人发布了一个基于高交互蜜罐的恶意代码捕获工具HoneyBow[8]，采用真实系统进行部署，捕获网络中传播的恶意代码。随着云计算的发展，在2010年，蜜网项目组提出了HoneyCloud的概念[10]。2012年，石乐义等人受生物界中生物保护色和警戒色现象的启发，提出拟态式蜜罐（mimicry honeypot）概念[11]，其在传统蜜罐的基础上增加保护色和警戒色，保护色对部署网络环境的特征进行模拟，迷惑攻击者无法识别出其为蜜罐而对其进行攻击，而警戒色是模仿蜜罐特征，使攻击者误认为是蜜罐而放弃攻击，本文提出的拟态构造蜜罐与此概念有所区别。2016年，Saadi等人提出一种云基础设施架构[12]，融合蜜罐、入侵检测等技术用于解决系统在入侵检测上的局限性，在云环境中增加入侵检测系统的知识，提高入侵检测系统的准确率。2018年，贾召鹏等人针对Web攻击威胁中攻击应用不在蜜罐中无法捕获攻击的情况，提出蜜罐簇的概念并实现原型系统ArkHoney[13]，将多个应用蜜罐组合协同进行攻击捕获。2019年，廉哲等人针对蜜网中流量控制不便，动态调整困难的问题，提出基于SDN技术的虚拟蜜网[14]，其中使用了拟态防御的思想，构建多个相同功能，不同实现方式的冗余蜜罐，当攻击者攻击真实业务系统时，可以使用流量重定向将攻击流量牵引冗余蜜罐中，并可伪随机的调整蜜网，破坏攻击链，阻碍攻击进行，此方案中的拟态防御思想与本文的拟态防御思想接近，但其方案中动态调整蜜网，使攻击者攻击无法进行，那么也就无法完整的收集攻击者的攻击手段和攻击意图，蜜罐无法发挥自身最大的用处。

当前对蜜罐自身安全的研究较少，大多集中在研究提高蜜罐仿真度，实现具体功能的蜜罐，适配各类应用场景蜜罐。但作为一种主动吸引攻击的安全资源部署在内网中，必须意识到蜜罐系统本身存在一定的风险，攻击者利用相关逃逸0Day对蜜罐系统展开逃逸攻击，获取蜜罐宿主机的控制权，将蜜罐作为跳板机对内网进行横向渗透，对真实业务展开攻击，或者攻击第三方，危害巨大。

虚拟机逃逸（Virtual Machine Escape）是攻击者突破虚拟机的防护限制，到达虚拟机所在的宿主系统中，并能在宿主操作系统中执行指令。虚拟化技术的安全关系到整个蜜罐系统的安全，当前蜜罐系统普遍采用虚拟化技术，在同一硬件设备上部署多种类型的蜜罐诱饵，虚拟化平台如果存在安全漏洞，那么整个蜜罐系统就构建在空中楼阁，成为攻击者突破内网安全防护的入口。

2008年，攻击者利用漏洞CVE-2008-0923能够对VMware公司的Workstation（版本6.0.2或

5.5.4上）进行虚拟机逃逸攻击，从虚拟机逃逸到虚拟机管理器上。2009年，在黑帽大会上，immunity公司现场演示了从VMware Workstation虚拟机环境下的逃逸攻击，并能在宿主系统上执行任意代码控制宿主系统。2014年，著名安全公司CrowdStrike发现QEMU相关的虚拟机软盘驱动代码存在漏洞，能够实现从虚拟机逃逸攻击到达宿主操作系统。2015年，Xen的物理内存资源管理上存在内存跨界访问漏洞（CVE-2015-7835），攻击者可以利用该漏洞可以由PV虚拟机中的页表管理超级调用。2016年，中国研究人员唐青昊在Pwn-Fest黑客大会上成功实现了VMware的虚拟机逃逸。2017年，VMware ESXi，Workstation、Fusion等软件中的SVGA驱动程序包含缓冲区溢出，攻击者可能利用该漏洞实现从guest虚拟机逃逸到宿主主机上执行代码。2018年，中国安全研究人员张焱宇利用VMware虚拟化平台的漏洞击穿Linux虚拟机获取宿主机系统的最高权限，成功实现虚拟机逃逸。2020年，QEMU存在内存越界漏洞CVE-2020-14364，该漏洞是利用QEMU USB模块的数组越界造成的，可实现虚拟机逃逸到宿主操作系统从而控制宿主系统。2020年，openmail邮件组发布了Linux内核权限提升漏洞CVE-2020-14386风险通告，影响目前使用内核虚拟化的产品有openshift/docker/kubernetes，攻击者利用该漏洞可能获取系统最高权限，实现虚拟机逃逸。

因此蜜罐依赖的虚拟化技术是并不是真正的安全可靠，不断有漏洞被暴露出来，如何在这种不安全的虚拟化平台之上构建安全的蜜罐系统是需要不断研究的方向。

网络空间拟态防御作为一种主动防御技术，由邬江兴院士在2014年首次提出了基于动态多变体运行机制为核心的拟态安全防御的原意和愿景[15]。其后在2016年对拟态防御思想进行深化丰富［16]，分析网络空间攻防不平衡的现状和本源问题，介绍了基于动态异构冗余（Dynamic Heterogeneous Redundancy，DHR）构造的拟态防御的具体实现。

在利用蜜罐作为内网主动防御过程中更要做好蜜罐自身系统的安全防护，如何将拟态防御思想与蜜罐系统相结合，实现具有内生安全的拟态构造蜜罐系统是本文的研究重点。

本文第2节介绍拟态构造蜜罐框架及输入代理、异构执行体、输出裁决和动态调度四个功能模块；在第3节对拟态构造蜜罐与移动目标防御和其他蜜罐进行比较分析，同时分析讨论了多异构执行体情况下的执行体时间不一致导致输出延迟的解决方案；第4节总结本文的研究内容。

2 拟态构造蜜罐

在本章中，我们通过将拟态防御思想与蜜罐主动防御系统相融合，构造具有拟态防御能力的蜜罐系统，防御蜜罐系统中存在的虚拟机逃逸攻击。

2.1 拟态构造蜜罐框架

蜜罐部署在内网中，诱导深入内网的攻击流量对其进行攻击，保护其他真实业务系统免遭攻击。蜜罐此时作为吸引攻击者攻击的目标，如果由于自身系统存在薄弱点，被攻击者利用相关逃逸漏洞对蜜罐系统展开攻击，导致攻击者获取蜜罐宿主机的控制权，那么部署在网络中的蜜罐系统将成为攻击者侵入内网的入口，并被当成跳板机对其他真实业务系统展开攻击，产生巨大危害。

为了克服蜜罐系统被攻陷后作为攻击源的问题，将拟态防御思想融入到蜜罐系统中来构建具有内生安全的拟态构造蜜罐系统。对现有蜜罐系统的基础上进行拟态化升级，如图1所示，攻击流量首先通过输入代理分发到相同功能不同实现方式的诱饵执行体中进行执行，执行结果进入输出裁决中进行一致性裁决，若裁决结果一致则输出，否则对异常诱饵执行体进行清洗或回滚，从诱饵资源池中调用新的诱饵执行体替换被攻陷的执行体。

这里有一个关键点，蜜罐作为一种主动防御系统，是需要被攻击者攻击的，这样才能记录攻击者的攻击手段和攻击意图，如果我们对原有的蜜罐构建异构执行体，举一个例子，见表1，相当于实现了三个不同的Web蜜罐，其中的中间件、操作系统不同，那么攻击者就能感知到同一种类型的蜜罐存在着不同的实现方式，从侧面被攻击者识别为蜜罐，这样就阻碍了攻击者的攻击行为。为了解决这个问题，我们考虑蜜罐宿主操作系统上运行的虚拟化技术采用异构实现，攻击者在不突破虚拟机的情况下，不会危害宿主机的安全，

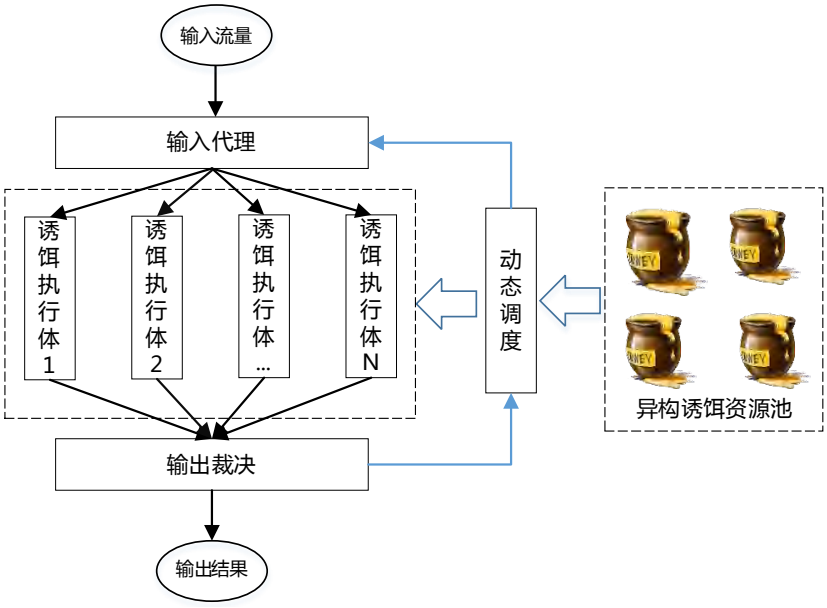


图 1　基于拟态构造的蜜罐模型

在突破虚拟机时，由于存在多个不同类型的虚拟机在同时运行，同一个漏洞一般针对具体的虚拟机类型，同一个漏洞能够攻击多个虚拟机平台概率极低。所以虚拟平台进行异构，但不同虚拟平台上运行的是相同的蜜罐，这个蜜罐不做异构，那么对于攻击者来说是很难探测自己同时是在三个蜜罐里。

表 1　异构Web执行体样例

编号	CMS	运行时	Web中间件	数据库	操作系统
1			IIS v8.0		Windows Server 2012 x64
2	Catfish	PHP v5.4.16	Apache v2.4.6	MySQL	CentOS 7.8.2003 x64
3			Nginx v1.19.1		Ubuntu 20.04 x64

下面对拟态构造蜜罐的四个功能模块：输入代理、异构执行体、输出裁决、动态调度分别进行介绍。

2.2　输入代理

输入代理是拟态构造蜜罐系统的流量入口，对进入拟态构造蜜罐的流量进行解包，提取信息分别传递给当前正在运行的诱饵执行体。

2.3　异构执行体

为了实现蜜罐的抗逃逸攻击，利用拟态防御机制，将承载蜜罐的虚拟机进行异构冗余处理，当前有KVM、VMware、Virtualbox、XEN等虚拟机，可以在服务主机安装多个虚拟化平台，其中以拟态Web蜜罐为例，如图2所示，将相同蜜罐诱饵安装在不同的虚拟化平台中，可以为服务主机提供相同服务的冗余蜜罐诱饵，当攻击流量进入拟态蜜罐时，拟态蜜罐对攻击流量进行分发，同时将流量分发到基于不同虚拟化平台相同功能的蜜罐诱饵执行体中执行，这一过程对攻击者无感，攻击者不会感受到多个拟态诱饵执行体的存在，始终使其只能感知到其中一个诱饵执行体。在每个蜜罐内部署监控程序，实时监控蜜罐内的变化。攻击者在进行逃逸攻击时，其利用的漏洞很大可能性不能同时应用到多个虚拟化平台，那么只能完成其中一个拟态执行体的逃逸，其他执行体仍然处于未逃逸状态，利用输出裁决对执行结果进行一致性比对，来发现差异。存在不一致情况时，则存在逃逸攻击，对不一致的执行体进行下线清洗，通过动态调度策略从诱饵资源池中调用新的执行体替换攻陷的执行体。

2.4　输出裁决

拟态蜜罐中异构诱饵执行体是上层应用相同底层虚拟化不同，不同虚拟机中的采集监控程序

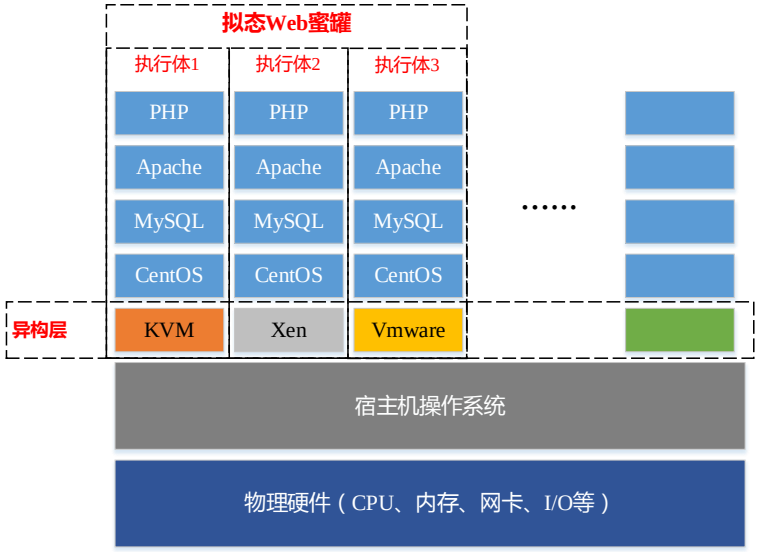


图 2　拟态 Web 蜜罐实例

独立实时采集各自异构诱饵执行体中的数据，并进行独立的逃逸检测，比对异构诱饵执行体逃逸检测结果，进而判断蜜罐逃逸攻击的存在。分析比对后，进行判断结果，并告知拟态蜜罐管理模块进行拟态蜜罐诱饵执行体的下线清洗。

2.5　动态调度

动态调度模块在接收到输出裁决输出的蜜罐诱饵执行体异常信息后，从异构诱饵执行体资源池中重新上线一个新的虚拟化平台，对被攻陷的虚拟化平台进行下线清洗，还原攻击者的攻击手段，获取攻击者利用的漏洞，对虚拟化平台进行更新升级修补漏洞后可重新上线。

3　分析讨论

本文将拟态防御思想与蜜罐主动防御系统相结合来构造具有内生安全的蜜罐系统，改变以往蜜罐系统对待 0Day 攻击束手无策的状况，为需要必须部署蜜罐而又需要确保蜜罐安全的场景提供一种解决方案。当前对于蜜罐自身的安全的研究较少，尤其在蜜罐系统防逃逸攻击主要依托于虚拟化技术的安全，而虚拟化技术不断曝出 0Day 漏洞，使我们并不能完全相信其安全，现在只有在 0Day 漏洞曝出后及时进行修补来减少损失，但这种情况不排除有些漏洞没有及时曝光或者没有及时进行漏洞修补都会危害系统安全。

与拟态防御相关防御解决方案有移动目标防御技术（Moving Target Defense，MTD）[17]，通过持续不断的动态变换，不同改变攻击面，迷惑攻击者，从而达到阻断攻击进行。针对蜜罐安全防护，利用移动目标防御思想，可以构建不同虚拟化平台在其上安装相同蜜罐，通过持续不断变换蜜罐来应对攻击者的攻击，达到防御攻击者的逃逸攻击，但此方法存在一个缺陷，蜜罐中的逃逸攻击发生的概率比较低，如果使用移动目标防御的方法进行持续不断的变换，其变换频率很难设定，变换频率过快，计算代价较大，变换频率较低，那么攻击者可能攻陷了宿主机并使用了一段时间，防御性能大大降低，同时移动目标防御方法会存在攻击行为阻断，不能完整捕获攻击行为，不便于溯源复盘。

与廉哲等人［14］提出的基于拟态防御机制的 SDN 虚拟蜜网相比，廉哲等人实现的拟态蜜罐是将上层蜜罐应用做成了异构，类似于表 1 中的方案，针对的是当攻击者利用某个漏洞攻击某一类应用时，通过牵引流量进入其他异构的蜜罐中，使其漏洞利用失败从而挫败攻击者的攻击行为，这里通过异构阻断攻击者的攻击链来实现主动防御，但其中的这样的结果是攻击者较容易发现所处的环境在不断变换，而放弃攻击，蜜罐则无法很好的收集到攻击者的攻击工具、使用的漏洞和攻击意图等。

在本文中提出的拟态构造蜜罐系统方案，是为了保留蜜罐原有功能的基础上尽量多收集攻击者的信息，同时保护蜜罐不被完全攻陷控制，实

现具有内生安全特性。

输出裁决中多个异构执行体对攻击请求响应在时间上存在差异，有些情况下以多个异构执行体执行完成输出结果进行裁决可能导致相对单个执行体的蜜罐时间延迟较长，可能被攻击者识别为蜜罐的问题。对此解决方案有：1、以单个执行体历史统计数据为依据，选择性能或者可靠性最高的异构执行体进行输出，其他执行体进行辅助判断是否存在逃逸情况，如果存在逃逸情况，则更换新的执行体为主要输出执行体；2、对于多个异构执行体，由于存在差异，很难满足某一个执行体在所有的服务执行上都是最快的，那么输出裁决以最先收到的结果进行输出，之后在收到所有结果时进行比较，如存在不一致则对执行体进行下线清洗。

4 结束语

本文提出的一种新型基于拟态构造的蜜罐系统设计思路，通过将拟态防御思想与蜜罐主动防御系统相结合，增强蜜罐系统的安全性，构建具有内生安全的蜜罐系统。当前拟态防御思想已经应用到Web服务、防火墙、交换机、路由器等产品中，证明了其实用价值。本文中将拟态防御思想应用到蜜罐系统中来增强蜜罐自身安全，也是针对当前蜜罐在实际部署存在被攻陷从而控制宿主机的情况的一种解决方案。当前本文提出的方案中还有较多细节需要完善，如拟态构造蜜罐怎么更好的对攻击者隐藏异构诱饵执行体的存在，怎么更好的判断攻击逃逸的存在等，这些都是我们以后需要研究的内容。

参考文献:

[1] Stoll C. The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage[M]. London：The Bodley Head Ltd.，1989.

[2] Cohen F. A note on the role of deception in information protection [J]. Computers & Security，1998，17(6)：483-506.

[3] Cohen F. The deception toolkit[EB/OL]. 2012. http://all. net/dtk/index. html
，1999

[4] The Honeynet Project, Know Your Enemy: Learning about Security Threats[M]. 2nd ed.，Boston: Addison-Wesley Professional，2004.

[5] Spitzner L. Honeypot farms[J]. 2012. http://www. symantec. com/connect/articles/honeypot-farms

[6] Spitzner L. Honeytokens: The other honeypots[J]. 2011. http://www. symantec. com/connect/articles/honeytokens-other-honeypot

[7] Provos N. A virtual honeypot framework[C]. In：Proc. of the 13th Conf. on USENIX Security Symp. Berkeley：USENIX Association，2004. 1-14.

[8] 诸葛建伟, 韩心慧, 周勇林, 等,HoneyBow：一个基于高交互式蜜罐技术的恶意代码自动捕获器[J]. 通信学报，2007，28(12)：8-13.

[10] The Honeynet Project. GSoC 2010 proposed ideas[EB/OL]. 2011. http://www. honeynet. org/gsoc2010/ideas

[11] Shi L, Jiang L, Liu D, et al. . Mimicry honeypots: a brief introduction [C]. WiCOM2012，IEEE Computer Society，Shanghai，2012，09：1-4.

[12] Saadi, Chaimae, and ChaouiHabiba. Cloud Computing Security Using IDS-AM-Clust, Honeyd, Honeywall and Honeycomb [C]. Procedia Computer Science (2016)：433-442.

[13] 贾召鹏, 方滨兴, 崔翔, 等. ArkHoney: 基于协同机制的Web蜜罐[J]. 计算机学报，2018，41(2)：413 - 425. doi: 10. 11897/SP. J. 1016. 2018. 00413.
JIA Zhaopeng, FANG Binxing, CUI Xiang, et al. ArkHoney: A web honeypot based on collaborative mechanisms[J]. Chinese journal of Computers，2018，41（2）：413 - 425. doi: 10. 11897/SP. J. 1016. 2018. 00413.

[14] 廉哲,殷肖川,席茜,等. 一种基于拟态防御机制的SDN虚拟蜜网[J]. 计算机工程与应用,2019,55(1):109-114.
LIAN Zhe, YIN Xiaochuan, XI Xi, et al. SDN virtual honeynet based on mimic defense mechanism[J]. Computer Engineering and Applications，2019，55(1)：109-114.

[15] 邬江兴. 拟态计算与拟态安全防御的原意和愿景[J]. 电信科学，2014，30(07)：2-7.

[16] 邬江兴. 网络空间拟态防御研究[J]. 信息安全学报，2016，1(04)：1-10.

[17] Jajodia, Sushil, GhoshAnup K. , SwarupVipin, Cliff Wang, and X. Sean Wang. Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats [M]. Berlin，; Springer，2011.

[作者简介]

胡先君（1989—），男，博士，工程师，主要研究方向为网络安全、密文域信号处理。

王涵（1989—），男，硕士，国家注册信息安全工程师，主要研究方向为网络与信息安全。

卜佑军（1978—），男，博士，副研究员，主要研究方向为网络空间安全。

基于谱平均线性判别分析的射频指纹识别方法

张展鹏，朱丰超，姚敏立
火箭军工程大学作战保障学院，陕西西安 710025

摘　要：大量无线设备的接入给物联网带来严重安全隐患。射频指纹识别技术通过提取无线设备硬件的细微特征来实现设备的唯一认证，能有效地解决无线网络安全问题。针对传统射频指纹识别方法的精度随着信噪比降低迅速下降的问题，本文提出了一种基于谱平均线性判别分析（Spectral Average Linear Discriminate Analysis，SALDA）的射频指纹识别方法。首先计算最优投影矩阵，然后提取候选设备连续多个前导的平均功率谱密度特征，最后在子空间内采用最小质心准则进行设备识别。在理论分析的基础上，本文在8个实际的软件无线电设备上进行实验，在15 dB以上取得了75%以上的精度，对比基准算法，平均提高了10%以上识别精度。

关键词：物理层安全、无线物理层认证、射频指纹、设备识别

Spectral Average Linear Discriminate Analysis for RF Fingerprint Identification

ZHANG Zhanpeng, ZHU Fengchao, YAO Minli
Academy of Combat Support, The Rocket Force Engineering University, Xi'an 710025, China

Abstract: Access to a large number of wireless devices brings serious security risks to the Internet of Things. The RFF identification technology realizes the device's unique authentication by extracting the subtle characteristics of the wireless device hardware, which can effectively solve the wireless network security problem. Aiming that the accuracy of traditional RFF identification methods declines rapidly with the decrease of SNR, this paper proposed an RFF identification method based on spectral average linear discriminate analysis (SALDA). First, we calculated the optimal projection matrix, then extracted the average power spectral density characteristics of multiple consecutive preambles of the candidate device, and finally used the minimum centroid criterion in the subspace for device identification. Based on theoretical analysis, we conduct experiments on eight actual software-defined radios and achieves an accuracy of more than 75% above 15 dB. Compared with the baseline algorithms, the identification accuracy is improved by more than 10% on average.

Key words: physical layer security; wireless physical layer authentication; radio frequency fingerprint; device classification

1　引言

随着无线通信技术的快速演进，无线网络逐渐成为现代信息生活的重要部分，并且广泛地用于民用和军事应用之中。但是，无线网络由于其特有的开放性和广播性，更容易遭受各种恶意攻击与入侵。近年来，5G网络进一步推进与部署，越来越多的物联网设备开始接入中心网络，无线网络存在巨大安全隐患。因此如何有效地对各种未知无线设备进行安全认证成为了热点问题。经典无线设备安全认证方法是定义在网络上层的密码机制。尽管5G认证体系针对4G网络的缺陷进行了增强，采用了公私钥加密体制，但本质上仍是基于密码学原理。随着计算机算力的飞速提升，这种基于密码机制的高层认证体系受到严重挑战。因此，基于物理层的无线认证技术得到了广大研究人员的关注[1]。

射频指纹（Radio Frequency Fingerprint，RFF）

基金项目：国家自然科学基金资助项目（No. 61601474）。

是指无线设备内在的本质特征，这种特征来源于模拟电子线路制造过程中产生的硬件误差，难以改变与模仿。类比人的生物指纹，射频指纹包含了无线设备独一无二的数字特征，因此可以用来对无线设备进行唯一识别[2]。由于其工作在物理层，可以作为增强措施兼容现有无线认证体系。在物联网应用中，中心设备具有足够的内存和算力，能够搭建射频指纹识别系统，减少边缘节点的复杂密码认证设计。在现有公开文献中，许多无线系统包括：ZigBee[3]、WiFi[4]和LTE[5]等等都已开展射频指纹识别的研究，射频指纹识别正在成为一种新兴的无线设备认证技术。

射频指纹识别一般可以分为两个阶段：训练阶段和分类阶段。在训练阶段，搭建在主机（中心接收机）上的射频指纹识别系统需要获取足够的样本用作本地数据库中的训练集，并训练合适的机器学习分类模型。在分类阶段，主机接从未知设备的通信信号中提取特征并通过训练好的模型识别未知设备。现有的射频指纹识别系统大多数都采用机器学习分类算法，其本质是数据驱动的，数据本身的质量和分布局限了射频指纹识别的精度和在实际通信场景中的使用。传统的射频指纹识别方法只考虑了发射机和接收机静止并保持固定距离的简单场景，即在单一信噪比（固定距离）下采集数据训练分类模型，在相同信噪比条件下进行识别。而当测试设备移动或者信道环境改变时则需要相应更新数据库[6]。文献[7]通过实验验证了训练数据库中样本信噪比的影响，实验结果表明在高信噪比下训练的模型仅仅能识别高信噪比下的测试设备，低信噪比下训练的模型更适合识别低信噪比下的测试设备。但是，在射频指纹识别系统的实际部署中，一般原始设备的登记过程通常工作在近距离，局限了训练样本信噪比多样性的来源，这种特性也局限了深度学习方法的在实际场景下的使用[8]。从已有的报道来看，现有的方法还不能很好的抵抗噪声的影响，随着信噪比的降低，识别精度开始迅速下降。这个问题进一步限制了射频指纹识别技术在实际通信通信系统中的部署。

本文提出了一种基于谱平均线性判别分析（Spectral Average Linear Discriminate Analysis，SALDA）的射频指纹识别方法，首先在近距离（高信噪比）采集的训练数据上进行训练，计算得到最优投影矩阵，然后提取同一个候选设备的连续多个前导并计算平均功率谱密度作为待识别信号的特征向量，最后在投影子空间内采用最小质心准则对候选设备分类。SALDA算法有两个显著的特点：一是将样本投影到最优子空间内减少特征向量冗余信息提高了可分性，二是通过谱平均有效地减小了待识别信号的噪声方差，并且将每类设备的质心向量作为模板指纹，减少了谱平均后待识别信号与真实类的偏差。

本文剩余部分的结构如下：第2节，简要介绍了射频指纹识别系统包括射频指纹来源及整个识别系统的构成；第3节，提出了基于SALDA的射频指纹识别算法并分析其理论基础；第4节，针对现有问题设计了多个对比实验，在8个实际的软件无线电设备上进行验证；第5节，结论。

2　射频指纹识别系统

2.1　射频指纹来源

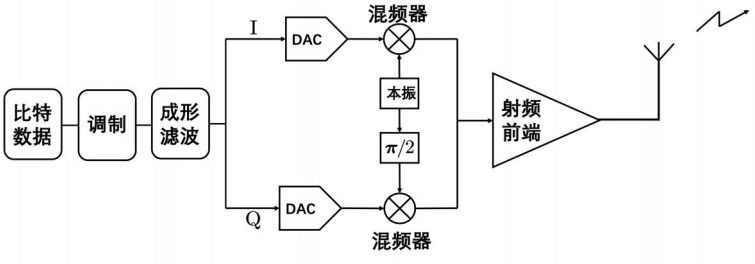


图1　典型的发射机内部结构

一种典型的发射机内部电路结构如图1所示。在发射机端，连续的二进制比特首先被映射为高阶符号并正交变换为复信号，$x_m = x_m^I + x_m^Q$，其中x_m^I和x_m^Q分别为正交和同相分量。每个符号持续的

时间由数据传输速率和调制方案决定，由于时钟缺陷导致定时不准确，每个符号实际持续的时间产生误差，即：时间间隔误差（Time Interval Error，TIE）。第 m 个符号实际持续时间为 $T_m = T + \sigma_{TIE}^m$。调制信号随即通过成形滤波器得到波形序列：

$$u[n] = \mathrm{mod}(x_m, h_s(t, T_m))。\tag{1}$$

在无线设备上，数模转换器（DAC）将基带信号序列 $u[n]$ 转换为连续时间模拟信号 $u(t)$，由于DAC的不理想往往会引入量化误差和积分非线性，输出的实际信号可以表示为：

$$y_u(t) = \sum_{-\infty}^{\infty}(u[n] + \Delta_n)g(\frac{t - nT_g}{T_g}) + \Delta_{INL},\tag{2}$$

$$g(\theta) = \begin{cases} 1, & 0 \leqslant \theta < 1 \\ 0, & \text{其他} \end{cases},\tag{3}$$

其中，Δ_n 是量化噪声，T_g 为DAC时间周期，Δ_{INL} 为积分非线性。在数模转换之后，模拟基带信号被混频器调制到带通：

$$z(t) = \mathrm{mix}(y_u(t), w_c, \xi),\tag{4}$$

其中，w_c 为载波的角频率，ξ 为引入的混频器正交误差。最后，信号通过射频前端的功率放大器和滤波器组将信号放大，经过天线辐射到空中。文献[6]指出，射频前端的非线性引入了最显著的一种射频指纹：

$$w(t) = h_{PA}(z(t), \tilde{\mathbf{a}}_{tx}) \otimes h_{BP}(t)\tag{5}$$

其中，$h_{PA}(\cdot, \tilde{\mathbf{a}}_{tx})$ 代表发射机前端的非线性函数，$h_{BP}(t)$ 为带通滤波器。有理论证明，随着信号被调制到载波上，射频前端的非线性在频域上的表现更为明显，因此，功率谱密度是一种区分度更大的射频指纹特征。

信号波形通过天线辐射到空中，在经过大规模路径衰减、无线信道各种噪声和延时的影响后，射频指纹被扭曲，这是射频指纹识别效果不好的主要来源。在接收端，信号经历了相反的信号处理过程，本文只考虑单一接收机条件下的射频指纹识别。

2.2 射频指纹识别系统流程

整个射频指纹系统可以简单地分为信号采集、前导信号提取、指纹特征提取以及射频指纹的分类识别几个部分。射频指纹识别系统流程图如图2所示。

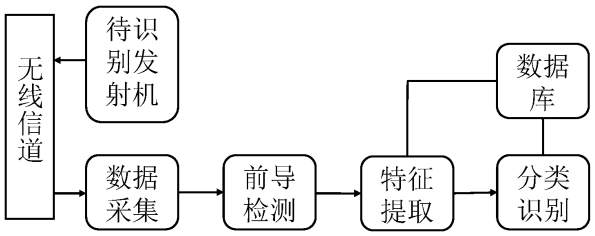


图2 射频指纹识别系统基本流程

待识别发射机不断地发射数据帧请求访问，接收机采集到的信号同时包含了设备的射频指纹和信道指纹，其中射频指纹包含了无线设备硬件的独特信息，不可模仿和改变，而信噪指纹表征着设备附近的环境因素即无线信道响应。真实的射频指纹极易受到信道的影响，在比较恶劣的高斯白噪声信道下，射频指纹识别精度也比较低。由于通信信号本身携带了许多比特信息，因此相同设备不同信号段的射频指纹特征也会受此影响，甚至可能会具有较大差异。而现代数字通信系统为了简化接收机操作几乎都在数据帧最前端规定了一段统一的码序列，即前导信号，在相同的协议下，前导信号都是一致的。前导信号的提取主要是早期的一些工作，其中比较经典的是基于方差阈值的检测算法[9]。而对于提取到的前导信号在进行特征提取时通常包括一些域变换和统计分析：频域、调制域、小波域和高阶统计量[10]。一般通过特征变换将原始时序信号变换为特征向量，通过大量采集训练数据集训练好分类识别算法模型对待识别发射机进行识别。

3 基于SALDA的射频指纹识别算法

3.1 线性判别分析原理

在射频指纹识别领域，使用频率最多的是基于fisher准则的线性判别分析（Linear Discriminant Analysis，LDA），LDA是一种典型的监督学习算法，充分利用了样本的标签信息。假设将经过特征变换后的数据集定义为 $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$，其中 $\boldsymbol{x}_i = (x_{i1}; x_{i2}; \cdots; x_{id})$，$y_i = \mathbb{R}$，数据集总共有 C 类，m 个样本。

LDA将所有的样本投影到子空间内，在这个子空间内不同类别的样本尽可能的分开，而相同类的样本尽可能的聚集，因此LDA的核心就是求解上述投影方向。LDA的目标函数可以表示为：

$$\mathbf{W} = \arg\max_{\mathrm{w}} \frac{\mathrm{tr}(\mathbf{W}^T\mathbf{S}_b\mathbf{W})}{\mathrm{tr}(\mathbf{W}^T\mathbf{S}_w\mathbf{W})}\tag{6}$$

$$\mathbf{S}_b = \sum_{k=1}^{c} m_k (\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu})(\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu})^T, \tag{7}$$

$$\mathbf{S}_w = \sum_{k=1}^{c} \left(\sum_{i=1}^{m_k} (\boldsymbol{x}_i^{(k)} - \boldsymbol{\mu}^{(k)})(\boldsymbol{x}_i^{(k)} - \boldsymbol{\mu}^{(k)})^T \right), \tag{8}$$

其中，$\mathrm{tr}(\cdot)$表示矩阵的迹，$\boldsymbol{\mu}$是总体样本均值向量，m_k是第k类样本的数量，$\boldsymbol{\mu}^{(k)}$是第k类样本的均值向量，$\boldsymbol{x}_i^{(k)}$是第k类样本的第i个样本。$\mathbf{S}_w$和$\mathbf{S}_b$分别为类间散度矩阵和类内散度矩阵，$\mathbf{W} = [w_1, \cdots, w_l]$。上述优化问题可以等价于寻找如下特征值求解问题的$l$个最大特征值对应的特征向量：

$$\mathbf{S}_b \mathbf{W} = \lambda \mathbf{S}_w \mathbf{W} \tag{9}$$

由于$\boldsymbol{S}_b$的秩最多为C - 1，则上述问题最多可以解出C - 1个非0特征值，即最多可以获得C - 1个投影向量。假设某一个测试样例为$\boldsymbol{x}$，将测试样本与投影矩阵$\mathbf{W}$相乘，得到$\mathbf{W}^{\mathrm{T}}x$，可以将测试样本投影到较低维度的子空间内，实现更好的分类效果，一定程度上去除了冗余信息，加快了分类识别的速度。

3.2 功率谱特征的方差分析

经过发射机、无线信道和接收机，原始信号嵌入了射频指纹，经过离线处理之后，将截取的前导正交序列表示为$x[n]$。通过上述理论分析，发射机最显著的射频前端非线性是信号畸变的主要来源，这种信号畸变在频域内最为显著，可以直接提取功率谱作为频域射频指纹。快速傅里叶变换（FFT）的提出大大加快了计算时间，在射频指纹识别领域中最常用周期图法来计算功率谱密度：

$$X_N(k) = \sum_{n=0}^{N-1} x[n] \exp\left(-j \frac{2\pi}{N} nk\right), \tag{10}$$

$$\hat{P}_{PER} = \frac{1}{N} \left| X_N(k) \right|^2, \tag{11}$$

其中N为FFT计算点数，在射频指纹识别中一般来说保持与前导序列$x[n]$样本点数一致即可。根据数字信号处理原理，对于固定的数据长度N，周期图法估计出的谱是一个有偏估计，当数据长度趋近于无穷时，估计谱的期望等于真实谱，因此又是渐近无偏的。根据进一步推导，估计谱的方差可以表示为：

$$\mathrm{var}[\hat{P}_{PER}(w)] = D + [E\{\hat{P}(w)\}]^2, \tag{12}$$

$$D = \left| \frac{1}{2\pi N} \int_{-\pi}^{\pi} P(\lambda) D_0(w - \lambda) D_0(w + \lambda) \mathrm{d}\lambda \right|^2, \tag{13}$$

其中，D_0为窗函数，$P(w)$为真实功率谱。由于实际使用时的数据长度总是有限的，周期图法估计的功率谱密度的方差总是大于估计值均值的平方。

3.3 SALDA算法

现有文献的工作通常在同一个信噪比环境下训练分类模型，并在相同的信噪比环境下测试。但是在实际使用中，各种准确信道条件下的数据难以获得。一般在初始化整个射频指纹系统（在数据库中登记已知设备）时，只能将发射机放在离中心设备（接收机）比较近的位置，这通常是比较高的信噪比环境。此时就出现了一个新的问题，训练集只能来源于比较高的信噪比条件下，而测试集的信噪比未知。随着测试样本信噪比的降低，测试样本的分布与训练集分布距离越来越大。在这样的系统条件下，各种分类算法的精度都迅速下降，无法达到基本的识别效果。从上述原理分析可知，信噪比的变化导致了样本分布的改变，尤其是高信噪比下的样本分布和低信噪比下的样本分布距离较大。对于传统的机器学习算法来说，在具有高质量的数据集下训练出来的模型，难以用于其他信噪比条件下的数据。因此，信噪比的降低给射频指纹识别带来两方面的影响：一是信号在低噪声条件下本身有效特征信息被模糊，样本的可识别度降低；二是样本分布产生变化，直接影响机器学习算法的性能。

假设通过前导提取后制成的数据集中每个样本是具有相同均值和方差的独立随机变量X_1，X_2，$\cdots$，X_L，则随机变量$X = (X_1 + X_2 + \cdots + X_L)/L$的方差将便为原来的$1/L$。根据这个简单的概率论原理可以发现平均的方法可以很好地改善周期图法估计功率谱的方差特性，通过计算多个连续帧的前导信号的功率谱密度，然后相加取平均得到平均功率谱密度：

$$\overline{P}_{PER}(w) = \frac{1}{NL} \sum_{i=1}^{L} \left| \sum_{n=0}^{N-1} x^i[n] \exp(-jwn) \right|^2, \tag{14}$$

其中，L表示连续选取的前导个数。

假设系统初始化训练阶段的训练集来自于良好信道条件的高质量前导信号，首先计算功率谱密度作为为特征向量，训练数据集可以表示为$\{(\boldsymbol{x}_{\mathrm{H}}^i, y_i)\}_{i=1}^{m}$，$y_i = \mathbb{R}$，总共$C$类，$m$个样本。首先在训练集上计算类间散度矩阵$\mathbf{S}_w$和类内散度矩阵$\mathbf{S}_b$，根据公式（6）求得该训练集下的投影矩阵

$\mathbf{W}$。在识别阶段，连续提取待识别设备连续多个前导并根据公式（14）计算平均功率谱密度：

$$\bar{\boldsymbol{x}}_L = (\boldsymbol{x}_L^1 + \boldsymbol{x}_L^2 + \cdots + \boldsymbol{x}_L^n)/n, \tag{15}$$

其中，n表示连续提取的前导数量，$\boldsymbol{x}_L^i$表示低信噪比下的某个样本，根据谱平均原理，$\bar{\boldsymbol{x}}_L$的方差能得到有效的降低，这个操作一定程度上提高了样本信噪比。

在计算出的投影矩阵和待识别设备平均功率谱密度特征的基础上，结合最近质心准则进行分类识别。首先将训练集投影到求得的最优子空间内，在该子空间内，不同设备的样本能尽量分开，相同设备的样本尽可能接近。然后将每一类设备样本的平均功率谱密度特征，即训练集类均值向量$\boldsymbol{\mu}^{(k)}$，投影到子空间内作为这一类设备的模板指纹（质心）。最后计算待识别设备的平均功率谱密度特征$\bar{\boldsymbol{x}}_L$并投影到子空间内，通过计算投影后的特征向量与训练集中每一类设备模板指纹的欧式距离，通过距离判断，距离哪一类设备比较近，就将未知设备判断为那一类，计算公式表达为：

$$y_L = \arg\min_{k \in C} \left\| \mathbf{W}^{\mathrm{T}} \bar{\boldsymbol{x}}_L - \mathbf{W}^{\mathrm{T}} \boldsymbol{\mu}^{(k)} \right\|^2 。 \tag{17}$$

其中，$\boldsymbol{\mu}^{(k)} = 1/m_k \sum_{i=1}^{m_k} \boldsymbol{x}_H^i$代表了高信噪比下的类均值向量。可以发现，SALDA算法有两个直观的优势：一是将样本投影到最优子空间内减少特征向量冗余信息，并且提高了可分性；二是计算低信噪比下连续多个前导信号的功率谱密度平均，降低了待分类样本$\bar{\boldsymbol{x}}_L$的方差，多个信号功率谱密度的平均在数学上与计算类均值向量$\boldsymbol{\mu}^{(k)}$方式一致，一定程度上减少了$\bar{\boldsymbol{x}}_L$与真实设备之间的偏差，如果用于平均的样本信号足够多，理论上$\bar{\boldsymbol{x}}_L$与真实类均值向量之间保持一个固定偏差，这个偏差由噪声等信道干扰在功率谱上的统计特性所体现。SALDA算法的具体计算流程如下表所示：

算法 1：谱平均线性判别分析SALDA

输入：训练数据：$\{(\boldsymbol{x}_{\mathrm{H}}^i, y_i)\}_{i=1}^m$

测试数据：$\boldsymbol{x}_L^1, \boldsymbol{x}_L^2, \cdots, \boldsymbol{x}_L^n$

模型参数：谱平均个数n

输出：测试数据分类结果y_L

开始：

1. 计算类内散度矩阵$\mathbf{S}_b$
2. 计算类间散度矩阵$\mathbf{S}_w$
3. 计算的最大的d个特征值和对应的d个特征向量，得到投影矩阵$\mathbf{W}$
4. 测试样本进行谱平均，并根据投影矩阵投影到最优子空间
5. 根据类质心距离最小准则计算分类结果y_L

4 实验结果

4.1 实验系统设置

ZigBee是一种低速率短距离传输上网协议，被广泛地运用于包括医疗保健、智慧家居、智能物流以及车对车通信等物联网应用中。本文利用9个Analog Devices（ADI）公司同一批次生产的ADALM-PLUTO软件无线电设备搭建了基于ZigBee协议的点对点通信平台（其中1个设备作为接收机，其余8个设备作为接收机）。实验设备如图3所示。ZigBee协议主要来源于IEEE 802.15.4低速率无线网络标准，本文采用定义在2.4GHz频段的偏移正交相位调制（Offset-QPSK）物理层技术。

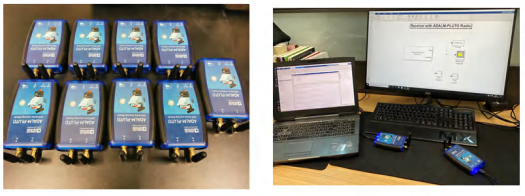


图 3 实验设备平台

发射机系统通过Matlab工具箱中提供的ZigBee协议通信工具箱在个人笔记本电脑上生成离散数字信号。首先将原始比特数据每4位映射为一个符号，然后通过直接序列扩频（DSSS）将每个符号映射为32位的chip值，最后使用基于半正弦脉冲成形滤波器（4个样本每符号）的O-QPSK将整个chip序列调制到载波上。这样得到了一个完整的数据帧波形，本文的载荷数据随机产生共120位比特，前导为8个全0符号即32个0比特，最后数据帧总长度为2690个样本点。其中前导部分长度为512个I/Q信号。波形数据通过Matlab提供的ADALM-PLUTO硬件辅助工具箱加载到设备上，经过数模转换和一系列滤波器组将信号搬移到2450MHz射频上，通过一个全向天线发射出去。软件无线电设备基带速率为4MHz，接收机系统分别距离发射机0.1m。采集的基带信号存储在计算机内进行离线处理，捕获共7600个实际前导信号（每个设备950个样本）。

4.2 实验结果分析

4.2.1 实验数据与实现细节

为了获得可靠的稳定算法模型，本文将80%的样本用作训练集，20%的数据进一步分为验证集和测试集，验证集用于调整模型参数，测试集验证算法的正确性。本文将0.1m下的80%的数据作为高信噪比条件下的原始数据，作为训练集训练算法模型。为了简化实验步骤，本文通过使用Matlab里面通信工具箱的高斯白噪声模块来模拟真实信道条件，那么可以分别得到了信噪比分别为30 dB、25 dB、20 dB、15 dB、10 dB、5 dB和0 dB的测试数据集。这里假设原始数据集信噪比较高当作纯净信号，所以这个信噪比为相对的信噪比，最后加噪声后的数据的信噪比实际中都要更低一些。实验考察SALDA算法及以下几种射频指纹识别基准算法在不同信噪比下的准确率：

• k近邻（k-Nearest Neighbor，KNN）[11]

• 线性判别分析（Linear Discriminant Analysis，LDA）[6]

• 支持向量机（Support Vector Machine，SVM）[12]

本文所提方法和三种基准方法在0至30dB条件下分别测试其识别性能。KNN的距离函数取最常用的欧式距离，近邻个数取10。KNN和SVM都按照Matlab统计学习工具箱提供的算法实现，LDA参考文献[13]的方法实现。

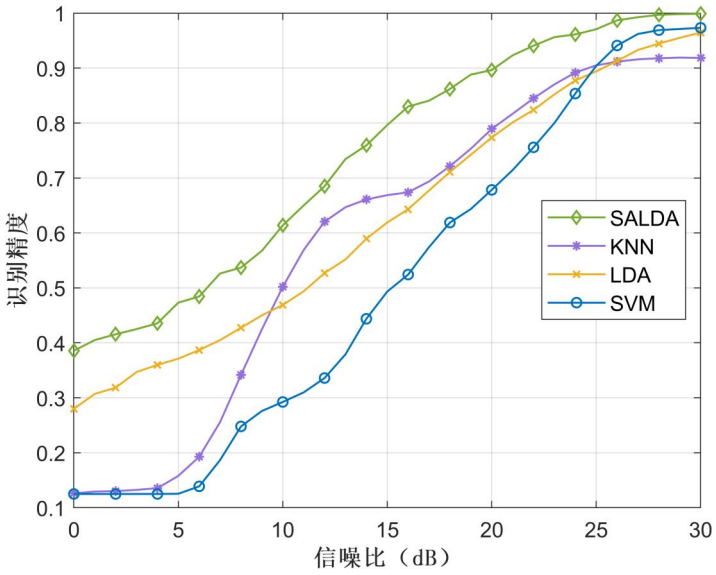


图4 SALDA、KNN、LDA和SVM在不同信噪比下识别精度

4.2.2 信噪比影响分析

图4首先对比分析了本章所提出的谱平均线性判别分析方法与其他几种基准算法在不同信噪比下的识别效果。从总体趋势来看本章提出了SALDA相比普通的LDA算法有了明显的提高，相比其他几种算法在所有信噪比下都具有明显的优势。从整体趋势来看，SALDA对原始LDA算法在10 dB至20 dB范围内提升较为明显，平均提高了10%至15%的精度提升，提升还是比较大。从图中我们也能看到SVM算法在25 dB以上的高信噪比下精度很高，但是随着信噪比下降，精度迅速下降。而KNN算法尽管在25 dB以上精度不是很高，但是在15 dB至20 dB之间保持了一定了稳定性。这种结果一定程度上反映了，经典SVM算法能更好的拟合训练样本，但是对未知的样本泛化性较差，而模型更加简单的KNN泛化性更强。但是在10 dB之后KNN算法迅速下降，这也说明了基于固定距离计算的模型难以在强噪声环境下进行识别。LDA算法一直在高信噪比下和低信噪比下一直保持稳定的趋势，并不会随着信噪比的降低迅速下降，说明了LDA算法本身对噪声的变化具有一定的鲁棒性。从表1中能看到典型信噪比下更为详细的精度对比结果。SALDA算法在20 dB以上基本能达到90%以上的识别精度，15 dB以上，接近超过80%的精度，这个结果已经能在常见的无线通信场景下实现一定的识别效果了。

表 1　SALDA、KNN、LDA 和 SVM 在典型信噪比下识别精度

信噪比	0 dB	5 dB	10 dB	15 dB	20 dB	25 dB	30 dB
KNN	0.1261	0.1579	0.5019	0.6688	0.7893	0.9049	0.9184
LDA	0.2800	0.3713	0.4689	0.6190	0.7736	0.8939	0.9645
SVM	0.1250	0.1254	0.2924	0.4933	0.6783	0.9039	0.9733
SALDA	**0.3856**	**0.4731**	**0.6138**	**0.7963**	**0.8963**	**0.9706**	**0.9988**

4.2.3　参数敏感性分析

影响 SALDA 算法的主要参数是参与谱平均前导样本的个数，因此通过增加前导样本的个数来分析算法参数的敏感性。主要分析几种典型的信噪比下（5 dB、15 dB、20 dB、25 dB）SALDA 算法的平均样本个数给识别精度带来的影响。

从图 5 中可以很容易的看出，随着平均样本个数的增加，在各个信噪比的测试环境中，识别精度均有不同程度的提升。在 5 dB、15 dB 和 20 dB 信噪比下，提升了超过 10% 的精度，在 25 dB 信噪比下，由于测试样本本身质量较高，精度提升稍微弱一些，不过也有 8% 的提升。从图中还可以发现，5 dB 信噪比以上使用 5 个前导样本的平均使得精度提升的最显著，超过 5 个数量的谱平均之后速度开始放缓，需要更多大量的样本才能实现更好的识别。由于实验样本数的有限，为保证平均后的测试样本总数，本文的平均个数上限设定为 12，可以看出即使在平均个数没有很多的情况下，SALDA 算法对识别精度已经有了可观的提升。如果进一步获得更多的样本，SALDA 算法在低信噪比下的性能将会得到更近一步的提升。

6　结束语

本文提出了一种基于 SALDA 的射频指纹识别方法。SALDA 算法在高信噪比数据下进行训练，并未满足传统机器学习对训练数据多样性的需求。在测试场景中提取连续多个数据帧的前导信号计算平均功率谱密度，降低了测试设备样本的噪声方差的同时减小了其与类质心之间的偏差。最后，本文在 8 个实际的软件无线电设备上进行实验，验证了 SALDA 方法的良好性能，对比传统的 LDA、KNN 和 SVM 算法，平均取得了超过 10% 以上的精度，显著的提高了识别精度。实验结果表明，本文提出的 SALDA 能够改善低信噪比条件下的射频指纹识别。

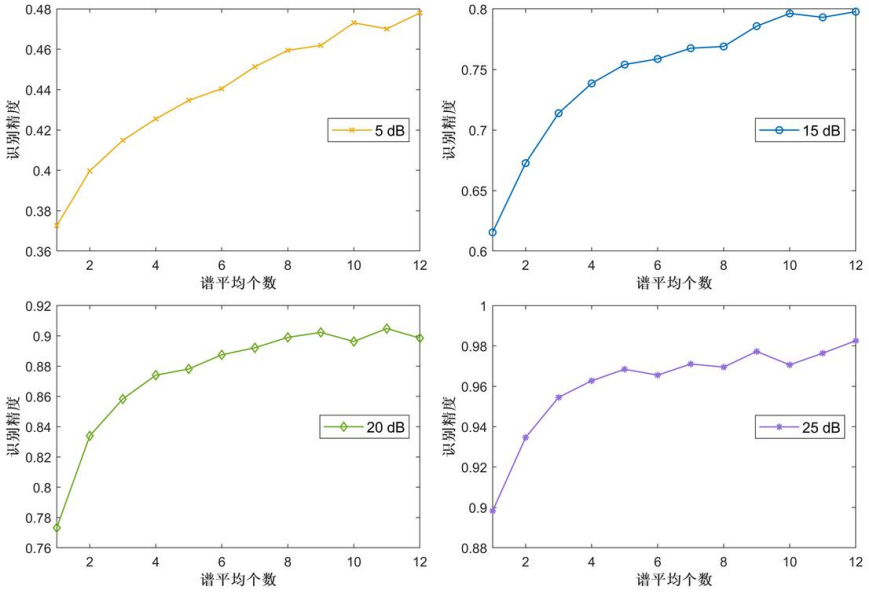


图 5　不同信噪比下谱平均对识别精度的影响

参考文献:

[1] FANG H, WANG X, TOMASIN S. Machine Learning for Intelligent Authentication in 5G and Beyond Wireless Networks［J］. IEEE Wireless Communications，2019，26(5)：55-61.

[2] SOLTANIEH N, NOROUZI Y, YANG Y, et al. A Review of Radio Frequency Fingerprinting Techniques［J］. IEEE Journal of Radio Frequency Identification，2020，4(3)：222-33.

[3] WANG J, ZHUANG L, CHENG W, et al. Analysis of Classification Methods Based on Radio Frequency Fingerprint for Zigbee Devices［M］. 2019：121-32.

[4] WHEELER C G, REISING D R. Assessment of the impact of CFO on RF-DNA fingerprint classification performance; proceedings of the 2017 International Conference on Computing，Networking and Communications (ICNC), F 26-29 Jan. 2017，2017［C］.

[5] DEMERS F, ST-HILAIRE M. Radiometric identification of LTE transmitters［M］. 2013.

[6] WANG W, SUN Z, PIAO S, et al. Wireless Physical-Layer Identification: Modeling and Validation［J］. IEEE Transactions on Information Forensics and Security，2016，11(9)：2091-106.

[7] REHMAN S U, SOWERBY K, COGHILL C. Effect of receiver Signal to Noise Ratio on the classification performance of RF fingerprinting；proceedings of the 2012 15th International Multitopic Conference (INMIC), F 13-15 Dec. 2012，2012［C］.

[8] RIYAZ S, SANKHE K, IOANNIDIS S, et al. Deep Learning Convolutional Neural Networks for Radio Identification［J］. IEEE Communications Magazine，2018，56(146-52.

[9] RASMUSSEN K B, CAPKUN S. Implications of radio fingerprinting on the security of sensor networks; proceedings of the 2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops - SecureComm 2007, F 17-21 Sept. 2007，2007［C］.

[10] REHMAN S U, ALAM S, ARDEKANI I T. An Overview of Radio Frequency Fingerprinting for Low-End Devices［J］. International Journal of Mobile Computing and Multimedia Communications，2014，6(3)：1-21.

[11] BALDINI G, GIULIANI R, STERI G, et al. Physical layer authentication of Internet of Things wireless devices through permutation and dispersion entropy；proceedings of the 2017 Global Internet of Things Summit (GIoTS)，F 6-9 June 2017, 2017［C］.

[12] YINGJUN Y, ZHITAO H, FENGHUA W, et al. Radio Specific Emitter Identification based on nonlinear characteristics of signal；proceedings of the 2015 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)，F 18-21 May 2015, 2015［C］.

[13] CAI D, HE X, HAN J. SRDA: An Efficient Algorithm for Large-Scale Discriminant Analysis［J］. IEEE Transactions on Knowledge and Data Engineering，2008，20(1)：1-12.

[作者简介]

张展鹏（1997—），男，硕士，主要研究方向：物理层安全，射频指纹识别等。

朱丰超（1985—），男，博士，副教授，主要研究方向：5G物理层安全，射频指纹识别，未来智能通信等。

姚敏立（1966—），男，博士，教授，主要研究方向：卫星通信，卫星动中通平板阵列天线，未来智能通信等。

Conditional Probability Voting Algorithm based on Heterogeneity of Mimic Defense System

Wei Shuai[1], Zhang Huihua[2], Ling Ouyang[1], Zhang Wenjian[1], Yu Hong[1]

1.National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China;

2.Wuxi Confidential Technology Service Center, Wuxi 214000, China

Key words: Condition probability voting algorithm; heterogeneous redundant executors; mimic defense architecture; system failure probability; scalability

Abstract. In recent years network attacks have been increasing rapidly， and it is difficult to defend against these attacks， especially attacks at unknown vulnerabilities or backdoors. As a novel method， Mimic defense architecture has been proposed to solve these cyberspace security problems by using heterogeneous redundant executors to perform the same task. How to choose appropriate executors and voting algorithm according to heterogeneities of these executors become the key issues of designing mimic defense architecture. Most of current researches are based on the 2-level similarity of executors， but the results are not accurate enough. This paper presents an attack model based on mimic defense architecture， abstracts binary division vector and relevant indexes to describe the heterogeneity of these executors， and innovatively proposes conditional probability voting algorithm， which is different from classic majority voting algorithm. This paper also analyzes the system failure probability and scalability of these voting algorithms， experiment results show that conditional probability voting algorithm is the best， both in system failure probability and scalability.

1 INTRODUCTION

With the continuous upgrading of information services， information system become more and more complex， and the amount of software and hardware code become larger and larger. Take operating system as an example， overall amount of code for Linux 2.0 is about 6MB， which increases to 28 MB for Linux 2.5， and 92MB for Linux 3.0. The latest Linux version is 5.4， total amount of code is 163 MB， which contain 25 million lines of code ［1］. According to the statistics of the United States， on average， within 1000 to 1500 lines of codes， human programmers will leave a software vulnerability ［2］， which shows that information systems have high security risks.

Since 2010， with the emergence of Stuxnet virus ［3］ which infected Iran's nuclear plant system and damaged some of the centrifuges， network security issues have been widely concerned. Since then， the network attacks have expanded from traditional Internet to energy， transportation and other fields which are related to the national economy and people's livelihood. With the arrival of the Internet of things， the security situation has become more and

more serious.

In order to keep cyberspace secure from network attack, some 'game-changing' technologies have been proposed, "Moving Target Defense" (MTD) [4] [5] [6] [7] [8] [9] is a typical of these technologies, which make network configurations, instructions, or code locations and so on changing over time, so the attacks for these targets are becoming difficult. However, in dealing with dark features hidden in the target system or unknown attacks through the hardware/software backdoors, there still exists the problem of ineffective mechanisms [10].

How to build a safe and reliable system based on devices with unknown vulnerabilities or backdoors, Jiangxing Wu and others put forward the idea of mimic defense architecture [11] [12] [13], using the principle of dynamic heterogeneous redundancy to break the static accurate environment which successful network attacks usually rely on. The classic model of mimic defense architecture is shown in Figure 1. The external input is distributed to the heterogeneous executors through input agent, and the heterogeneous executors perform the same task and send their results to the scheduler, the scheduler uses majority voting or alternative voting strategy to choose the correct result as final output. Then scheduler can carry out corresponding feedback control according to the results of working executors. If a certain number of results generated by an executor are wrong, then the executor is considered to have been attacked, and should be cleaned. The attacked executor will recover and be added to the work queue after cleaning. Control, system relevant parameters and system operation status can be viewed or controlled by negative feedback controls.

The mimic defense system uses heterogeneous redundant executors to perform the same task, and compares results they generated, to make sure the results are correct. The principle is that heterogeneous executors will not or rarely generate the same error. In fact, with the development of open source and agile

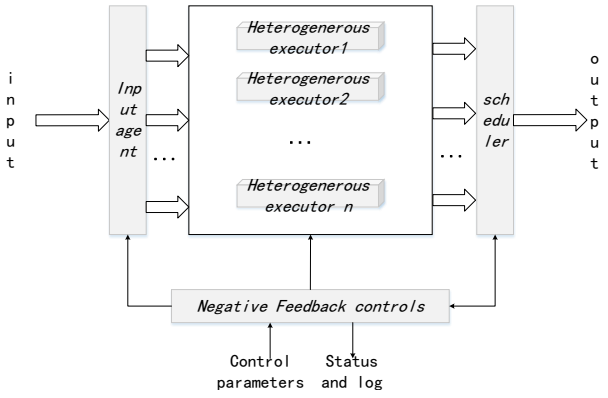


Fig. 1 classic model of mimic defense architecture

development technology, different heterogeneous executors inevitably have similarities. Take application program for example, there are many high-quality software in open source community, such as GitHub. After many users' testing and improvement, these software programs are relatively stable and reliable. In order to save time and reduce costs, more and more developers tend to use open source software for application development, which results in the similarity between different application programs. The same is true for hardware, such as CPU, which also uses a large number of public IP in the design process, so that there exist similarities between different hardware.

At present, many studies have been done on how to choose executors for working based on heterogeneity [14] [15] [16] [17], to make mimic defense system more effective in defending network attacks. The heterogeneity discussed in these researches is basically based on 2-level similarity, which is, suppose there are three executors (1, 2, 3), the 2-level similarities are referred to the similarities for 1&2, 1&3 and 2&3. The sum of similarities is lower, the system is considered to be safer. There are mainly three kinds of algorithms to choose executors: the maximum heterogeneous algorithm (MHA) [14] and the optimal mean distance algorithm (OMDA) [15], and random seeds scheduling method proposed by qinrang Liu [16] which require the similarity of the working executors below a certain value. All these studies are

base on 2-level similarities. However， no attack model has been proposed to prove their defense effect. In fact， as the number of working executors grows， 2-level similarity become less important. Jiexin Zhang measures heterogeneity based on diversity combined with the second entropy of species differences， gives a new evaluation standard of heterogeneity， and concludes that the heterogeneity will reach max when there are 3 executors in the mimic system ［17］， and the heterogeneity will drop as executors number increase when the number is above 3. But only heterogeneity is taken into consideration without relevant attack model.

At present， only the heterogeneity has been introduced into the voting strategy of the mimic defense system， and consider the system is safer when the executors are more heterogeneous. However， There are only assumption， no research has been done to describe the network attack model targeting the heterogeneous mimic defense system， How heterogeneities between executors takes effect in defending against such attacks， how to Quantitatively analyze system failure probability based on heterogeneity of executors and network attack model.

In this paper， we analyze the mimic defense system， supposing executors are composed of component implementations， so a mimic defense system can be described as a matric of component implementations. We consider the characteristic of network attack， do some assumptions and abstract a network attacking model. We analyze the key factors that influence the defense effect of the system， extracts the component diversity， binary division vector and relevant indexes as the key indicators. We innovatively introduce conditional probability voting strategy， compare it with majority voting algorithm （MVA） ［18］ and maximum heterogeneous algorithm （MHA） ［14］ by system failure probability and scalability， prove it is best both in theory calculation and experiments. Our contributions are：

We abstract an attack model for mimic defense system， which provides basis for computing system failure probability of different voting algorithms.

We extract binary division vector and relevant indexes to measure heterogeneity of mimic defense system， which also provide basis for computing system failure probability of different voting algorithm.

We innovatively propose conditional probability voting algorithm （CPVA）， which has lower system failure probability compared with classical MVA. we also provide the method to compute the system failure probability.

We prove their scalability of conditional probability voting is much better， the system failure probability does not increase as the number of working executors increase， not the same as MVA.

We do some experiments to prove our calculation， and prove conditional probability algorithm is best both in system failure probability and scalability.

In the rest， we begin by our problem abstraction in section 2， in which we present executor composition and network attack assumption. We present binary division vector and relevant indexes， based on these indexes and the attack assumption， we introduce CPVA， MVA and their system failure probability/scalability in section 3， In section 4 we present experiments and some evaluations. We conclude our paper in section 5.

2 Problem abstraction

2.1 Composition of Heterogeneous Executors

With the development of open source technology， more and more software and hardware adopt agile development approach to system integration. Reusing mature units can effectively reduce the cost of development， easily achieve the design goal， and improve system reliability. However， the development of code reuse technology makes it difficult in choosing heterogeneous executors. The shared units of different executors will bring homologous vulnerabilities/backdoor. The correlation between different executors can be analyzed through source code analysis， source tracing and gene analysis， etc. Generally， the more shared codes are reused， the more vulnera-

ble mimic defense system will be.

Executors are usually composed of a series of modules, such as CPU, operating system, middleware, application, etc. Each module is composed of several components. For example, the application can be divided into component 1, component 2, ..., component M, etc. A component is the smallest unit of the executor, which is atomic (cannot be divided), and its implementations are different from each other. Then each executor can be represented by a component implementation vector $z^i = (g_1^i\ g_2^i \cdots g_N^i)$, where N is the number of components contained in an executor and each component may have several implementations.

In reality, the combinations of component implementations are limited due to compatibility problems between different modules or components, such as, only several specific types of middleware are suitable for a specific type of application. Actually, only mature combinations can be used due to efficiency or reasonable costs. Because the executors are composed of component implementations, so we describe a mimic defense realization by a matrix as shown below. There are 5 executors in the mimic defense system, each executor has 4 components, each component has 3 implementations.

$$\begin{pmatrix} z^1 \\ z^2 \\ z^3 \\ z^4 \\ z^5 \end{pmatrix} = \begin{pmatrix} a_1 & b_1 & c_3 & d_3 \\ a_2 & b_2 & c_2 & d_1 \\ a_1 & b_2 & c_1 & d_2 \\ a_2 & b_1 & c_2 & d_2 \\ a_3 & b_3 & c_1 & d_1 \end{pmatrix}$$

2.2 network attack assumption

We assume that input agent, scheduler, and feedback controller shown in Fig 1 are safe from network attacks because they are realized by hardware logic. Besides, the input agent is only responsible for input data distribution. Scheduler and feedback controller are responsible for data comparison and configuration interaction, their logic is simple, and can be proved safe through formal analysis. So, failure probability of the mimic defense system is only related to the vulnerability of the executors.

Hypothesis 1: every attack against different vulnerability/backdoor will produce different abnormal output.

For the common attack methods, such as a middleware with a hidden background listening port, the communication docking with it will inevitably produce different output. Most attacks need to determine the attack effect through the output, except some unconventional attacks such as side channel attacks, but the probability is low, which is not considered in this paper.

Hypothesis 2: when the high-level vulnerability / backdoor is attacked, the executors who share the vulnerability will generate the same output.

This assumption is true in most cases, such as a middleware with a hidden background listening port. But may not be true in other cases, such as buffer overflow attack. Although applications have the same buffer overflow vulnerability, due to different underlying processors, the same binary code can only be effectively executed on a specific platform (such as arm architecture processor), but not on another platform (such as X86 processors). However, attackers can also use other advanced attack methods such as branch prediction, making them to generate the same results on different platforms.

Hypothesis 3: each vulnerability / backdoor of the system have the same probability being attacked.

As described in section 1, Human programmers will leave one software vulnerability within every piece of 1000 to 1500 lines of code. Then the total number of vulnerability/backdoor of each component can be estimated by its code volume, and the attacking probability of each component can be estimated by the total number of vulnerabilities / backdoors contained in the component.

Hypothesis 4: different implementations of each component have the same attacking probability.

Due to the unpredictability of vulnerability/backdoor, the functions of the same component are the same, so the amount of code is basically the same. If the skill levels of programmers are similar, the number of vulnerabilities will be proximate. So, the

error probabilities of each component's different implementations are basically the same.

Hypothesis 5: only one vulnerability / back door can be attacked at a time, and the attacked executor will be cleaned in a short time.

As the multithreading on a single core computer is actually serial at the micro-level, the attack is also serial microscopically in most cases, so only one vulnerability/backdoor can be attacked at a time. It is easy for a single executor to be breached. But the cleaning time is generally short, and the probability of another successful attack in a short time is low. Therefore, this paper does not consider the case of degraded attack, that is, there is no successful attack at another executor in the cleaning process, otherwise the mimic defense system will eventually fail.

According to the above assumptions, we can assume the probability of successfully attacking each component is β_i, which should satisfy $\sum_i \sum_{\varphi_i} \beta_i = 1$, φ_i is the number of implementations for component *i*. The mimic defense system will generate wrong output if some shared vulnerability / backdoor is attacked and its result is voted through by scheduler. If an executor is judged to have produced an error, it will be cleaned.

C Problem to be solved

The mimic defense system can effectively defend attacks against known or unknown vulnerability/ backdoor. Take the 5-executors system in part A for example, suppose MVA is adopted, that is, the output result is considered to be correct if it is shared by majority number of executors. Then it can be seen that no matter which component implementation is attacked, at most two executors will generate the same incorrect results, and the incorrect result will not be voted through, so the system will be safe.

Voting algorithm [18] [19] [20] plays an important role in system failure probability of the mimic defense system. For the example shown in part A, if other voting algorithms are adopted, such as a weighted voting algorithm, that is, the results of executors 1 and 3 are preferred if they are the same. If this algorithm is adopted, when the vulnerability/ backdoor in component implementation a_1 is attacked, it will cause the wrong result to be voted through and the system fails to generate correct output. So, two main problems are considered in this paper:

When the output results of executors in the mimic defense system are inconsistent, which voting algorithm should be used to minimize the failure probability of the system? Is MVA the best one?

How scalability of the voting algorithm will be with the increase of executors? Can the increase of executors reduce the failure probability of the system? Which candidate executors should be selected to reduce the failure probability?

3 Conditional probability voting algorithm and majority voting algorithm

3.1 Binary Division vectors

In order to effectively identify the degree of heterogeneity of the mimic defense system, the concepts of diversity and binary division vector are extracted to characterize the heterogeneity of components and executors.

Component diversity φ_i: the implementation number of the component g_i, which can be calculated by formula $\left| \bigcup_i g_k^i \right|$.

Taking the component g_1 in section 2.1 as an example, its implementation set is $\left(a_1\ a_1\ a_1\ a_2\ a_3\right)^T$, then union the contents and get the set $\left\{a_1\ a_2\ a_3\right\}$, which contain 3 elements, so the diversity of component *a* is 3.

Binary division vector η_i^k: for a component, divide the same implementations into one group and other implementations into another group, which is called a binary division, and use a vector to represent. Assign corresponding values in the vector of the same implementations to 1 and others to 0, which is called the binary division vector of the component implementations.

The number of binary division vectors for a component is equal to the diversity of the component. For example, the implementations of $g_1$in section 2.1 is $(a_1\ a_1\ a_1\ a_2\ a_3)^T$, then there are 3 binary division vectors for component g_1, the binary division vector of a_2 is $(0\,0\,0\,1\,0)^T$, the binary division vector of a_3 is $(0\,0\,0\,0\,1)^T$, the binary division vector of a_1 is $(1\,1\,1\,0\,0)^T$.

Complement binary division vector $\sim\eta_i^k$: which is reversing every element in the binary division vector. if adding binary division vector to its complement binary division vector, the result should be a vector whose elements are all 1.

Based on the binary division vector $(1\,1\,1\,0\,0)^T$ of component implementation a_1, reverse all the elements in it, and its complement vector will be $(0\,0\,0\,1\,1)^T$.

Isomorphic number of binary division vector λ_i^k: the number of elements whose value is equal to 1.

There are three 1 in the binary division vector of component implementation a_1, which is $(1\,1\,1\,0\,0)^T$, then there are three a_1, and Isomorphic number of $(1\,1\,1\,0\,0)^T$is 3.

Property 1: Assuming that there are T executions in the mimic system, there will be at most 1 implementation whose isomorphic number is not less than $\left\lfloor\frac{T+1}{2}\right\rfloor$. If the diversity of a component is 2, there must be one implementation whose isomorphic number is not less than $\left\lfloor\frac{T+1}{2}\right\rfloor$.

3.2 Majority voting algorithm

MVA adopts majority rule. If the number of executors generating the same result is the largest, the corresponding result will be adopted.

（A） Voting algorithm

The decision strategy of majority voting algorithm is shown as follows:

If there is only one result, the result is considered to be the final output result;

Divide the executors by their results, put executors with the same result into a group G_k. According to the hypothesis in in section 2 only one vulnerability/backdoor is attacked at a time, so there are usually 2 groups, suppose they are G_1 and G_2.

If $|G_1|>|G_2|$, then select the result of G_1 as the final output; otherwise, select the result of G_2 as the final output;

Clean the executors which have been arbitrated to be abnormal.

If the similarity of most executors in the mimic defense system was 0, the failure probability of the system will be 0. Assuming that there are T executions in the mimic system, and the Isomorphic number λ_i^k of all the binary division vectors does not exceed $\left\lfloor\frac{T+1}{2}\right\rfloor$. According to the attack assumption in section 2, the erroneous results are the same only when attacking at the executors with the same vulnerability/backdoor. Then if all the binary division vectors does not exceed $\left\lfloor\frac{T+1}{2}\right\rfloor$, once there is an attack microscopically, most executor doesn't have the vulnerability/backdoor, and will generate the correct results, then the system is absolutely safe. So, if the 2-level similarity of 3 executors' system is 0, or 3-level similarity of 5 executors' system is 0, then the failure probability of these system is 0.

（B） system failure probability

Input: component number N, executor number T, probability of component being successfully attacked β_k, component diversity φ_i, isomorphic number of binary division vector λ_i^k

Output: failure probability of MVA *msum*

1） Vg=NULL

2） for i = 1: N

3） for k = 1: φ_i

4） If（λ_i^k>=$\left\lfloor\frac{T+1}{2}\right\rfloor$）

5） add i in Vg

6） endfor

7） endfor

8） msum = 0

9） for each index in *Ug*

10） $msum = msum + \beta_k$

11） endfor

The program mainly consists of two parts. Line 2-7 saves all the components with implementations whose Isomorphic number is no less than $\left\lfloor \frac{T+1}{2} \right\rfloor$ in stack *Vg*， line 9-11 sums the attacking probability corresponding to the components in the stack to get system failure probability.

From the algorithm we can see only component that has $\left\lfloor \frac{T+1}{2} \right\rfloor$ and above implementations matters for system failure probability of *T*-executors mimic defense system. Other levels of similarity will have no effect.

（C） scalability

The majority voting algorithm is relatively simple， compare the number of executors that share the same result and choose the result with largest number as the final output. However， there is a disadvantage about this algorithm， that is， increasing an executor with shared component implementation may increase the failure probability of the system， such as the following example.

$$\begin{pmatrix} a_1 & b_1 & c_1 & d_1 \\ a_1 & b_1 & c_1 & d_1 \\ a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \end{pmatrix}$$

Although the failure probability of the subset（executors at rows 3， 4， 5） is 0， but as a whole（executors at rows 1， 2， 3， 4， 5） the system failure probability is 1/3 if attacking probability of each component is equal. Because the Isomorphic number for component implementation a_1， b_1， c_1， d_1 is more than half， if the vulnerability/back door in components a_1， b_1， c_1， d_1 is attacked， errors will be generated and the system will output incorrect results.

Although the failure probability of the system may not decrease with the increase of executors， the failure probability of the system may decrease by selecting appropriate executors. Take the executors shown in section 2.1 for example， any 3 executors from the system have a certain failure probability， but the overall 5 executors would never fail.

According to the characteristics of MVA， similar to the Buckets effect， avoiding the long and making up the short， the following conditions are generally required in order to reduce the failure probability of mimic defense system using MVA：

Do not add the same implementation of a component so that its Isomorphic number exceeds $\left\lfloor \frac{T+1}{2} \right\rfloor$.

Add different implementation of a component or balance the same implementation of a component so that its maximum implementation is less than $\left\lfloor \frac{T+1}{2} \right\rfloor$.

3.3 Conditional probability voting algorithm

The basic idea of CPVA is when the results generated by executors are different， we should analyze exactly which components will cause current groping of results， remove the impossible components that may cause the grouping， and find components group with less error probability， adopt the corresponding result as the final output. Because this algorithm determines the output according to the results grouping， it is called conditional probability voting algorithm.

（A） Voting algorithm

The decision strategy of CPVA is described as follows：

If there is only one result， take the result as the output result；

The executors which generated the same results are divided into one group G_K， generally there are only two groups， assumed as G_1 andG_2；

Calculation β_{G_1} and β_{G_2}， $\beta_{G_1} = \sum_k \beta_k, if\ k \in \bigcap_{i \subset G_1} (z^i - \bigcup_{j \subset G_2} z^j)$ ， $\beta_{G_2} = \sum_k \beta_k, if\ k \in \bigcap_{i \subset G_2} (z^i - \bigcup_{j \subset G_1} z^j)$

If $\beta_{G_1} > \beta_{G_2}$ ， the result of G_2shall be used， otherwise， the result of G_1shall be used；

Clean the abnormal executors which have gener-

ated wrong result.

CPVA groups executors according to their results， analyzes the components grouping situation， remove those components that are not able to cause the grouping situation， and select the set of components that have low probability. According to the assumption in section 2， in each case only one vulnerability/backdoor could be attacked， so the final result of the system can be divided into G_1 andG_2. We do elimination and comparison mainly in three steps：

If there is the same implementation of one component in both G_1 and G_2， the error cannot be caused by attacking vulnerability/backdoor in this component. So the same component implementation in the G_1 and G_2 need to be removed， then we can get two eliminated sets G'_1 and G'_2， that is， $z^i - \bigcup_{j \subset G_2} z^j$and $z^i - \bigcup_{j \subset G_1} z^j$.

If there are multiple implementations of one component in G'_1 or G'_2 ， the error cannot be caused by vulnerability/backdoor in these different component implementations， or there will be different results in G'_1 or G'_2 ， we use intersection to eliminate different component implementations， we can get two eliminated sets $G''_1$and G''_2 ， that is， $\bigcap_{i \subset G_1} (z^i - \bigcup_{j \subset G_2} z^j)$ and $\bigcap_{i \subset G_2} (z^i - \bigcup_{j \subset G_1} z^j)$，

Now we get the shared component implementation， G''_1 and G''_2 ， which could cause this kind of grouping situation， then accumulate probabilities of components contained in G''_1 and G''_2， and select the result with lower probability as final output， so the overall system failure probability will be relatively lower.

（B） system failure probability

Input： component number N， executor number T ， probability of component being successfully attacked β_k， component diversity φ_i， binary division vector η_i^k， isomorphic number of binary division vector λ_i^k

Output： system failure probability of CPVA *csum*

1） *Gu*=NULL
2） for $i = 1$： N
3） for $k = 1$： φ_i
4） if （$\lambda_i^k >= \left\lfloor \frac{T+1}{2} \right\rfloor$）
5） *Gu* = union （*Gu*， η_i^k）
6） endif
7） endfor
8） endfor
9） csum = 0
10） $G_L = G_S$=NULL
11） for each vctorl in *Gu*
12） for i = 1： N
13） for k = 1： φ_i
14） if（*vctorl* == η_i^k）
15） add *i* in G_L
16） else if （~*vctorl* == η_i^k）
17） add *i* in G_S
18） endif
19） endfor
20） endfor
21） *lsum* = *ssum* = 0
22） for each *k* in G_L
23） *lsum* = *lsum* + β_k
24） endfor
25） for each *k* in G_S
26） *ssum* = *ssum* + β_k
27） endfor
28） if *lsum* < *ssum*
29） *csum* = *csum* + *lsum*
30） else
31） *csum* = *csum* + *ssum*
32） endif
33） endfor

The program is mainly divided into two parts， 1-8 lines union all the binary division vector whose corresponding isomorphic implementations are not less than $\left\lfloor \frac{T+1}{2} \right\rfloor$， save the union result in a stack *Gu*. 9-33 lines take out every element *vctorl* in the stack *Gu*， compare them with all binary division vectors in the system， save sequence number of the

component who contain one implementation whose binary division vector is equal to *vctorl* in G_L， save sequence number of the component who contain one implementation whose binary division vector is complemental to *vctorl* in *Gs*， and then compare sums of attacking probabilities for the components in G_L and *Gs*， accumulate the relatively lower probability， and finally get system failure probability.

From the description of the algorithm， it is easy to see that the failure probability of CPVA is smaller than that of MVA. Because when all the failure probabilities in G_L is not greater than those in *Gs*， the failure probability of CPVA is equivalent to the majority voting algorithm. Otherwise CPVA can get a lower failure probability.

It is easy to see that when the binary division vector is $(11\cdots10)^T$， that is， Isomorphic number is *N*-1， there must exist another binary division vector $(00\cdots01)^T$. But conversely， when the partition vector is $(00\cdots01)^T$， there may exist multiple binary division vectors， such as $(10\cdots00)^T$ ， $(01\cdots00)^T$， and so on. So， the number of binary division vector $(00\cdots01)^T$ must be more than binary division vector $(11\cdots10)^T$in the system. In the mimic defense system composed of 3 heterogeneous executors， the binary division vectors in G_L are always chosen， then CPVA is equivalent to MVP.

（C） scalability

CPVA can effectively perform better scalability than MVA， such as the example of 5 executors in section 3.2. If the system adopts CPVA， the same grouping which makes MVP wrong， that is， one grouping of executors 1， 2 and 3， and another grouping of executors 4 and 5， Since there is no shared component implementations for executors 4 and 5， the conditional probability voting algorithm will choose results of executor 4 and 5 as the final output， so system failure probability is 0.

In general， it can be proved that system failure probability will not increase with increasing executors in the system， that is， the failure probability of any executors set is no less than that of its extended set. Because if binary division vectors are equal， their subsets must be equal. So， the number of shared binary division vectors in G_L and $G_S$must be more than that of the extended set. System failure probability comes from shared binary division vector， so system failure probability of the extended set must be no greater than system failure probability of the original set.

Although CPVA has good scalability， increasing the number of executors will not increase the system failure probability， but it does not always reduce system failure probability. To reduce the failure probability of heterogeneous executors， one of the following conditions should be satisfied：

Executor with different implementation of the component whose diversity number is 1 in the original system.

Executor with different implementation of the component in the group （G_L or G_S） with lower probability when the diversity number is more than 1 in the original system.

4 experiments and analysis

Our experiment assume that every executor has N components， implementations of each component are randomly generated within a certain range of M. M for each component is different to distinguish different components， such as［101，110］，［201，220］ and so on. It is assumed that attacking probability for every component implementation is the same， so system failure probability of different voting algorithm is easy to compute. Experimental program is written with MATLAB. Three kinds of voting algorithms are tested， the maximum heterogeneous algorithm （MHA） [14]， MVA as shown in section 3.2， CPVA as shown in section 3.3.

For mimic defense system， if the number of executors -N in the system increases， the system cost， power consumption， etc. would also increase. So， during the realization of mimic defense system， the number of executors would be relatively

small， because of that we limit the number of executors within 10 for system failure probability test and scalability test.

4.1 3-executors experiment

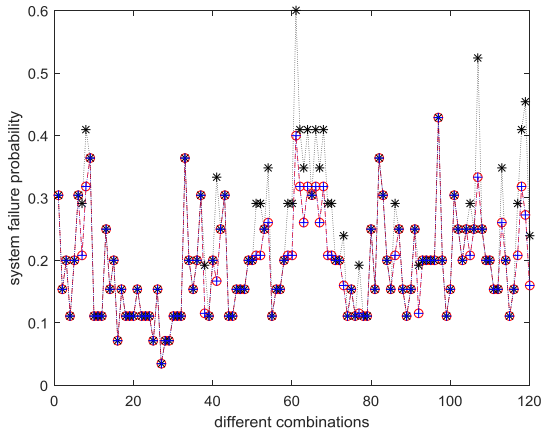


Fig. 2 system failure probiblities of MHA， MVA and CPVA when N=10 M=5

We choose 10 and 100 components， 5 and 10 component implementations range for tests， so，（N， M） combinations are （10， 5），（100，5），（10，10），（100，10）. We randomly generate a 10-executor set， and it contains 120 3-executor subsets， all the subsets are evaluated. The results of MRA are shown with black * sign， results of MVA are shown with red O sign， and results of CPVA are shown with blue + sign.

It is shown in the four examples that system failure probability of MVA completely coincides with system failure probability of CPVA， which is consistent with the conclusion in section III. In the case of 3 executors， CPVA is equivalent to MVA. It also can be seen that when N=10， the failure probability of MRA is basically the same as that of MVA， but when N=100， t the failure probability of MRA is far better than that of MVA. Because when the number of components is large， there is a great probability that the same implementation will be shared by 3 executors. In this case， MRA doesn’t take 3-level similarity into consideration， so it will result in inexact evaluation. When M increases， we can see the difference between MRA and MVA/CPVA become smaller， because with the increase of component implementations range， the probability that the same im-

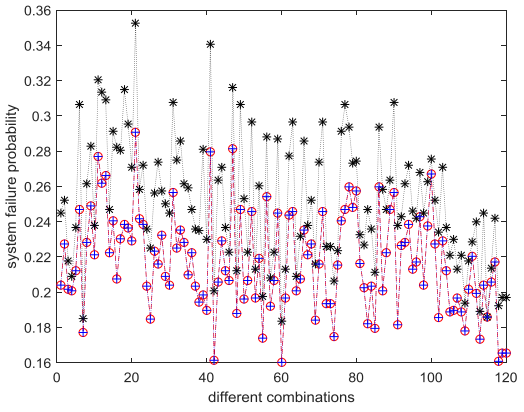


Fig. 3.system failure probiblities of MHA， MVA and CPVA when N=100 M=5

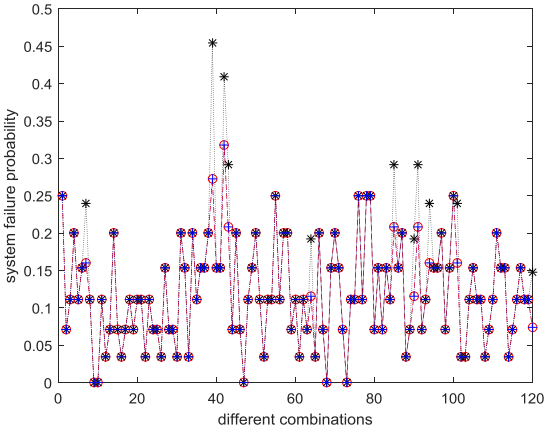


Fig. 4 system failure probiblities of MHA， MVA and CPVA when N=10 M=10

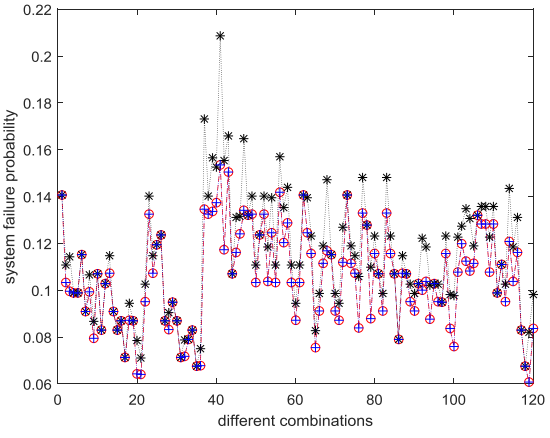


Fig. 5 system failure probiblities of MHA， MVA and CPVA when N=100 M=10

plementation of a component being shared by 3 executors become lower.

4.2 5-executors experiment

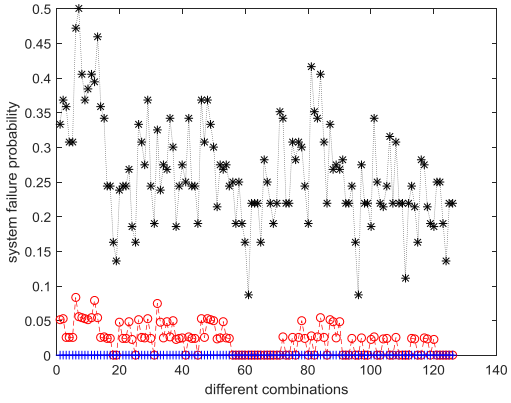


Fig. 6 system failure probiblities of MHA， MVA and CPVA when N=10 M=10

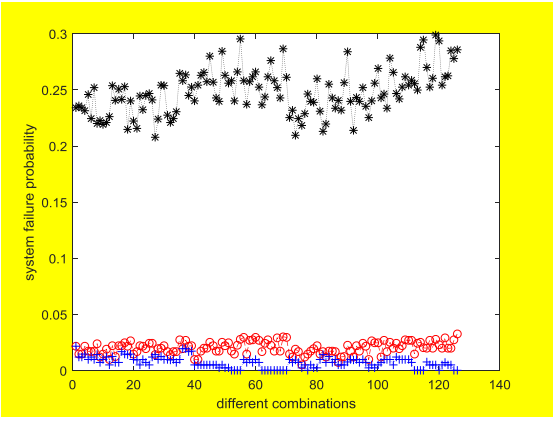


Fig. 7 system failure probiblities of MHA， MVA and CPVA when N=100 M=10

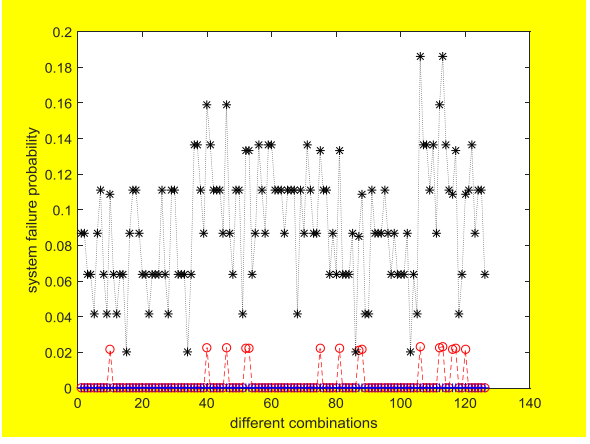


Fig. 8 system failure probiblities of MHA， MVA and CPVA when N=10 M=20

We choose 10 and 100 components， 10 and 20 component implementations range for tests， that is， （N， M） sets are （10， 10），（100，

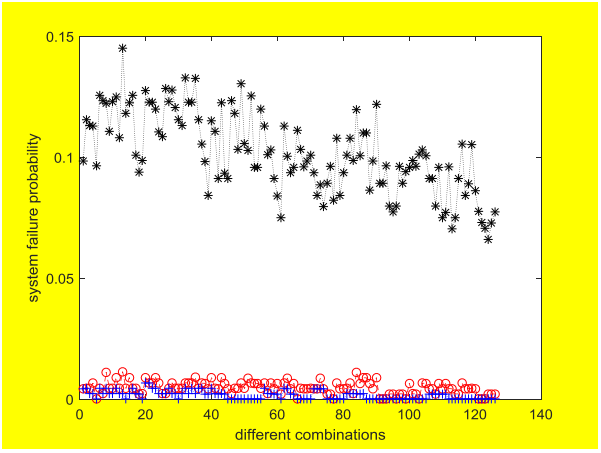


Fig. 9 system failure probiblities of MHA， MVA and CPVA when N=100 M=10

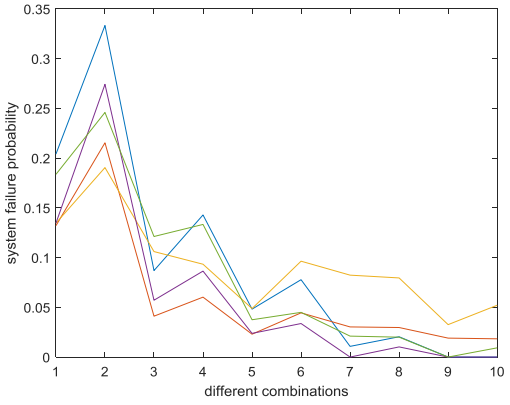


Fig. 10 system failure probiblities of MHA with excutors increase

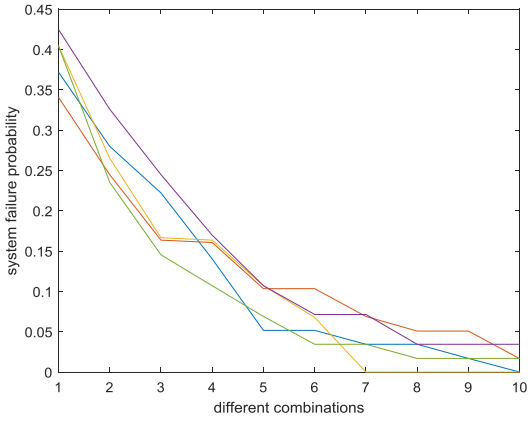


Fig. 11 system failure probiblities of CPCA with excutors increase

10）， （10，20） and （100，20）. We randomly generate a 9-executor set， and it contains 126 5-executor subsets， all the subsets are evaluated. The results of MRA are shown with black * sign， results of MVA are shown with red O sign， and results of CPVA are shown with blue + sign.

Apparently different from the phenomenon shown in 3-excutors system， the system failure probability of MVA is remarkably lower than that of MRA. Because in 5-excutor system， 3-level and above similarities are determining factors for system failure probability， but MRA only takes 2-level similarity into consideration. So the result is remarkably wrong. And it can be predicted that the error will be larger with the increase of executors. It is shown in these four examples that system failure probability of CPCA is different from system failure probability of MVA. It performs lower and better when M become larger or N become smaller. Because with the increase of components or decrease of component implementations range， the probability that 3-level and above shared component implementation become bigger， the imbalance between GL and GS also increase， therefore CPCA will benefit more from the imbalance.

4.3 Scalability experiment

First 3 random executors are created as the basic mimic defense system， and then randomly created executors are added gradually to the existing system， calculate the failure probability of MVA and CPVA， and then analyze the scalability of the algorithm. The results are shown in Fig 10 and Fig11， Lines of different colors in the figure represent different random tests using the same method. It can be seen that the failure probability of MVP will decrease with the increase of executors in general， but it is not strictly decrease， and it will randomly jitter. While he failure probability of CPVA will strictly decline both totally and partly without jitters， which is consistent with the analysis in section 3.

In the large number voting strategy， it can be seen that the system failure probability of even number （2k） actua-tors are significantly higher than that of 2k-1 executors， so generally the odd number of executors will be adopted in the MVA， but the CPVA does not have this problem. The system failure probability of even number executors 2k is not greater than that of 2k-1 executors， and in some cases fail-

ure probability will significantly decline.

5 Conclusion

Based on the principle and architecture of mimic defense system， this paper analyzes the voting algorithms of the scheduler， including majority voting algorithm and conditional probability voting algorithm， summarizes the advantages and disadvantages of various voting algo-rithms， and does corresponding experiments. Experimental results show that conditional probability voting algorithm can improves the security and reliability of the system greatly.

However， the assumption of the attack model in this paper is relatively strict. There are some cases in which the executors with the same vulnerability generate different results and with different vulnerabilities generate the same results. The probabilities of different implementations of the same component being attacked may be different. Therefore， the algorithm can be extended to be suitable for more examples.

6 Acknowledgments

This work was supported by the the National Natural Science Fund for Creative Research Groups Project （no. 61521003）. We thank Prof. Jiangxing Wu， an academician of Chinese Academy of Engineering， who has proposed the main idea of mimic defense system.

REFERENCES:

[1] The Linux Kernel Orgnization, The Linux Kernel Archives，https：// cdn. kernel. org/pub/linux/kernel/. 2020.

[2] Wang xu. Research on the Semantic Annotation of Software Vulnerability Source Codes［D］, Master Thesis，Dept. of Software Eng. ，Northwestern Polytechnical Univ. ，Xian，China，2018.

[3] LAMGNER R，“Stuxnet: Dissecting a cyberwarfare weapon," IEEE Security & Privacy，vol. 9，no. 3，pp. 49－51，2011.

[4] Boyd S. W. , Gaurav S. K. , Locasto M. E. , et al. : 'On the general applicability of instruction-set randomization', IEEE Trans. Dependable Secur. Comput. ，2010，7,（3），pp. 255－270

[5] Nguyen-Tuong A. , Evans D. , Knight J. C. , et al. : 'Security through redundant data diversity'. IEEE/IFPF Int. Conf. Dependable Systems and Networks，June 2008

[6] Manadhata P. K. , Wing J. M. : 'An attack surface metric', IEEE Trans. Softw. Eng. ，99 (PrePrints)，2011，37,（3），pp. 371－386

[7] Zhuang R. , DeLoach S. A. , Ou X. : 'Towards a theory of moving target defense'. Proc. First ACM Workshop on Moving Target Defense. ACM，2014，pp. 31－40

[8] Zhuang R. , Bardas A. G. , DeLoach S. A. , et al. : 'A theory of cyber attacks'. Proc. Second ACM Workshop on Moving Target Defense. ACM，2015

[9] Hong J. B. , Kim D. S. : 'Assessing the effectiveness of moving target defenses using security models', IEEE Trans. Dependable Secur. Comput. ，2015，10,（11），pp. 1545－5971

[10] Jiangxing W. Cyberspace Mimic Defense-Generalized Robust Control and Endogenous Security，Berlin，Gemany: Springer International Publishing，pp. 15-205，2020.

[11] Jiangxing W. Research on cyber mimic defense［J］. Journal of Cyber Security，2016，1(4)：1-10.

[12] Hu H , Wu J , Wang Z , et al. Mimic defense: a designed-in cybersecurity defense framework［J］. IET Information Security，2018，12(3)：226-237.

[13] WEI S, YU H, GU Z Y, et al. Architecture of mimic security processor for industry control system［J］. Journal of Cyber Security，2017，2（1）：1-12.

[14] Conglin S, Shuangxi C, Chunming W. Adaptive mimic defensive controller framework based on reputation and dissimilarity［J］. Journal of Communications，2018，39(S2)：173-180.

[15] YAO W B, YANG X Z. Design of selective algorithm for diverse software components［J］. Journal of Harbin Institute of Technology，2003，35(3)：261-264.

[16] Qinrang L, Senjie L, Zeyu G. Heterogeneous redundancies scheduling algorithm for mimic security defense［J］. Journal of Communications，2018，39（07）：192-202.

[17] Zhang JX, Pang JM, Zhang Z. Quantification method for heterogeneity on Web server with mimic construction. Ruan Jian Xue Bao/Journal of Software，2020，31(2)：564−577.

[18] , B. Voting algorithmsParhami［J］. IEEE Transactions on Reliability，1994，43(4)：617-629.

[19] Tong Z , Kain R Y . Vote assignments in weighted voting mechanisms［J］. IEEE Transactions on Computers，1991，40(5)：664-667.

[20] JAMALI N, SAMMUT C. Majority Voting: Material Classifiction by Tactile Sensing Using Surface Texture［J］. IEEE Trans on Robotics，2011，27(3)：508-521.

About the authors

Shuai Wei was born in Dengzhou， Henan， China， in 1984. He received the B. S. degree in computer science and technology， the M. S. degree in Software Engineering， and the Ph. D. degree in High performance computing and Parallel Compiling in 2005， 2008， and 2012， respectively. He is currently an Research Assistant with NDSC. He has authored over 40 applied patents and published articles. His research interests include high performance computing， cyber securi-

ty， and machine learning.

Huihua Zhang was born in Taizhou， Jiangsu， China， in 1983. He received the B. S. degree n computer science and technology， the M. S. degree in Software Engineering in 2005， 2008， respectively. He is currently a Research Assistant with NDSC. He has authored over 40 applied patents and published articles. His research interests include high performance computing， cyber security， and machine learning.

Wenjian Zhang was born in Shangqiu， Henan， China， in 1987. He received the B. S. degree in communication engineering， the M. S. degree in communication and information systems in 2010， 2013 respectively. He is currently a Research Assistant with Wuxi Confidential Technology Service Center. He has authored over 10 applied patents and published articles. His research interests include high performance computing， cyber security， and machine learning.

Hong Yu was born in Ziyang， Sichuan， China， in 1988. She received the B. S. degree in computer science and technology， the M. S. degree in Software Engineering in 2011， 2013 respectively. He is currently an Research Assistant with NDSC. He has authored over 20 applied patents and published articles. Her research interests include high performance computing， cyber security， and machine learning.

Machine Learning Algorithms in Encrypted Traffic Classification: An overview

ZHANG Surong[1], BU Youjun[1], ChenB [1], ZhangQ [1,2], LuXY [1], WangH [3]

1.Information Technology Institute, PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, China;

2.Zhongyuan Network Security Research Institute, Zhenzhou University, 450000, China;

3.Network Communication and Security Purple Mountain Laboratory, 211100, China

Key words: encrypted traffic; traffic classification; machine learning; deep learning

Abstract. In the big data era， the rapid growth of traffic， especially encrypted traffic， has brought great challenges to network management based on traffic classification. The emergence and rise of machine learning provides a good solution to this problem. After decades of development， machine learning algorithms are constantly optimized and updated. A series of more advanced improved algorithms have emerged， and many of them have been successfully applied to the encrypted traffic classification. In this paper， the machine learning algorithms involved in encrypted traffic classification is classified based on extensive reading of literature. At the same time， the new representative research accomplishments of the algorithm in this field is summarized in detail as far as possible. Finally， there are some discussions about the existing problems and their challenges， as well as opportunities for encrypted traffic classification， in order to provide help for the follow-up research.

1 Introduction

The Cisco Visualized Network Index Forecast Research Report ［1］ pointed out that， global IP traffic will increase approximately three times at a compound annual growth rate of 24% in the five years from 2016 to 2021. According to *the 45th Statistical Report on Internet Development in China* ［2］， the scale of China's Internet users has reached 904 million by March 2020.

With the substantial increase in the scale of Internet users and the Internet penetration rate， the network security and data protection awareness of people have been continuously increased. *The Cisco Encrypted Traffic Analysis White Paper* ［3］ shows that， as more and more enterprises achieve digitalization， a large number of services and applications use encryption as the main method of protecting information. The encrypted traffic growth trend is shown in Fig. 1.

Due to the rapid development of traffic， especially encrypted traffic， traffic classification technology has received extensive attention of academia and industry. Traffic classification， essentially belonging to the traffic identification and classification， is of great significance to the network management， network planning， and network flow model research. At present， there are three main traffic classification methods： port number-based method， payload-based method ［4］ and machine learning-based

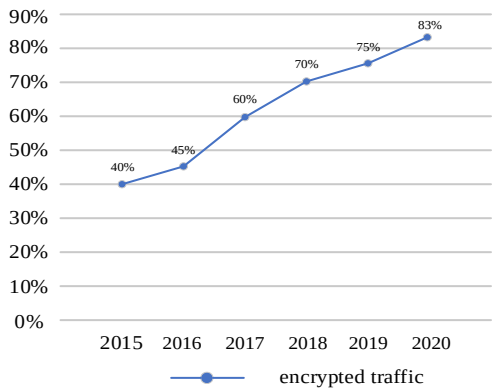


Fig. 1 The encrypted traffic growth trend

method. The first two methods are not suitable for encryption traffic classification due to the decrease of accuracy, privacy problem and other issues. The third method is the mainstream method in this field, which has many advantages such as fast detection speed, good performance, high flexibility and robustness.

Machine learning is an artificial intelligence implementation method drived by data. It learns knowledge and regularities from sample data, and then makes practical inference and decision. The origin of machine learning technology can be traced back to 1958 [5] or even earlier. In the past few decades, machine learning has experienced a series of periods from emergence to rise, then to vigorous development. There have been traditional supervised learning and unsupervised learning, deep learning, ensemble learning, transfer learning and other algorithms, which make machine learning flourish in the current era of artificial intelligence and shine brilliantly in various technical fields.

Machine learning began to be gradually applied in the traffic classification around 2005, and there have been many research accomplishments [6-8] at present. In this paper, based on an in-depth review of existing research, we classify and summarize these machine learning methods used in encrypted traffic classification, point out the advantages and disadvantages of various methods, and look forward to future improvements and development directions.

The main contributions of this paper are summarized as follows:

1) We introduce the general process framework of encrypted traffic classification using machine learning algorithms, to help researchers understand the algorithm working principle more quickly.

2) We classify and summarize the machine learning algorithms used to encrypted traffic classification and analyze their advantages and disadvantages, providing help for the follow-up studies in this field.

3) Based on the summary, we propose improvements and development directions for the application of various machine learning algorithms in encrypted traffic classification.

2 General Framework for Encryption Traffic Classification

Encrypted traffic classification is to learn various features of encrypted traffic data through a model, and use them to classify the data into the correct category. In order to overcome the challenge of invisible encrypted traffic payload, researchers have proposed many machine learning algorithms. On this basis, Rezaei et al. [9] proposed a general process framework for traffic classification, comprising seven steps. These steps are shown as follows: clear classification goals, data collection, data processing, feature selection, model selection, training and evaluation, and regular evaluation updates.

The steps in the above framework are independent of each other, which are suitable for step-by-step encryption traffic classification models. But it fail to cover the later integrated end-to-end models, such as an end-to-end based one-dimensional CNN encrypted traffic classification method proposed by Wang Wei et al. [10] This method directly learns features from the original traffic data, without separate data feature extraction and other sub-steps. Considering this problem, Zhai et al. [11] further summarized on the basis of literature [9] and proposed a "six-step method" framework model suitable for most existing research, as shown in Fig. 2. The

steps in the "six-step method" are: research target positioning, data collection, data processing, model selection, training and evaluation, as well as application testing and improvement. We will briefly summary the content of each step proposed by the author as follows.

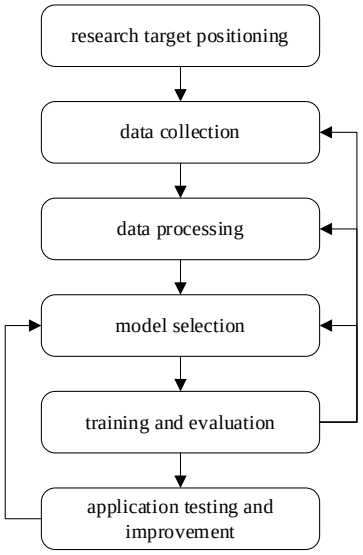


Fig. 2 [11] "six-step method" framework model

Research target positioning. It usually divided into encrypted and unencrypted identification, encryption protocol identification, encrypted service identification and malicious traffic detection.

Data collection. Constructing a suitable dataset is the basic and important link of traffic classification. The author pointed out that the data collection methods include direct collection method, script generation method and hybrid method. At the same time, the paper also summarizes the encrypted traffic dataset in the table form.

Data processing. This step includes data preprocessing and feature extraction. Data preprocessing converts the original data into the required form. Feature extraction is closely related to model classification accuracy, which is one of the most important step of all. In encrypted traffic classification, features are generally divided into three categories: basic features including quintuple, protocol number and so on, statistical features including the number of source packets and destination packets and other information, and time series features including packet duration and packet arrival time, etc.

Model selection. According to the size and dataset type, the appropriate algorithm model is selected, including traditional machine learning model which depends on artificial feature extraction and deep learning model which can automatically extract feature.

Training and evaluation. N-fold cross validation is usually used in training, which is generally 10 fold. The evaluation indexes used to evaluate the classification performance mainly include accuracy index and real-time index. The accuracy index includes accuracy rate, recall rate, precision, false positive rate and F1 score, while the real-time index refers to the time taken to complete the accurate detection of the first n packets in the flow.

Application testing and improvement. The model is used in the actual network environment to test its effectiveness and robustness. It is also regularly updated and improved to achieve stable and good performance.

The details of the specific steps are not repeated here. Readers can refer to literature [11] for in-depth understanding.

3 Machine Learning Algorithm for Encrypted Traffic Classification

As described in Section 1, after decades of development, there are a variety of improved machine learning algorithms, in which many have been successfully applied to encrypted traffic classification. We classify and summarize the machine learning algorithms involved in this field, in order to help the researchers have a clearer understanding of the existing research accomplishments, and can enlighten the future research direction. The classification framework of machine learning algorithms in this paper is shown in Fig. 3.

3.1 Traditional Supervised Machine Learning Algorithm

Supervised learning uses a set of known categories samples to adjust the classifier parameters,

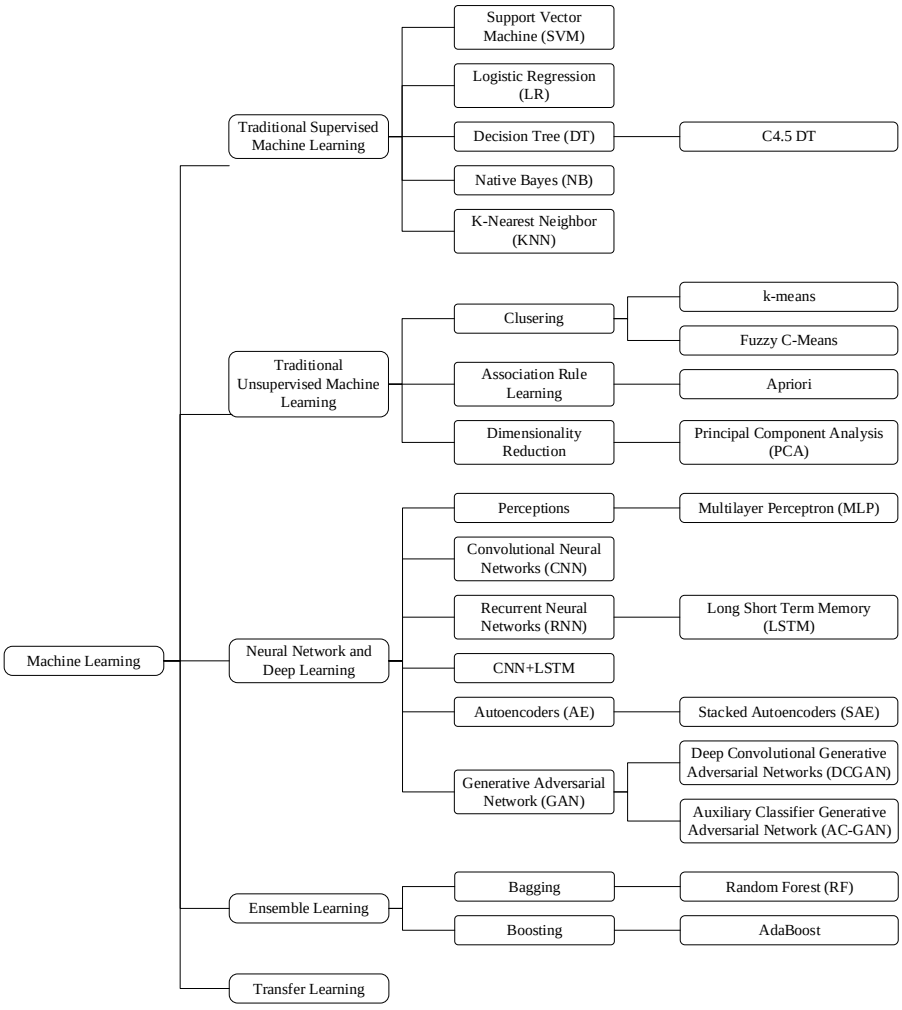


Fig. 3 The classification framework of machine learning algorithms in this paper

which aim to achieve the required performance. In supervised learning， training data has both features and labels. Through training， the machine learning model can predict its label after giving data.

In this paper， the traditional supervised machine learning is further subdivided into support vector machine （SVM）， logistic regression （LR）， decision tree （DT）， native Bayes （NB） and k-nearest neighbor （KNN） to introduce. The details are summarized in Table 1.

1) Support Vector Machine (SVM)

SVM uses the hinge loss function to calculate the empirical risk， and adds regularization term in the solving system to optimize the structural risk. As a sparse and robust classifier， its purpose is to find the separation hyperplane which can correctly partition the training dataset and has the largest geometric interval. So it is a geometry based approach.

In encrypted traffic classification， SVM is often used as a classifier to classify the input data into two categories： encrypted and unencrypted traffic identification， such as Skype-nonSkype traffic identification ［12］， SSH-nonSSH traffic identification ［13］ and VPN-nonVPN traffic identification ［14］. The focus of these researches usually do not lie in SVM itself， but using it as an auxiliary tool to prove the effectiveness of the new methods used in other steps of encrypted traffic classification. Although it has been proved that SVM has advantages in classification performance and robustness， it can not adapt to large-scale data samples and has great difficulties in solving multi-classification problems.

2) Logistic Regression(LR)

Although logical regression is called regression， it is actually a classification model. Essentially it assume that the data obey a certain distribution， and then use the maximum likelihood estimation method to estimate the parameters. So LR is a statistical method， which is often used in binary classification， and softmax function can be added to used for multi-class processing. The results of LR are easy to explain because there is a one-to-one correspondence between the data attributes and the model parameters.

In encrypted traffic classification， LR can be used to dimensionality reduction for extracted features ［15］， or as a classifier for input data ［16］［17］. Logistic regression models are usually combined with regularization methods which has different norms to reduce the model complexity and enhance its generalization ability. Among them， L1 logistic regression model and L2 logistic regression model are most commonly used. Logistic regression has the advantages of simple model， fast training speed and strong interpretability， but it also has some shortcomings such as unable to solve the nonlinear problem and unbalanced data problem.

3) Decision Tree (DT)

Decision tree is a tree structure， which usually has three steps： feature selection， decision tree generation and decision tree pruning. In the classification process， it first tests a certain feature of the sample from the root node， and allocates the sample to its subnodes according to the test results. At this time， each subnode corresponds to a feature value. It recursively tests and allocates the sample in this way until reaching the leaf node， and finally divides the sample into the class of the leaf node. Decision tree generation algorithms include ID3， C4.5 and CART. ID3 uses information gain as the criterion of feature selection， which can only deal with discrete attributes and is only suitable for binary classification problems. By introducing information gain ratio used to select features， C4.5 solves the ID3 problem that it can only deal with discrete attributes and tends to select features with high quantity. Cart is different from ID3 and C4.5 in that the tree generated by cart must be a binary tree. In other words， in any case， the internal nodes can only be bisected according to the attribute value.

In encrypted traffic classification， C4.5 decision tree model is most widely used， which is usually used as classifier in classification system. Like other traditional machine learning algorithms， decision tree models are not the focus of research in recent literatures. It usually appear as a comparison model of new methods to prove their advanced nature ［18］， or as classifiers to output results ［19］， which demonstrate the performance of the new methods used in other steps of encrypted traffic classification.

Visual analysis can be carried out in the DT algorithm， resulting in strong interpretability. At the same time， its time complexity is relatively low， and a good model can be trained on a dataset with a large amount of data in a short time. But it also does not apply to unbalanced datasets， and it is easy to ignore the inherent correlation between samples in the dataset.

4) Native Bayes (NB)

Naive Bayes is one of Bayes classifiers， which is the simplest and most commonly used classifier. The reason why it is called naive is that it assumes that the input attributes are independent of each other， and there will not be a situation in which the attribute with a large proportion affects the classification result. However， it leads to some constraints in practical application， that is， if there is a correlation between attributes， the classification accuracy will be reduced. Fortunately， the naive Bayes still does well in most cases. The basic steps is as follows： firstly， the joint probability distribution from input to output is learned through the given training set with the assumption of independence between feature attributes， and then based on the learned model， the output with the maximum posterior probability can be obtained by utilizing the input ［24］.

Naive Bayes is often used as classifier ［25-

26], which outputs the probability value of each class, and takes the class with the largest probability as the classification result. Although it is simple and has stable classification efficiency, naive Bayes is not widely used in encrypted traffic classification due to its assumptions.

5) K-Nearest Neighbor (KNN)

K-nearest neighbor is one of the simplest algorithms in machine learning. Its idea is that if a sample has k most similar samples in the feature space and most of them belong to a certain category, then the sample belongs to this category. The algorithm has the advantages of high precision and no input data assumption, but it has high computational and space complexity, resulting in a long computation time. Due to the emergence of algorithms with good performance such as deep learning, k-nearest neighbor algorithm is not widely used in encrypted traffic classification, which is usually used as a comparative experiment [18] [27].

Table 1 Traditional Supervised Machine Learning Algorithm Summary

Algorithm	Typical Application	Applied Literature	Dataset
SVM	classifier	Ref[12]	Collage Campus data set collected by the author, not published
		Ref[13]	Network Information Management and Security Group(NIMS) dataset, with no download addresses provided
		Ref[14]	UNB ISCX network traffic dataset, with no download addresses provided
LR	Feature dimension reduction classifier	Ref[15]	Dataset consisting of 10000 network flow samples, not published
		Ref[16]	The dataset collected by the author, not published
DT	classifier	Ref[17]	The data set collected by the author, not published
		Ref[18]	The dataset collected by the author, not published
		Ref[19]	NLANR dataset [20]
			MAWI dataset [21]
			DARPA99 dataset [22]
			Italy dataset [23]
			The NIMS dataset collected by the author, not published
NB	classifier	Ref[25]	The SSL/TLS traffic dataset collected by the author, not published
		Ref[26]	The dataset collected by the author, not published
KNN	classifier	Ref[18]	The dataset collected by the author, not published
		Ref[27]	ISCX-IDS 2012 dataset

3.2 Traditional Unsupervised Machine Learning Algorithm

The main difference between unsupervised learning and supervised learning is whether the dataset has labels. The unsupervised learning method only has the dataset itself and no corresponding labels. If it is found that the dataset presents a certain aggregation, it can classified according to this character, but not for the purpose of corresponding to some pre-classified label. The traditional unsupervised machine learning algorithm can be subdivided into clustering algorithm, association rule learning algorithm and dimension reduction algorithm.

In encrypted traffic classification, unsupervised machine learning algorithm does not have many application achievements. The main reason is that it can only achieve clustering, but can not give the specific class name, which is inconsistent with the goal of encrypted traffic classification. At present, only k-means [28-29] and fuzzy c-means clustering [30] belonging to clustering algorithm, Apriori [31] belonging to association rule learning algorithm and principal component analysis (PCA) [14] belonging to dimension reduction algorithm are applied to encrypted traffic classification. The summary is shown in Table 2.

Table 2 Traditional Unsupervised Machine Learning Algorithm Summary

Algorithm	Applied Literature	Dataset
K-means	Ref[28]	The UCIS dataset, with no download addresses provided
	Ref[29]	The dataset collected by the author, not published
Fuzzy c-Means	Ref[30]	The dataset collected by the author, not published
Apriori	Ref[31]	Auckland VI traffic dataset [32] Wireless traffic data provided by Community Resource for Archiving Wireless Data at Dartmouth (CRAWDAD) [33]
PCA	Ref[14]	UNB ISCX network traffic dataset, which includes VoIP, VPN-VOIP, P2P, VPN-P2P and other 14 types of traffic, with no download addresses provided

From the above work, the traditional unsupervised machine learning algorithm in encrypted traffic classification has hardly been updated and developed in recent years. Especially it is gradually replaced after the emergence of deep learning and neural network technology, which is also in line with the development trend of technology upgrading.

3.3 Neural Network and Deep Learning Algorithm

In the case of small data amount, traditional machine learning algorithms can show better performance. But with the advent of the big data era, the rapid growth of data makes these algorithms no longer applicable, replaced by more powerful neural networks and deep learning algorithms.

Neural network is a kind of machine learning technology that simulates human brain in order to realize artificial intelligence. A classic neural network structure is shown in Fig. 3, including input layer, intermediate layer (also known as hidden layer) and output layer. The circle in the figure represents neurons, and its fine structure is shown in Fig.4. In this figure, the connecting lines represent the connection between "neurons", corresponding to a weight, which is obtained through the training process. The nonlinear function is the activation function, which introduces nonlinear factors to the neural network, so that it can arbitrarily approximate any nonlinear function and be applied to many nonlinear models.

Since its advent, neural network has experienced the development process from single-layer neural network (perceptron) to multilayer neural network (deep learning). The training process of deep learning uses pre-training and fine tuning techniques, which greatly reduces the training time. At the same time, deep learning has the advantages of automatic feature extraction without manual intervention, which makes it favored by researchers and develops rapidly. At present, the most commonly used deep learning algorithms are CNN, recurrent neural networks (RNN) and auto encoders (AE), etc. In addition, the generative adversarial network (GAN) is also one of the popular research points in the deep learning.

In this section, we no longer summarizes the algorithms according to the supervised or unsupervised way, but further subdivides the machine learning algorithms into multi-layer perceptron (MLP), CNN, RNN, AE and GAN, etc. The application and research results of these algorithms in encrypted traffic classification will be introduced respectively.

1) Multilayer Perceptron (MLP)

MLP is a kind of feedforward neural network model, which is extended from perceptron, using supervised back-propagation algorithm to train. One of its important features is multi-layer. The first layer is the input layer, the last layer is the output layer, and the middle layer is called the hidden layer. MLP does not specify the number of hidden layers, so the appropriate number of them can be selected according to our requirment. The different layers of MLP neural network are fully connected, that is, all the neurons in the upper lay00000000er are connected with

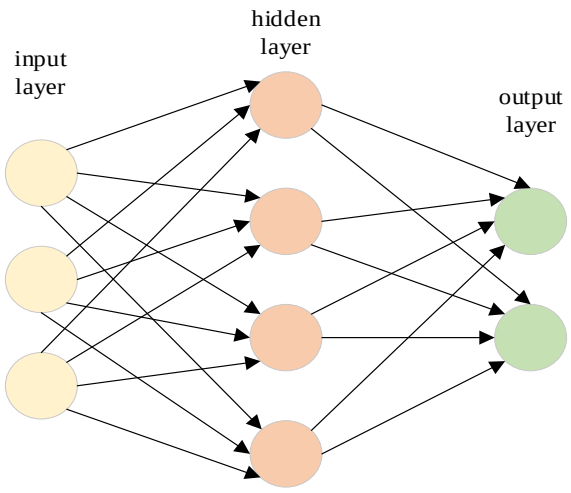


Fig. 3 Neural network diagram

all the neurons in the next layer. So it also brings too many parameters, complex structure and difficulty in training. In encrypted traffic classification, MLP is rarely used alone because of its complexity and low accuracy, which is usually used as a comparative experiment [34-35].

2) Convolutional Neural Networks (CNN)

CNN is the most popular deep learning algorithm, which can automatically learn features from large-scale data and generalize the results to unknown data of the same type. The standard CNN consists of convolution layer, linear rectification layer, pooling layer and full connection layer. The convolution layer is composed of several convolution units whose parameters are optimized by supervised back-propagation algorithm. The purpose of convolution is to extract different features of input. After that is the linear rectification layer, also called the activation layer. The bias is usually added and the ReLU is usually used as the activation function. The pooling layer is used to divided the features obtained by convolution layer into several regions, taking their maximum or average value to obtain new features with smaller dimensions. The full connection layer combines all local features to global features, which are used to calculate the score of each category.

In encrypted traffic classification, CNN is one of the most widely used deep learning algorithms, including one-dimensional CNN and two-dimensional CNN. When one-dimensional CNN is used, the original traffic data packet is preprocessed to a one-dimensional vector as the CNN input. When using two-dimensional CNN, the original traffic data packet is usually preprocessed to a two-dimensional image of p*q dimension, which is normalized to a gray-scale image within the pixel value range of [0, 1] for subsequent operation. The p and q values are determined by the researchers themselves.

Wang et al. [10] proposed an end-to-end encrypted traffic classification method based on one-dimensional CNN. It integrates feature extraction, feature selection and classifier into a unified end-to-end framework. The first 784 bytes of each flow or session are used as input. Then the model automatically learns the nonlinear relationship between the original input and the expected output. The method is verified on ISCX VPN-nonVPN traffic dataset. The experimental results show that 11 of the 12 evaluation indexes are better than the latest method, which proves the effectiveness of this method.

Rezaei et al. [36] utilized a semi-supervised method based on one-dimensional CNN to eliminate the need for a large number of labeled datasets. First-

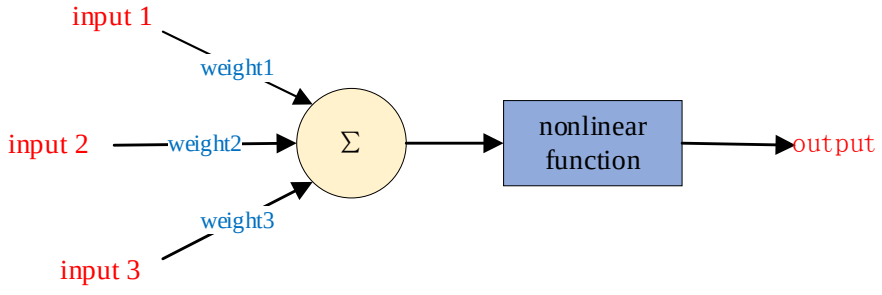


Fig. 4 Neuron model

ly， the unlabeled datasets are used to train the CNN based model. Then， the weight of convolution layer is transferred to the new model with more linear layers. Finally， the new model is retrained on a small labeled dataset. The datasets used were QUIC dataset［37］， Waikato VIII ［38］ and Ariel dataset［39］. The experimental results show that the method has a high accuracy， which is close to the accuracy of the model trained on the large label dataset.

Seq2Img， a traffic classification framework based on two-dimensional CNN， was proposed in reference ［40］. Its basic idea is to use the reproducing kernel Hilbert space （RKHS） to transform the original part of the flow sequence into multi-channel images， and then input it into CNN to obtain the traffic classification results. Two traffic datasets including five protocols and five applications are used to test the performance of the proposed method.

The core of the SDN-HGW framework proposed by Wang et al. is three different deep learning classifiers ［35］. In their work， CNN is two-dimensional， which is composed of three convolution layers， two maximum pooling layers and a full connection layer with softmax as classifier. The flow packet data is preprocessed to two-dimensional image as the CNN input. Finally， the average accuracy rate of CNN in ISCX VPN-nonVPN encryption traffic dataset reached 98.47%. The novelty of this paper is that the author puts the encrypted traffic classification technology into the practical application scenario of home smart network， instead of discussing it separately. It makes this technology have application background and become more vivid.

In addition to the above-mentioned papers， many researchers have done a lot of studies in CNN based encrypted traffic classification ［41-43］. Although the models they use have different choices in terms of convolution layer number， pooling operation， filter size， etc.， their basic processes are consistent.

3) Recurrent Neural Networks (RNN)

RNN is a deep learning model whose input is sequence data. The reason why it is called recurrent neural network is that in the process of its operation， the network will memorize the previous information and apply it to the calculation of the current output. The input of the hidden layer includes not only the output of the input layer， but also the output of the previous hidden layer. Long short term memory （LSTM） is a typical RNN model， which can solve the "long-term dependence" problem of standard RNN. This problem is that， when the current predicted position is far from the position of relevant information， the standard RNN will unable to learn such information. Its specific structure and principle will not be repeated here.

Traffic is composed of traffic bytes， packets and network flows layer by layer in the order from small to large. The temporal relationship between different bytes of a packet and between different packets can be used as a structural feature. LSTM is the preferred algorithm for time series feature detection. For example， Zeng et al. ［42］ proposed a lightweight framework， called deep full range （DFR）， which uses deep learning to encrypted traffic classification and intrusion detection. DFR takes the original traffic as the input， and the LSTM based DFR classifier is actually based on a three-layer LSTM model with 256 LSTM units in each layer. The time-related features leared by LSTM model are sent to a softmax classifier to get the final traffic classification results. On the ISCX 2012 IDS dataset， this approach achieves 99.41% accuracy.

In addition to classification， RNN can also be used for intrusion detection. A deep learning method for intrusion detection using RNN was proposed in reference ［44］. The experimental results on NSL-KDD dataset show that the model not only has strong intrusion detection capability， but also has high accuracy in binary classification and multi classification. Compared with traditional classification methods （such as J48， naive Bayes and random forest）， this method has higher accuracy and detection rate， as well as lower false positive rate.

4) Combination of CNN and LSTM

In recent years, the accuracy of encrypted traffic classification using CNN or LSTM model alone has reached the upper limit. Based on in-depth analysis of traffic structure, researchers proposed a series of architectures combining CNN and LSTM to detect encrypted traffic [45-47], in which CNN is used to learn the spatial features and LSTM is used to learn the time series features. In order to improve the classification accuracy, the hierarchical spatiotemporal feature extraction is used. For example, Zou et al. [45] proposed an abnormal network traffic detection system which can automatically learn the features of network traffic, combining convolution network and recurrent network to improve the classification accuracy. In this model, the input can be the pcap file or the actual traffic captured on the router. For each flow, the author extracts three consecutive packets and generates three packet images after data preprocessing. Then, the three packet images are input into CNN to mine the internal features hidden in a single packet image. The three feature vectors from CNN are input into LSTM together to extract the temporal features between the three packets. Finally, the softmax classifier is used to output the classification results. On ISCX VPN-nonVPN traffic dataset, this method can achieve 91% accuracy and recall rate.

5) Autoencoder (AE)

Autoencoder is a kind of neural network which can reproduce the input as much as possible. It first compresses the high-dimensional vector into the low-dimensional vector through the encoder, and then decompresses the low-dimensional vector through the decoder to reconstruct the original sample. Autoencoder is an unsupervised machine learning algorithm. That means, the input data has no corresponding label to calculate the loss function in train process. To solve this problem, autoencoder introduces the "reconstruction error", which is the error between the output of decoder and the original input data. By adjusting the parameters of the encoder and decoder, autoencoder can minimize the reconstruction error and get the trained model. After the training process, the decoder is almost useless, only the output of the encoder is needed. SAE is one of the most widely used autoencoders. It is a deep neural network composed of multi-layer autoencoders, which uses greedy layered method and fine-tuning technology to train.

In encrypted traffic classification, SAE is usually used as a feature extraction method to obtain simple high-order features of input data, which are input into the classifier to get results. Hochst et al. [48] proposed an unsupervised traffic classification method based on SAE, which realized the traffic classification independent of known categories. Experimental results show that the algorithm can detect 7 different types of mobile service flows with an average accuracy rate of 80% and an average recall rate of 75%. In reference [41], the author used SAE and CNN for comparison, in which SAE consists of five completely connected layers stacked together. These layer has 400, 300, 200, 100 and 50 neurons respectively. To prevent over-fitting, the dropout rate of 0.05 was adopted after each layer, that is, 5% of neurons were randomly set to zero in the training stage. In the last layer of SAE, softmax classifier is added to output the final classification results. Finally, on the ISCX VPN-nonVPN traffic dataset, its F1 score reached 0.92. In addition, there are many literatures using SAE to compare with other algorithms [35], but there are few studies using SAE alone for experiments.

6) Generative Adversarial Network (GAN)

GAN [49] is a model to generate data by confrontation. It originated from the Nash equilibrium theory of game theory, which consists of a generator and a discriminator. The generator captures the potential distribution of real data samples and then generates new data samples. The discriminator is used to determine whether the input is real data or generated samples. The learning optimization process is to find a Nash equilibrium between them.

In encrypted traffic classification, GAN is usu-

ally used to process unbalanced datasets. It generates data based on a small number of samples to balance the quantity difference between various data types. Vu et al. [13] used an algorithm called auxiliary classifier generative adversarial network (AC-GAN) to solve unbalanced data. The main difference between AC-GAN and GAN is that AC-GAN takes random noise and class label as input to generate the required samples. The author uses SVM, RF and DT algorithm as classifiers. The experimental results show that the classifier performs better on the enhanced dataset than on the original dataset. Wang et al. [50] proposed a method called FlowGAN. It makes full use of the GAN advantages in data enhancement to generate comprehensive traffic data based on classes with a small number of samples. The authors used MLP as classifier to evaluate the performance of FlowGAN. The experimental results on ISCX VPN-nonVPN traffic dataset show that, compared with unbalanced dataset, the recognition accuracy, recall rate and F1 score of FlowGAN are improved by 13.2%, 17.0% and 15.6% respectively.

Radford et al. [51] proposed a semi-supervised deep learning algorithm combining CNN and GAN in 2015, called deep convolutional generative adversarial network (DCGAN). The principle of DCGAN is almost the same as GAN, but the generator and discriminator in classical GAN are replaced by two convolutional neural networks, whose structure is changed to improve the samples quality and the speed convergence. The details of DCGAN can be referred to reference [51]. We only focuses on its application in encrypted traffic classification. Iliyasu et al. [52] used the DCGAN architecture [53] improved by feature matching technology to perform semi-supervised encryption traffic classification. Similarly, the DCGAN consists of a generator and a discriminator. The generator has five layers, which are a dense layer, a reshape layer and three two-dimensional transpose convolution layer. The discriminator is like a common CNN based classifier, but it replaces the maximum pooling operation with the global average pooling operation. The discriminator also includes five layers, namely three two-dimensional convolution layer, a global average pooling layer and the last fully connected dense layer. On the ISCX VPN-nonVPN dataset, the method can achieve 93% accuracy when 50% of the data is marked.

In recent years, GAN and various improved GAN algorithms are gradually applied to encrypted traffic clasiification, and more related research results [54-55] have emerged. It has become one of the most popular research points recently. Although GAN can solve the imbalanced datasets, it is difficult to train converge, which is a research direction that the follow-up researchers can pay attention to.

3.4 Ensemble Learning Algorithm

As the name implies, ensemble learning is to integrate multiple individual machine learning algorithms into a better and more comprehensive model according to a certain strategy. There are two kinds of ensemble learning algorithms: bagging and boosting. In the bagging algorithm, the basic learners participating in training are independent of each other and integrated in parallel. In this way, the average error probability is significantly reduced. The representative algorithm is RF. In the boosting algorithm, the basic learners participating in training use the dependency relationship between them to integrate in order. By giving higher weight to the samples with error marks in the previous training, the overall prediction effect can be improved. Its representative algorithm is AdaBoost. In encrypted traffic classification, only RF [13] [17] [56-57] and AdaBoost [19] are currently used as one of the classifiers for comparison with other algorithms. The details will not be repeated here.

3.5 Transfer Learning Algorithm

For human beings, transfer learning is the ability to draw inferences from one instance; for machine learning algorithms, it is to use existing knowledge to learn new knowledge. The existing knowledge is called the source domain, while the new knowledge is called the target domain. Using the

similarity between data， tasks or models， transfer learning applies the models learned from the source domain to the target domain. In this way， it can solve the problem of missing label or no label in the target domain.

In encrypted traffic classification， Rezaei et al. ［36］ proposed a semi-supervised detection method based on transfer learning. Firstly， a model is pre-trained on a large unlabeled dataset， where the input is the time series features of some sampled data packets. Then the learned weights are transferred to a new model， which will be trained on a small labeled dataset. The experiments on different datasets show the effectiveness of this method. Reference ［58］ focuses on the classification of encrypted discrete sequential protocol messages （DSPM）， proposing a LSTM-TL method based on LSTM and transfer learning （TL）. This method transfers the pre-trained LSTM model on the source domain to the target domain， where it can classify unlabeled DSPM data without any labeled data. Experiments are conducted on text， ACARS， HTTP&SSH ［59］ and AIS datasets. On the first three datasets， the F1 score and accuracy are all greater than 0.96， while only 50% and 67% respectively on AIS dataset. The reason is that the difference between the source domain used for pre-training and the target domain is too large. This also proves that choosing the right source domain is critical for high performance transfer learning.

With the development of deep learning， transfer learning based on deep neural network has become one of the research hotspots. Researchers pay more attention to the transportability of deep neural network （i.e. which layers should be transferred or fixed） and fine tuning technology. In addition， to solve the adaptive problem of the deep neural network， a large number of scholars are studying the joint training by modifying the deep neural network and adding the domain adaptive layer. A series of new depth transfer algorithms have emerged， such as deep domain confusion （DDC） ［60］， deep adaptation networks （DAN） ［61］， joint adaptation network （JAN） ［62］ and parameter transfer unit （PTU） ［63］. In encrypted traffic classification， there are few researches on the application of transfer learning. However， the problems of encrypted traffic data sets， such as less resources and difficult labeling， can be solved by transfer learning in theory. Therefore， this research direction is also worthy of attention.

4 Summary and Conclusion

The continuous growth of encrypted traffic not only provides users with more privacy protection， but also brings great challenges to network management based on traffic analysis. Encrypted traffic classification or encrypted traffic detection has become one of the foundations of network security， whose importance is increasingly prominent. We first introduces the current general framework of encrypted traffic classification， and then classifies and summarizes the machine learning algorithms involved in encrypted traffic classification. For each algorithm， we summarizes the relatively new representative research results as much as possible after a brief description of the principle. We believe our work can enable researchers to have a general grasp of encrypted traffic classification， understand its development status， and peep out its development tendency and new valuable research directions from the existing accomplishments.

In general， machine learning algorithm application in encrypted traffic classification has achieved a lot of research accomplishments， and new results are still emerging. However， there are still some problems to be solved， which are also the research directions that can be paid attention to in the future.

1) Dataset Problem

Datasets are the key to machine learning algorithms， but currently the public data set of encrypted traffic is less， and there are problems such as imbalance and no labels， resulting in difficulty for high-precision experiments. On the one hand， more accurate labeling methods and more comprehensive auto-

matic feature extraction techniques can be used as follow-up research points; on the other hand, unsupervised machine learning algorithms that do not require labels, semi-supervised machine learning algorithms that require only a few labels, and transfer learning algorithms is worthy of in-depth study, which is also a popular research point of machine learning algorithms.

2) Real-time Detection Problem

At present, almost all encrypted traffic classification methods can only achieve offline dataset detection, but not online real-time detection in the actual network environment. The real-time performance is still a huge challenge in this field.

3) Model Interpretability Problem

Although machine learning algorithms have high accuracy, it is difficult to explain how they work, which features are used for traffic identification and classification, as well as how much impact each feature has on the final results. How to provide users with interpretable, relevant or understandable solutions is a problem worth studying.

References:

[1] The Cisco Visualized Network Index Forecast Research Report [EB/OL]. https://www. cisco. com/c/dam/global/zh_cn/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360. pdf

[2] The 45th Statistical Report on Internet Development in China [EB/OL]. http://www. cnnic. net. cn/hlwfzyj/hlwxzbg/hlwtjbg/202004/P020200428596599037028. pdf

[3] Cisco Encrypted Traffic Analytics[EB/OL]. https://www. cisco. com/c/dam/en/us/solutions/collateral/enterprise-networks/enterprise-network-security/nb-09-encrytd-traf-anlytcs-wp-cte-en. pdf

[4] BUJLOW T, CARELA-ESPAÑOL V, BARLET-ROS P. Independent comparison of popular DPI tools for traffic classification [J]. Computer Networks, 2015, 76: 75-89.

[5] Samuel A L . Some studies in machine learning using the game of checkers[J]. IBM Journal of Research and Development, 2000, 44 (1/2):p. 206-226.

[6] Wang Z. The applications of deep learning on traffic identification [J]. BlackHat USA, 2015, 24(11): 1-10.

[7] CHA, Seunghun; KIM, Hyoungshick. Detecting encrypted traffic: a machine learning approach. In: International Workshop on Information Security Applications. Springer, Cham, 2016. 54-65.

[8] Smit D, Millar K, Page C, et al. Looking deeper: Using deep learning to identify internet communications traffic [C]//2017 Australasian Conference of Undergraduate Research (ACUR). 2017.

[9] Rezaei S, Liu X. Deep Learning for Encrypted Traffic Classification: An Overview[J]. IEEE Communications Magazine, 2019, 57(5): 76-81.

[10] Wang W, Zhu M, Wang J, et al. End-to-end encrypted traffic classification with one-dimensional convolution neural networks[C]. intelligence and security informatics, 2017: 43-48.

[11] ZHAI M F, ZHANG X M, ZHAO B. Survey of encrypted malicious traffic detection based on deep learning [J]. Chinese Journal of Network and Information Security, 2020, 6(3): 59-70.

[12] Zhang H, Gu Z, Tian Z. Skype traffic identification based SVM using optimized feature set [C]//2010 International Conference on Information, Networking and Automation (ICINA). IEEE, 2010, 2: V2-431-V2-435.

[13] Vu L , Bui C T , Nguyen Q U. A Deep Learning Based Method for Handling Imbalanced Problem in Network Traffic Classification[J]. 2017:333-339.

[14] Saber A, Fergani B, Abbas M. Encrypted traffic classification: Combining over-and under-sampling through a PCA-SVM[C]//2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS). IEEE, 2018: 1-5.

[15] Meng J, Yang L, Zhou Y, et al. Encrypted traffic identification based on sparse logistical regression and extreme learning machine[M]// Proceedings of ELM-2014 Volume 2. Springer, Cham, 2015: 61-70.

[16] Anderson B, McGrew D. Identifying encrypted malware traffic with contextual flow data[C]//Proceedings of the 2016 ACM workshop on artificial intelligence and security. 2016: 35-46.

[17] Anderson B, McGrew D. Machine learning for encrypted malware traffic classification: accounting for noisy labels and non-stationarity [C]//Proceedings of the 23rd ACM SIGKDD International Conference on knowledge discovery and data mining. 2017: 1723-1732.

[18] Draper-Gil G, Lashkari A H, Mamun M S I, et al. Characterization of encrypted and vpn traffic using time-related[C]//Proceedings of the 2nd international conference on information systems security and privacy (ICISSP). 2016: 407-414.

[19] Alshammari R, Zincir-Heywood A N. Can encrypted traffic be identified without port numbers, IP addresses and payload inspection? [J]. Computer networks, 2011, 55(6): 1326-1350.

[20] NLANR. http://pma. nlanr. net/special

[21] MAWI. http://tracer. csl. sony. co. jp/mawi/

[22] DARPA 1999 intrusion detection evaluation data. http://www. ll. mit. edu/IST/ideval/docs/1999/schedule. html(last accessed March, 2008.

[23] tracesSkype. http://tstat. tlc. polito. it/traces-skype. shtml (last accessed August, 2009).

[24] Zhang H. Chinese text classification and python implementation based on Naive Bayes [D]. Shandong Normal University, 2018.

[25] Sun G L, Xue Y, Dong Y, et al. An novel hybrid method for effectively classifying encrypted traffic [C]//2010 IEEE Global Telecommunications Conference GLOBECOM 2010. IEEE, 2010: 1-5.

[26] Coull S E, Dyer K P. Traffic analysis of encrypted messaging services: Apple imessage and beyond［J］. ACM SIGCOMM Computer Communication Review，2014，44(5)：5-11.

[27] Atli B G, Miche Y, Kalliola A, et al. Anomaly-based intrusion detection using extreme learning machine and aggregation of network traffic statistics in probability space［J］. Cognitive Computation，2018，10(5)：848-863.

[28] Bacquet C, Zincir-Heywood A N, Heywood M I. Genetic optimization and hierarchical clustering applied to encrypted traffic identification ［C］//2011 IEEE symposium on computational intelligence in cyber security (CICS). IEEE，2011：194-201.

[29] Du Y, Zhang R. Design of a method for encrypted P2P traffic identification using K-means algorithm［J］. Telecommunication Systems，2013，53(1)：163-168.

[30] Liu D, Lung C H. P2P traffic identification and optimization using fuzzy c-means clustering［C］//2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011). IEEE，2011：2245-2252.

[31] Chaudhary U K, Papapanagiotou I, Devetsikiotis M. Flow classification using clustering and association rule mining［C］//2010 15th IEEE International Workshop on Computer Aided Modeling，Analysis and Design of Communication Links and Networks（CAMAD）. IEEE，2010：76-80.

[32] WAND Trace Catalogue. http://www. wand. net. nz/wits/catalogue. php.

[33] DavidKotz, TristanHenderson, IlyaAbyzov, andJihwangYeo. CRAWDADtracedart-mouth/campus/tcpdump/fall01 (v. 2004-11-09
). Downloaded from http://crawdad. cs. dartmouth. edu/dartmouth/campus/tcpdump/fall01, November 2004.

[34] Aceto G, Ciuonzo D, Montieri A, et al. Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges［J］. IEEE Transactions on Network and Service Management，2019，16(2)：445-458.

[35] Wang P, Ye F, Chen X, et al. Datanet: Deep learning based encrypted network traffic classification in sdn home gateway［J］. IEEE Access，2018，6：55380-55391.

[36] Rezaei S, Liu X. How to achieve high classification accuracy with just a few labels: A semi-supervised approach using sampled packets［J］. arXiv preprint arXiv:1812. 09761，2018.

[37] QUIC Dataset. 2018. https://drive. google. com/drive/folders/1Pvev0hJ82usPh6dWDlz7Lv8L6h3JpWhE ［Online; accessed 30-Oct-2018].

[38] Waikato VIII Dataset. 2013. https://wand. net. nz/wits/ ［Online; accessed 24-Oct-2018].

[39] Ariel University Dataset. 2016. https://drive. google. com/drive/folders/0Bynah7-gERTldG5UZ2NhNkJMMlk ［Online; accessed 24-Oct-2018].

[40] Chen Z, He K, Li J, et al. Seq2Img: A sequence-to-image based approach towards IP traffic classification using convolutional neural networks［C］//2017 IEEE International Conference on Big Data (Big Data). IEEE，2017：1271-1276.

[41] Lotfollahi M, Siavoshani M J, Zade R S H, et al. Deep packet: A novel approach for encrypted traffic classification using deep learning［J］. Soft Computing，2020，24(3)：1999-2012.

[42] Zeng Y , Gu H , Wei W , et al. Deep-Full-Range: A Deep Learning Based Network Encrypted Traffic Classification and Intrusion Detection Framework［J］. IEEE Access，2019：1-1.

[43] Wang W, Zhu M, Zeng X, et al. Malware traffic classification using convolutional neural network for representation learning［C］//2017 International Conference on Information Networking (ICOIN). IEEE，2017：712-717.

[44] Chuan-Long Y , Yue-Fei Z , Jin-Long F , et al. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks［J］. IEEE Access，2017，5(99)：21954-21961.

[45] Zou Z , Ge J , Zheng H , et al. Encrypted Traffic Classification with a Convolutional Long Short-Term Memory Neural Network［C］// 2018 IEEE 20th International Conference on High Performance Computing and Communications；IEEE 16th International Conference on Smart City；IEEE 4th International Conference on Data Science and Systems（HPCC/SmartCity/DSS）. IEEE，2018.

[46] Wang W, Sheng Y, Wang J, et al. HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection［J］. IEEE Access，2017，6：1792-1806.

[47] Lopez-Martin M, Carro B, Sanchez-Esguevillas A, et al. Network traffic classifier with convolutional and recurrent neural networks for Internet of Things［J］. IEEE Access，2017，5：18042-18050.

[48] Höchst J, Baumgärtner L, Hollick M, et al. Unsupervised traffic flow classification using a neural autoencoder［C］//2017 IEEE 42nd Conference on Local Computer Networks (LCN). IEEE，2017：523-526.

[49] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets［C］//Advances in neural information processing systems. 2014：2672-2680.

[50] Wang Z X, Wang P, Zhou X, et al. FLOWGAN: Unbalanced Network Encrypted Traffic Identification Method Based on GAN［C］//2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications，Big Data & Cloud Computing，Sustainable Computing & Communications，Social Computing & Networking（ISPA/BDCloud/SocialCom/SustainCom）. IEEE，2019：975-983.

[51] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks［J］. arXiv preprint arXiv:1511. 06434，2015.

[52] Iliyasu A S, Deng H. Semi-Supervised Encrypted Traffic Classification With Deep Convolutional Generative Adversarial Networks［J］. IEEE Access，2020：118-126.

[53] SalimansT. , GoodfellowI. , ZarembaW. , CheungV. , RadfordA. , and ChenX. , "Improved techniques for training GANs,Jun". 2016, arXiv: 1606. 03498. ［Online］. Available: https://arxiv. org/abs/1606. 03498

[54] Fathi-Kazerooni S, Rojas-Cessa R. GAN Tunnel: Network Traffic Steganography by Using GANs to Counter Internet Traffic Classifiers［J］. IEEE Access，2020，8：125345-125359.

[55] Wang P, Li S, Ye F, et al. PacketCGAN: Exploratory study of class imbalance for encrypted traffic classification using CGAN［C］//ICC 2020-2020 IEEE International Conference on Communications

(ICC). IEEE，2020：1-7.

[56] Lashkari A H, Draper-Gil G, Mamun M S I, et al. Characterization of tor traffic using time based features[C]//ICISSp. 2017：253-262.

[57] Barradas D, Santos N, Rodrigues L. Effective detection of multimedia protocol tunneling using machine learning [C]//27th {USENIX} Security Symposium ({USENIX} Security 18). 2018：169-185.

[58] Li Q, Ju Y, Zhao C. Classification of Discrete Sequential Protocol Messages Based on LSTM Network and Transfer Learning[C]//2020 5th International Conference on Computer and Communication Systems (ICCCS). IEEE，2020：424-430.

[59] The Mid-Atlantic Collegiate Cyber Defense Competition (MACCDC)，Accessed November 25，2019. http://maccdc. org/.

[60] Tzeng E, Hoffman J, Zhang N, et al. Deep domain confusion: Maximizing for domain invariance [J]. arXiv preprint arXiv: 1412. 3474，2014.

[61] Long M, Cao Y, Wang J, et al. Learning transferable features with deep adaptation networks [C]//International conference on machine learning. PMLR，2015：97-105.

[62] Long M, Zhu H, Wang J, et al. Deep transfer learning with joint adaptation networks [C]//International conference on machine learning. PMLR，2017：2208-2217.

[63] Zhang Y, Zhang Y, Yang Q. Parameter Transfer Unit for Deep Neural Networks[C]//Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer，Cham，2019：82-95.

About the authors

Zhang Surong was born in October 1996. She received her B. E. degree in communication engineering from Shandong University. She is now a master student of People’s Liberation Army Strategic Support Force Information Engineering University. Her research interests include network space security and deep learning algorithm. （Email： 1091003035@ qq. com）

Bu Youjun was born in March 1978. He received his PhD degree from PLA Strategic Support Force Information Engineering University and is an associate researcher there. His research interests include network security， mimicry defense technology and artificial intelligence technology. （Email： 13140186091@126. com）

SR网络中基于深度强化学习的流量工程

陈博，孙鹏浩，兰巨龙，张鹏，卜佑军
中国人民解放军战略支援部队信息工程大学信息技术研究所 河南郑州 450000

摘　要：流量工程在网络策略设计中具有重要地位。近年来，分段路由等新型网络技术的兴起对流量工程策略提出了新的场景和需求。本文针对目前现有分段路由网络中的静态流量工程策略灵活性差的问题，提出了基于深度强化学习的流量调度方案。本文所提方案通过多路径流量传输实现，能够根据不同网络流量视图使用度强化学习算法动态优化网络流量在不同路径上的分配比例，从而进一步减少网络拥塞，提升网络性能。实验结果表明，本文所提方案能够比现有方案提升9%的吞吐量。

关键词：SDN、深度强化学习、分段路由、流量工程

Traffic Engineering based on deep reinforcement learning in SR network

Chen Bo, Sun Penghao, Lan Julong, Zhang Peng, Ma Hailong
Information Technology Institute, PLA Strategic Support Force Information Engineering University, Zhengzhou Henan, 450000

Abstract: Traffic Engineering (TE) plays an important role in the network policies. Recently, the emergence of new network technologies such as Segment Routing (SR) places new scenarios and requirements for the TE in networking. Aiming at the low flexibility problem of static TE policies in the current SR networks, we propose a Deep Reinforcement Learning (DRL) based TE scheme. The proposed scheme employs multi-path transmission, which can use DRL to dynamically adjust the traffic splitting ratio on different paths based on the network traffic distribution. As a result, the network congestion can be reduced and the performance of the network is improved. Simulation results show that our proposed scheme can improve the throughput of the network by up to 9% than existing schemes.

Key words: SDN; deep reinforcement learning; segment routing; traffic engineering

1　引言

随着5G、物联网、云计算等网络技术的发展，运营商、数据中心等网络服务提供商面临着日益增长的服务差异化需求，如高带宽、低时延、低丢包率等等。传统的网络结构采用流量工程解决流量动态调度的问题，从而提高网络资源利用率[1]。但是MPLS、RSVP-TE[2][3][4]等流量工程方案需要维护大量的状态信息，影响路由设备的处理能力并占用网络带宽。

近几年，分段路由（Segment Routing，SR）[5]技术作为一种新的网络技术被提出。SR具有源路由[6]和状态只存在于边缘的特点，使其可以支持超大规模的流量工程，同时其具有与SDN技术的原生匹配性，便于实现应用驱动的网络。基于分段路由的流量工程将用户的意图（如延迟、不相交路径、SRLG、带宽等）转换为Segment列表，然后将Segment列表编程到单域/跨域网络的边缘设备上，同时引导流量至Segment列表所对应的路径上，从而实现“基于意图的网络（Intent-Based Networking， IBN）[7]”，完成传统网络向下一代网络平台的演进。

基于分段路由的流量工程能够很好的满足用户的服务差异化需求，可以根据网络管理员指定

基金项目：国家重点研发计划项目(2020YFB1804803，2017YFB0803201，2016YFB0801200)；国家自然科学基金项目(No. 61572519, No. 61802429,No. 61521003)

的路径进行传输。网络流量在SR节点之间按照最短路径进行转发，当存在多条相同代价的最短路径（ECMP）时，流量可在多条路径上平均分布。然而，当采用静态的多路径传输方案例如ECMP时，网络中的流量分布依赖于实际的流量矩阵和网络拓扑，此类方案在异构的网络拓扑场景下表现差强人意，不能取得较优的网络流量分布[8]。

近年来，随着机器学习技术的发展成熟，基于机器学习算法实现网络流量动态调度逐渐称为网络领域的研究热点。其中，在机器学习算法各种类别中，深度强化学习（Deep Reinforcement Learning， DRL）算法由于其适用于产生自动控制策略、且能够不依赖于大量标签数据进行训练，因此在网络应用领域中得到了广泛的重视，在流量调度等场景中取得了较好的效果。例如，Xu等人[9]将深度强化学习用于域内流量工程问题中提出了基于深度强化学习的流量工程方案DRL-TE，将流量工程问题划分为静态多路径求解以及在线动态调整路径分流比两部分，相比传统路由和流量工程算法，在时延、吞吐量上都具有明显优势。Valadarsky等人[10]尝试利用深度强化学习单元根据历史流量数据对未来的网络流量进行预测，并基于强化学习模型的流量预测能力计算出恰当的路由配置，仿真实验结果表明对于具有明显规律特征的流量矩阵，强化学习模型能够通过流量预测来实现良好的路由配置，达到优于流量无关最优路由并且接近最优的路由配置效果。

基于上述背景，本文提出了基于深度强化学习的混合IP/SR网络流量调度方案DRL-MPTE。在给定网络拓扑和SR路径的条件下，当网络中SR节点之间存在多条相同代价的最短路径时，利用深度强化学习确定每条路径的权重，分配合适的流量到每一条路径，从而最大化网络效用，保证SR节点间的多路径传输公平性。本文开创性地将深度强化学习应用于SR网络的多路径流量调度中，其中本文的主要贡献如下：

1）设计了智能化SR流量工程网络架构，通过结合SDN控制器的全局策略部署功能和DRL算法的自主探索能力，实现了SR网络中的智能化流量工程调度。

2）设计了DRL-MPTE算法，根据网络中多路径的实时链路状态对网络流量进行调度，提升了多路径传输带宽利用率。

3）在omnet++网络仿真模拟器上对提出的方案进行了实验，实验结果表明，相比ECMP方法，DRL-MPTE算法能够提升网络带宽利用率XX%。

论文的组织结构如下：第二章介绍了研究内容的相关背景，包括分段路由和深度强化学习；第三章描述了研究问题的数学表示；第四章是本文的核心部分；详细阐述了基于深度强化学习的流量调度方案；第五章对所提出的方案进行了实验验证；第六章对全文工作进行总结。

2 背景

2.1 分段路由

分段路由思想基于源路由，其主要实现过程为通过在源节点向每个数据包的IP头中插入一个有序的分段列表，引导数据包按照指定的路径传输，中间节点无需存储数据流的状态信息，从而减少了网络节点的表项负载。分段路由机制包括两种类型的段：节点段和邻接段。节点段表示一个路由，当节点段位于分段列表的顶部时，数据包沿着最短路径传输到该节点。邻接段表示一个节点的出接口，可以认为是网络中的一条链路。当邻接段位于分段列表的顶部时，数据包沿着本链路传输到下一节点。

如图1显示了分段路由的数据转发过程，图中节点A作为源节点，F作为目的节点，分段列表为（B，E，F），表示中间经过节点段B和E，箭头表示经过的最短路径。节点A首先将数据包发送到节点B，节点B收到数据包后，弹出标签B，看到最上面的标签E，根据最短路径将数据包发送到节点C，节点C不做处理直接转发数据包到D，同样D将数据包发送到E，节点E收到数据包后，弹出标签E，看到最上面的标签是F，根据最短路径将数据包经过G转发到节点F。

多条相同代价的路径称为Equal Cost Multi-Path （ECMP），如图2所示，如果分段列表是（F），则有三条相同代价的路径：（A，B，G，F），（A，B，C，D，E，G，F）和（A，C，D，E，G，F），如图2中红色、蓝色和绿色的三条路径。SR网络中对ECMP的处理默认采用负载平均分配的方式。当有一条单位大小的流从A到F时，流在每一个分路节点平均分割，如图2中每条链路旁的

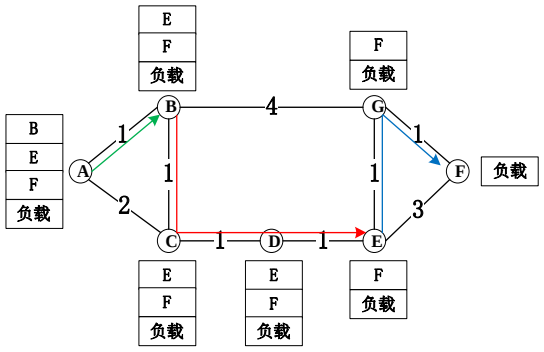


图1　分段路由数据转发示意图

分数。从图中可以看出，流量并没有在三条路径上最优分布，B-G链路没有得到有效利用。

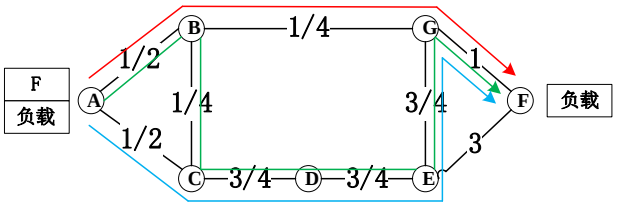


图2　带ECMP的分段路由示意图

2.2　深度强化学习

强化学习算法旨在学习长期策略，从而做出决策动作使优化问题中的目标函数最大化。与监督学习相比，强化学习不需要大量的带标签数据集，而是通过探索状态和动作空间，在奖励函数引导下进行策略迭代更新来学习最佳的决策动作。在强化学习中，状态空间是一组状态的集合**S**，动作空间是一组动作的集合**A**，在每一个时刻t，agent观察环境状态$s_t \in \mathbf{S}$，采取动作$a_t \in \mathbf{A}$，获得反馈奖励r_t，同时产生一个新的状态$s_{t+1} \in \mathbf{S}$。强化学习的目标是找到一个策略函数π（s），将状态映射到动作，并最大化折扣累积奖励$R_0=\sum_{t=0}^{T}\gamma^t r_t$，其中$\gamma \in [0, 1]$是奖励折扣因子。

为了处理连续高维状态决策空间上存储空间爆炸的强化学习问题，研究者们将近几年快速发展的深度学习模型引入强化学习框架，设计了多种深度强化学习模型。DeepMind公司的Mnih等人提出了DQN（Deep Q-Learning Network）[11]，采用一个深度神经网络（Deep Neural Network， DNN）来代替原本Q-Learning中的Q值表来近似估计Q函数。与基于Q函数估计的DQN方法相对应的是策略梯度方法，策略梯度法利用深度学习模型作为策略函数，通过计算策略梯度的方式直接优化策略函数。为了进一步提升策略梯度方法的性能，加速强化学习模型的收敛速度，可以将Q值学习与策略梯度方法结合起来，通过价值估计函数来预测当前状态下采用行动后续会得到的价值，并利用预测结果对策略模型进行训练，这就是强化学习的Actor-Critic（AC）框架。

3　网络模型

本节描述混合IP/SRv6网络的流量工程问题。

考虑一个混合IP/SRv6网络，所有节点运行传统的网络协议栈，支持IPv6和OSPFv3协议，其中只有一部分节点支持SRv6，为简单起见，本文考虑SR节点是网络拓扑的连通子图，组成一个SR域[12]，数据流只进入一次SR域。如图3所示，节点c、d、e、f是SRv6节点，组成SR域。当有一条数据流从a到i时，数据沿着箭头的方向转发，当数据从a转发到c时，c到f有两条相同代价的路径（c，e，f）和（c，d，f），根据一定的策略将数据按照一定的比例分流到两条路径，数据到达f后，f直接转发到目的节点i。

本文将混合IP/SRv6网络建模为一个有向图$G(V, E)$，其中V表示网络中的节点集合，对应网络中的路由器或者交换机，E表示节点间互连的链路集合，V中包含普通节点和SRv6节点。考虑流量矩阵T表示网络中流量需求的长期预测，D_f表示流f的流量需求，链路e的链路容量为C_e。K_{ij}表示SR域中节点i和j之间的加权最短多路径数量，不同节点对的多路径数量可能不同，需基于网络拓扑分析提前确定多路径数量，如图3所示，节点i和j之间的多路径数量为K_{ij}=3，节点i和k之间的多路径数量为K_{ik}=2。本文目标是研究混合IP/SR网络中链路容量带宽分配的流量工程问题，优化每条流在SR域内K_{ij}个路径上的流量分割比例。假设流f在SR域中经过节点i和j，节点i和j之间第k条SR路径的流量分割比为$w_{ij,\,k}$，其中$\sum_{k=1}^{K_{ij}} w_{ij,k} = 1$。表1是本文用到的一些符号和定义。

α-公平性模型广泛用于网络效用最大化（Network Utility Maximization，NUM）[13]，根据这个模型，本文采用的效用函数$U_\alpha(x) = (\frac{x^{1-\alpha}}{1-\alpha})$，特别地，当$\alpha \to 1$时，$U_1$（$x$）的极限是$\log x$[14]。当$\alpha$>0

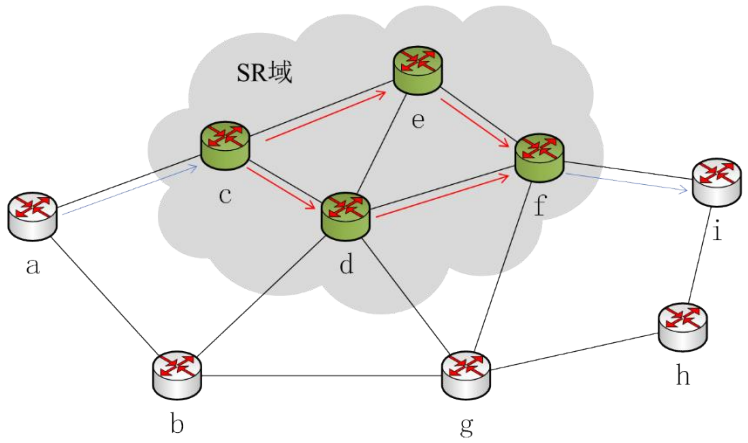


表1　变量定义列表

变量名称	定义
G	网络拓扑图
V	网络节点集合
V'	网络中SR节点集合
E	网络节点间的链路集合
K_{ij}	SR域中节点i和j之间的多路径数量
D_f	流f的带宽需求
C_e	链路容量带宽
$w_{ij,k}$	SR域中节点i和j之间第k条SR路径的流量分割比
z_f	流f的延迟
t_e	链路e的负载
$\boldsymbol{s, a, r}$	状态、动作和奖励
θ^{π},θ^{Q}	actor网络($\pi(\cdot)$)和critic网络($Q(\cdot)$)的神经网络参数

时，$U_\alpha(x)$随x单调递增。α用于平衡公平性和有效性，如果α=1，则目标是实现按比例公平，该公平性被广泛用于资源分配。我们定义流f的效用函数$U(\bullet)$为：

$$U(z_f) = -U_\alpha(z_f)$$

其中z_f为流f端到端的延迟，流量工程的目标是最大化所有网络流的总效用，即$\sum_f U(z_f)$。

网络中的SR节点根据节点重要度$R(v)$[13]选择。节点重要度$R(v)$定义为：

$$R(v) = \mu d_v + \delta SP_v, \mu + \delta = 1$$

其中d_v表示节点v的度，SP_v表示通过节点v的最短路径的数量，μ和δ分别为d_v和SP_v的权重系数。

4　基于DRL的流量调度方案

4.1　调度方案

本文设计了智能化SR流量工程网络架构，系统框架图如图4所示。网络拓扑节点包括SR路由器和普通路由器，SR路由器之间采用最短路径优先算法寻找下一个SR路由器，统一传输路径上的相邻SR路由器之间存在多条路径时，通过多路径路由进行传输，传输分流比通过DRL算法确定。普通路由器之间通过最短路径优先算法确定单一路由。路由过程分两个阶段：第一阶段是SR域外路由，第二阶段是SR域内路由。SR域外路由指源节点到SR节点和SR节点到目的节点之间的路由，SR域内路由指SR域内两个SR节点之间的路由。SR域外路由是单路径传输，SR域内路由是多路径传输。两阶段路由共同组成网络中任意两个节点之间的路由。

智能化SR流量工程网络架构利用SDN技术收集网络链路状态并控制链路流量分配，在控制器之上运行DRL agent程序，完成网络流量的智能控制调度。

SDN控制器从网络拓扑采集链路利用率作为状态信息s_t，作为策略/值DNN的输入，DNN读取状态信息s_t，输出流量调度策略π_t，控制器根据调度策略π_t采取相应的动作a_t，即SR域中多路径的流量分流比。奖励系统受到SDN控制器输出动作的反馈，生成相应的奖励r_t。s_t、a_t、r_t一起存入经验缓存区，用于随后生成训练学习信号，更新DNN。

4.2　DRL算法实现

本节详细介绍提出的基于DRL的混合IP/SR网络流量调度框架DRL-SRTE。

为了利用DRL技术，首先需要设计状态空间、动作空间和奖励函数。

状态空间：状态是网络中所有链路的负载，状态向量为：$\mathbf{s}=[t_1, t_2, \cdots, t_{|e|}]$。

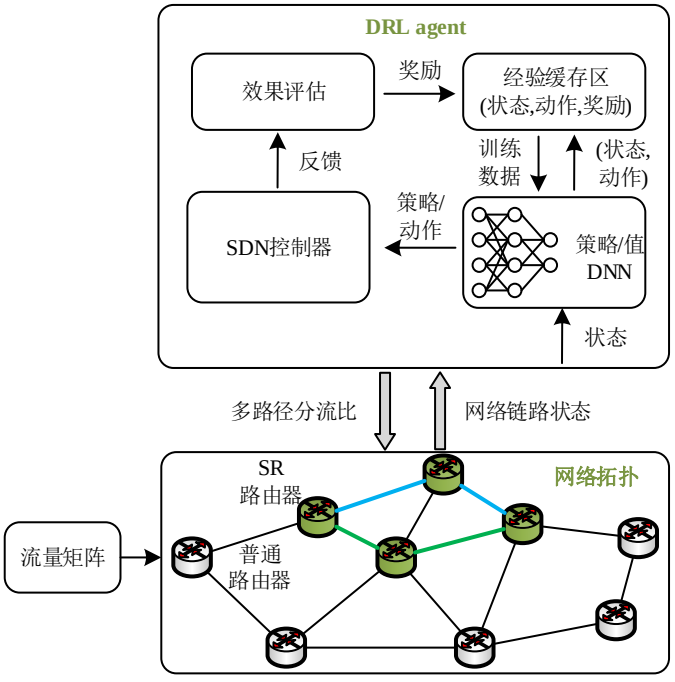


图4　DRL-SRTE 系统架构图

动作空间：动作定义为网络中节点 i 和节点 j 中间的多路径流量分割比例，动作向量的正式表示为：$\mathbf{a}=[w_{12,1}, w_{12,2}\cdots w_{12,k}\cdots w_{12,|K12|}\cdots w_{ij,k}\cdots w_{ij,|Kij|}\cdots]$，其中，$w_{ij,k}$ 表示节点 i 到节点 j 的第 k 条路径分流比，$\sum_k w_{ij,k}=1$。

奖励函数：奖励是流量工程问题的目标，定义为所有网络流的总效用，即 $\boldsymbol{r}=\sum_f U\left(x_f, z_f\right)$。

状态空间、动作空间和奖励函数的设计与 DRL 方法的性能密切相关。本方案收集了网络状态和流量工程问题的关键信息，且不包括无用冗余的信息。流量调度框架的核心是一个运行 DRL 算法的 agent，该算法在每一个状态寻找最优动作，然后执行该动作改变网络状态，并收集奖励信息和转换样本。

流量调度问题是一个连续控制问题，我们选择当前在连续控制领域应用最为广泛的算法 DDPG（Deep Deterministic Policy Gradients）[16] 作为本文的 DRL 算法。

DDPG 算法是一种 actor-critic 算法，利用了深度神经网络和确定性策略梯度进行连续控制。DDPG 的核心思想是同时维护两个函数：一个作为参数化的 actor 函数 $\pi(s_t|\theta^{\pi})$，用于生成动作 a，另一个是参数化的 critic 函数 $Q(s_t, a_t|\theta^{Q})$，用于评估 actor 函数选择的动作。Actor 函数通过深度神经网络实现，以状态作为输入，输出最好的动作。Critic 函数也是通过深度神经网络实现，以状态-动作对作为输入，输出相应的 Q 值。

算法 1 展示了 DRL-SRTE 的运行过程。

5　实验评估

为了验证本文提出的 DRL-SRTE 算法在混合 IP/SR 网络中的有效性，我们对各种网络拓扑下的性能进行了实验评估，并与 ECMP 算法、最短路径算法（Shortest Path First， SPF）进行了对比实验。基于 Omnet++网络仿真软件进行实验环境构建，并使用 Keras+TensorFlow 实现了 DRL-SRTE，在一台配置 Intel Core i7 3.0GHz CPU 和 32GB 内存的台式机上进行仿真实验。算法训练 100 回合，每回合设置训练步数为 1000。

对于网络环境，本文在 SDNlib [18] 中选择了三个实际电信网络场景网络拓扑进行实验，分别对应于不同国家城市的真实的电信网络拓扑，表 2 是仿真实验用到的拓扑信息。拓扑中链路的带宽确定为 1Gbps，流量矩阵中的每条业务量的需求在（8，30）Mb/s 随机生成，业务流的源节点和目的

算法1：DRL-SRTE

输入

输出

使用随机 θ^{π} 和 θ^{Q} 初始化 actor 网络 $\pi(\cdot)$ 和 critic 网络 $Q(\cdot)$；

使用随机 $\theta^{\pi'}$ 和 $\theta^{Q'}$ 初始化 target_actor 网络 $\pi'(\cdot)$ 和 target_critic 网络 $Q'(\cdot)$，其中 $\theta^{\pi'} = \theta^{\pi}, \theta^{Q'} = \theta^{Q}$；

初始化 replay buffer；

for episode t=1 to T do

  获取网络的初始状态 s_0；

  for i=1 to N do

    基于 ε-贪心策略选择动作 a_t；

    采取动作 a_t，获得下一时刻状态 s_{t+1} 和收益 r_t；

    存储转换样本 (s_t, a_t, r_t, s_{t+1}) 到 replay buffer；

    从 replay buffer 中采样 N 个转换样本 (s_i, a_i, r_i, s_{i+1})；

    计算 critic 网络的目标值：$y_i = r_i + \gamma Q'\left(s_{i+1}, \pi'\left(s_{i+1}|\theta^{\pi'}\right)|\theta^{Q'}\right)$；

    更新 critic 网络 $Q(\cdot)$ 的权重 θ^{Q}；

    更新 actor 网络 $\pi(\cdot)$ 的权重 θ^{π}；

    更新 target_actor 网络 $\pi'(\cdot)$ 和 target_critic 网络 $Q'(\cdot)$ 的权重 $\theta^{\pi'}$ 和 $\theta^{Q'}$；

    $\theta^{\pi'} \leftarrow \tau\theta^{\pi} + (1-\tau)\theta^{\pi'}$

    $\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'}$

  end for

end for

节点也是随机生成的。

表2　实验拓扑信息

网络拓扑	节点	链路
Nobel-Germany	17	22
Geant	22	36
France	25	45

如图5展示了三种算法在不同网络拓扑下的网络吞吐量对比，从中可以看出，本文提出的DRL-SRTE算法明显优于ECMP和SPF算法，相比ECMP算法最高能够提高9%的吞吐量，相比SPF算法最高能够提高12%的吞吐量（France网络拓扑）。在三个网络拓扑中，France网络拓扑的性能提升最大，说明随着网络拓扑的增大和网络中SR多路径的增多，DRL-SRTE算法能够带来更高的吞吐量提升。

图6所示为在三个网络拓扑中，本文提出的DRL-SRTE算法相比ECMP和SPF算法，在不同的平均链路利用率时的吞吐量性能提升比例。从图中可以看出，当平均链路利用率较小时，网络链路基本没有发生拥塞，此时DRL-SRTE算法的性能提升比例较小，当吞吐量逐渐增大时，DRL-SRTE算法相比ECMP和SPF的性能提升逐渐增大，但当网络吞吐量达到一定阈值后，DRL-SRTE算法相比ECMP的性能提升比例逐渐下降。

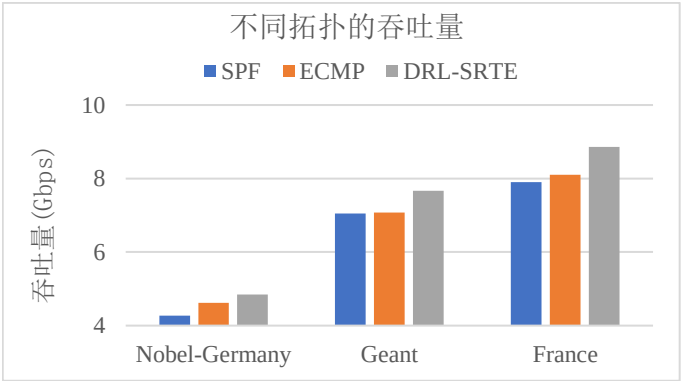


图5　三种算法在不同拓扑下的吞吐量对比

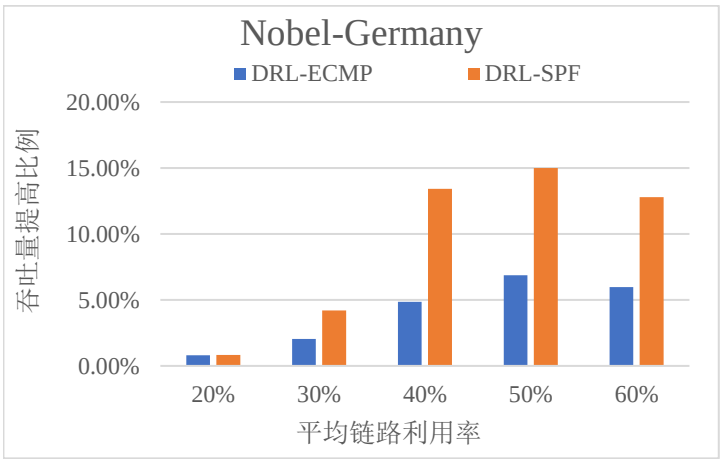
图6（a） Nobel-Germany拓扑在不同平均链路利用率下DRL-SRTE算法相比ECMP和SPF算法的性能提升比例

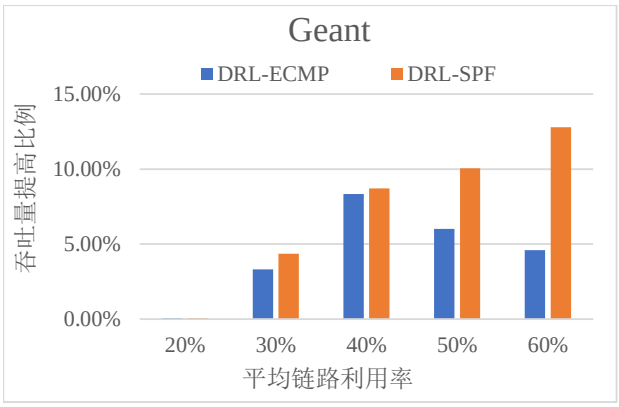
图6（b） Geant拓扑在不同平均链路利用率下DRL-SRTE算法相比ECMP和SPF算法的性能提升比例

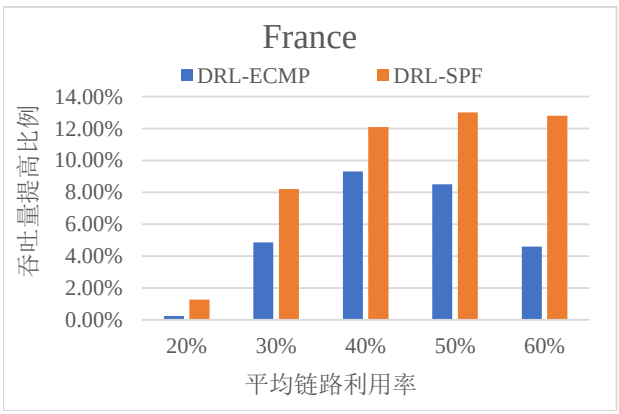
图6（c） France拓扑在不同平均链路利用率下DRL-SRTE算法相比ECMP和SPF算法的性能提升比例

6　总结

本文面向混合IP/SR网络流量调度问题，提出了一种基于深度强化学习的流量工程算法DRL-SRTE，该算法通过分析网络中的链路带宽利用率，输出SR域内多路径上的流量分流比，克服了传统的ECMP存在的问题，可以根据网络流量模型很







好的控制网络流量分布，达到更好的网络效用。利用omnet++网络仿真软件在两个知名的网络拓扑——Nobel-Germany 和 Geant——进行实验评估结果表明，当 SR 域内存在多路径时，相比传统的 ECMP 的平均分配流量，DRL-SRTE 算法智能分配流量能够有效提高网络吞吐量，从而提高网络整体效用。

参考文献:

[1] Fortz B. , Rexford J. , Thorup M. 2002. Traffic engineering with traditional IP routing protocols . https://tools. ietf. org/html/rfc3272，2002-05

[2] AwducheD. , MalcolmJ. , AgogbuaJ. , O'DellM. , and McManusJ. . 1999. RFC2702: Requirements for Traffic Engineering Over MPLS. RFC Editor, USA.

[3] Xipeng Xiao, HannanA. , BaileyB. and NiL. M. , "Traffic engineering with MPLS in the Internet，" in IEEE Network，vol. 14，no. 2，pp. 28-33，2000.

[4] https：//tools. ietf. org/html/rfc3209

[5] FilsfilsC. , NainarN. K. , PignataroC. , CardonaJ. C. and FrancoisP. , The Segment Routing Architecture［C］. 2015 IEEE Global Communications Conference (GLOBECOM)，San Diego，CA，2015，pp. 1-6.

[6] https：//en. wikipedia. org/wiki/Source_routing

[7] https：//www. cisco. com/c/en/us/solutions/intent-based-networking. html.

[8] Moreno E, Beghelli A, Cugini F. Traffic engineering in segment routing networks[J]. Computer Networks，2017，114：23-31.

[9] Xu Z. , Tang J. , Meng J. , Zhang W. , Wang Y. , Liu C. ，& Yang, D. Experience-driven Networking: A Deep Reinforcement Learning based Approach ［C］//IEEE Conference on Computer Communications，Honolulu，HI，2018：1871-1879.

[10] Valadarsky A. , Schapira M. ，& Shahaf, D. Learning to Route[C]// In Proceedings of the 16th ACM Workshop on Hot Topics in Networks (HotNets-XVI). Association for Computing Machinery，New York，NY，USA，2017：185－191.

[11] MnihV. , et al. , Human-level control through deep reinforcement learning, Nature, Vol. 518, No. 7540，2015，pp. 529－533

[12] Cianfrani A. , ListantiM. and PolveriniM. , Incremental Deployment of Segment Routing Into an ISP Network：a Traffic Engineering Perspective. IEEE/ACM transactions on networking，2017. 25(5)：p. 3146-3160.

[13] LowS. H and LapsleyD. E. , Optimization flow control. I. Basic algorithm and convergence ［J］. IEEE/ACM Transactions on

networking，Vol. 518，No. 6，1999，pp. 861－874.

[14] WinsteinK. and BalakrishnanH.，Tcp ex machina：Computer-generated congestion control, *ACM SIGCOMM*'2013，pp. 123－134.

[15] GangYirou, Pei Zhang, Xiaohong Huang, Tianle Yang. Throughput Maximization Routing in the Hybrid Segment Routing Network［A］. In Proceedings of the 2nd International Conference on Telecommunications and Communication Engineering (ICTCE 2018)［C］, ACM，2018:262－267.

[16] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning［J］. Computer Science，2015，8（6），pp. A187.

[17] Christian R.. Kai H.. A library of test instances for Survivable fixed telecommunication Network Design［EB/OL］. http://sndlib. zib. de/home. action

.

[作者简介]

陈博（1989-），男，博士研究生，现为国家数字交换系统工程技术研究中心助理研究员，主要研究方向为流量工程，网络智慧化。

孙鹏浩（1992-），男，博士研究生，现为国家数字交换系统工程技术研究中心博士研究生，主要研究方向为网络智慧化，可编程网络。

兰巨龙（1962-），男，博士，现为国家数字交换系统工程技术研究中心教授、博士生导师，主要研究方向为新一代信息网络关键理论与技术。

张鹏（1982-），男，学士，现为国家数字交换系统工程技术研究中心副教授，主要研究方向为网络安全。

卜佑军（1978-），男，博士，现为国家数字交换系统工程技术研究中心副研究员，主要研究方向为网络与信息安全，机器学习。

基于软件定义的卷积神经网络可重构电路设计

夏云飞[2]，张丽[1]，李沛杰[1]，许立明[2]
[1]国家数字交换系统工程技术研究中心，河南郑州450002；
[2]天津市滨海新区信息技术创新中心，天津300457

摘　要：本文针对多种卷积神经网络模型处理特点，归一化处理流程并设计出一种可以通过软件定义动态重构卷积层网络与运算模式的加速电路。采用SoC双总线架构，在FPGA上实现了卷积神经网络的流水计算。在网络模型与输入数据集相同的条件下，对CPU、GPU以及本设计FPGA电路进行运算效率与能耗之间的比较，并实时利用软件切换在FPGA上完成两种不同网络结构下的数据集处理。实验分析表明，在200MHz工作频率，本设计FPGA电路运算处理能力为CPU的10倍，运算能耗比为GPU的2倍，并且电路结构通过软件配置完成两种网络模型的工作模式切换。

关键词：卷积神经网络（CNN）、软件定义、动态可重构、FPGA、流水计算、SoC

The Re-configurable Circuit Design of Convolutional Neural Network Based on Software Definition

XIAYun-fei [2], ZhangLi [1], LiPei-jie [1], XuLi-ming [2]
1.China National Digital Switching System Engineering＆Technological R＆D Center, Zhengzhou, Henan 450002,China;
2.InformationTechnology Innovation Center of Tianjin Binhai New Area, Tianjin 300457,China

Abstract: Aiming at the processing characteristics of various convolutional neural network models, this paper normalized the processing flow and designed an accelerating circuit that can dynamically reconstruct the convolutional layer network and operation mode by software definition. The flow computation of convolutional neural network is realized on FPGA by SoC dual bus architecture. Under the condition that the network model and the input data set are the same, the calculation efficiency and energy consumption of CPU, GPU and the designed FPGA circuit are compared, and the data set processing under two different network structures is completed on the FPGA by software switching in real time. The experimental analysis shows that, at the operating frequency of 200MHz, the designed FPGA circuit has an operation processing capacity of 10 times that of CPU, and an operation energy consumption ratio of 2 times that of GPU. Moreover, the circuit structure completes the working mode switch between the two network models through software configuration.

Key words: Convolutional Neural Network(CNN); Software Definition; Dynamically reconfigurable; FPGA; pipeline calculation; SoC

1 引言

随着计算机视觉技术的发展，神经网络凭借深度学习技术使极大地提高了算法性能。其中卷积神经网络（Convolutional Neural Network，后文简称CNN）作为该项技术的基础模型得到了广泛关注，并越来越多地应用于图像识别、视觉拟合、视频搜索等领域[1]。

目前CNN算法通常采用软件方式，在CPU或GPU等通用处理器上实现，但在CNN的算法结构中，层内运算互不相关，层之间各自独立，运算数据流可以建模为流水线处理[2]。而通用处理器主要针对逻辑事务等分支跳转应用场景进行了性能特化，所以并不适合CNN算法海量据搬移以及密集迭代运算的特征，并且通用CPU中通用运算单元也难以满足CNN算法大量重复乘累加计算的需求[4]。因此，越来越多的研究人员采用AISC或FPGA电路来对CNN算法进行硬件加速实现。

专用ASIC电路虽然具有主频高与运算能耗比高等特点，但由于其电路架构相对固定，所以只适用于算法模型与应用固化的场景。但ASIC电路由于其本身缺乏算法配置能力，使得该实现方式无法应对CNN算法快速衍进更新的需求[5]。FPGA（现场可编程逻辑阵列）作为一种可重构电路在相对灵活地实现CNN算法的同时，也具备硬件电路的高性能并行计算能力[6]。但FPGA电路可重构性建立在重新构建电路并加载BIT流文件的基础上实现的，电路重构时间长属于一种静态可重构电路，在架构实现上还存在优化空间。

本文通过研究卷积神经网络的数据结构特点，并归一化不同算法中的原子算粒，设计出了一种基于软件定义的卷积神经网络可重构电路。即通过可编程算粒引擎与路由总线构建算粒网络，同时利用多通道DMA高效实现不同搬移规律的二维数组展开。最终实现针对不同的卷积神经网络模型地软件快速重构，并保持高性能运算处理能力。

2 卷积神经网络特征分析

2.1 卷积神经网络算粒解析

卷积神经网络是一种以多层神经为基础的监督型学习算法，通常由卷积层、全连接层、池化层、以及激活层组成[7][8][9]。通常的工作流程为将图像作为输入，交替迭代进行卷积层提取特征于池化层子抽样得出特征图，最终经全连接层运算得出结果。

卷积层主要功能是完成图像的特征提取运算，处理原理类似数字滤波器，以大小为M×N的卷积核为计算核心，对目标图像进行滑窗点乘遍历，抽象公式如下所示：

$$X \otimes Y = \sum_{i=1}^{M}\sum_{j=1}^{N} X(i,j) \cdot Y(i,j)$$

池化层负责对数据抽样处理，实现对特征的深度抽象，减少特征图数据维度。池化处理主要分为均值池化与最大值池化，即对特征图中选定的均分区域（N×N）进行均值计算或求最大值，这两种处理过程可以阐述为如下公式：

最大值池化：$A^{(1)}(i,j) = \max\left[I^{(1)}_{K\times K(i,j)}\right]$

均值池化：$x_j^l(i,j) = f\left(\frac{1}{n}\sum_{i\in M_j} x_i^{l-1} + b_j^l\right)$

激活层完成对数据的输出控制，即阈值选通传输操作。在CNN网络中常见的激活函数包括：纠正线性函数ReLU、双曲正切函数tanh、S形sigmoid函数、以及符合函数等[10]，运算过程为在激活函数上对将激活前的输出值进行乘叠加后输出，常见的激活函数公式如下表所示：

表1 常见激活函数公式

函数名称	激活函数公式
纠正线性函数ReLU	$ReLU(x) = \max(0,x)$
S形sigmoid函数	$Sigmoid(x) = \frac{1}{1+e^{-x}}$
双曲正切函数tanh	$Tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$
符号函数	$Sign(x) = \begin{cases} +1 & x \geqslant 0 \\ -1 & x < 0 \end{cases}$

全连层通常处于CNN神经网络的最后处理部分，通过将特征图逐行读入并于权重参数矩阵相乘后，完成图像的最终分类处理，整个处理过程对应的数学公式如下：

$$A^{(1)}(j) = B^{(1)}(j) + \sum_{i=1}^{N_{in}^{(1)}} B^{(1)}(i) \cdot W(i,j)$$

2.2 卷积神经网络结构分析

自从1994年世界上第一个卷积神经网络模型LeNet-5被提出以来，多种不同的CNN网络模型相继问世，例如AlexNet、VGG、GoogleLetNet等等

模型[12]。不同的算法模型分别在卷积层与子抽样层互连/迭代层数、卷积核运算尺寸、激活层函数处理方式等诸多处理环节进行不同的优化。AlexNet模型以LeNet-5模型为蓝本，通过增加卷积层进一步提高了模型图像识别率；VGG模型通过使用较小卷积核模拟更大尺寸卷积核的方式，在保证层级结构统一且简单的基础上，进一步增加了模型运算深度，并提高识别精度[13]；而GoogleLetNet模型则通过设计不同尺寸并行卷积模块，并配合传统的池化模块与卷积模块的拼接，在降低运算量与参数量的基础上，仍然保持了图像的高识别率[14] [15]。

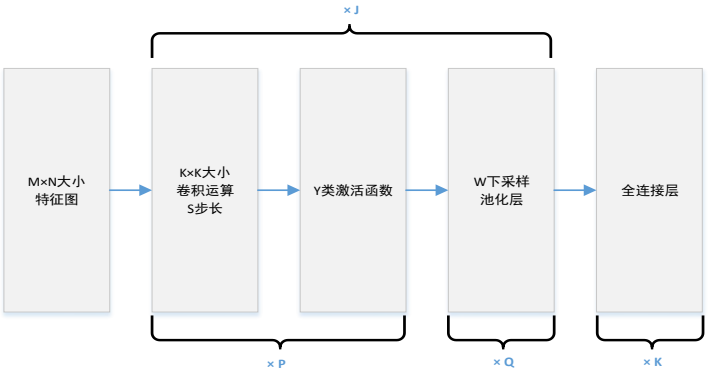


图1卷积神经网络处理流程

尽管上文分析的CNN算法处理特性与运算规模不尽相同，但仍然可以通过一个参数化的处理流程进行归一化总结。如图1所示，其中K×K大小的卷积核构成了CNN网络运算核心，以不同步长S与Y类型的激活函数配合，实现卷积层处理；再配合Q个下采样数W的池化层，可堆叠出J深度的连续卷积处理；最终与K个全连接层互连，完成整个CNN流程处理。如果可以抽象出以上流程中各参数实现方式，那么便可以实现一种软件可定义的可重构卷积神经网络架构。

3　可重构CNN电路设计

3.1　总统架构设计

根据上文对卷积设计网络模型的分析，我们可以发现其本身涉及的运算种类相对固定，访存地址规律固定，数据结构规整。同时，整体运算与数据可重用率高，所以本文通过卷积核尺寸、卷积滑窗步长、卷积迭代层数、池化下采样抽取尺寸以及激活函数类型等功能特性，进行基于软件定义的通用性设计，以增加部分迭代流程为代价，归一出一种适应多种CNN网络的电路结构，整体结构如图2所示。

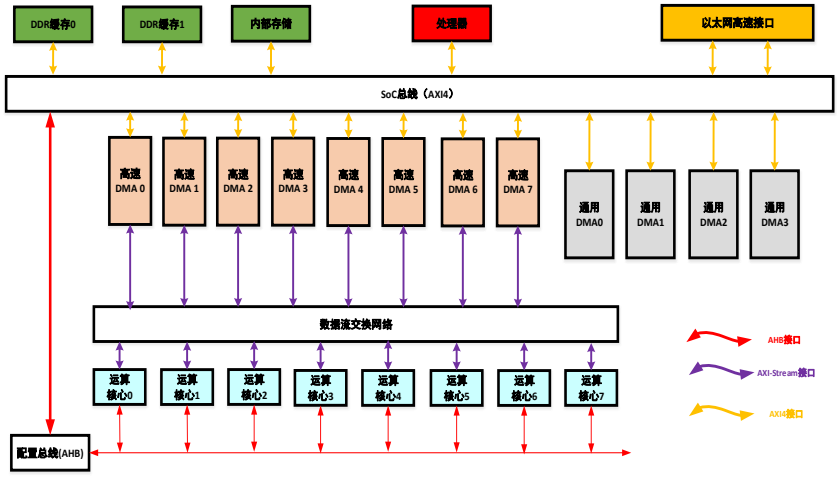


图2动态可重构CNN网络电路架构框图

本文所设计的可重构卷积神经网络是基于SOC架构设计，通过处理器的软件配置，实现整体电路的动态可重构。本架构通过高性能DMA与通用DMA，将数据缓存中不同尺寸的二维特征图与卷积核参数转换为一维数据流，并分别通过数据交换网络与配置总线进入到核心运算模块，完成卷积运算；而且数据交换网络采用基于路由的全交换结构，可以实现任意运算核心以及高速DMA之间的数据流互连。因此，本架构电路可以针对不同的CNN网络模型的处理流程，软件定义数据流路由，以及配置各运算核心工作模式，处理器启动各DMA进行数据流搬移，经过数据流迭代搬移，实现深度卷积层处理。

3.2 重配置运算核心

两输入，单输出；其余与XE相同，二维展开运算的核心。

针对于卷积神经网路而言，其主要运算类型相对固化，主要的计算类型总结而言如下：

（1） 乘法：两个一维向量数据流相乘运算，用于卷积运算中乘法以及因子缩放；

（2） 加减法/累加：实现卷积运算中累加运算，也可以实现池化运算中偏置常量加法；

（3） 比较：一维向量数据比较结果输出，或非等置位等处理，可以实现激活函数或池化采样等处理；

（4） 最值统计：求取一维向量中最大/小值，可实现ReLU激活函数操作；

（5） 移位：通过对数据的额左移/右移，实现输入数据的2的幂次方运算，实现数据缩放，节省乘法资源；

（6） 向量复制：将输入的一维向量复制多出一路，实现向量平方运算，用于可能的超越运算泰勒级数展开处理；

（7） 序列生成：针对输入的常数，展开为任意长度的一维数据流，实现矩阵常数乘法；

（8） 数据直通：将输入的数据不做任何处理，直接透传到下一级运算核心。

本论文中运算核心处理模块，主要负责完成以上总结的处理，运算核心整体数据流采用基于握手的流水线处理，所有运算核心输入输出均与数据网络互连，因此可以配合数据交换网络完成核心间数据流交换。针对以上所有处理进行分类，将移位、向量复制、序列生成等处理统一放入运算核心数据入口处作为数据预处理模块；将乘法与加减法/累加等操作作为运算处理模块归为一类；最后将最值统计与比较操作做为后处理子模块实现，整体模块框图如图2所示：

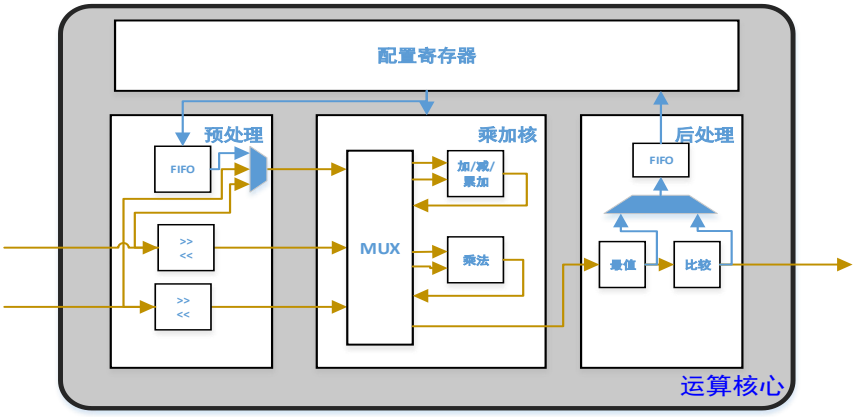


图2重配置运算核心框图

运行核心模块主要划分为四个子功能模块：预处理模块、乘加核模块、后处理模块以及配置寄存器模块，各模块主要功能如下：

预处理模块：负责移位、复制以及序列生成等功能，子模块为2输入3输出设计，新增路数据路径用于传输复制或常数序列。内部设计下行FIFO，通用DMA可以通过AHB配置总线向FIFO中自动写入数据，以实现将FIFO参数进行常数展开。移位操作可以通过配置为0移位实现数据透传。

乘加核：内部设计全互联数据MUX模块，可以通过配置实现数据流通过2个运算、1个运算、或直接透传，并且可以决定进入乘法与加法运算

核的先后顺序，以实现不同算法公式；同时乘法与加法模块实现采用单精度浮点运算。

后处理模块：主要完成对对任意尺寸的1维输出数据的最值提取，以及比较后降采样操作，并且设计上行FIFO，通用DMA可通过配置总线读取FIFO中最值或比较值结果，上送SoC总线后缓存至内部存储中。

配置寄存器：实现对运算核心的工作模式，实现对1维数据尺寸、移位位数、加法/累加/减法、最大/小值、数据复制、比较阈值等模式的切换，实现运算核心的重配置。

整个运算核心采用两数据流输入一数据输出接口数量，所有输入输出接口均与数据交换网络相连，数据来源为高速DMA从DDR中读出的数据缓存。数据流输出接口附带目前模块路由信息，用于数据交换网络进行路由。同时，内部设计与配置寄存器相接的上行与下行FIFO，FIFO与通用DMA设计握手连接信号，实现通用DMA对FIFO状态的感知，以实现数据流量调节，这样便可实现常数数据的批量1维展开，实现各种卷积神经网络的运算处理。

3.3 软件定义DMA

本论文中的高速DMA与通用DMA是数据搬移核心，完成数据从SoC总线上存储设备与运算核心之间的数据搬移。本架构中内置一个处理器，用来管理两种DMA的指令队列，用于DMA内数据通道的配置，DMA功能总结如下：

所有DMA均可以通过SoC总线访问总线上所有存储；

所有DMA内置指令队列，用来定义源地址、目的地址、数据长度、burst个数、地址变化方式等描述符任务；

所有通用DMA支持4个逻辑通道同时数据搬移；

高速DMA支持固定地址、顺序地址、跳跃地址、跳跃片段地址等地址跳跃形式；

高速DMA内部针对读写各有一个数据通道，实现复杂的地址循环，将SoC总线读取的数据直接转化为数据流送入数据交换网络中，负责批量读取特征图数据与卷积核参数；而通用DMA中采用多逻辑通道时分复用的形式，负责低速参数的自动搬移。工作过程中两种DMA同时被处理器启动，高速DMA将一维化特征图数据经过总线与数据交换网送人运算核心，与通用DMA送入的参数或其他一维化特征图进行运算，可实现二维数据的流水式运算。

3.4 数据交换网络

数据交换网络主要完成对连接其上模块进行交换路由，实现点对点之间数据流的交互，当前接口采用AXI4-Stream协议，可实现整个网络接口之间通过握手机制形成流控。AXI4-Stream协议接口中分为master与slave接口，master接口为输出接口，slave为输入接口，两种接口互为对称衔接。数据交换网络对所有master接口进行编制，从slave接口输入对应的路由地址，即可实现网络内对应slave到master之间的路由。

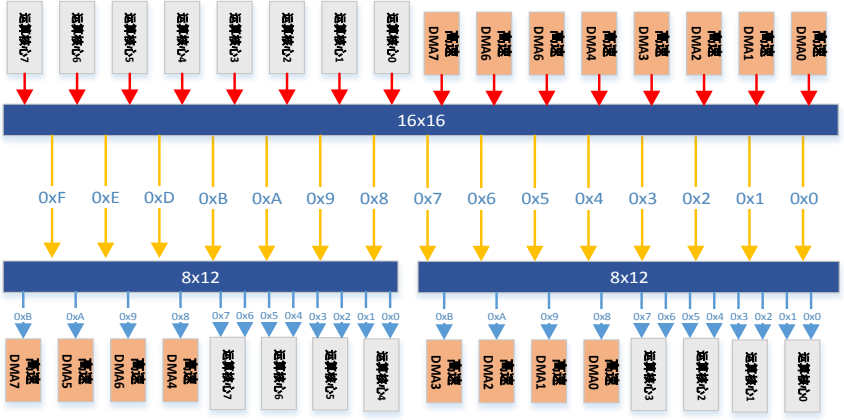


图4数据交换网络拓扑图

数据交换网络根据不同算法流程规划的运算核心个数为8个，同时每个运算核心都是二输入一输出的端口数量，合计master接口数量16个，

slave 数量 8 个，并且在计算 8 个高速 DMA 的 AXI4-Stream 接口的 master 与 slave 各 8 各接口。整个交换网络的需要 24 个 master 接口，以及 16 个 slave 接口。由于采用的总线 IP 最大支持 16×16 规模交换网络，所以采用两级拓扑接口实现 16×24 数量接口的数据网络。如图 4 所示，每个 master 接口均被编址。

4 FPGA 实现与结果分析

4.1 FPGA 实现

本文进行硬件设计的开平台为 Xilinx 公司的 ZC706 开发板，其 FPGA 芯片包含 PS 和 PL 两种硬件资源，利用其 PS 内的 ARM 双核处理器、AXI4 总线、以及 SDMA 等 IP 分别实现本论文中处理器、SoC 总线、以及通用 DMA 功能；在 PL 部分实现运算核心、数据交换网络与高性能 DMA。测试集为 MNIST 集合中 200 张 256×256 大小的图片，每个 Pixel 的取值范围是 0～255 之间的整数，在进行实现前将所有像素转换为单精度浮点数后进行试验测试。测试算法包括 AlexNet 与 VGG-16 两种不同算法分别在本论文电路架构上实现运算，并使用相同算法与输入数据集分别在通用处理器 Intel CORE i5-7200U 与 NVIDIA GTX1050 GPU 上进行处理时间的性能比对，并且 CPU 与 GPU 采用与本设计电路相同的数据精度，均为单精度浮点数据。

4.2 实验结果

本文设计中的 FPGA 芯片内部处理主频为 200MHz，由于在 FPGA 中实现的卷积神经网络电路可以通过 ZYNQ 中 ARM 处理器的配置，完成电路的动态重构。所以在切换不同网络模型时，FPGA 无需改变电路结构，可以采用与 CPU 或 GPU 相同的方式直接切换不同工作软件即可。下表为整个 FPGA 芯片资源使用情况：

下面三种运算平台使用 AlexNet 网络模型，针对 200 张 256×256 大小的单精度浮点图像进行运算，计算速度比对如表 3 所示。经过比对可以看出，本设计的 FPGA 处理电路是通用 CPU 处理速度的近 10 倍，但相比 GPU 并无性能优势，不过考虑到主频差异，本设计 FPGA 在性能功耗比上占据一定优势，GPU 能耗比约为 200÷1.722÷300≈0.387（FPS/W），可重构 FPGA 设计为 200÷31.84÷7.8≈0.8053（FPS/W），约为 GPU 的 2.08 倍。

表 2 FPGA 资源使用表

资源类型	资源数量	占用比例%
LUT	188831	85.17
Flip-Flops	258938	59.22
DSP48E	804	89.24
Block RAM	677	62.03

表 3 AlexNet 模型算法效率对比

平台类型	图像数量	处理时间（s）	功耗（W）
Intel CORE i5-7200U	200	310.07	25
NVIDIA GTX1050 GPU	200	1.722	300
本设计 FPGA	200	31.84	7.8

切换三个平台网络模型处理软件，切换为 VGG-16 算法，并对 100 张 256×256 大小的单精度浮点图像进行运算，三个平台处理算法速度比如表 4 所示。三个平台处理效率比率相同，并且本设计 FPGA 平台实现了软件配置切换实现不同卷积神经网络算法。

表 4 VGG-16 模型算法效率对比

平台类型	图像数量	处理时间（s）	功耗（W）
Intel CORE i5-7200U	100	252.55	25
NVIDIA GTX1050 GPU	100	1.412	300
本设计 FPGA	100	26.10	7.8

5 结束语

本文设计的基于软件定义的卷积神经网络可重构电路设计，归一化多种 CNN 网络模型处理流程，并采用 SoC 两层总线架构，通过可编程 DMA 配合可重构运算核心实现动态可重构多种 CNN 算法的目的，并最终在 Xilinx 公司的 ZYNQ 平台完成 FPGA 电路开发。同时，经过同通用 CPU 与 GPU，在相同算法模型下比对相同数量处理效率比对实验中，达到加速性能高于 CPU 近 10 倍、计算效能高于 GPU 约 2 倍的效果。既实现了通过软件定义实现 CNN 网络模型动态可重构，也保证较高的加速处理性能。在未来的工作中，会进一步整合并添加其他新型 CNN 算法模型运算核心，以进一步提高本电路的应用普适性。

参考文献:

[1] Singh K , Tiwari S C , Gupta M . A closed-loop ASIC design approach based on logical effort theory and artificial neural networks [J]. Integration，2019，69:10-22.

[2] Chen J , Prodic A , Erickson R W , et al. Predictive digital current programmed control [J]. IEEE Transactions on Power Electronics，2003，18(1):411-419.

[3] Wu D , Pigou L , Kindermans P J , et al. Deep Dynamic Neural Networks for Multimodal Gesture Segmentation and Recognition[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence，2016，38(8):1583-1597.

[4] Lecun Y , Bottou L . Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE，1998，86(11):2278-2324.

[5] Chellapilla K, Puri S, Simard P. High performance convolutional neural networks for document processing [C]//10th International Workshop on Frontiers in Handwriting Recognition，2006

[6] Yu Y , Wu C , Zhao T , et al. OPU: An FPGA-Based Overlay Processor for Convolutional Neural Networks[J]. IEEE Transactions on Very Large-Scale Integration (VLSI) Systems，2019，28 (1) : 35-47.

[7] Liu Z , Dou Y , Jiang J , et al. An FPGA-based processor for training convolutional neural networks[C]// 2017 International Conference on Field Programmable Technology (ICFPT). IEEE，2018.

[8] 方睿, 刘加贺, 薛志辉, et al. 卷积神经网络的FPGA并行加速方案设计[J]. 计算机工程与应用唯一官方网站，2015，51(8):32-36.

[9] 王巍, 周凯利, 王伊昌等. 卷积神经网络(CNN)算法的FPGA并行结构设计[J]. 微电子学与计算机，2019，036(004):57-62,66.

[10] 梁爽. 可重构神经网络加速器设计关键技术研究[D].

[11] 仇越, 马文涛, 柴志雷. 一种基于FPGA的卷积神经网络加速器设计与实现[J]. 微电子学与计算机，2018，035(008):68-72.

[12] 翟社平, 邱程, 杨媛媛, et al. 基于FPGA的卷积神经网络加速器设计与实现[J]. 微电子学与计算机，2019(8):83-86.

[13] 窦阳, 卿粼波, 何小海,等. 基于FPGA的CNN加速器设计与实现[J]. 信息技术与网络安全，2019，038(011):P. 96-101.

[14] 周瑛, 张铃. 模糊集方法在检索评价系统中的应用[J]. 计算机技术与发展，2007，17(001):111-113.

[15] Venugopal S , Castro-Pareja C R , Dandekar O S . An FPGA-based 3D image processor with median and convolution filters for real-time applications[C]// Proc of Spie-is&t Electronic Imaging. 2005.

人脸检测技术综述

朱灵灵[1, 2]，高超[1]，陈福才[1]，李辉[1]

[1]中国人民解放军战略支援部队信息工程大学，河南省 郑州市 450002；

[2]郑州大学 中原网络安全研究院，河南省 郑州市 450001

摘　要：人脸检测算法是计算机视觉信息处理的经典问题，其主要是对人脸进行定位的任务，经过多年的研究已经相对比较成熟。本文分别阐述了人脸检测技术目前面临的主要难点、常用检测指标、常用数据库、人脸检测算法的分类以及应用场景等五个方面。其中根据对人脸检测算法发展历程和现状的研究，将检测算法主要概括为三种类型进行介绍和分析，分别为早期算法、AdaBoost框架、深度学习框架。

关键词：人脸检测、早期算法、AdaBoost方法、深度学习框架

A Review: Face Detection Algorithms

Abstract: Face detection algorithm is a classic problem of computer vision, which is mainly for the task of face location, and after years of research has been relatively mature. This paper describes the main difficulties of face detection technology, detection indicators, databases, detection algorithms and application scenarios. According to the research on the development of face detection algorithms, they are divided into three categories: early algorithm, AdaBoost framework and deep learning framework.

Key words: face detection; early algorithm; AdaBoost framework; deep learning framework

1　引言

生物特征识别技术已经成为近年来识别个体的主要方式，而人脸检测对比其他生物特征识别方法更具优势，无需用户配合检测即可通过摄像头从远处获得，从而得到较为广泛的应用。人脸检测算法是计算机视觉信息处理的经典问题，一直得到国内外学者的关注与研究，并被认为是图像分析领域中最成功的应用之一[1]。虽然目前人脸检测技术得到了快速发展，但其在现实生活的应用中仍面临着多方面的考验。如何更好地解决面临的难点是各研究学者将人脸检测技术应用于实现中一直需要解决的问题。

2　人脸检测的难点

人脸检测算法是计算机视觉信息处理的经典问题，由于其广阔的应用场景及实际需求，一直深受各研究学者的关注。得益于先验知识，人脸检测对于人类来说较为简单、迅速；而相对于计算机，该技术的研究出发点是在降低计算量和参数量的前提下，提高检测的精确率和召回率。而目前人脸检测技术在现实生活中的应用面临的难点可以概括为以下三个方面：（1）复杂背景和光照条件：光照强度的变化对人脸检测性能的影响较大；（2）表情问题：面部表情变化过大也会在一定程度上对人脸检测的性能带来扰动；（3）遮挡问题：可以分为自遮挡和外部遮挡两种类型。这些遮挡在现实生活中都较为普遍，这给人脸检测任务带来了较大的困难。

3　评价指标

评价人脸检测算法的效果一般采用如下三个指标来衡量：召回率、精确率、检测速度。

1　召回率

通俗来讲，检测器能够正确检测出来的人脸数量越多表明检测性能越好，而由于每幅图像中所包含的人脸数量不确定，所以需要用检测出来

基金项目：郑州市协同创新重大专项（162/32410218）

的人脸数的比例来衡量，即为召回率Recall。由公式（1）得出召回率，

$$R = \frac{TP}{TP + FN} \tag{1}$$

式中，R召回率，FN人脸被错误预测为背景的数量。

2 精确率

检测器能够正确检测出人脸的同时，也有可能会把其他东西误认为人脸，我们同样采用正确检测出的人脸数的比例来衡量检测性能，即为精确率Precision。公式（2）得出精确率，

$$P = \frac{TP}{TP + FP} \tag{2}$$

式中，P表示精确率，TP被正确划分的人脸数量，FP背景被错误划分为人脸的数量。

3 检测速度

评价检测器的检测效果的同时，还要衡量其检测速度，从多个角度综合评价人脸检测器的性能。检测器检测速度越快越好，通常表示为每秒帧率（frame-per-second， FPS），即平均每秒内可以处理的图像数量。对比检测速度时，需要注意保证测试环境、实验配置等因素相同，比如，CPU的型号和主频的差异、多核多线程差异、GPU型号的差异等都会对检测速度的衡量带来影响。故对比数据时，为了保证对比结果的科学性和可靠性，通常需要确保其他因素的一致性。

4 常用数据库

1 FDDB

FDDB（Face Detection Data Set and Benchmark）[2]数据集总共2845张图像且标记5171个脸部注释，主要用于研究人脸在非约束环境下的检测问题。数据集中人脸的难度主要包含表情、光照条件变化、配饰、发型、是否遮挡等难点。同时该数据集具有的特点：一般将图像的较长边缩放到450，得到的图像分辨率通常较小；包含彩色、灰色两种图像；数据集中的多数图像中包含的人脸数较少。大多只含一个人脸。

2 WIDER FACE

另一种数据集为WIDER FACE [3]，通常采用WIDER FACE数据集来对人脸检测定位任务和人脸分类任务进行训练，该数据集含有32203张图像，其中一共标记393703个人脸，其中包含化妆、遮挡、角度、不同尺度、姿势等情况下变化的图片，如图1所示，数量较大，其广泛应用于评估卷积神经网络的性能，是一个人脸检测基准数据集。

图1 Wider face中不同尺寸、姿势、遮挡、化妆等情况下变换的图片示例

5 人脸检测技术的常用方法

1 早期算法

模板匹配是早期算法中的主流方法，即，该方法首先人工定义一个人脸模板，对比待检测图像的各个位置并匹配，进而判断每个位置中是否包含人脸。机器学习算法后来被用于该问题，包括神经网络、支持向量机等。这些都是针对图像中的某个区域进行人脸-非人脸的二分类问题。

Yuille [4]设计一种变形模型作为目标模板，

之后比较模板和待检测图片间的同位置处的像素差异从而进行模板的匹配。Kass［5］设计了Snake模型，并利用该模型来估计各面部器官的轮廓曲线，后采用相应的方法确定具体的面部特征，设定当获得的面部特征达到设置的固定值时，则视为在图片中检测到人脸。

Rowley［6］［7］提出的方法在早期具有代表性，文献［6］通过对人脸和非人脸数据集训练得到一个多层感知器，进而判断是否为人脸。但该方法具有局限性，不能很好地检测出非正面的人脸图像，针对这一问题，文献［7］进一步设计一种检测模型，该模型由连个神经网络组成，第一个网络输出人脸的倾斜角度，后利用这个角度将检测框旋转，之后输入第二个网络对倾斜后的图像判断是否为人脸。

2 AdaBoost框架

Boost算法主要是基于PAC（probably approximately correct）的方法，其思想是通过多个简单的弱分类器级联成准确率较高的分类器，精度和速度相比早期算法都得到了有效的提高。

文献［8］设计的VJ框架使用Harri-like特征和级联的AdaBoost分类器构造，这种方法有效提升了检测速度和精度。类似地，AdaBoost目标检测框架主要是使用多个AdaBoost分类器组成级联结构来对候选框图像判定其是否为人脸。如图2所示，在AdaBoost分类器中，靠前的分类器可以快速筛选出非人脸的候选框，但同时也会将背景图像误判为人脸。若一个候选框最终通过了所有分类器，则判定该候选框为人脸，否则，反之。

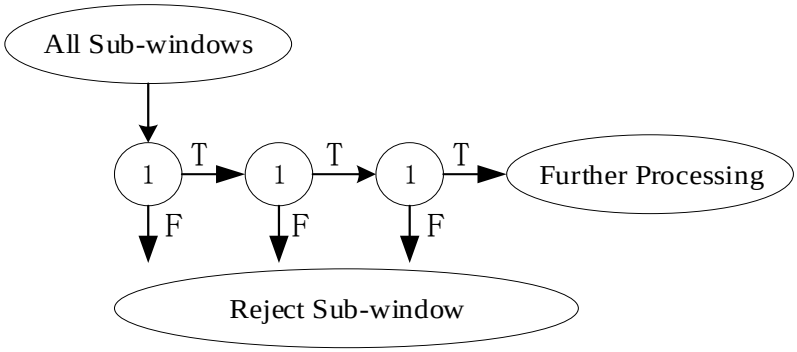


图2 AdaBoost分类器检测图

此外，可变形组件模型（Deformable Part Model，DPM［9］）也是比较典型的方法，可以看做是一种基于组件的方法，检测效果相对较好。但该模型的计算复杂度高，检测速度慢，因此在工业界中很少使用，一般采用AdaBoost框架。

这种基于手工设计特征的方法稳定性不好，易受检测环境的影响，故在复杂场景下较难保证人脸检测的性能，应用场景受到限制。而深度学习的方法提出后，深度卷积神经网络具有抗干扰能力强、学习特征能力强等特点，可以得到很好地应用。下面主要介绍深度学习框架的人脸检测方法。

3 深度学习框架

深度学习纪元从2012开始一直持续到现在，2006年，深度学习鼻祖Hinton就提出了深度信念网络，而在2012年Hinton利用深度卷积网络训练的分类器夺得了ImageNet比赛冠军，在一定意义上开启了图像分类和识别的新纪元。图3展示了人脸检测系统的主要步骤，人脸检测将图像窗口分为两个部分：一个包含人脸、一个包含背景。理想的人脸检测器能够在任何光线条件下、任何背景下检测出任何面孔的存在。

3.1 基于深度卷积网络的算法

相比传统方法，由于深度学习具有明显的优势，人脸检测方法越来越趋向于采用基于卷积神经网络（Convolutional Neural Networks，CNN）的方法。Garcia等人［10］设计了一种可以处理部分不同的人脸姿势和角度变化的单一多层网络对人脸进行检测，可以看做是传统技术和深度网络相

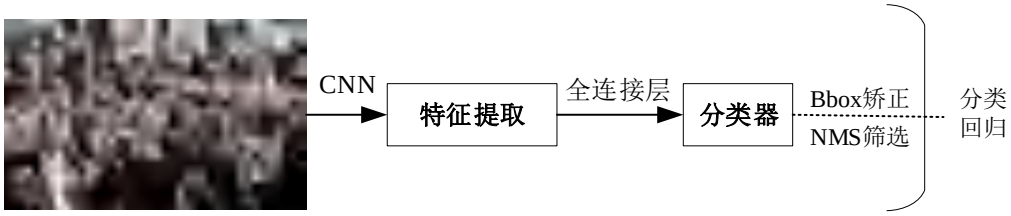


图3　人脸检测主要步骤

结合的一个代表。Osadchy 等人［11］提出的联合训练神经网络检测人脸，显著提高人脸检测模型的性能。Li 等人［12］提出一种基于级联方法的CNN，但该模型需要校准人脸的候选框边界，这会消耗额外的计算量。

DenseBox [13] 检测算法，设计使用全卷积网络直接预测输入图像中的目标对象的位置坐标和类别置信度。Kaipeng [14] 提出的MTCNN方法是用于人脸检测任务的多任务神经网络模型，主要借鉴级联的思想，即采用三个PNet、RNet和ONet网络顺序对人脸图像进行处理并预测，同时加入了关键点回归进而提高检测性能，目前在很多工业级场景中得到了应用。Face R-CNN人脸检测 [15] 则是借鉴Faster R-CNN算法，并对人脸形状的做出针对性的优化。M. Najibi［16］等人提出的SSH算法，通过对骨干网络VGG不同卷积层的输出做分支，可以变相地实现多尺度的检测。PyramidBox算法 [17] 设计了一种基于锚点的上下文信息辅助方法PyramidAnchors，引入上下文信息来学习其他较难提取的人脸特征，如部分遮挡、低分辨率等。

3.2　轻量级人脸检测算法

随着移动设备的快速普及，人脸检测算法需要应用在手持设备、车载设备等移动终端上。然而这些终端设备在计算能力、存储能力等方面相对服务器和个人计算机设备局限性很强。为了能使基于深度网络的人脸检测算法较好地运行在移动终端设备上，基于轻量级卷积神经网络的算法得到越来越多人的关注。

2016年，文献［20］提出Inception v3，该方法将一个二维卷积拆分为两个相对较小的卷积，如将7*7卷积拆分成1*7和7*1卷积，从而可以降低参数量；同年，文献［18］提出的SqueezeNet是采用1*1卷积核代替3*3卷积核，有效减少了输出特征图的通道数。文献［19］提出轻量级卷积网络MobileNet，在尽量保证检测精度相当的基础上，减少该网络的参数量和计算量。而Xception［21］严格意义上并不是轻量化模型，仅仅是其借鉴深度卷积，而深度卷积又是上述几个轻量化模型的关键点。

同时近年来，科学家将研究重心转移到轻量级人脸检测算法的研究，从而可以更好地将人脸检测技术应用在移动端设备上。ZQCNN［25］是一款由国内开发者ZuoQing开源的人脸检测、关键点定位的代码库，主要基于MTCNN算法实现人脸检测，可以检测出侧脸、低头，但精度不高，同时人脸框不是特别稳。2019年10月，用户Linzaer［22］在Github上开源了一款适用于边缘计算设备、移动端设备以及PC的超轻量级通用人脸检测模型（Ultra-Light-Fast-Generic-Face-Detector-1MB），该模型文件大小仅1MB，其设计是针对边缘计算设备或低算力设备设计的，可以在低算力设备中如用ARM进行实时的通用场景的人脸检测推理。Libfacedetection［23］于2019年3月份左右宣布开源，可用于商业开发，该库将模型文件转化为C的静态变量，同时提供了INT8的量化模型参数，加速预测过程。其模型结构较简单，即轻量级的SSD架构，模型体积仅有3.34M。文献［24］提出的LFFD检测算法在人脸检测领域并不是精度最高的，但作为近来轻量级人脸检测模型的代表，却是极具实用价值的。其不仅适用于人脸检测，实则是通用的一类目标检测器，可扩展到行人检测、人头检测、车辆检测等。Retinaface-Mobilenet-0.25项目是将RetinaFace［26］中的骨干网络Resnet50替换为MobileNet0.25，该模型大小仅1.68MB。CenterFace模型［27］可以算是CenterNet针对人脸检测任务的特例，这一点和RetinaFace作为RetinaNet的在人脸任务的特例有异曲同工之妙。

6 人脸检测的应用场景

随着深度神经网络的发展，人工智能领域取得了重大进展，以及各类高清图像采集设备的普及，人脸检测、识别技术得到迅速发展和应用，近几年准确率已达到甚至超越人类水平。人脸检测、识别技术作为生物识别技术一种，目前较热门的应用领域可以分为三个方面：

（1） 身份认证与安全防护

由于人脸检测和识别的方便性和安全性，以脸部信息作为主要特征的识别认证得到广泛应用，具有研究价值和商业价值。目前的主要应用在嫌犯追踪系统、楼宇门禁系统、智能办公室，同时智能手机的支付系统等。

（2） 媒体与娱乐

由于面部信息的特殊性，人们与脸部特征的互动越来越多，用于日常生活的娱乐活动中。如智能手机、数码相机的美颜相机APP中应用人脸检测识别技术，可以增加用户的互动。

（3） 图像搜索

传统的图像检索技术主要是基于文本，而基于人脸识别的图像搜索技术近年来得到发展，如2010年提出的百度识图，输入一幅图像，检测是否包含人脸，如果是，则搜索近似的图像，并全网搜索出现过的相似人脸图像。

7 结束语

本文简单概括了人脸检测的相关难点、评价指标、常用数据库，并将人脸检测方法分为三类进行介绍和分析，最后对人脸检测的应用场景做出了讨论。目前而言，相对于不同框架导致的人脸检测算法检测率的提高程度（如CNN类的算法比传统算法提高检测率），工程中更多的需要考虑的是摄像头的光圈、曝光、宽动态等问题，这些对人脸检测和识别的结果影响更大。

参考文献:

[1] ZhaoW.，alet（2000）"Face recognition: a literature survey", Technical Report CAR-TR-948, University of Maryland，October 2000.

[2] Jain V, Learned-Miller E. Fddb: A benchmark for face detection in unconstrained settings［R］. Technical Report UM-CS-2010-009，University of Massachusetts，Amherst，2010.

[3] Yang S, Luo P, Loy C, et al. Wider face: A face detection benchmark［C］// CVPR，2016：5525-5533.

[4] Yuille A L ,Hallinan PW, Cohen DS. Feature extraction from faces using deformable templates［J］. International Journal of Computer Vision，1992，8(2)：99-111.

[5] Kass M , Witkin A , Terzopoulos D . Snakes: Active contour models［J］. International Journal of Computer Vision，1988，1(4)：321-331.

[6] RowleyHenry A, BalujaShumeet, KanadeTakeo. Neural network-based face detection. 1998，IEEE Transactions on Pattern Analysis and Machine Intelligence.

[7] RowleyHenry A, BalujaShumeet, KanadeTakeo. Rotation invariant neural network-based face detection. 1998，computer vision and pattern recognition.

[8] S. Z. Li , L. Zhu , Z. Q. Zhang , A. Blake , H. J. Zhang , H. Y. Shum. Statistical learning of multi-view face detection. In：Proceedings of the 7-th European Conference on Computer Vision. Copenhagen, Denmark：Springer，2002. 67-81.

[9] MathiasM. , BenensonR. , PedersoliM. and Van GoolL. . Face detection without bells and whistles. ECCV 2014.

[10] Garcia C, Delakis M. Convolutional face finder: A neural architecture for fast and robust face detection［J］. IEEE Transactions on pattern analysis and machine intelligence，2004，26（11）：1408-1423.

[11] Osadchy M, Cun Y L, Miller M L. Synergistic face detection and pose estimation with energy-based models［J］. Journal of Machine Learning Research，2007，8(May): 1197-1215.

[12] Li H, Lin Z, Shen X, et al. A convolutional neural network cascade for face detection［C］// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015：5325-5334.

[13] Lichao Huang, Yi Yang, Yafeng Deng, Yinan Yu. DenseBox: Unifying Landmark Localization with End to End Object Detection. 2015，arXiv: Computer Vision and Pattern Recognition

[14] Kaipeng Zhan, Zhanpeng Zhang, Zhifeng L, Yu Qiao. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. 2016，IEEE Signal Processing Letters.

[15] Face R- CNN - H. Wang, Z. Li, X. Ji, Y. Wang. Face R-CNN. arXiv preprint arXiv：1706. 01061，2017.

[16] - MSSH. Najibi, P. Samangouei, R. Chellappa, L. Davis. SSH：Single Stage Headless Face Detector. IEEE International Conference on Computer Vision (ICCV)，2017.

[17] PyramidBox - X. Tang, Daniel K. Du, Z. He, J. Liu PyramidBox: A Context-assisted Single Shot Face Detector. arXiv preprint arXiv：1803. 07737，2018.

[18] Sandler M. , Howard A. , Zhu M. , Zhmoginov A. , Chen L. C. ：Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. arXiv preprint arXiv：1801. 04381（2018）

[19] Howard A. G. , Zhu M. , Chen B. , Kalenichenko D. , Wang W. , Weyand T. , An- dreetto M. , Adam H. ： Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv：1704. 04861（2017）

[20] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception

architecture for computer vision［C］//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016：2818-2826.

[21] Chollet, F.: Xception: Deep learning with depthwise separable convolutions. arXiv preprint (2016）

[22] 2019，项目网址https://github. com/Linzaer/Ultra-Light-Fast-Generic-Face-Detector-1MB

[23] https://github. com/ShiqiYu/libfacedetection，2019

[24] He Y , Xu D , Wu L , et al. LFFD: A Light and Fast Face Detector for Edge Devices［J］. 2019.

[25] https://github. com/zuoqing1988/ZQCNN，2018

[26] DengJ. , GuoJ. , ZhouY. , YuJ. , KotsiaI. , and ZafeiriouS. . Retinaface: Single-stage dense face localisation in the wild. arXiv：1905. 00641，2019. 1，3

[27] https://github. com/Star-Clouds/CenterFace，2019

[作者简介]

朱灵灵（1995-），女，硕士，研究生，计算机视觉

高超（1982-），男，博士，助理研究员，计算机视觉

陈福才（1974-），男，教授，研究员，网络安全

李辉（1996-），男，硕士，研究生，网络大数据分析

拟态多执行体调度算法研究进展

朱正彬，刘勤让，刘冬培，王崇
解放军战略支援部队信息工程大学，河南郑州 450002

摘　要：拟态防御是一种基于DHR架构的新型主动防御技术，具有动态、异构、冗余以及负反馈内生安全特性，能显著提高系统的鲁棒性和安全性。其中拟态调度算法是拟态防御技术的关键，极大提高了系统抗广义不确定扰动的能力，其主要考虑系统动态性、异构性以及冗余度。本文从调度对象、调度数量以及调度时机等三方面对现有拟态容错调度算法进行了总结归纳，并详细分析了它们的优点与不足，最后展望了拟态调度未来的研究方向与趋势。

关键词：拟态防御、内生安全、拟态调度

Research Progress of Mimic Multi-execution Scheduling Algorithm

Zhu Zhengbin, Liu Qinrang, Liu Dongpei, Wang Chong
Information Engineering University, Zhengzhou, Henan, 450002

Abstract: Mimic defense is the new active defense technology based on the DHR architecture. With dynamic, heterogeneous, redundant and negative endogenous security features, it can significantly improve the robustness and security of system. Among them, the mimic scheduling algorithm is the key to mimic defense technology, which greatly improves the ability of system to resist generalized uncertain disturbances,mainly focusing on system dynamics, heterogeneity and redundancy. From three aspects, the existing mimic fault-tolerant scheduling algorithms are summarized in this paper: scheduling object, scheduling quantity and scheduling timing, and their advantages and disadvantages are analyzed in detail. Finally, the future research direction and trend of mimic scheduling are prospected.

Key words: mimic defense; endogenous security; mimic scheduling

1　引言

随着互联网技术的不断发展和更深层次的应用，当今社会已进入到“互联网+”和“万物互联”的时代，网络遍及我们生活的各个角落。与此同时网络空间时刻存在未知漏洞和后门等不确定性威胁，使得恶意攻击者利用少量的资源或代价就能肆意践踏个人乃至公众的隐私权，造成当今网络空间易攻难守的非对称局面。传统网络安全技术主要通过亡羊补牢式的策略来防范网络中频繁出现的各种网络威胁，如防火墙[1]、IDS[2]、IPS[3]等。同时漏洞挖掘、特征提取、蜜罐技术[4]、沙箱技术[5]等防御手段主要基于攻击手段和行为特征等先验知识的精确获得。但在现实生活中对于未知的攻击无法精确感知，同时对于所有软硬件漏洞后门无法穷尽。据研究统计平均一千行到一千五百行代码程序员就会在留下一个漏洞[6]，系统漏洞无法避免且容易被攻击者发现并加以利用，同时人为预留后门也存在极大安全威胁。

针对网络空间易攻难守的非对称局面，许多机构和学者提出了网络防御新思想，不再一味追求精确已知的漏洞后门，转而采取动态、容错的新型网络空间主动防御技术。如可信计算[7]、移动目标防御[8]等。可信计算致力于从芯片、硬件结构和操作系统等硬件底层做起，提供系统的可

基金项目：国家核高基重大专项基金资助项目(NO.2017ZX01030301)。

靠性、可用性、信息和行为安全性。移动目标防御（MTD）采用有效地址突变[9]、IP地址随机化[10]、端口随机化[11]、加密随机化[12]等多样性技术增强系统脆弱攻击面的不确定性、动态性，有效限制系统脆弱性暴露及被攻击的机会，增加攻击者扫描攻击难度，目前该技术已成功应用于软件定义网络（SDN）[13]。但目前MTD技术发展面临许多问题，例如缺乏一定的效能评估机制，系统开销过高影响服务性能以及虚拟环境中移动目标的安全和弹性技术等问题。

基于网络主动防御思想及移动目标防御技术，参考自然界中拟态章鱼能根据不同环境调节自身色彩、行为等来隐藏自己原本特征以此来躲避风险。邬江兴院士进一步提出拟态防御理论[14]，通过向系统引入动态、异构、冗余等特性增强系统广义鲁棒性和安全性。相比MTD技术，拟态防御技术通过引入仲裁和负反馈机制，使得系统在具有内在攻击面不确定性的同时，根据仲裁信息有针对性的调整系统内部结构，对外呈现动态性和广义不确定性，极大增加了攻击者的攻击难度，目前拟态防御技术已成功应用于路由器[15]、交换机[16]、网络操作系统[17]等。联合测试证明[14]在功能等价异构冗余的多维动态重构机制作用下，网络空间拟态防御（CMD）几乎不可能实现可靠、持续的协同逃逸，实验结果表明对于未知的漏洞后门有很好的防御效果。同时拟态调度是实现拟态防御的关键一环，本文在总结拟态相关研究的同时，重点关注拟态调度相关算法的研究。

本文组织结构如下：引言部分对当前网络生态及网络安全防御技术发展新趋势进行了分析概括；第2节对拟态架构、调度模型及调度算法要实现的目标进行详细阐述；第3节对现有拟态容错调度算法进行全面论述；第4节综合评估现有调度算法；第5节基于现有调度算法存在的不足，展望未来的研究趋势和方向；第6节对全文进行总结。

2 拟态调度架构

拟态防御基于动态异构冗余（DHR）构建动态、异构、冗余且具有负反馈特性的系统架构和运行机制，结合仲裁机制和多执行体调度策略实现对系统漏洞和后门的容忍。如图1[14]所示，在拟态防御架构中，主要包括输入代理、异构执行体池、异构构件池、拟态调度器、在线执行体集以及拟态裁决器等。其中，输入代理负责将系统输入数据复制分发，即将输入数据复制成k份分发给k个功能等价结构相异的执行体集，各在线异构执行体并行执行，并将结果发送给拟态裁决器；拟态裁决器采用如全体一致表决算法[18]、多数表决算法[19]，复数表决算法、一致性表决算法[20]等表决算法，计算各异构执行体输出来产生最终输出结果。同时拟态裁决器将各执行体的状态反馈给调度器，调度器决定是否需要根据当前态势使用特定的调度算法从异构执行体池中选择上线执行体，并对下线执行体进行清洗恢复等操作。DHR的异构性、动态性、冗余性以及负反馈特性使得系统在时间和空间上具备不确定性，攻击者难以掌握系统的脆弱点，进而使得拟态防御系统具备内生防御特性和天然免疫能力，拟态防御技术有望从根本上摆脱目前网络空间“易攻难守”的困局。

拟态防御技术注重系统架构、调度空间以及调度时机对系统整体安全性的影响，从而使系统对外呈现动态性和广义不确定性。其中多执行体调度算法是实现拟态防御的关键一环，执行体调度使系统保持高动态性和高异构度，能够避免攻击者长时间探测和协同逃逸，实现系统高鲁棒性和可靠性。一般调度策略可分为以下三步：

（1） 调度策略确定在线执行体余度，依据异构度等选取执行体上线；

（2） 确定在线执行体变换门限，不定时替换异常在线执行体，下线执行体清洗；

（3） 根据裁决信息反馈确定下一次变换时机和执行体冗余度。

传统主副版本容错调度[21-22]算法主要用于处理确定任务的调度进而实现容错，拟态多执行体调度算法主要从系统安全可靠性角度出发兼具系统运行效率等提出的一系列调度算法。综合考虑调度算法要实现的相关功能和安全性，一般来讲，主要从以下几个方面对调度算法进行评价：

2.1 动态性

容错调度算法使系统不断变换从而具有动态性，即对外呈现测不准效应。文献[23]提出算法动态性可以体现在调度方案的平均周期，即相同调度方案产生之间的时间间隔。理想的调度方案要

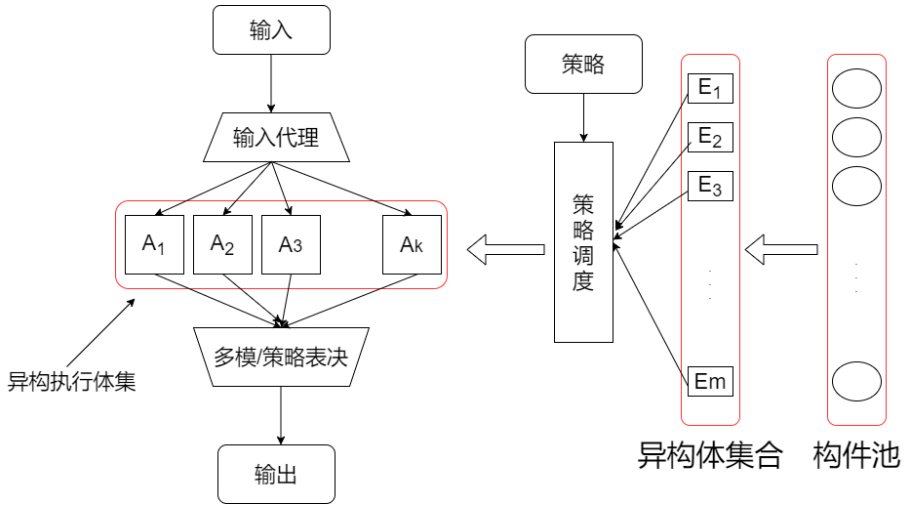


图1　拟态防御架构

求执行体集池足够大，完全相同的调度方案几乎不可能产生，现实中由于执行体余度的限制调度方案必然存在重复的可能性。在不影响系统原本功能，同时尽可能追求大的调度周期即提升系统动态性，是衡量某一调度算法的主要指标。

2.2　可信任度

拟态容错主要研究拟态架构能否在偶然或恶意的失效情况下连续、可靠、正常地执行。在拟态系统中各执行体的可信任度是系统可靠性的基础，不同的调度策略是系统可靠性的重要组成部分。对各执行体安全性量化以及系统整体性可信任度研究是调度算法亟待解决的问题。

2.3　异构度

拟态架构要求功能等价、结构相异的在线异构执行体，执行体集相异度越大存在共有漏洞的可能性就越低，被同一攻击者利用同一漏洞攻破造成的瞬时逃逸的概率就越小，执行体集相异性能显著提高系统安全性[24]。调度上线功能等价执行体异构度越大，系统安全增益越大，拟态容错调度算法追求最大执行体异构度，以避免系统发生瞬时逃逸。

2.4　系统开销

理想的调度算法单纯基于系统安全性考虑，不考虑系统开销及时间代价。现实研究中如果仅单纯追求系统安全性往往会花费极大的代价而只得到很少的系统安全增益。系统开销是衡量调度算法的重要指标，在追求系统安全性的同时减小运算复杂度、降低时间空间代价进而降低系统整体开销。

2.5　服务质量

拟态架构期望在不影响乃至提高系统原有功能的基础上，使得系统具备内生安全特性。但往往拟态防御机制的引入会对系统本身的服务属性造成一定的负面影响。考虑调度算法的同时要综合考虑系统的服务质量，在追求系统整体安全性最大化的同时，不降低甚至能提升系统本来的服务质量。

3　拟态调度算法

拟态调度算法主要从动态、异构、冗余出发结合系统内生安全特性，实现拟态系统的高鲁棒性和安全性。目前还没有相关调度算法全面的总结评估工作，经过大量阅读研究相关文献，我们对现有拟态调度算法进行了整理归纳，包括各调度算法的调度指标、执行效率等进行了分析比对，并对其优缺点进行了评估。本文综合分析各调度算法的主要调度思想，接下来将从调度对象、调度数量及调度时机等三方面出发详细论述现有调度算法。

3.1　调度对象

拟态架构在线执行体集要求功能等价、结构相异的异构执行体，在线执行体集相异度越大存在共有漏洞的可能性就越低，被同一漏洞攻破造成的瞬时逃逸的概率就越小。因此现有大多数调度算法集中于度量软件异构度并结合其他评价指标，从时间和空间两个维度出发以期望得到最大的在线执行体集异构度避免瞬时逃逸的发生。

3.1.1 基于软件异构度度量

基于对软件异构度的量化，文献[25]提出最长相异性距离组件选择（MD）算法和最佳平均相异距离的软件组件选择（OMD）算法。MD算法在选择软件时始终选取相异距离最长的组件，OMD算法选取具有最佳平均相异距离的软件组件，但该算法仅仅考虑内部组件之间的相异距离总和可能会与平均相异距离存在矛盾，可能找不到足够的组件组成相异性系统，对系统阈值设置要求比较高且缺乏动态性。

由于MD和OMD算法缺乏对于历史信息的考虑，吕迎迎等[26]提出基于历史信息的负反馈调度算法，利用系统监测得到的探测数据进行非一致性检验和非重复检验，分析该攻击类型进而根据不同攻击类型选择最合理的调度。仿真实验证明该算法较传统静态结构失效率相比非均匀攻击可降低50%，相比非重复攻击可降低58%。但该算法仅考虑外部探测对于执行体的影响，没有进一步研究上线执行体间的异构度和具有同一漏洞的概率。

为了使执行体调度对外呈现动态性的同时具备内部可控性，张震骁等[27]提出基于正态分布的拟态防御动态调度策略。采用序号系数n、安全系数s、时间系数t、人工控制系数a、综合系数c和概率系数p六个异构执行体属性计算某一执行体被选中的概率值：

$$p_n = \frac{1}{\sqrt{2\pi}} e^{-\frac{c_n^2}{2}}$$

该算法以正态分布为载体函数来计算某一执行体被调度具体概率，以达到对外呈现动态测不准，对内概率可控的广义调度算法，具有很好的随机性和概率可控性。但该算法时间复杂度较高，其载体函数值得进一步优化。

缺乏对于执行体当前安全性考虑，Li等[28]从系统安全性角度出发提出了一种SDN多控制器的动态自学习调度算法。通过控制器历史表现评估当前时刻该控制器可靠性，控制器组S_l可靠性可表示为：

$$Q(Z) = \begin{cases} -\eta(Z)^2 \times \dfrac{F(Z)}{U(Z)} & U(Z) \neq 0 \\ 0 & U(Z) = 0 \end{cases}$$

其中F（Z）/U（Z）表示工作中平均故障率，$\eta(Z)$表示集合S中元素的个数。根据历史表现自适应更新迭代，同时考虑负载约束利用贪心式启发算法GA求解出最优化调度集。该算法结合历史信息具备一定的负反馈特性，很好实现了对执行体当前安全度的考量，但在控制器增多时该算法复杂度聚增且缺乏进一步的实验验证。

3.1.2 基于异构体组件度量

缺乏对冗余体内各组成组件间相似性研究，刘勤让等[23]提出了一种新的冗余体间相似度的量化方法，通过衡量各组件间的相似度进而量化r余度冗余体集合整体相似度：

$$S|_{\Omega'} = \frac{1}{C_r^2} \sum_{i=1}^{r-1} \sum_{j=i+1}^{r} \sum_{l=1}^{m} \xi_l L_l^i Y_l L_l^{j\,T}$$

其中L_k^i表示执行体P_i中第k个组件对应的特征向量，ξ_l为系统组件相似度权重，Y_l表示组件特征相似度矩阵。通过设定各执行体间最小相似度阈值，结合最小相似度算法提出随机种子最小相似度算法（RSMS）。该算法在平均调度周期较于MD提升了300%，平均失效率较随机调度算法减低67.32%。但该算法在执行体数量较少时动态性值得进一步研究，同时缺乏对于历史裁决信息的考虑没有很好体现出拟态的负反馈特性。

王晓梅等[29]采取对比整体差异性的方式，将线下执行体和在线执行体的差异性求解均值和方差，提出基于BSG博弈的拟态Web服务器调度算法。提出衡量执行体C_i、C_j间差异性：

$$w_{ij} = C_{ij} * P^T$$

其中C_{ij}表示执行体C_i到C_j软件栈各层的差异性，P^T代表执行体软件栈各层差异性的加权系数。通过求解均值方差再结合BSG博弈收益函数，量化新的异构执行体间异构度，通过BSG策略增强了服务器的动态性和随机性。但该算法复杂度较高需进一步优化，同时对攻击者类型和攻击方式统计缺乏一定的理论判别。

文献[30]提出多系统异构性可通过复杂性、差异性来衡量即：

$$HETEROGENEITY = COMPLEXITY \times DISPARITY$$

基于多系统异构性研究，张杰鑫[31]采用上述异构性量化方法，并结合二次熵对执行体集差异性进行量化，最终通过M类构件集的异构性来计算执行体集异构性：

$$H_A = \sum_{K=1}^{M}\left(1 - \sum_{i=1}^{s} P_{ki}^2\right) \times \left(\sum_{i=1}^{s}\sum_{j=1}^{s} d_{kij}^2 P_{ki} P_{kj}\right)$$

d_{kij}表示构件集k中i、j构件之间的差异性，P_{ki}为构件的丰富度。结合Web服务质量提出了基于最大异构性和Web服务质量的随机种子调度算法RSMHQ. Web服务质量WS与综合评价指标ES较随机调度算法分别提升15.32%，37.45%，有效实现了安全性和服务质量的平衡。在具体实际应用环境中，可以根据优化算法进一步确定安全性和服务质量权重以到达最佳平衡效果。

不同于上述执行体异构度量化方法，文献[32]采用共有漏洞指标和共享代码数量分析度量执行体间的相似程度。普黎明等[33]采用文献[32]在时间和空间两个维度来衡量面向拟态云服务的异构执行体相似度，提出基于优先级和时间片的执行体调度算法（PSPT）。该算法通过定义时间权重因子α和空间权重因子β，基于执行体共有漏洞指标定义执行体相似度：

$$CVI_{ly}\left(C_{lp},C_{lq}\right) = \sum_{i\text{-}y\text{-}years+1}^{y} \alpha_i v_i\left(C_{lp},C_{lq}\right)$$

$$CVI_{y}\left(E_j,E_k\right) = \sum_{l\text{-}1}^{tiers} \beta_l CVI_{ly}\left(C_{lp},C_{lq}\right)$$

其中CVI_{ly}表示基于共有漏洞同一种类构件的相似程度，CVI_{y}表示执行体间相似度。PSPT算法实现了很好的动态性是RSMS算法的1617倍，且时间复杂度为$o(1)$，实现了极高的系统动态性和线性算法复杂度。 同样Wu[34]基于执行体共有漏洞采用Jeccard距离描述任意两执行体E_i、E_j间异构属性：

$$He_{ij} = He\left(E_i,E_j\right) = 1 - \frac{\left|V\left(E_i\right) \cap V\left(E_j\right)\right|}{\left|V\left(E_i\right) \cup \Gamma\left(E_j\right)\right|}$$

其中$V\left(E_i\right)$为执行体E_i的漏洞集合。结合综合调度指标CS，提出一种面向拟态防御系统的基于执行体异构度、性能和历史置信度的随机种子调度算法（RSHPC），相较于随机调度算法该算法异构度提升了88.68%，系统性能提升了20.80%，最后综合指标CS提升了42.59%。

3.1.3 基于MOSS度量

出于与上述异构度衡量的不同，QIU[35]提出软件相似度度量（MOSS）方法，该方法利用类图的结构相似性和属性相似性相结合，将两种相似性结合到迭代更新过程中计算软件相似性得分。综合考虑执行体负载，顾泽宇等[36]基于MOSS算法衡量系统间异构程度，并采用凹形指数函数描述攻击成功概率：$p\left(v\left(C_{ei},t\right)\right) = \int \lambda e^{-\lambda t} dt$. 基于最大执行体集安全系数和最小调度体集负载方差：

$$\begin{cases} \max \quad \xi\left(C_e S,t\right) = \dfrac{1}{m}\sum\limits_{i \epsilon C_e S} \xi(t)_i \\ \min\ l\left(C_s S,t\right) = \dfrac{1}{h}\sum\limits_{j \epsilon C_s S} \left(l(t)_j - \overline{l(t)}\right)^2 \end{cases}$$

其中$\xi(t)_i$代表集合$C_e S_i$中任意元素C_{ei}在时间段τ内的安全系数，$l(t)$表示调度体负载，最后提出负载感知安全调度算法LA-SSA确定执行体调度策略，该算法较于安全优先调度算法SPSA实现了很好的系统安全增益，同时负载均衡性优于SPSA，在一定范围内有效解决了系统安全性与计算性能平衡问题。

同样高明等[37]提出一种基于拟态防御的差异化反馈调度判决算法，利用MOSS算法得到执行体E_i、E_j间异构度为$\sigma \epsilon [0,1]$，同时定义执行体集异构度为：

$$\sigma^* = \frac{1}{2m} \times \sum_{i}^{m}\sum_{i}^{j} \sigma\left(E_i,E_j\right)$$

其中$\sigma\left(E_i,E_j\right)$表示$E_i$、$E_j$的异构度。并根据历史判决器反馈结果量化执行体安全防御系数，以形式化的数学推导最优化问题得出最后调度结果，更精确推导执行体集相似度，3执行体时系统输出异常率平均值为0.1058，5个时仅为0.0673，实现了很好的安全性。该算法最优化算法可进一步研究，优化迭代确定异构度和执行体安全系数的广义平衡。

3.2 调度时机

调度时机是实现系统动态性、对外呈现测不准效应的重要因素，调度时机问题是如何选择一个最佳的在线执行体集变换时间点，大多数采用固定时间间隔、固定异常触发次数等来进行执行体集变换。基于对调度时间问题的研究，卢振平等[38]刻画了一种闭环控制器调度安全流程，将动态调度的时间问题建模为随机理论中的更新过程，提出了一种最优调度算法确定调度时间OSA，即根据输入的调度代价、攻击损耗以及攻击分布函数计算最佳调度时间。通过定义单位代价：

$$\gamma(T_i^d)=\frac{E(R_i)}{E(X_i)}=\frac{C_s^d+A_{loss}^d F(T_i^d)}{\int_0^{T_i^d} xF(x)dx+T_i^d(1-F(T_i^d))}$$

其中$E(R_i)$、$E(X_i)$表示第i次调度的总代价及所用时间期望值，T_i^d为计算防御者第i次调度的时间间隔，调度目标是寻求第i次调度最佳时间T_i^d以最小化单位代价$\gamma(T_i^d)$。OSA算法综合权衡执行开销以及攻击损耗两个重要决定因素，较于固定周期与随机调度算法，实现了较小的调度代价且调度算法平均运行时间仅为1.91s。在一定程度上解决了调度过程中时机选择的问题，但缺乏对执行体本身安全性、异构度等方面的考虑。

通过对异常门限S和调度周期T的综合考虑，Guo[39]引入滑动窗口机制提出基于滑动窗口模型的调度序列控制方法。可变门限S和周期T，基于内部资源和外部攻击将时间和门限耦合联动控制调度过程。采用Δt_i与s_i窗口共同驱动：

$$\begin{cases}e_1=(t-t_i\geqslant\Delta t_i)\wedge(c_t>0)\\ e_2=(\varphi_i\geqslant s_i)\wedge(c_t>0)\end{cases}$$

该算法采用异步并行方式使能调度模块和窗口滑动模块，时间复杂度和空间复杂度低。同时从时间层面考虑调度，该算法能够在不同场景情况下调度参数调整提升系统安全性、高效性和鲁棒性，具有很好的自适应能力。但该算法仅考虑一次替换单一执行体上线及随机选择，有待进一步研究。

3.3 调度数量

拟态容错主要是基于n模冗余机制，借以裁决机制实现对执行过程中部分执行体被攻破的容错方法，大多数调度算法从动态、异构方面出发，通过衡量异构度和动态性增加系统鲁棒性和安全性，缺乏对于冗余度的研究考虑。魏帅[40]基于安全增益、成本代价以及系统安全性等方面综合得出三模冗余容错在安全性和成本代价方面获得较为综合的最佳效果。但冗余度和动态性综合缺乏更加深入的研究。

基于对冗余度和动态性的考虑，李军飞[41]提出基于效用的动态弹性调度策略，根据当前网络环境进而确定下一步在线执行体的数量。下一次执行体调度数量u和下一次调度间隔v由当前网络中异常执行体数量n和故障时间t采用概率密度函数[42]和轮盘法[43]来确定：

$$P(u=a)=\int_{\frac{a-1}{N}}^{\frac{a}{N}} f(n,x)dx\ ,a\epsilon[1,N]$$

$f(n,x)$是与n相关的概率密度函数。最后确定各个u值的轮盘区间：

$$\begin{cases}S(u=a)=(0,p(1)], & a=1\\ S(u=a)=(\sum_{N-1}^{a-1}P(u),\sum_{N-1}^{a}P(u)], & a\epsilon[2,N]\end{cases}$$

利用随机数生成$r\epsilon[0,1]$，确定下一次调度执行体数量。

同样基于判决反馈结果综合考虑动态性及冗余性，高明等[37]提出基于判决反馈的调度数量算法权衡系统代价与安全性。以执行体输出的可靠度占比为判决依据，根据U_1前后时刻变化更新调度个数：

$$m(t)=[(1+\alpha(U_1(t-2)-U_1(t-1)))\times m(t-1)]$$

其中U_1表示执行体输出各类结果最大占比，$m(t-1)$表示上一时刻的调度个数。并综合考虑了系统负载，增加了系统动态性、可靠度同时显著降低了系统失效率和系统代价。但上述两种算法仅考虑了改变执行体数量的方法，但缺乏对具体变换时间和变化条件的研究以及执行体数量变换后续裁决算法的更新，值得进一步研究。

4 算法评估

动态性、异构度及冗余度能显著提高系统安全性，现有调度算法主要从执行体调度时机、执行体选取策略以及执行体调度数量等方面来实现系统动态性变化，对外呈现不确定性对内实现概率可控。在考虑系统安全性的同时，执行效率、服务质量、执行负载等调度目标也被相关算法考虑，以期望在不牺牲系统原有功能和效益的同时，通过拟态架构使系统获得显著的安全增益。总体来说，现有调度算法一方面实现了系统安全性的显著提升，另一方面基于调度目标的考量实现了很好的适应性能准确扩展到其它应用中。基于调度算法普遍考虑的调度目标，本节从动态性、可信任度（平均失效率）、异构度、系统开销以及服务质量等五方面来综合比较现有算法。

其中综合考虑其它调度目标，现有拟态调度算法都对系统动态性、执行体间异构度以及算法复杂度有一定的研究并实现了较好的效果。基于

原文实验数据，本节定义动态性衡量标准为平均调度周期大于80为高，中高为50-80之间，中为20-50之间，中低为5-20之间，小于5为低。同时异构度度量基于原文数据，本文对于实验得出的不同取值进行了归一处理如表1所示（MOSS表示用MOSS方法衡量执行体间异构度，画斜线表示该方法未考虑该指标）：

表1　算法综合性对比

	动态性	平均失效率	异构度	系统开销	服务质量
MD[25]	★	1.1419×10^{-4}	0.1143	$o(1)$	/
OMD[25]	★	3.5989×10^{-4}	0.2183	$o(1)$	0.4503
随机调度[23]	★★★★★	7.6647×10^{-4}	0.2723	$o(1)$	/
RSMS[23]	★★	2.7664×10^{-4}	0.155	$o(n)$	/
PSPT[33]	★★★★★	/	0.249	$o(1)$	/
RSMHQ[31]	★★	/	0.3768	$o(n)$	0.5193
RSHPC[34]	★★	/	/	$o(1)$	/
基于正态分布[27]	★★★★	/	/	$o(2^n)$	/
基于BSG[29]	★★	/	MOSS	$o(n)$	/
基于反馈判决[37]	★★★	/	MOSS	$o(n)$	/
基于自学习[28]	★★★	/	/	$o(2^n)$	/
基于安全策略[36]	★★	/	/	$o(n)$	/
滑动窗口[39]	★★★★	/	/	$o(n)$	/

（注：★-低，★★-中低，★★★-中，★★★★-中高，★★★★★-高）

从表1可以看出前文所提的调度目标在现有调度算法中都有被考虑，但各算法在不同应用场景中考虑方面有所不同，主要集中于研究系统动态性、执行开销和执行体异构度量化等，以期望在实现系统动态性的同时降低系统开销，并结合其他评价指标衡量该算法的安全增益和系统原有功能。总的来说现有调度算法大多都实现了很好的系统动态性及安全性，很好体现了拟态防御的动态、冗余以及负反馈特性，并通过形式化的数学公式量化出执行体异构度，对于拟态架构实现了很好的安全增益。但大多数算法只考虑了本文所提五个调度指标中的一两个，而缺乏对于整体调度指标的考虑以及各调度指标的综合考虑与评估，即本文所总结调度指标的优先级以及权重。上述算法及评价指标还有待进一步考虑验证，本文所综合提出的五个调度目标并不能完全衡量一个调度算法的好坏，并且各调度指标权重值得进一步研究。同时随着执行体异构度量化遇到瓶颈，执行体调度时机和调度数量研究逐渐受到重视，如何实现调度时机和执行体数量变化的有机结合以及后续裁决算法有很大研究空间。算法复杂度和执行体异构度在特定场景中有待进一步研究优化，并结合具体服务在实现系统高质量安全增益的同时不损害甚至提高实际应用的具体效能。

5　问题与展望

本文对现有拟态调度算法进行了详尽的介绍和分析，可以看出现有研究主要集中于考虑执行体异构度、负载率、调度时机以及调度数量等方面，在一定程度上使得拟态系统容错率、安全性得到了提高，但目前仍存在一些问题和不足值得进一步研究讨论：

（1）现有调度算法衡量执行体异构度大都基于执行体间共有漏洞的数量，但计算时仅仅只能依据已发现的漏洞，缺乏对执行体间未知的漏洞的考虑，同时在不同环境中各漏洞表现不一。衡量执行体异构度可以从自身结构、功能等方面结合基于度量[44]、字符串[45]、树[46]、类图[35]等方法进一步研究。

（2）现有调度算法只单方面研究调度时机或者调度数量的变化，根据不同场景变化综合考虑系统动态、异构、冗余等方面并结合最优化算法，寻求调度时机和调度数量的有机结合以实现最大的系统安全增益，有望成为下一个研究热点。

（3）现有调度算法在量化执行体集异构度时基本通过采用执行体间两两异构度累加的方式来计算，忽略了高阶相似性漏洞[6]的存在，如何量化执行体集高阶异构度以及与两阶异构度的分配权重有待进一步研究。

（4）现有调度算法主要是依据实验结果从而衡量某一算法对系统的安全增益，缺少形式化的数学量化推导及求解。可进一步考虑结合博弈论收益模型量化该算法的安全收益并结合最优化算法求解，利用数学推导直观严谨的求解出执行体调度策略。

6 总结

随着网络入侵窃密事件频发，拟态防御凭借其主动防御思想在网络安全防御方面逐渐显示出强大的能力，拟态调度算法也显示出越来越重要的作用。同时随着拟态技术的不断发展应用，在不同应用环境中新的拟态调度问题也不断涌现，值得进一步创新和改进。本文在现有文献的基础上，以各调度算法研究点为主线并考虑算法时间先后顺序，详细研究了拟态调度的工作机理及其安全机制，分析了现有各调度算法的创新点以及存在的不足，并在最后对拟态调度的趋势和研究方向进行了展望。

参考文献:

[1] Oppliger R. Internet security: firewalls and beyond [J]. Communications of The ACM，1997，40(5)：92-102.

[2] Intrusion detection systems [M]. Springer Science & Business Media，2008.

[3] Panda B K, Pradhan M, Pradhan S K. Intrusion Prevention System [M]//Network Security Attacks and Countermeasures. IGI Global，2016：245-258.

[4] Perdisci R , Dagon D , Lee W , et al. Misleading worm signature generators using deliberate noise injection[C]// IEEE Symposium on Security & Privacy. IEEE，2006.

[5] ReisCharles, BarthAdam, PizanoCarlos. Browser Security: Lessons from Google Chrome[M]. ACM，2009.

[6] 魏帅,张辉华,苏野,薛鹏飞,闻亮. 基于高阶异构度的大数裁决算法及性能分析[J/OL]. 计算机工程：1-7.
Wei S, Zhang H H, Su Y, Xue P F, Wen L. Majority voting algorithm and performance analysis based on high level heterogeneity [J/OL]. Computer Engineering：1-7.

[7] 沈昌祥,张大伟,刘吉强,叶珩,邱硕. 可信3. 0战略:可信计算的革命性演变[J]. 中国工程科学，2016，18(06)：53-57.
Sheng C X, Zhang D W, Liu J Q, Ye X, Qiu S. The strategy of TC 3. 0: A revolutionary evolution in trusted computing [J]. Engineering Sciences，2016，18(06)：53-57.

[8] Zheng J, Namin A S. A Survey on the Moving Target Defense Strategies: An Architectural Perspective [J]. Journal of Computer Science and Technology，2019，34(1)：207-233.

[9] Jafarian J H, Alshaer E, Duan Q, et al. An Effective Address Mutation Approach for Disrupting Reconnaissance Attacks [J]. IEEE Transactions on Information Forensics and Security，2015，10(12)：2562-2577.

[10] Jafarian J H, Alshaer E, Duan Q, et al. Adversary-aware IP address randomization for proactive agility against sophisticated attackers [C]. international conference on computer communications，2015：738-746

[11] Luo Y, Wang B, Cai G, et al. Effectiveness of Port Hopping as a Moving Target Defense [C]. international workshop on security，2014：7-10.

[12] Azab M, Eltoweissy M. ChameleonSoft: Software Behavior Encryption for Moving Target Defense [J]. Mobile Networks and Applications，2013，18(2)：271-292.

[13] Sakic E, Deric N, Kellerer W, et al. MORPH: An Adaptive Framework for Efficient and Byzantine Fault-Tolerant SDN Control Plane [J]. IEEE Journal on Selected Areas in Communications，2018，36(10)：2158-2174.

[14] 邬江兴. 网络空间拟态防御研究[J]. 信息安全学报，2016，1(04)：1-10.
Wu J X. Research on cyber mimic defense [J]. Journal of Cyber Security，2016，1(04)：1-10.

[15] 马海龙,江逸茗,白冰,张建辉. 路由器拟态防御能力测试与分析[J]. 信息安全学报，2017，2(01)：43-53.
Ma H L, Jiang Y M, Bai B, Zhang J H. Tests and analyses for mimic defense ability of routers [J]. Journal of Cyber Security，2017，2(01)：43-53.

[16] 宋克,刘勤让,魏帅,张文建,谭力波. 基于拟态防御的以太网交换机内生安全体系结构[J]. 通信学报，2020，41(05)：18-26.
Song K, Liu Q R, Wei S, Zhang W J, Tan L B. Endogenous security architecture of Ethernet switch based on mimic defense [J]. Journal of Communications，2020，41(05)：18-26.

[17] Hu H, Wang Z, Cheng G, et al. MNOS: a mimic network operating system for software defined networks [J]. Iet Information Security，2017，11(6)：345-355.

[18] Parhami B. Voting algorithms[J]. IEEE Transactions on Reliability，1994，43(4)：617-629.

[19] Jamali N, Sammut C. Majority Voting: Material Classification by Tactile Sensing Using Surface Texture [J]. IEEE Transactions on Robotics，2011，27(3)：508-521.

[20] Mcallister D F, Sun C E, Vouk M A, et al. Reliability of voting in fault-tolerant software systems for small output-spaces [J]. IEEE Transactions on Reliability，1990，39(5)：524-534.

[21] Reis G A, Chang J, Vachharajani N, et al. SWIFT: Software Implemented Fault Tolerance [C]. symposium on code generation and optimization，2005：243-254.

[22] 彭浩, 陆阳, 孙峰,等. 副版本不可抢占的全局容错调度算法[J]. 软件学报，2016，27(12)：3158-317.
Peng H, Lu Y, Sun F, et al. Fault tolerant global scheduling with non-preemptive backups [J]. Journal of Software，2016，27 (12)：3158-317.

[23] 刘勤让, 林森杰, 顾泽宇. 面向拟态安全防御的异构功能等价体调度算法[J]. 通信学报，2018，39(07)：188-198.
Liu Q R, Lin S J, Gu Z Y. Heterogeneous redundancies scheduling algorithm for mimic security defense[J]. Journal of Communications，2018，39(07)：188-198.

[24] 韩进,臧斌宇. 软件相异性对于系统安全的有效性分析[J]. 计算机应用与软件，2010，27(09)：273-275+300.
Han J, Zang B Y. Analyzing the effectiveness of software diversity for system security [J].

Computer Applications and Software，，2010，27(09)：273-275+300.

[25] 姚文斌, 杨孝宗. 相异性软件组件选择算法设计[J]. 哈尔滨工业大学学报，2003，35(3)：261-264.
YAO W B, YANG X Z. Design of selective algorithm for diverse software components [J]. Journal of Harbin Institute of Technology，2003，35(3)：261-264.

[26] 吕迎迎,郭云飞,王禛鹏,程国振,王亚文. SDN中基于历史信息的负反馈调度算法[J]. 网络与信息安全学报，2018，4(06)：45-51.
Lv Y Y, Guo Y F, Wang Z P, Cheng G Z, Wang Y W. Negative feedback scheduling algorithm based on historical information in SDN [J]. Chinese Journal of Network and Information Security，2018，4(06)：45-51.

[27] 张震骁. 拟态防御动态调度策略研究[D]. 郑州大学，2018.
Zhang X Z. Research on dynamic scheduling strategy for mimic defense[D]. Zhenzhou University，2018.

[28] Li J, Wu J, Hu Y, et al. DSL: Dynamic and Self-Learning Schedule Method of Multiple Controllers in SDN[J]. Etri Journal，2017，39(3)：364-372.

[29] 王晓梅,杨文晗,张维,杨镇. 基于BSG的拟态Web服务器调度策略研究[J]. 通信学报，2018，39(S2)：112-120.
Wang X M, Yang W H, Zhang W, Yang Z. Research on scheduling strategy of mimic Web server based on BSG[J]. Journal of Communications，2018，39(S2)：112-120.

[30] Twu P, Mostofi Y, Egerstedt M, et al. A measure of heterogeneity in multi-agent systems [C]. advances in computing and communications，2014：3972-3977

[31] 张杰鑫,庞建民,张铮,郤铭,张浩,聂广来. 面向拟态构造Web服务器的执行体调度算法[J]. 计算机工程，2019，45(08)：14-21.
Zhang J X, Pang J M, Zhang Z, Tai M, Zhang H, Nie G L. Executors Scheduling Algorithm for Web Server with Mimic Structure[J]. Computer Engineering，2019，45(08)：14-21.

[32] Garcia M, Bessani A, Gashi I, et al. Analysis of operating system diversity for intrusion tolerance [J]. Software - Practice and Experience，2014，44(6)：735-770.

[33] 普黎明,刘树新,丁瑞浩,王凯. 面向拟态云服务的异构执行体调度算法[J]. 通信学报，2020，41(03)：17-24.
Pu L M, Liu S X, Ding R K, Wang K. Heterogeneous executor scheduling algorithm for mimic cloud service[J]. Journal of Communications，2020，41(03)：17-24.

[34] Wu Z, Wei J. Heterogeneous Executors Scheduling Algorithm for Mimic Defense Systems[C]. international conference on computer and communication engineering，2019：279-284.

[35] Qiu D, Li H, Sun J L, et al. Measuring software similarity based on structure and property of class diagram [C]. ieee international conference on advanced computational intelligence，2013：75-80.

[36] 顾泽宇,张兴明,林森杰. 基于安全策略的负载感知动态调度机制[J]. 计算机应用，2017，37(11)：3304-3310.
Gu Z M, Zhang X M, Lin S J. Load-aware dynamic scheduling mechanism based on security strategies[J]. Journal of Computer Applications，2017，37(11)：3304-3310.

[37] 高明,罗锦,周慧颖,焦海,应丽莉. 一种基于拟态防御的差异化反馈调度判决算法[J]. 电信科学，2020，36(05)：73-82.
Gao M, Luo J, Zhou H Y, Jiao H, Ying L L. A differential feedback scheduling decision algorithm based on mimic defense[J]. Telecommunication Science，2020，36(05)：73-82.

[38] 卢振平,陈福才,程国振. 软件定义网络中控制器调度时间机制设计与实现[J]. 网络与信息安全学报，2018，4(01)：36-44.
Lu Z P, Chen F C,Chen G Z. Design and implementation of the controller scheduling-time in SDN[J]. Chinese Journal of Network and Information Security，2018，4(01)：36-44.

[39] Guo W, Wu Z, Zhang F, et al. Scheduling Sequence Control Method Based on Sliding Window in Cyberspace Mimic Defense[J]. IEEE Access，2019，8：1517-1533.

[40] 魏帅,于洪,顾泽宇,张兴明. 面向工控领域的拟态安全处理机架构[J]. 信息安全学报，2017，2(01)：54-73.
Wei S, Yu H, Gu Z Y, Zhang X M. Architecture of mimic security processor for industry control system[J]. Journal of Cyber Security ，2017，2(01)：54-73.

[41] 李军飞. 软件定义网络中拟态防御的关键技术研究[D]. 战略支援部队信息工程大学，2019.
Li J F. Research on key technologies of mimic defense in software-defined network[D]. Information Engineering University，2019.

[42] Parzen E. ON ESTIMATION OF A PROBABILITY DENSITY FUNCTION AND MODE [J]. Annals of Mathematical Statistics，1962，33(3)：1065-1076.

[43] Lipowski A, Lipowska D. Roulette-wheel selection via stochastic acceptance[J]. Physica A-statistical Mechanics and Its Applications，2012，391(6)：2193-2196.

[44] Tamada H, Nakamura M, Monden A, et al. Java Birthmarks — Detecting the Software Theft— [J]. IEICE Transactions on Information and Systems，2005，88(9)：2148-2158.

[45] Park H, Choi S, Lim H, et al. Detecting code theft via a static instruction trace birthmark for Java methods [C]. international conference on industrial informatics，2008：551-556.

[46] Baxter I D, Yahin A, De Moura L, et al. Clone detection using abstract syntax trees [C]. international conference on software maintenance，1998：368-377.

[作者简介]

朱正彬（1996-），男，信息工程大学硕士研究生，主要研究方向为网络空间安全，拟态防御。

刘勤让（1975-）男，博士，信息工程大学信息技术研究所研究员，主要研究方向为网络空间安全、宽带信息网络及芯片设计。

刘冬培（1985-），男，博士，信息工程大学信息技术研究所助理研究员，主要研究方向为SoC芯片测试与验证。

王崇（1995-），男，信息工程大学博士研究生，主要研究方向为拟态防御，缓存侧信道防御。

基于诱捕的软件异常检测研究

傅建明，刘畅，解梦飞，罗陈可
武汉大学国家网络安全学院，武汉大学空天信息安全与可信计算教育部重点实验室，武汉 430072

摘　要：高级持续威胁（APT）利用复用代码攻击，可以绕过堆栈的不可执行限制，这依然是网络安全的重要威胁。传统的控制流完整性和地址随机化技术虽然有效抑制了APT的步伐，但软件的复杂性和攻击演化使得软件仍存在被攻击的时间窗口。为此，以资源为诱饵的诱捕防御是确保网络安全的必要补充。论文从诱饵类型、诱饵生成、诱饵部署、诱饵度量方面刻画了诱捕防御的机理。同时，剖析了诱捕防御在勒索软件检测、漏洞检测、Web安全方面的应用，并提出了用于检测勒索软件的诱饵动态更新方法。最后，讨论了诱捕防御面临的挑战，希望诱捕防御可以为发现未知攻击、溯源攻击意图提供理论和技术支持。

关键词：高级持续威胁、代码复用攻击、控制流完整性、地址随机化、诱捕防御

A Survey of Software anomaly Detection Based on Deception

Fu Jianming, Liu Chang, Xie Mengfei, Luo Chenke
Key Laboratory of Aerospace Information Security and Trusted Computing,School of Cyber Science Engineering, Wuhan University, Wuhan 430072, China

Abstract: Advanced Persistent Threats (APT) exploits code reuse to bypass the non-executable stack&heap protection, which is still an essential threat to network security. Traditional control flow integrity and address space randomization technologies have effectively prevented the pace of APT. However, the complexity of the software and the evolution of attacks make the software still being vulnerable. For this reason, deception defense with resources as bait is an indispensable supplement for network security. This paper describes the mechanism of deception defense from bait type, bait generation, bait deployment, and bait measurement. Simultaneously, deception defense applications in ransomware detection, vulnerability detection, and Web security have been analyzed. Finally, the deception defense challenges have been discussed, and we hope that deception defense can provide theoretical and technical support for discovering unknown attacks and attack attribution.

Key words: advanced persistent threat; code reuse attack; control flow integrity; address randomization; deception defense

1　引言

互联网和物联网的发展，以及智能设备的广泛应用，提高了人们的生活质量和工作效率，使得人类的工作和生活逐渐数字化、自动化和智能化。随着时间的推移，网络世界的信息资产和财富不断累积。这些资产和财富吸引了大量的攻击者或者攻击集团，网络安全事件不断涌现。360公司于2018年发布了478篇高级持续威胁（Advanced Persistent Threat，APT）分析报告［1］，APT攻击一般会使用漏洞（如Windows漏洞CVE-2018-8611、Flash漏洞CVE-2018-15982）实现攻击代码的自动加载和攻击行为的隐藏。而且，网络攻击从过去的隐蔽潜行逐渐走向公开对抗［2］。例如，臭名昭著的Wannacry勒索病毒使用了永恒之蓝漏洞（CVE-2017-0143）实现病毒自动化的网络传播。防病毒工具［3］、入侵检测系统、防火墙都是软件外部的安全机制，它们利用签名或者启发式知识检测已知威胁，如木马、计算机病毒、不合规的流量等。这些预定义的策略和知识独立

基金项目：国家自然科学基金资助项目（61972297, U1636107）

于软件的内部逻辑，难以检测零日漏洞攻击和新的恶意软件。

蜜罐（HoneyPot）和蜜网（HoneyNet）是网络安全的一种主动防御机制［4~6］。该防御机制通过部署虚假的目标（Honey-token）和模糊化真实目标，干扰攻击者对目标和防御者的认知，引诱攻击者使用其部署的诱饵服务和诱饵网络，动态捕获其攻击行为和攻击工具，解析其攻击方法，推测攻击者的攻击意图和攻击动机。网络诱捕［4］利用主机、服务和信息作为诱饵，检测网络入侵、恶意代码、僵尸网络和蠕虫传播、垃圾邮件传播，并提取网络攻击特征，为入侵检测系统和防火墙提供威胁情报。目前，企业界有许多公司研制基于诱骗的主动防御产品，如Rapid7（美国）、LogRhythm（美国）、Attivo Networks（以色列）、Cymmetria（以色列）、Smokescreen Technologies（印度）、北京元支点（中国）、长亭科技（中国）、默安科技（中国）、锦行网络科技（中国）等等。他们利用软件定义网络或虚拟机实现终端、网络、应用、工控等蜜罐产品，这些产品可以从系统或网络的外部异常行为上感知攻击过程，但缺少与主机软件和主机数据关联的诱捕机制。

蜜罐和蜜网作为一个独立于业务系统的诱捕环境，其目的是诱使攻击者攻击暴露的主机和服务。把蜜罐思想引入软件安全，在主机环境部署诱饵文件。一旦软件访问这些诱饵文件，就触发软件异常报警。此时，诱捕防御是一种基本的安全能力，与已有的防病毒、防火墙、入侵检测系统互为补充。诱捕防御可以模糊化防守方的代码资源和数据资源，增强防守方的情报能力和防御筹码，感知未知的安全威胁。

本文第二节介绍研究背景和相关的工作。第三节详细阐述诱捕机理，第四节介绍诱捕机制的应用场景。最后，讨论诱捕防御的挑战。

2　背景与相关工作

APT攻击的简化流程包括目标探测、攻击代码传输、攻击代码触发、攻击代码执行和敏感资源获取［7］，如图1所示。这里省略了逃逸、提权、边界移动等过程，逃逸是为了对抗主机安全检测和网络安全检测，提权是为了获得更多的敏感资源，边界移动的目的是探测和获取外网不可见的资源，深度挖掘目标的高价值资源。

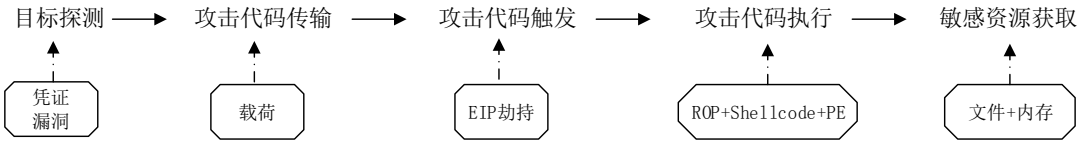


图1　APT攻击的基本流程

目标探测可分为远程探测（remote probing）和本地探测（local probing）两种类型。远程探测可以通过手动或自动化工具进行，如直接猜测目标的登录凭证（user name &password）、扫描系统的版本或脆弱性［7］，扫描或探测数据包中的特有签名、探测频率或行为模式都可以作为检测目标探测攻击的依据［8］。一旦攻击者熟悉这些依据，就可以绕过这类检测。针对远程探测，可以部署一些虚假的登录凭证、人为植入的漏洞。当这些凭证或漏洞被利用时，则可以提前感知攻击行为［9］［10］。

攻击者借助远程探测的结果，如弱口令或系统漏洞，可以初步远程访问目标系统，但攻击者在目标系统中执行的代码是目标系统定制的，这种定制受限于目标软件的内部逻辑，无法满足攻击者获取敏感资源的需求。为此，攻击者需要远程注入自己精心构造的代码到目标系统中，然后执行注入代码，以达到攻击目的。

远程注入的代码分为ROP（Returned-Oriented Programming）、Shellcode、可执行代码模块（如PE、ELF）三类。ROP复用软件中的已有代码，构建攻击者的攻击载荷［11］［12］，以绕过栈和堆中代码不可执行的限制（Data Execution Prevention，DEP）。ROP可以让Shelllcode获得可执行权限。Shellcode是由可执行的代码片段构成，其作用是加载可执行的代码模块［13］。可执行代码模块是攻击者完成攻击任务的核心，用来探测软件环境、加载任务模块，并最终从磁盘和内存中获

取目标的文档资源、网络资源或者各种登录凭证等。目前，大多数ROP防御聚焦于控制流完整性（Control Flow Integrity，CFI）保护［14］和地址空间布局随机化防御（Address Space Layout Randomization， ASLR ）［15］，前者阻止软件间接代码指针的非预期跳转，后者破坏ROP代码地址不变的假设。针对ROP的构造，可以在软件中部署一些诱饵代码（booby trap）［16］。一旦这些诱饵代码被利用，则可以检测未知的攻击。

众所周知的计算机病毒、勒索软件、木马等都属于模块级的攻击代码［17］，这些攻击代码探测磁盘中的敏感文件、内存中的敏感数据，这种探测属于本地探测。针对本地探测，可以部署一些诱饵文件，一旦这些诱饵文件被访问，则可以检测未知的攻击［18］。

在控制了内网机器后，攻击者往往会进行横向移动，例如读取其他系统用户的数据、扫描当前网段内的其他主机，以便控制包含高价值目标的主机。另外，攻击者还会尝试攻击域服务器，从而获取域管理员权限。对于域控攻击，同样可以通过布置诱饵进行检测。例如对于Kerberoast攻击，攻击者会查询高权限域用户下的服务主体名称（SPN），导出并尝试破解服务票据。因此防守方可以创建一个诱饵Kerberoast服务账户，并伪造SPN，检测对于该诱饵服务票据的请求行为。

为了隐藏攻击行为，攻击者会使用加密通信，匿名代理等方式来传输数据，从而加大溯源的难度。另外，在完成攻击任务后，攻击者还会清理目标主机和网络中的相关记录，例如删除系统日志和文件、修改注册表等。

APT攻击流程的每个阶段都可以部署诱饵资源，实施诱捕防御。多级防御或纵深防御同样可以应用于诱捕防御。文献［19］提出了由网络服务仿真、安全漏洞伪造、操作行为控制和文件系统镜像组成的深度欺骗策略， 以突破单层欺骗防御的局限性。文献［20］针对APT侦察、破袭主机、破袭服务器等多阶段攻击过程，设计用户层、本地层、全局层的诱饵资源：诱饵用户、诱饵文件和诱饵服务器，构建了一种纵深的诱捕防御体系。

文献［6］认为网络欺骗利用骗局干扰或误导攻击者的认知，以抑制攻击活动。同时，描述了网络欺骗的生命周期、工作流程，剖析了设备层、网络层、数据层、应用层上的欺骗技术。该文献从认知层面介绍诱骗防御，讨论了网络层面和系统层面诱骗防御存在的挑战。

文献［21］从社会工程学角度定义信息欺骗。信息欺骗是利用用户兴趣偏好、利益互惠、摆脱恐惧等方面的弱点，以误导用户操作，获取敏感信息、下载和执行恶意代码等。同时，详细分析了社交欺骗、网站欺骗、应用欺骗、邮件欺骗等欺骗场景。

本文聚焦软件或代码访问的资源，从诱饵类型、诱饵生成、诱饵部署、诱饵度量等方面刻画了诱捕防御的机理。同时，剖析了诱捕防御在勒索软件检测和漏洞检测等方面的应用。与文献［6］比，本文属于应用层和数据层交叉的欺骗，从软件安全角度介绍诱骗防御机制。文献［21］以攻击者的信息欺骗为主，分析应对的防御措施。与之对比，本文采用信息欺骗的思想以确保软件安全，其视角和内容都不同。

3 诱捕机制

文献［6］定义了网络欺骗的4个要素｛Defender， Asset， Trick， Attacker， Profit｝，其中Trick表示骗局，是4个要素的核心。骗局有两种策略：以真似假和以假拟真。

以真似假：改变已有资源（Asset）的特征，或者伪装为已知的诱饵系统。该掩饰策略让攻击者难以发现目标，最终因失去攻击目标自动放弃攻击。

以假拟真：构建虚假资源，吸引攻击者访问这些虚假资源，同时检测和观察攻击活动，挖掘攻击意图。该模拟策略让攻击者陷于骗局，消耗攻击者的资源，延缓或消化攻击。

本文探索以假拟真，其诱捕机制包括骗局中诱饵设计和攻击检测两部分。

定义1（诱饵）：诱饵（Bait）是一种防御资源，B=｛R， T， L｝。其中，R为诱饵资源集合，T为资源的类型，L为资源的位置。

诱饵跟域名一样，也是一种IOC（Indicator Of Compromise）［22］。传统IOC，如域名、邮件地址等，都是攻击代码自身携带的、固有的签名。而诱饵是防御者设计和部署的，是一种非对称的

防御资源，攻击者难以绕过。

定义2（轨迹）：在给定时间窗，进程访问资源的轨迹S=｛P， R， O， C｝。其中，P是用户进程集；R是用户使用或访问的资源集合；O是资源访问类型，包括读、写、执行等；C是资源访问的上下文，包括时间、触发条件等。

定义3（异常）：在给定的时间窗，进程p的资源访问轨迹与诱饵资源关联，则提示进程p异常。换言之，S.R∩B.R不为空，表示进程异常。

因此，只要资源访问的轨迹符合**定义3**，则检测到可疑攻击。

3.1 诱饵类型

诱饵［23］是诱捕防御的战略资源。在日常数字世界中，用户或软件使用的任何信息都可以作为诱饵，如网络服务的特定端口、数据库/网站/远程主机的登录凭证、Email地址、电话号码、文档等。诱捕防御的基本思路是感知与诱饵的交互行为，推断可能的未授权访问或者恶意攻击。

HoneyFile［24］引入了三类诱饵：**文件诱饵、用户名和口令诱饵、用户邮件名诱饵，**以此提高入侵检测的准确性。口令诱饵（password.txt）和邮件名诱饵用以弥补文件诱饵单一性。Decoy-document［25］采用了更多的Honey-token（诱饵的一种表示），如contract.pdf含有beacon（俗称web Bug）。Beacon是一种自动远程访问防御者服务器的方式，如文档中的远程链接。攻击者一旦打开该文档，Beacon会自动连接远程服务器，借此可以用来检测内部攻击和异常。此外，Bogus-program［26］作为Honey-token可以有效检测程序盗版。Honey-token也广泛应用于其他安全防御中，例如口令猜测攻击检测［9］、漏洞触发的攻击流量转发检测［10］、诈骗电话和广告电话检测［27］［28］、勒索软件检测［29］［30］等。

Honey-token同样可以被引入到软件安全中防御代码复用攻击［16］。通过向软件中注入一些指令序列作为Trap（陷阱），这些指令序列被定义为HoneyGadget［31］［32］，其功能是警示用户，终止软件执行等。一旦攻击者使用这些注入的指令，则会触发报警。Readactor++在函数表中注入指向HoneyGadget的虚假指针［31］，增加HoneyGadget触发的机会。

表1描述了诱饵类型、使用方式和用途。诱饵类型分为File、Data、Code三种类型。File表示诱饵文件（文档），是指部署在文件系统中的特定格式文件，如桌面上的通信录文件、或者文档文件夹中的技术报告、软件文件等。一旦发现有对这些文件的访问操作，安全工具就提示异常。而Data属于一种元数据，表示有语义的一串字节序列，指可以从公开网络环境获取、从远程网络访问的诱饵，包括但不限于网络端口、登录凭证、电话号码等，数据诱饵可以嵌入在文件、程序、数据库中。防御方有意或无意在社交平台、论坛留言板等地方泄露这些诱饵，诱使攻击者使用，从而观察攻击行为和攻击意图。Code表示可以直接执行的机器代码。攻击者一旦使用这些代码会触发报警。诱饵使用方式描述了攻击者使用诱饵的位置，分为本地（Local）和远程（Remote）。前者表示攻击代码已经在目标主机运行，准备获取目标中的敏感信息；后者表示攻击者远程使用这些诱饵，试图远程访问目标系统。诱饵用途描述诱饵在软件安全中的作用，如检测网络异常访问、文件非授权访问、软件盗版、勒索软件、漏洞攻击等。

3.2 诱饵的生成与部署

诱饵的生成直接关系到诱饵的有效性。目前大多文献对诱饵生成没有特别说明，仅仅说明诱饵的属性更新，诱饵内容需要包含一些敏感数据［23］，如登录凭证、社会保险号、邮件地址等。如果这些敏感数据在流量日志中被发现，则判定为从网络层检测到异常或攻击。文献［24］单独区分文件诱饵、登录凭证诱饵、邮箱诱饵。上述两篇文献均未介绍诱饵构造的方法。

文献［25］［26］引入了一种新的诱饵（beacon），可以远程检测主机系统受到了入侵。其基本思路是利用办公软件的宏或PDF的Javascript将一个URL植入到文档或软件中，该URL随文档或软件的打开自动远程访问防御者服务。

个人身份信息（Personally Identifiable Information，PII）包含个人隐私，如电话号码、住址等，一直是攻击者希望获取的信息。文献［34］分析了姓名、性别、职业、收入等属性的分布，以及收入与职业、地域（经纬度）的关联，解析了身份证号码和电话号码的构成，设计和实现了一个自动的个人身份信息生成方法。利用搜集的信息，

表1　诱饵类型和诱饵的用途

诱饵	诱饵类型	使用方式	用途	备注
Email地址、登录凭证[23]	Data	Remote	异常告警	
HoneyFile[24]	File+Data	Local+remote	文件和数据非授权访问	
Decoy-document [25]	File+Data	Local+remote	文件和数据非授权访问	beacon
honeytable[33]	Data	Remote	DB的非授权访问	
Bogus-program [26]	File+data	Local+remote	软件盗版	beacon
HoneyWord[9]	Data	Remote	口令猜测	口令
Honey-patches[10]	Data	Remote	漏洞攻击重定向	
Honey-pot[27] Mobipot[28]	Data	Remote	钓鱼/推销电话	电话号码
R-locker[43]	File	Local	勒索软件	管道
Codearmor[32] Trap[31]	Code	Local	漏洞检测	ROP gadget 虚假指针

该方法可以直接输出常见的个人身份信息，这些信息满足各个属性的分布关系以及属性之间的关联约束。因数据缺失和隐私保护等原因，该文献没有讨论信用卡、邮件地址、医疗记录、购买偏好、娱乐偏好的生成。

数据库在各个组织和单位广泛应用，如存放公司的客户数据、大学的学生成绩等。从网络诱饵元素，如网络端口、IP地址、域名等，很容易延伸到数据库的数据记录。HoneyGen［35］给出了一种伪造数据库元组的构造方法，利用伪造的数据记录感知异常和入侵。该方法希望产生高质量的伪造数据记录，让攻击者无法识别真伪。该方法分为约束抽取、元组生成和元组选择三个阶段。约束抽取收集数据表的字段属性，区分敏感字段和一般字段，构建外键、主键、字段之间的关联约束、连续字段的边界、离散字段的值集。元组生成利用混淆和泛化技术生成候选元组，同时，这些元组满足抽取的约束。最后，利用对数似然估计来计算候选元组与真实元组的相似度，选择相似度较高的生成元组作为诱饵数据。图2描述了Honeytoken DB的生成过程，其约束生成和混淆泛化直接影响诱饵的质量。

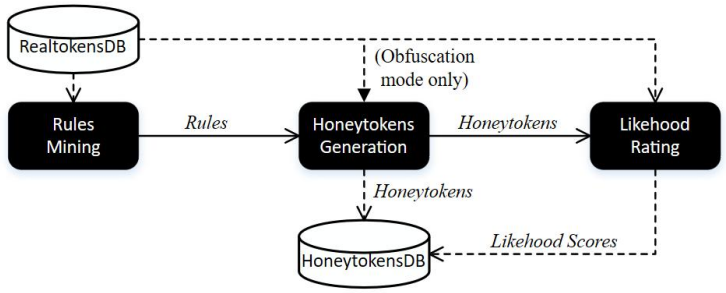


图2　数据表诱饵的生成框架

文献［26］研究利用程序变换（code obfuscation）从给定软件（proprietary source code ）生成bogus software， 其人工构造的beacon隐藏在文档说明（pdf）、API使用说明（html）、以及代码库中。其程序变换包括符号变换（类、方法、变量）、代码等价变换、垃圾代码注入、结构变化（增加新类、方法、变量、控制流混淆）等。当攻击者获取了这些bogus software，执行编译后的程序，或者打开含有beacon的文档时，beacon会自动向服务器发送告警，表明该beacon对应的程序（软件）被盗。如果被盗的文档或软件在断网或沙箱环境中打开，则此beacon失效。如果beacon有

较强的触发约束，则攻击者难以检测到beacon，从而提升诱饵的价值。

HoneyWord［9］提出了Honeywords方案来保护口令文件，通过在口令文件中增加额外N−1个假的身份凭据来增强安全性。该方案在已有的口令尾部追加给定长度随机值作为诱饵口令，也可以分析已有口令的模式（如W4D1W5，表示口令前4位为字母、中间1位为数字、后5位为字母），随机生成符合该模式的诱饵口令。如果口令文件被偷并被破解，攻击者将会面对N个不同的口令，其中，只有一个是正确的。如果攻击者使用任意一个假的口令，将会触发一个告警，系统管理员就会知道口令数据库已经被盗并被破解。

3.3 诱饵的度量

诱捕防御本身是一种概率防御机制。如果部署的诱饵没有任何访问行为，则该防御失效。该机制的有效性依赖于诱饵的构造和诱饵的部署。诱饵应该尽可能部署在攻击者必经的搜索路径上，同时诱饵对攻击者而言是难以辨识的。HoneyFile［24］讨论了诱饵文件MAC（Modified Time， Access Time， Created Time）时间属性和文件访问日志的更新，防止攻击者从文件的时间属性鉴别出诱饵文件。诱饵的数量和位置直接影响异常检测的准确性和及时性［36］。因此，攻击者和防御者在诱捕防御上不断对抗、演化，诱饵度量定性上反应了诱饵的质量。

当诱饵生成和部署后，攻击者、用户都可能会访问它，而安全工具会探测它。对攻击者而言，诱饵是真实的、动态的、有诱惑且容易发现的。对用户而言，诱饵是可辨识的、无干扰的。对检测工具而言，诱饵的访问是可检测的，可区分的。文献［24］给出了诱饵度量的维度。

真实性（Believability）：诱饵从形式和内容上无法与真实资源区分。与真实文件比，文件诱饵的文件名、MAC、访问日志是无差别的，且文件诱饵的内容也是无差别的。在实际检测中，只要诱饵被访问，就会发出告警。有时，诱饵内容的真实性在现实系统中会忽略。

动态性（Variability）：诱饵需要随着时间和用户活动而变化。诱饵的属性，如访问时间会更新；诱饵的数量随用户活动而更新。同时，不同系统上部署的诱饵不同。

诱惑性（Enticement）：诱饵的诱惑性依赖于攻击者的兴趣或意图，即诱饵必须蕴含攻击者的目标。攻击者关注高价值的文档，如包含口令、社会保险号、银行账户信息、在线支付信息等。Windows操作系统的资源管理器提供文档的关键词搜索，攻击者利用该搜索功能就可以定位到他认为的高价值文档。此外，攻击者可以通过浏览目录发现诱惑性高的文件名，如password.txt。因此，从诱饵的属性和内容两个方面体现诱惑性。

醒目性（Conspicuousness）：诱饵命中攻击者搜索的次数或概率。当攻击者在受害者主机中不断进行文档搜索，防御者希望其每次搜索结果中都含有诱饵。换言之，诱饵容易被攻击者发现并访问。醒目性依赖于诱饵部署的位置（目录）、诱饵的诱惑性、攻击者的搜索策略（文件的属性搜索、内容搜索）。

无干扰（Non-interference）：诱饵不会影响正常用户的活动。用户可有效识别出系统中的诱饵，从而避免访问诱饵。同时，即使用户无意访问诱饵，用户也能及时发现，并终止对诱饵的访问。换言之，尽量减少诱饵在系统中的副作用。

可检测性（Detectability）：检测性是指安全工具及时检测到诱饵访问的能力。一旦发现诱饵访问，能立刻进行捕获。诱饵的真实性、动态性、诱惑性、醒目性、无干扰是保证攻击检测的基础。这种检测性需要排除正常用户的误用，以及对抗攻击者的绕过技术。

文献［6］定义了网络欺骗的4个属性：**机密性（confidentiality）**、**可鉴别性（authentication）**、**可用性（availability）**、**可控性（controllability）**。其中**confidentiality**对应**Believability（真实性）**，**authentication**对应**Detectability（可检测性）**，**availability**对应**Non-interference（无干扰性）**。特殊情况下，攻击者可能会识别出诱饵和伪造的环境，并利用伪造环境提升攻击能力。为此，防御者必须确保诱饵和伪造环境的可控性。这里可控性超出了软件安全确保的范围，本文没有单独讨论。

攻击者可能会在获取诱饵资源后进行分析和识别，从而绕过诱捕防御。我们可以把鉴别诱饵真实性的能力分为低（Low）、中（Medium）、高（High）三类。

低级：攻击者没有鉴别诱饵，默认这些资源都是真实的。遇到诱饵资源时，会直接访问这些资源。

中级：攻击者获得这些文档，以及文档中的敏感数据后，进行简单的验证，如时间属性的验证［37］、折旧验证［38］，然后利用网络搜索引擎等途径验证敏感数据（凭证）的真实性；

高级：攻击者采用人工智能和大数据分析，鉴别资源的真实性。

在诱捕防御设计中，一般会考虑攻击者具备中级鉴别能力，以此提高系统的有效性。

4　应用场景

在软件安全领域，诱捕防御首先用于检测单位内部的威胁［23］。内部人员知晓单位网络结构和安全策略，有便捷访问内部资源的优势。当安全策略存在缺陷时，内部攻击者会出现非授权或不当的资源访问。防御者把高价值的文件、数据库记录、医疗记录、包含登录凭证和社会保险号等作为诱饵，部署在文件系统和数据库系统中。一旦检测到对诱饵的访问行为，则报警。此外，HoneyFile［24］作为诱饵部署在文件服务器，辅助入侵检测系统（IDS），降低入侵检测的误报和漏报。Decoy document［36］的误报源于合法用户利用资源管理器显式浏览或者应用软件隐式浏览诱饵，白名单机制可以降低误报。本节重点介绍诱捕防御在勒索软件检测、漏洞攻击检测等方面的应用。

4.1　勒索软件检测

根据搜索策略，勒索软件会遍历文件系统，搜索和加密符合搜索策略的文件，直到遍历结束。已有文献从勒索软件的文件访问行为和文件内容的熵两个角度检测勒索软件［39］［40］［41］［29］。它们的检测思路是，前者利用文件访问行为频繁、访问会跨越不同的路径，以及存在明显的文件内容加密操作和文件写操作等特点；后者则依据加密文件的熵值普遍偏高。这两种判据在压缩工具和正常加密工具环境下存在误报。而且这些实时检测工具依赖于采集的数据量，采集足够的数据量才能触发报警。当报警发生时，部分文件已经加密，存在检测响应的延迟。

文献［37］分析了一千多个勒索软件，解析了它们的文件加密、文件删除和锁屏勒索行为。同时，该文献讨论了利用文件访问活动检测勒索软件的措施，但攻击者可以模仿正常用户的文件访问绕过行为检测。例如，勒索软件可以选择访问和加密部分文件。为此，该文献首次提到了可以设置诱饵资源以捕获勒索软件。UNVEIL［42］首次设计了诱饵资源环境，监控诱饵的删除操作和写操作，实现了检测勒索软件的沙箱。其诱饵资源环境包括文档、密钥和证书、压缩文件、图片、音频、视频，并介绍了文件内容、文件名、文件时间、文件目录、文件个数的构建思路和原则。每一个待分析的样本对应一种诱饵资源环境，以此增强环境的多样性、减少诱饵环境被识别的机会。

文献［43］在用户主目录创建一个命名管道作为核心诱饵文件。为了提高诱饵在文件系统中的覆盖率，在每个目录创建该命名管道的符号链接。一旦发现有对该管道的读行为时，系统判定检测到勒索软件并报警。如果勒索软件识别出该类管道文件，则该检测方法容易被绕过。

RWGuard［29］在文件的实时监控中部署了txt、doc、pdf、ppt、xls五类诱饵文件。其文件内容来源于文件夹中的同类型文件，以增强诱饵的真实性。诱饵文件与正常文件的相似性可能会导致用户难以区分诱饵文件，影响用户对诱饵的识别。

文献［44］根据Windows系统的文件遍历机制，设计以“sw”（最先遍历的符号）和“ew”（最后遍历的符号）开始的文件名和目录名作为诱饵。不管勒索软件采用何种文件遍历方式，都可以最大概率捕获到该样本。文献［45］把ASCII表中字节值小的打印字符’!’、‘〃’、’ #’作为诱饵文件名的第一个字符，并在用户的桌面和文档文件夹、磁盘根目录以及网络驱动器中部署这些诱饵。然而，其添加文件名前缀的方式单一，容易被攻击者猜测到。

文献［30］提出了一种栈式文件系统DcyFS，对可信进程提供真实的文件资源，对半可信进程提供部分真实的文件资源和诱饵资源，利用这些资源检测数据盗用和数据勒索。

文献［46］为了首要保护用户关键文件数据，提出在用户最近访问的10个文件夹和文件数量最

多的10文件夹中署诱饵文件。但文件数量最多的文件夹不一定是重要的文件夹，如日志文件夹。同时，该部署方式难以预测用户对文件的更新活动。

文献［36］通过大量用户调查和实验分析表明，当单个文件夹内诱饵文件数量与所有文件总数之比小于20%时，诱饵文件对正常用户的工作干扰可以忽略不计。

针对诱饵的生成、部署和检测技术，文献［18］提出识别诱饵文件的三种特征。其一是诱饵文件的静态特征。诱饵文件可能包含某些共有且易观测的特征，如文件的隐藏属性、文件内容为空、文件类型为管道、文件名或文件内容中的特定字符或字符串。其二是诱饵文件的文件修改时间特征。诱饵文件一般不会被用户或正常软件修改，诱饵文件的最后修改时间通常远早于系统中正常文件的最后修改时间，可以将最后修改时间远小于系统文件最后访问时间的文件标记为诱饵文件。其三则是诱饵文件的位置特征。诱饵文件通常部署在文件夹的首部或尾部（按照文件名的首字节的ASCII值排序），以确保诱饵优先被勒索软件访问。因此，勒索软件可以采取随机选择文件夹的文件加密，以绕过诱饵文件的定向部署。

基于诱饵文件的勒索软件检测研究目前仍未提出系统性的诱饵文件设计与部署方法，且往往由于诱饵文件部署不理想导致检测误报率上升。本文从诱捕防御出发，重点关注诱饵文件的动态更新，提出了Windows平台上一种基于诱饵文件和进程文件行为的勒索软件早期检测方法。检测框架如图3所示，主要由进程事件监控模块、诱饵部署与检测模块构成［47］。进程事件监控模块主要通过Windows事件追踪（ETW）和内核文件事件监控来记录系统中所有进程产生的各类事件，对收集到的事件进行类型过滤，并将处理后的事件交由诱饵部署与检测模块进行进一步分析。诱饵部署与检测模块主要负责在系统中的关键位置部署诱饵文件，并检测对诱饵文件的可疑访问行为，任何对诱饵文件进行写入或删除的进程都被认定为勒索软件。

在检测前，首先由诱饵部署与检测模块进行诱饵部署。诱饵部署包含两部分操作，其一是在勒索软件最先访问的文件夹中部署静态诱饵，其二是在当前用户最近访问过的文件夹中部署动态诱饵。部署完成后则将所有已部署诱饵添加至诱饵监控列表用于勒索软件检测。在检测过程中，由进程事件监控模块实时监控来自进程的各类事件。若捕获到的事件与用户最近访问文件夹的变化有关，则通知诱饵部署与检测模块，使其根据变化情况实时地增加或删除动态诱饵，并更新诱饵监控列表。若事件是写入或删除已部署的诱饵文件，则认为该事件来自于勒索软件，由诱饵部署与检测模块收集相关进程的信息并将其标记为勒索软件。

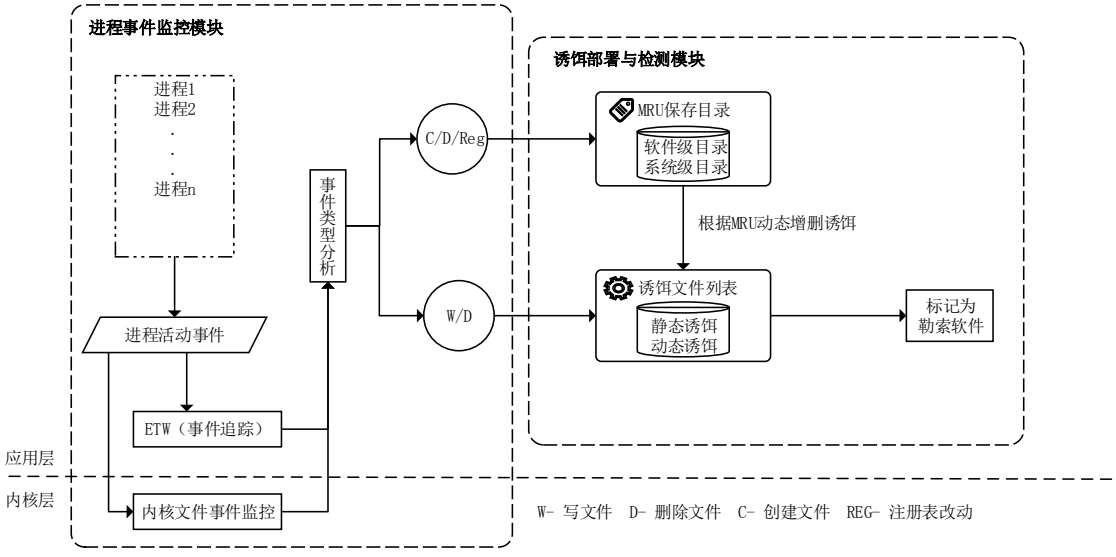


图3 基于诱捕的勒索软件检测框架

4.2 漏洞攻击检测

除数据诱饵和文件诱饵外，还有代码诱饵。ROP攻击会重用软件的代码。通过在软件中部署代码诱饵来实施诱捕防御，以检测ROP攻击。同时，当发现漏洞攻击后，可以把攻击流量或攻击载荷重定向到蜜罐系统中，在蜜罐系统中进一步观察攻击轨迹、识别攻击意图。

A-ROP防御

攻击者一般利用内存错误［48］在目标软件中注入ROP攻击代码。ROP编程从起初为复用库函数的返回到库函数（return-to-libc）的攻击［49］，逐渐发展为返回导向编程（ROP）［50］［51］、跳转导向编程（JOP）［52］、虚表导向编程（COOP）［53］等。不管ROP的编程如何演化，其攻击成功都是依赖于EIP的跳转和ROP预设链接地址的准确性［54］。控制流完整性保护（CFI）技术［55］确保EIP会跳转到合法的代码区域，但细粒度的CFI的开销比较大，且难以应用到复杂结构和动态代码的代码指针保护［56］。

ROP需要复用已知的代码和数据。随机化防御就是扰动目标软件的内存布局，使得攻击者获得的代码地址无效［57］。但内存泄露打破了内存布局随机化的假设［58］［59］，提高了ROP攻击的成功率。直观的对抗方法是缩短内存泄露信息的时效性，即动态随机化技术［60］［61］［62］［63］。但动态指针的跟踪和修订会带来较多的时间开销［60］。此外，动态随机化的空隙依然为攻击者提供了可乘之机。

文献［16］根据ROP复用目标软件中的代码这一特性，讨论了在目标软件中注入诱饵代码（即HoneyGadget）的设计思路。当检测到这些HoneyGadget被使用时，系统可以警告用户或终止进程。代码诱饵即使在内存泄露环境下，仍然是防御ROP攻击的一种有效选择。

CodeArmor［32］在保留初始版本外，随机化整个代码区域，并克隆多个随机化代码版本，利用系统调用和信号机制触发随机化版本的切换，其延迟较小。每次只有一个激活版本是有效的，其他版本都是诱饵。CodeArmor在切换版本时，将原位置替换为honey gadget来吸引攻击者。Readactor++［31］在虚函数表中插入了大量虚假的虚函数指针作为诱饵引诱攻击者，一旦攻击者执行了这些虚假指针指向的代码，则会落入其陷阱中。

文献［64］提出在二进制文件中插入nop指令和honey gadget，并维护一个记录了插入gadget的地址表。一旦目标应用程序执行了敏感的函数调用，HoneyGadget将暂停其执行并读取最近分支记录（LBR）缓冲区，以检查LBR记录是否与地址表中的地址匹配。如果记录匹配，则说明可能存在ROP攻击。

B-漏洞攻击劫持

攻击者攻击打补丁的Web服务时，其漏洞探测和漏洞攻击会失效。此时，无法感知攻击者的攻击意图，也没有办法收集攻击工具（远程控制工具）的信息。为了吸引攻击者和获取攻击载荷，文献［10］借助蜜罐的思想，设计了honey-patches和honey-patching。honey-patches在补丁代码中注入诱骗服务的触发点。当攻击者利用漏洞攻击载荷时，honey-patching把攻击载荷转发给诱骗服务器，实现引诱攻击者的目的。同时，利用内存修订和网络连接保持以确保Web自身的安全性和诱骗服务的真实性。该方法需要人工辅助才能设计已知漏洞对应的honey-patches，难以自动化部署，不能转发零日漏洞攻击。

4.3 Web安全

为了评估钓鱼攻击，研究受害者、钓鱼者以及研究者的行为，Phisheye［65］构建一个包含脆弱性的Web服务网站（即蜜罐服务），诱使攻击者攻击该网站，并在该网站安装钓鱼攻击包。从近五个月数据日志中，发现了474个网络钓鱼攻击包，初步刻画了钓鱼攻击包的完整生命周期。此外，29%的攻击者通过搜索引擎查询到部署的蜜罐，超过40%攻击者通过社交网络来共享蜜罐服务。同时，反网络钓鱼服务GSB和PhishTank对蜜罐服务上钓鱼网站的平均检测延迟约12天。

Tripwire［66］构建邮件系统的登录诱饵凭证（用户名和口令），然后利用这些诱饵凭证自动注册为第三方网站的用户。一旦有人利用网站的凭证登录给定邮件服务器，则可以检测到攻击，并推断出该凭证关联的网站被攻破。

DorkPot［67］利用GHDB（Google Hacking DataBase）的dork构建蜜罐网页，并把这些蜜罐网页部署在云服务平台，对外提供Web访问。攻击

者利用谷歌的搜索引擎，搜索并访问这些蜜罐网页。其实验结果统计了dork的使用频率，攻击者喜爱搜索网络摄像头和路由器等在线设备、密码和数据库等重要数据。

5 讨论

传统的防御方式，如流量入侵检测等，属于被动的防护手段。而诱捕防御属于主动防御技术，通过在APT攻击的各个阶段部署诱饵，可以快速检测和响应对主机的攻击。与传统防御手段对比，诱捕防御并不局限在特定的位置，而是贯穿于防御体系的各个环节。它可以和传统防御相结合，例如作为控制流完整性和地址随机化技术的补充手段，以对抗利用内存破坏漏洞的攻击，形成纵深的安全体系。

攻击者实施攻击时利用的资源和需要获取的资源都可以作为诱饵资源。IOT设备广泛应用于智能家庭、智慧工厂和智慧城市。为了获取攻击IOT设备的样本，可以虚拟各种IOT设备作为诱饵［68］［69］，并部署到云端，实时监控这些诱饵设备的访问情况，捕获攻击样本，并跟踪识别攻击意图。电话号码和设备同样可以用作诱饵［27］［28］，利用人工智能技术实施自动电话应答，尽可能多收集电话滥用信息，检测和发现语音钓鱼、电话营销、拒绝服务等电话威胁。此外，Honey-USB可以转发恶意USB固件访问主机的流量，可以识别攻击意图［70］；Honey page［71］［72］用于检测浏览器扩展的信息泄露。

工业界从基于蜜罐和蜜网的网络安全防御逐渐延伸到基于诱捕的主机安全保护。为对抗勒索软件，安全工具在主机上部署文件诱饵［73］［74］［75］，但这些诱饵文件命名方式和文件所在位置比较单一，容易被攻击者绕过。insightidr［76］使用用户访问凭证和用户文件作为诱饵资源，以此检测异常用户行为。其诱饵资源的多样性还需要实证分析。

同时，诱捕防御也存在一定的局限性，诱捕防御本身是一种概率机制。为了提高其成功概率，需要确保诱饵与真实资源的相似性、诱饵的多样性、部署位置的随机性。这些是诱捕防御机制将来的研究热点。文献［6］指出诱饵资源与被保护的业务系统高度一致。为了确保诱饵资源的相似性和多样性，诱饵资源还需要与用户行为高度一致，确保诱饵的动态性和新鲜性，使得攻击者难以识别诱饵资源。部署位置的随机性是指诱饵所在目录、路径是难以被攻击者猜测的。同时，这种随机性可以确保攻击者尽可能地访问这些目录和路径。另外，诱捕防御也有被绕过或者利用的可能。例如攻击者通过识别虚拟化环境和系统监控工具来判断当前环境是否安全。在部署诱饵的主机被控制后，攻击者也可以将它作为跳板，进而攻击其他主机和域服务器。

6 结论

控制流完整性、地址随机化和诱捕防御三者构建了内存破坏漏洞的立体防御体系。前两者是软件自身完整性和内存布局的多样性的结合，目的是抑制攻击代码的执行。诱捕防御部署在代码上，可以抑制攻击代码的执行；部署在数据资源、文件资源上，可以从系统层面主动感知攻击。因此，从防御层看，诱捕防御是控制流完整性和地址随机化的有效补充。

本文从APT攻击流程出发，分析和总结了APT攻击的每一个环节都可以部署诱饵，并实施诱捕防御。同时，总结了诱饵类型，剖析了诱饵的生成、部署和度量。接着，分析了诱捕防御在勒索软件检测、漏洞检测、Web安全等安全场景中的应用。最后，诱捕防御还可以应用到物联网安全、电话安全等领域。确保诱饵的真实性、动态性、新鲜性、多样性依然是未来研究工作的重点。

参考文献:

[1] 全球高级持续性威胁(APT)2018年总结报告. https://ti. 360. net/uploads/2019/01/02/56e5630023fe905b2a8f511e24d9b84a. pdf

[2] Zavarsky, Pavol, and LindskogDale. "Experimental analysis of ransomware on windows and android platforms: Evolution and characterization. "Procedia Computer Science 94（2016）: 465-472.

[3] Ye, Yanfang, et al. "A survey on malware detection using data mining techniques. "ACM Computing Surveys (CSUR) 50. 3（2017）: 41.

[4] 诸葛建伟, et al. "蜜罐技术研究与应用进展. "软件学报 24（2013）.

[5] Jajodia, Sushil, et al. Cyber deception. Springer，2016.

[6] 贾召鹏, et al. "网络欺骗技术综述. "通信学报 38. 12（2017）: 128-143.

[7] HOLMES: Real-time APT Detection through Correlation of Suspicious Information Flows. SP 2019.

[8] Jianming Fu, Lin Li, Yingjun Wang, Jianwei Huang, and Guojun

Peng. Web Scanner Detection Based on Behavioral Differences. 5th International Symposium，SocialSec 2019，Copenhagen，Denmark，July 14－17，2019

[9] Juels, Ari, and RivestRonald L. . "Honeywords: Making password-cracking detectable. "Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. ACM，2013.

[10] Araujo, Frederico, et al. "From patches to honey-patches: Lightweight attacker misdirection, deception, and disinformation. "Proceedings of the 2014 ACM SIGSAC conference on computer and communications security. ACM，2014.

[11] 乔向东, 郭戎潇, and 赵勇. "代码复用对抗技术研究进展. "网络与信息安全学报 4. 3（2018）：1-12.

[12] 彭国军, et al. "软件二进制代码重用技术综述. "软件学报 28. 8（2017）：2026-2045.

[13] 梁玉, 傅建明, 彭国军, 彭碧琛, 张焕国. S-Tracker:一种基于栈异常的shellcode检测方法. 华中科技大学学报，2014，(11）39-46.

[14] AbadiMartín, BudiuM. , and LigattiJ. . "Control-flow integrity. "Acm Conference on Computer & Communications Security 2005.

[15] Snow, Kevin Z. , et al. "Just-in-time code reuse: On the effectiveness of fine-grained address space layout randomization. " 2013 IEEE Symposium on Security and Privacy. IEEE，2013.

[16] Crane, Stephen, et al. "Booby trapping software. "Proceedings of the 2013 New Security Paradigms Workshop. ACM，2013.

[17] DING Shuang, FU Jianming, PENG BiChen. ModuleGuard: A Gatekeeper for Dynamic Module Loading Against Malware. Wuhan University Journal of Natural Sicence，2013，18(6)：489~498

[18] Genç Z. A. , Lenzini G. , Sgandurra D. (2019) On Deception-Based Protection Against Cryptographic Ransomware. In：Perdisci R. ，Maurice C. ，Giacinto G. ，Almgren M. (eds) Detection of Intrusions and Malware, and Vulnerability Assessment. DIMVA 2019. Lecture Notes in Computer Science，vol 11543. Springer, Cham

[19] 姚兰, 王新梅. 基于欺骗的网络主动防御技术研究[J]. 国防科技大学学报，2008，30(3)：65-69.

[20] Wang W, Bickford J, Murynets I, et al. Detecting targeted attacks by multilayer deception［J］. Journal of Cyber Security and Mobility，2013，2(2)：175-199.

[21] 刘秀文,傅建明,黎琳,等. 面向用户交互场景的信息欺骗分类及其威胁抑制机制[J]. 武汉大学学报(理学版)，2019，65(2)：126-138

[22] LiaoX. , YuanK. , WangX. , LiZ. , XingL. and BeyahR. ，"Acing the IOC game: Toward automatic discovery and analysis of open-source cyber threat intelligence"，
Proc. ACM SIGSAC Conf. Comput. Commun. Secur，pp. 755-766, Oct. 2016.

[23] Spitzner, Lance. "Honeypots: Catching the insider threat. "Computer Security Applications Conference，2003. Proceedings. 19th Annual. IEEE，2003.

[24] Yuill, Jim, et al. "Honeyfiles: deceptive files for intrusion detection. "Information Assurance Workshop，2004. Proceedings from the Fifth Annual IEEE SMC. IEEE，2004.

[25] Bowen, Brian M. , et al. "Baiting inside attackers using decoy documents. "International Conference on Security and Privacy in Communication Systems. Springer，Berlin，Heidelberg，2009.

[26] Park, Younghee, and StolfoSalvatore J. . "Software-based Decoy System for Insider Threats. "Proceedings of the 7th International Symposium on ACM Symposium on Information，Computer and Communications Security (ASIACCS 2012)，Seoul，May 2-4，2012. ACM Press，2012.

[27] Gupta, Payas, et al. "Phoneypot: Data-driven Understanding of Telephony Threats. "NDSS. 2015.

[28] Balduzzi, Marco, et al. "Mobipot: Understanding mobile telephony threats with honeycards. "Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security. ACM，2016.

[29] Mehnaz S, Mudgerikar A, Bertino E, et al. RWGuard: A Real-Time Detection System Against Cryptographic Ransomware［C］. recent advances in intrusion detection，2018：114-136.

[30] Taylor, Teryl, et al. "Hidden in Plain Sight: Filesystem View Separation for Data Integrity and Deception. "International Conference on Detection of Intrusions and Malware， and Vulnerability Assessment. Springer，Cham，2018

[31] Crane, Stephen J. , et al. "It's a TRaP: Table randomization and protection against function-reuse attacks. "Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM，2015.

[32] Chen, Xi, BosHerbert, and GiuffridaCristiano. "CodeArmor: Virtualizing the code space to counter disclosure attacks. " 2017 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE，2017.

[33] Cenys, RainysD. , RadvilaviciusL. , and GoraninN. , "Implementation of Honey token Module In DBMS Oracle 9ir2 Enterprise Edition for Internal Malicious Activity Detection, " IEEE Computer Society'S TC on Security and Privacy，2005.

[34] White, Jonathan. "Creating personally identifiable honeytokens. " Innovations and Advances in Computer Sciences and Engineering. Springer，Dordrecht，2010. 227-232. //names，addresses，telephone numbers and social security numbers

[35] Bercovitch, Maya, et al. "HoneyGen: An automated honeytokens generator. "Intelligence and Security Informatics (ISI)，2011 IEEE International Conference on. IEEE，2011.

[36] Salem, BenMalek, and StolfoSalvatore J. . "Decoy document deployment for effective masquerade attack detection. "International Conference on Detection of Intrusions and Malware， and Vulnerability Assessment. Springer，Berlin，Heidelberg，2011.

[37] Kharraz A. , Robertson W. , Balzarotti D. , Bilge L. , Kirda E. : Cutting the Gordian Knot：a look under the hood of ransomware attacks. In：Almgren，M. ，Gulisano，V. ，Maggi，F. （eds. ）DIMVA 2015. LNCS，vol. 9148，pp. 3 － 24. Springer, Cham（2015）.

[38] Spotless Sandboxes: Evading Malware Analysis Systems using Wear-and-Tear Artifacts. SP 2017.

[39] Scaife N. , Carter H. , Traynor P. ，Butler, K. R. : CryptoLock (and Drop It): stopping ransomware attacks on user data. In: IEEE International Conference on Distributed Computing Systems (ICDCS) (2016）

[40] Continella A. , Guagnelli A. , Zingaro G. , De Pasquale G. , Barenghi A. , Zanero S. , Maggi, F. : ShieldFS: a self-healing, ransomware-aware filesystem. In: Proceedings of the 32nd Annual Conference on Computer Security Applications，pp. 336 – 347. ACM (2016）

[41] Kharraz A, Kirda E. Redemption: Real-Time Protection Against Ransomware at End-Hosts[C]//International Symposium on Research in Attacks，Intrusions，and Defenses. Springer，Cham，2017：98-119.

[42] Kharraz A. , Arshad S. , Mulliner C. , Robertson W. , Kirda, E. UNVEIL: A large-scale, automated approach to detecting ransomware. In: 25th USENIX Security Symposium (2016）

[43] Gómez-Hernández J. A. , Álvarez-GonzálezL. , and García-TeodoroPedro. "R-Locker: Thwarting ransomware action through a honeyfile-based approach. "Computers & Security，73（2018）：389-398.

[44] Lee J, Lee J, Hong J. How to Make Efficient Decoy Files for Ransomware Detection? [C]//Proceedings of the International Conference on Research in Adaptive and Convergent Systems. ACM，2017：208-212.

[45] El-Kosairy A. , & Azer M. A. （2018，April）. Intrusion and ransomware detection system. In 2018 1st International Conference on Computer Applications & Information Security (ICCAIS) (pp. 1-7). IEEE.

[46] Voris J, Song Y, Salem M B, et al. Active authentication using file system decoys and user behavior modeling: results of a large scale study[J]. Computers & Security，2019.

[47] 杨铮,傅建明,罗陈可,黄坚伟. 一种基于诱饵文件的勒索软件及时检测方法. 武汉大学学报(理学版)，2020，66(5)：230~240.

[48] SZEKERES L, PAYER M, WEI T, et al. Sok: Eternal war in memory [C] Proceedings of the 34th IEEE Symposium on Security and Privacy (S&P'13). 2013 ：48 – 62

[49] SHACHAM H. The geometry of innocent flesh on the bone: return-into-libc without function calls (on the x86)[C] // Proceedings of the 14th ACM conference on Computer and Communications Security (CCS'07). 2007 ：552 – 561.

[50] BUCHANAN E, ROEMER R, SAVAGE S, et al. Return-oriented programming: Exploitation without code injection[J]. Black Hat，2008，8.

[51] ROEMER R, BUCHANAN E, SHACHAM H, et al. Return-oriented programming: Systems, languages, and applications [J]. ACM Transactions on Information and System Security (TISSEC'12)，2012，15(1)：2.

[52] BLETSCH T, JIANG X, FREEH V W, et al. Jump-oriented programming: a new class of code-reuse attack[C] // Proceedings of the 6th ACM Symposium on Information，Computer and Communications Security(AsiaCCS'11). 2011 ：30 – 40.

[53] SCHUSTER F, TENDYCK T, LIEBCHEN C, et al. Counterfeit Object-oriented Programming: On the Difficulty of Preventing Code Reuse Attacks in C++ Applications[C]// Proceedings of the 36th IEEE Symposium on Security and Privacy (S&P'15). 2015 ：745 – 762

[54] 彭国军,梁玉,张焕国,傅建明. 软件二进制代码重用技术综述. 软件学报，2017，28(8)：2026−2045.

[55] Hu, Hong, et al. "Enforcing Unique Code Target Property for Control-Flow Integrity. "Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. ACM，2018.

[56] Xu, Xiaoyang, et al. "{CONFIRM}: Evaluating Compatibility and Relevance of Control-flow Integrity Protections for Modern Software. "28th USENIX Security Symposium，2019.

[57] 傅建明, 林艳, 刘秀文,张旭. 云计算环境下基于随机化的安全防御研究. 计算机学报，2018，41(6)：1207~1224

[58] Seibert, Jeff, OkhraviHamed, and SöderströmEric. "Information leaks without memory disclosures: Remote side channel attacks on diversified code. "Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. ACM，2014.

[59] 傅建明, et al. "内存地址泄漏分析与防御. "计算机研究与发展 53. 8 (2016)：1829-1849.

[60] Bigelow, David, et al. "Timely rerandomization for mitigating memory disclosures. "Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM，2015.

[61] Chen, Yue, et al. "Remix: On-demand live randomization. "Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy. ACM，2016.

[62] Lu, Kangjie, et al. "How to Make ASLR Win the Clone Wars: Runtime Re-Randomization. "NDSS. 2016.

[63] Williams-King, David, et al. "Shuffler: Fast and deployable continuous code re-randomization. "12th USENIX Symposium on Operating Systems Design and Implementation (OSDI16). 2016.

[64] Huang X. , Yan F. , Zhang L. , & Wang K. （2019，August）. HoneyGadget: A Deception Based ROP Detection Scheme. In International Conference on Science of Cyber Security (pp. 121-135）Springer, Cham.

[65] Han, Xiao, KheirNizar, and BalzarottiDavide. "Phisheye: Live monitoring of sandboxed phishing kits. "Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM，2016.

[66] DeBlasio J. , Savage S. , Voelker G. M. , & Snoeren A. C. （2017，November）. Tripwire: Inferring internet site compromise. In Proceedings of the 2017 Internet Measurement Conference (pp. 341-354).

[67] Quinkert, Florian, LeonhardtEduard, and HolzThorsten. "DorkPot: A Honeypot-based Analysis of Google Dorks. " Proceedings of the Workshop on Measurements, Attacks, and Defenses for the Web (MADWeb '19), San Diego, CA. 2019.

[68] Pa Y M P, Suzuki S, Yoshioka K, et al. IoTPOT: analysing the rise of IoT compromises [C]//9th USENIX Workshop on Offensive Technologies (WOOT 15). 2015.

[69] Dang F. , Li Z. , Liu Y. , Zhai E. , Chen Q. A. , Xu T. , Chen Y. , and Yang J. Understanding fileless attacks on linux-based iot devices with honeycloud. In International Conference on Mobile Systems, Applications, and Services (Mobisys) (2019), pp. 482--493.

[70] Dave Jing Tian, BatesAdam, and ButlerKevin. 2015. Defending

against malicious USB firmware with GoodUSB. In Proceedings of the 31st Annual Computer Security Applications Conference. ACM, 261--270.

[71] Mengfei Xie, Jianming Fu, Jia He, Chenke Luo, and Guojun Peng. JTaint: Finding Privacy-Leakage in Chrome Extensions. ACISP 2020.

[72] Chen Q.， Kapravelos, A.: Mystique: Uncovering information leakage from browser extensions. In：Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 1687-1700. ACM (2018）

[73] Cybereason RansomFree. https：//www. cybereason. com/

[74] CyberSight RansomStopper. https：//cybersight. com/

[75] ZoneAlarm Anti-Ransomware. https：//www. zonealarm. com/anti-ransomware/

[76] insightidr. https：//www. rapid7. com/products/insightidr/features/deception-technology/

[作者简介]

傅建明（1969生），男，博士，教授，主要研究方向为系统安全、移动安全。

刘畅（1998生），男，硕士生，主要研究方向为移动安全。

解梦飞（1997生），男，硕士生，主要研究方向为系统安全。

罗陈可（1996生），男，博士生，主要研究方向为系统安全、二进制安全。

拟态防御技术在量子时代面临的机遇与挑战

何明[1]，王俊超[1，2]，刘晓楠[1]，庞建民[1]，单征[1]，卫今[3]，张帆[1]

[1]国家数字交换系统工程技术研究中心 河南 郑州 450002；

[2]中国科学技术大学物理学院 合肥230026；

[3]复旦大学计算机科学技术学院 上海 200433；

[4]通讯作者：王俊超

摘　要：网络空间拟态防御技术（Cyberspace Mimic Defense， CMD）是一种颠覆传统防御技术的新型技术手段。与传统的技术手段不同，拟态防御技术引入了动态、异构、冗余（Dynamic、Heterogeneous、Redundant，DHR）的特性，使得攻击者在攻击时难以像传统安全一样能够通过不断试错来达到攻击目的。在现有高性能计算机技术发展的当下，解决了传统安全防御攻防不平衡的现状。然而，随着量子时代的到来，传统的密码体系逐渐变得不再安全。一旦大规模量子计算机制造出来，那么现有的所有密码体系将会失效。由于量子计算机具有天然的强大并行处理能力，如果攻击者能够利用量子计算机强大的算力来实时穷举所有的可能性，将会对拟态防御技术造成巨大的安全威胁。当然，如果量子计算技术能够与拟态防御技术相结合，将有效弥补拟态技术在DDoS攻击防御方面的不足。在量子计算技术飞速发展的当下，本文论述了现有量子计算机的发展现状及问题；对网络空间拟态防御技术在量子时代所面临的巨大挑战和机遇进行了分析；并对拟态防御技术与量子计算技术的融合进行了展望。

关键词：拟态防御技术、量子计算机、DDoS攻击、量子霸权

Opportunities and challenges of mimicry defense technology in the Quantum Age

He Ming[1], Wang Junchao[1,2], Liu Xiaonan[1], Pang Jianmin[1], Shan Zheng[1], Wei Jin[3], Zhang Fan[1]

1.National Engineering and Technology Research Center for Digital Switching Systems, Zhengzhou 450002, China;

2.School of Physics, University of Science and Technology of China hefei 230026, China;

3.School of Computer Science and Technology, Fudan University, Shanghai 200433, China

Abstract: Cyberspace Mimics Defense (CMD) is a new technical means to overturn the traditional Defense technology. Different from traditional techniques, mimicry defense technology is introduced into the Dynamic, Heterogeneous, redundancy (Dynamic, Heterogeneous, Redundant, DHR) features, makes it hard for the attacker in the attack as a traditional security can achieve through continuous trial and error to attack purposes. In the existing high-performance computer technology development today, has solved the traditional security defense of offensive and defensive imbalance situation. However, with the advent of the era of quantum, the traditional password system gradually become no longer safe. Once a large-scale quantum computer is built, all existing cryptographic systems will become ineffective. Because the quantum computer has strong natural parallel processing ability, if an attacker can exploit the powerful computing power of quantum computers to exhaust all possibilities in real time, will the safety of the mimicry defense technology poses a

great threat. Of course, if quantum computing technology can be combined with mimicry defense technology, it will effectively make up for the shortcomings of mimicry technology in DDoS attack defense. In quantum computing technology rapid development today, the paper discuss the existing current situation and problems of the development of quantum computer; analyzes the great challenges and opportunities faced by the cyber space mimicry defense technology in the quantum age; and fusion of mimicry defense technology and quantum computing technology is prospected.
Key words: mimicry defense technology; quantum computer; DDoS attack; quantum supremacy

1 背景

随着网络化和信息化的快速发展，人们在社会活动中得到极大的便利。但是，网络空间安全问题同样存在于社会活动的每一角落。在网络环境中常常出现诸多不确定威胁，如软硬件系统存在的漏洞，植入软硬件系统的后门。使得少数不法组织或人员利用软硬件漏洞或后门侵犯公众的隐私权，同时对网络基础设施和公共服务系统进行威胁，这将对网络空间及社会稳定产生极大威胁。传统网络空间防御系统是以威胁特征感知的精确防御为基础的体系，是在“已知风险”或“已知的未知风险”的前提条件上建立起来的，需要多方面先验知识的支撑，这种防御为“后天获得性免疫”。除了一些加密认证外，在遇到漏洞后门等具有不确定性的攻击下，传统网络空间防御是不进行防御的，这彰显了防御体制的弱点。因此，网络空间安全呈现一种“易攻难守”的态势。在此背景下，邬江兴院士提出了网络空间拟态防御（CyberSpace Mimic Defense，CMD）技术。作为一种颠覆传统网络空间防御技术的一种新型技术手段，拟态防御技术引入了动态、异构、冗余的特性，保证系统在受到攻击前能够主动识别，并随时应对可能出现的攻击[1]。

在现有高性能计算机技术发展的当下，拟态防御解决了系统存在的漏洞、后门、木马等问题，将不确定性威胁转化为已知的概率性可靠事件，从而改善传统安全防御攻防不平衡的现状。然而，随着量子时代的来临，传统密码体系逐渐变得不再安全。如果大规模量子计算机被制造出来，那么现有的全部传统密码体系将会失效。

拟态防御技术本质上是一种引入随机性、动态性、冗余性等的一种技术，使得攻击者在攻击时难以像传统安全空间一样能够通过试错法来达到攻击目的。在量子领域，量子计算机具有天然的强大并行处理能力，如果攻击者能够利用量子计算机强大的算力来实时穷举所有的可能性，将会对拟态防御技术造成巨大的安全威胁。然而，挑战与机遇并存。在拟态防御技术中的关键技术上配以对应的量子算法，将能够极大地提升拟态防御系统性能，从而解决对于复杂的攻防对弈问题。例如，当量子计算技术能够与拟态防御技术相结合时，可以有效弥补拟态技术在DDoS攻击防御方面的不足。在软件安全方面，量子计算能够利用其并行算法促进多样化编译变体的生成。

近几年，量子计算机作为一种新型的计算技术登上舞台。与经典计算机相比，量子计算机在处理某些问题上，如大数据搜索，因式分解等方面更具有优势。特别是在2019年9月，谷歌宣称实现“量子霸权”，研发的量子处理器“Sycamore”在随机量子电路采样时能够在200s完成目前最先进的超级计算机需要1万年处理的问题。然而，对于量子计算机而言，所能实现的量子比特的规模目前还比较小。虽然量子比特的数量与经典计算机相比显得不多，但是所能描述和操作的量子信息却非常大。世界上最强大的计算机也难以完成拥有53位量子比特的量子计算机能够完成的任务。当计算规模进一步上升，就到了量子优越性阶段，即量子计算机能够完成的计算任务在任何经典计算机中都难以完成。因此，国内外顶尖量子研究团队都在努力制造量子比特数目更多的量子计算机。

量子计算机基于量子力学原理能够以极大的速度提升和改善量子计算机的处理性能。然而，相较于传统计算机而言，量子计算机仍然处于初级阶段。不同的量子计算机实现方案层出不穷，但是，目前并没有统一的标准来衡量量子计算机不同方面、不同视角的性能指标。

2 原理

2.1 拟态防御技术原理

自然界中，常出现一种生物在色彩、纹理和

形状等特征上模拟另一种生物或环境，从而使一方或双方受益的生态适应现象，在生物学中称为拟态现象[2]。拟态现象又被称为“拟态伪装”，是一种基于内生机理的主动防御现象。在网络空间主动防御中，由于大多数信息系统并不能隐藏云计算、路由交换、文件存储等服务功能和性能，同时需要让外界对其使用方法和功能细节以及考虑使用习惯的延续问题有相关的了解。因此，拟态伪装并不能作为网络空间主动防御的基本概念。除此之外，目标对象的系统架构、运行机制、核心算法、异常表现以及可能存在的不确定漏洞或后门等行为，都可以通过类似拟态伪装的方式进行主动隐匿。拟态防御是由除隐藏目标对象服务功能之外的拟态伪装定义的。

在异构冗余体制中，非相似余度构造（Dissimilar Redundancy Structure，DRS）[3]是一个非常典型的例子。但DRS设计复杂，成本高昂，并且其本身是是静态不变的，攻击者若掌握目标过程信息，就有可能实现跨物理层的侧信道攻击。因此，拟态防御技术引入了动态异构冗余（Dynamic Heterogeneous Redundant，DHR）[4]的思想。如图1所示，为动态异构冗余结构示意图。对于DHR，要求系统具有视在结构表征的不确定性，来增加攻击者的成本。

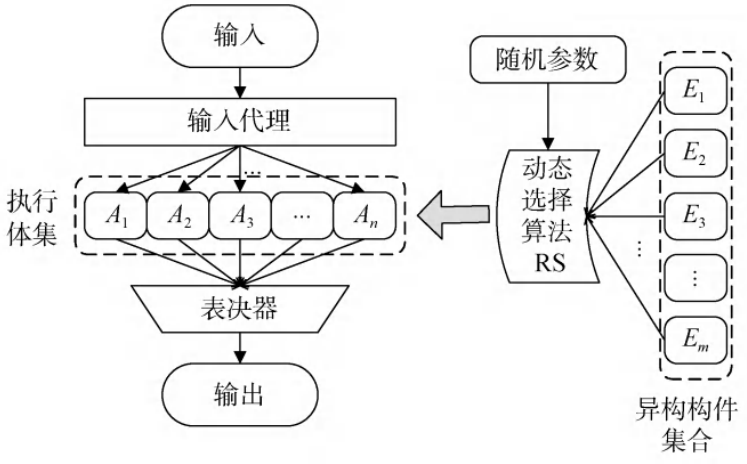


图1　DHR结构示意图[5]

2.2　量子计算机简介

早在1900年，普朗克通过对黑体辐射实验进行分析，提出辐射量子假说。即假定电磁场和物质交换能量是以间断的形式（能量子）实现的，能量子的大小同辐射频率成正比，比例系数为普朗克常数。这一假使得量子力学正式亮相于现代物理学范畴内。作为现代物理学的基础理论之一，量子力学是描述原子和亚原子尺度的物理学理论。

80年代初期，阿岗国家实验室的P. Benioff提出二能级的量子系统可以用来仿真数字计算，这是最早的量子计算概念。而后费曼也对这个问题行进研究，并在1981年于麻省理工学院举行的First Conference on Physics of Computation中演讲，勾画以量子现象实现计算的愿景。

1985年，牛津大学的D. Deutsch提出量子图灵机（Quantum Turing Machine）[6]的概念，使得量子计算具备数学的基本型式。然而，这一阶段的量子计算研究多半局限于探讨计算的物理本质，比较抽象，尚未进一步跨入发展算法的阶段。

直到1994年，贝尔实验室的应用数学家P. Shor提出用于将一个很大的整数分解成质因子的乘积的Shor算法[7]。由于算法对常用的经典公钥密码体制造成威胁，所以引起广泛关注。1996年，同在贝尔实验室的Grove提出了Grove算法[8]。对于在无序数据库中搜索若干特定目标时，Grove算法可以对一些启发式算法起到二次加速作用[9]。

因此，量子计算机可以被定义为遵循量子力学原理进行高速数学和逻辑运算、存储及处理量子信息的一类物理装置。

3　量子计算机发展现状

3.1　量子芯片发展现状

量子计算机的容错定律表明：只有当门错误率很低和串扰足够弱时，量子计算机才能够正常

的工作[10]。因此，量子计算机的工程化实现必须要突破容错率问题。而解决量子容错的基本方法首先需要明确量子计算机中误差的基本模型，并使用某种技术手段进行测量，在得到测量结果之后对误差采用相应的量子校准和操控技术修正和调校量子计算机，最终再次检验校准结果。具体来说，为了确保量子计算机能够根据设计好的算法正确工作，需要完成以下的具体工作：

（1）设计出测评方法来识别对量子计算机影响最大的噪声；

（2）建立精确的误差模型；

（3）根据容错量子计算相关的指标参数来量化错误率；

（4）尽可能的使用较少的资源进行量子计算机的校准和修正；

因此，为了实现量子计算机，需要对其进行精确、全面的测评。表1给出了目前世界上著名的量子计算机中量子芯片的6个指标参数：量子比特数目、连通度、T_1[11]、T_2[12]、单量子比特门保真度[13]、两量子比特门保真度[14]和读出保真度[15]。从表1中可以看出，即使采用相同的物理实现方式（如IBM的超导量子计算机），其测评的指标也存在较大差异。

表1　目前世界上著名的量子计算机中量子芯片的6个指标参数表

序号	机器	国家	物理实现	量子比特数目	量子比特连通度			T1(微秒)			T2(微秒)			单量子比特门保真度			两量子比特门保真度			读出保真度		
					最小值	最大值	平均值	最小值	最大值	平均值	最小值	最大值	平均值	最小值	最大值	平均值	最小值	最大值	平均值	最小值	最大值	平均值
1	IBM Q5 Tenerife	西班牙	超导	5	2	4	2.4	34.3	49.7	43.6	4.8	53.3	25.1	99.83%	99.93%	99.89%	93.78%	97.74%	95.54	64.90%	95.70%	84.70%
2	IBM Q5 Yorktown	美国	超导	5	2	4	2.4	36.04	73.66	58.83	25.16	73.8	54.65	99.87%	99.92%	99.90%	98.46%	98.89%	98.66%	97.45%	98.70%	98.31%
3	IBM Vigo	西班牙	超导	5	1	3	1.6	71.09	124.55	100.45	18.39	127.74	80.77	99.86%	99.92%	99.89%	98.81%	99.29%	99.08%	93.60%	98.50%	97.22%
4	IBM Ourense	西班牙	超导	5	1	3	1.6	69.04	117.51	92.66	30.14	126.47	60.5	99.90%	99.95%	99.93%	99.04%	99.27%	99.20%	96.30%	98.90%	97.44%
5	IBM Q16 Melbourne	澳大利亚	超导	14	1	3	2.57	25.5	88.5	52.2	15.6	105	64.8	96.26%	99.80%	99.20%	84.52%	97.11%	92.99%	89.30%	96.59%	94.64%
6	IBM Q20 Poughkeepsie	美国	超导	20	2	3	2.3	39.4	123.3	73.2	10.8	123.6	66.2	99.72%	99.95%	99.89%	93.39%	98.89%	97.75%	TBD	TBD	TBD
7	IBM Q20 Tokyo	日本	超导	20	2	6	3.9	42.2	148.5	84.3	24.3	78.4	49.6	99.39%	99.94%	99.80%	92.88%	98.53%	97.16%	N/A	N/A	91.72%
8	IBM Q20 Austin	美国	超导	20	2	6	3.9	N/A	N/A	102.6	N/A	N/A	111.26	N/A	N/A	N/A	N/A	N/A	98.47%	N/A	N/A	91.55%
9	IBM Q System One	美国	超导	20	2	6	3.9	38.2	132.9	73.9	39.2	100.8	69.1	99.92%	99.98%	99.96%	97.15%	99.03%	98.31%	TBD	TBD	TBD
10	IBM Almaden Boelingen Singapore	美国/德国/新加坡	超导	20	1	3	2.3	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD
11	IBM Rochester	美国	超导	53	1	3	2.15	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD
12	Rigetti 16Q Aspen-4	美国	超导	16	2	3	2.25	N/A	N/A	25.24	N/A	N/A	19.89	N/A	N/A	95.50%	N/A	N/A	90.35%	N/A	N/A	93.02%
13	Rigetti Aspen-7	美国	超导	28	2	3	2.375	N/A	N/A	41	N/A	N/A	35	N/A	N/A	99.23%	N/A	N/A	95.20%	N/A	N/A	96.40%
14	Rigetti Aspen-8	美国	超导	31	2	3	2.375	N/A	N/A	29	N/A	N/A	18	N/A	N/A	99.70%	N/A	N/A	95.66%	N/A	N/A	N/A
15	IonQ 11 Qubit	美国	离子阱	11	11	11	11	>10^10	>10^10	>10^10	TBD	TBD	3*10^6	99.18%	99.64%	99.46%	95.10%	98.90%	97.50%	N/A	N/A	99.30%
16	Honeywell 4 Qubit	美国	离子阱	4	4	4	4	TBD	TBD	TBD	TBD	TBD	TBD	99.986%	99.991%	99.989%	99.12%	99.28%	99.20%	99.70%	99.80%	99.70%
17	Honeywell 64 Quantum Volume	美国	离子阱	>=6	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	99.99.7%	TBD	TBD	TBD	TBD	TBD	TBD	TBD
18	Google Sycamore	美国	超导	53	1	4	3.25	9.7	27.8	16.04	N/A	N/A	N/A	99.66%	99.92%	99.84%	N/A	N/A	99.38%	84.40%	99.60%	96.20%
19	QuTech Spin-2	荷兰	超导	2	1	1	1	N/A	N/A	>20000	N/A	N/A	>6	N/A	N/A	99%	N/A	N/A	>90%	N/A	N/A	85%
20	QuTech Starmon-5	荷兰	超导	5	1	4	2.4	8.9	24	17.5	14.6	26.8	23	99.80%	99.90%	99.84%	95.30%	97.90%	97.15%	95.30%	98.70%	97.08%
21	QuTech&Intel Horse Ridge	荷兰	超导	49	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD
22	KF C6-130	中国	超导	6	TBD	TBD	TBD	TBD	TBD	5	TBD	TBD	2	TBD	99.70%	99.50%	TBD	TBD	90%	TBD	TBD	90%

量子计算机中的量子比特数目是衡量量子计算机计算能力的重要指标参数。量子比特数目越多，量子计算机所能描述的信息量越大。量子霸权（Quantum Supremacy）的概念直接与量子比特数目相关。当要解决的问题规模达到一定程度时，经典计算机难以在有限的时间内完成的问题可以用量子计算机很快解决。经典计算机中的比特相对稳定，非常容易操作；而量子特性广泛存在于微观世界，操控起来非常复杂。目前，量子比特的物理实现方式主要包括：超导、离子阱、光、NMR等。例如，在使用钙离子实现量子比特时，需要利用射频和直流电极将离子困在一个腔体内。这个腔体由激光降温至几mK的低温。激光和微波信号能够精确制备出量子的“0”、“1”状态。量子状态可以通过荧光（一种光致冷发光现象）读取出来。逻辑操作可以采用微波脉冲的组合来实现。目前，这种制备量子初态的方法和读取误差可以控制在0.07%以内，量子逻辑门操作误差控制在10^{-6}以内[16]。

连通度的概念与图论中的连通度概念较为接近，是指量子计算机量子拓扑图中最大的量子比特的度数。图2给出了IBM公司的5量子比特Tenerife[17]，16量子比特Melbourne[18]，20量子比特Tokyo和20量子比特Johannesburg四台量子计算机的量子比特拓扑图。

量子拓扑图中每个节点表示了一个量子比特。

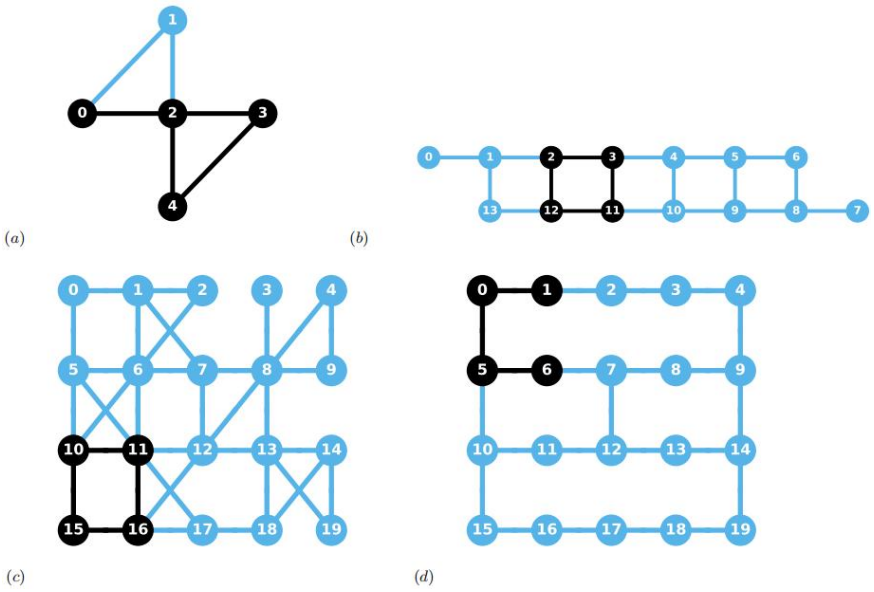


图2　IBM公司四台量子计算机的量子比特拓扑图

两个量子比特之间存在边，则说明多门操作可直接作用于这两个量子比特之上。当两个量子比特之间不存在边，则需要上层量子软件（如量子编译器）通过增加一系列SWAP操作，来实现非连通量子比特之间的相互作用。量子计算机的连通度越高，则表明量子计算机的硬件可编程性和可操作性更强。若量子计算机中的每个量子比特都与其他所有的量子比特之间存在通路，这种量子计算机称为全连通量子计算机。在这种量子计算机上执行门操作非常灵活，且能够减少SWAP门带来的额外门操作，从而降低量子线路深度[19]。

量子计算机的最大挑战在于如何最大限度的保留其量子态叠加时间，因为这将有利于保留更长时的量子信息，从而有助于开发更可靠的超级量子计算机。在全球量子芯片质量评测中，T_1是一个非常重要的指标。T_1是与一个量子态从高能级|1>衰变到基态|0>的时间相关的指标。受到外界环境的影响，高能级的能量会泄漏到外部环境中。

3.2　量子霸权

2019年10月，Google在Nature杂志正式发布了其实现量子霸权的论文[20]。量子霸权（Quantum Supremacy）其更准确的内涵是量子优越性（Quantum Advantage），是指对于某些问题，当问题规模达到一定程度时，经典计算机难以完成量子计算机能够完成的任务。Google首先在量子硬件上有所突破，设计了具有54位量子比特的芯片，然而，其中一个量子比特不能正常工作，因此，在他们所实现的量子霸权论文中，只用了53位量子比特。也就是说，用这53位量子比特能够同时表示2^{53}个状态。量子中的状态通常包括实部和虚部，如果都是双精度的浮点来描述状态，那么总共需要$16=2^4$个字节来表示一个状态。

验证量子霸权的测试基准（benchmark）中需要完成四项工作：

（1）定义一个计算问题。

（2）设计量子算法来解决该问题。

（3）与经典算法能够运行的最好结果进行对比。

（4）根据复杂性理论分析，证明没有经典算法能够比现有的算法更好。

目前，较为通用的用于验证量子霸权的测试集包括：

(1)　Shor算法

Shor算法能够利用量子计算机解决大数分解问题。给定n位的二进制整数，只需要O（n^3）即可完成数据分解，而最优的经典算法则需要2O（$n^{1/3}$）。大数分解问题在经典计算机中最优的上界复杂度为O（$2^{n/3}$+O（1））。

Shor算法是第一个能够利用量子计算机在多项式时间内解决现实问题的量子算法。大数分解问题是RSA加密体系的基础，因此，一旦量子计算机研制成功，且Shor算法能够正确运行，那么，

RSA加密体系就会瞬间瓦解，区块链技术所依赖的基础也将不复存在。然而，由于目前的量子计算机中资源远远没有达到分解一个大数的需要，因此，Shor算法在目前并没有被用于验证量子霸权。

(2) Boson采样

Boson采样[21]是指通过线性光学网络发送玻色子的概率分布进行采样，能够解决某些搜索问题。然而，包含足够大噪声的Boson采样很难能够在经典计算机上进行高效模拟。

目前最大的Boson采样物理实现能够一次处理6个光子。一个具有n个光子的系统和m个输出模式的系统的Boson采样在经典计算机上的最优时间复杂度能够达到O（$n2^n + mn^2$）。

(3) 随机量子线路输出采样

目前已知的最优的模拟量子随机线路的算法时间复杂度随着量子比特数目的增加呈指数级增长。2019年，Google发布了53位的量子芯片sycamore，证明了该任务在量子计算机上执行只需要200秒，而在世界上最快的经典超级计算机上运行需要10000年[22]。线性方程组求解之所以能够作为高性能计算机的测评，是由于其具有代表性。而Google所验证的量子霸权只是用了一种随机线路。

事实上，量子计算机在某些问题上能够有加速。如果将经典高性能计算的测试集移植到不存在误差的量子计算机上来做，的确有方法可以实现。然而，哪怕是一个很简单的二元方程组用HHL算法[23]进行求解，其量子线路的深度都能达到326。随着问题规模的增大，量子线路深度就会急剧增加。即使精度比较高的门操作，误差随着线路深度增加而不断放大，最终可能导致不能够得到正确的结果。因此，一个具有现实意义的问题在现有的量子计算发展早期阶段还难以用来测评量子计算机的性能。

对于量子霸权的质疑声一直未断。IBM声称Google验证的量子霸权，如果用一台超级计算机，采用某种优化手段只需要2.5天就能完成。此外，由于量子计算机中存在大量的噪声，那么要对量子计算机进行纠错或许也需要非常多的资源。虽然量子计算机运行只需要200秒，但是纠错或许需要10000年时间。

3.3 量子计算发展所面临的问题

量子计算机几乎可以说是目前最为复杂的计算系统，涉及到物理、化学、数学、计算机、电子工程等多个领域学科相关的知识和背景。在解决诸如化学、物理模拟、机器学习、密码学、线性代数、金融、推荐、网络等产生的经典难题，量子计算机被认为是超越摩尔定律的最有前途的计算范式之一。然而，目前真正意义上实用的量子计算机还未完成。

（1）底层物理基础难以满足量子计算理想模型。量子计算的基础理论事实上相对研究比较完备了。本质上讲，量子图灵机与现有的图灵机在计算能力上并没有特别大的差别（这里所说的计算能力并不是算得多快，而是能不能算）。正如图灵机理论模型与现有的电子计算机工程实现一样，量子计算理论模型也需要与某种具备良好量子操控特性、具有噪声容忍的物理实现相对应。这也是目前量子计算所面临的重大问题，这部分的工作需要不同领域的科学家勠力同心才能有所突破。例如，2020年4月，霍普金斯大学的研究者采用了一种新型材料，或许能够极大推动量子计算机工程化实现。

（2）在量子算法设计方面，研究者需要用量子的思维去思考问题。与传统计算机编程模型不同，量子计算机的编程思维有着很大的区别。量子算法近年来的发展速度其实并不快。真正的有着较大影响力的算法包括用来进行素数分解的Shor算法和用来搜索的Grover算法等，当然还有近似优化的QAOA算法等。一旦量子计算机研制成功，那么，这些问题将会得到巨大的加速。例如，当问题规模大到一定程度，用量子计算机进行搜索将会极大的提高搜索效率。

（3）最重要的一点，是应用的驱动！世界上第一台电子计算机的出现是为了加速曼哈顿工程和服务于二战。就以人工智能领域为例，神经网络模型几乎在世界上第一台电子计算机ENIAC问世不久就被提出，而人工智能的几次高潮都是由大公司的广告效应所推动的：一次是深蓝，一次是AlphaGo。这两个为机器学习推广的广告效应赚足了全球人的眼球。而量子计算机也需要应用的驱动。虽然量子计算在金融、化学、智慧城市、区块链、人工智能等诸多领域有着巨大的应用潜

力，但是由于目前还没有真正能够投入实用计算的量子计算机，以及真正只有量子计算机能够完成而现有高性能计算机不能完成的贴近于人们现实生活中的例子。试想一下，推动深度神经网络发展其实在于算力的不断提升，而量子计算这种具有巨大计算潜力的新型计算模型，或许有可能带来一次新的科技革命。

4 拟态防御技术在量子时代的挑战

随着量子时代的到来，如果攻击者利用量子计算机强大的并行处理能力来实时穷举所有攻击的可能性，将会对拟态防御技术造成巨大的安全威胁。拟态防御之所以能够抵御已知与未知漏洞攻击，其核心的技术要素在于差模表决，即面向同一攻击方式，对所有变体都能进行攻击。下面从暴力破解攻击、DDoS攻击、漏洞快速挖掘三个方面对拟态防御技术在量子时代面临的挑战进行分析。

4.1 暴力破解攻击

拟态防御技术是一种打破传统攻防平衡的新型防御手段。为了提升系统的安全性，必然会以一定的成本提升及性能损耗为代价。在攻击者攻击手段不断丰富和变化的当下，只有通过在用户端引入动态性、冗余性及异构性等，才能够使得攻击者的攻击成本增加。就像密码防御一样，现有的密码防御核心在于攻击者的算力跟不上密码设定的长度，一旦攻击者具有极其强大的算力，必然会使得密码防御系统失效。例如，地址空间布局随机化技术（Address Space Layout Randomization，ASLR）[24] 能够一定程度上抵抗内存攻击，通过引入地址随机化使得攻击者难以精确构建出有效的shellcode。然而，随后攻击者设计了相应的堆喷射（Heap Spray）攻击及基于返回地址攻击（Return-Oriented Programming，ROP）。这两种攻击方法通过暴力穷举以及暴力搜索现有代码来构建出有效的漏洞利用程序。

在拟态防御系统中，如果攻击者能够对所有的变体实施攻击，那么，就有可能构造出一种攻击来对所有的变体攻击生效。在表决端这些被攻击的变体就有可能呈现出同样的输出，从而隐藏了攻击者的攻击效果。即使在系统防御中引入了动态性和用于防止差模逃逸的表决模块，量子计算机通过强大的算力，也能够快速地定位并构造出一种攻击方式，使得不同的执行体在攻击载荷下展现出相同的异常行为，从而躲过表决器的裁决。

4.2 DDoS攻击

DDoS攻击（Distributed Denial of Service）[25]，其前身为DoS攻击。DoS攻击（Denial of Service，DoS）是指无法正常的使计算机或网络给用户提供服务或资源。而DDoS是指利用Internet上已知设备与系统漏洞，攻占联网主机，使其成为攻击者的代理，随后通过发送指令操作联网主机向目标主机和网络发起DoS攻击，使系统资源和网络带宽被大量消耗，从而导致系统或网络瘫痪。由于DoS攻击往往是由攻击者所精心构造的畸形请求，在形态上与正常的数据包相差不多。

对于拟态防御技术来说，可以通过引入动态、冗余、异构等特性，防止攻击者出现共模逃逸的情况。而DoS攻击在目前以服务为中心的云计算环境下是较为常用的攻击方式。因此，在输入代理端，攻击者一旦拥有了极为强大的算力，将有可能构造出足够多的能够躲避DoS防火墙检查的合理攻击请求，而这些数量极大的合理请求将会使得输入代理段无法满足正常的用户请求。在这种情况下，拟态防御技术对于拒绝服务攻击（DoS）并没有提供较好的解决方案。

4.3 漏洞快速挖掘

目前，已经有不少的自动化漏洞挖掘工具。漏洞挖掘本质上是通过尽可能穷举程序的外部输入，来发现系统漏洞。而量子计算机强大的算力，将会极大地缩短Fuzzing的过程，大大提升Fuzzing的效率。

虽然拟态防御技术对于单个0-Day漏洞具有较强的防御能力，但是，一旦软件系统端出现千疮百孔的漏洞，将会使得攻击者拥有更为多样的攻击手段和方法，再次打破攻防不平衡的状态，从而使得攻防再次出现“易攻难守”的局面。

5 拟态防御技术在量子时代所面临的机遇

与经典计算机相似，攻击者与防御者双方之间的竞争本质上在于算力的竞争。在拟态防御技术中的关键技术上配以对应的量子算法，将能够极大地提升拟态防御系统的性能，从而能够解决

拟态防御对于复杂的攻防对弈问题。这里从软件安全、输入代理、表决器三个方面对拟态防御技术在量子时代面临的机遇进行分析。

5.1 软件安全

一直以来，软件安全（Software Security）是攻防两方之间的博弈。在软件保护方面，防御者通常采用加壳、加密、指令替换等被动方法来增加软件攻击的难度。但攻击者仅需要找其中一个漏洞，就能利用所有漏洞系统进行攻击，而对于防御者来说，阻止所有被利用的漏洞是非常困难的。因此这种软件保护方法并不是有效的。

拟态防御技术下的软件保护，常用的方法是利用多样化编译生成变体集合[26]。在多样化编译系统中，量子计算机能够提供更为强大的算法，促进多样化变体的生成，增加软件运行时状态的动态性和不确定性，使得攻击者不能知晓分配到特定用户变体的内部结构，从而对软件保护起促进作用。

5.2 输入代理

在DHR结构示意图（图1）中，输入代理和5.3节提到的多模表决器被称作拟态括号（Mimic Bracket，MB），存在于IPO模型之间（如图3所示）。

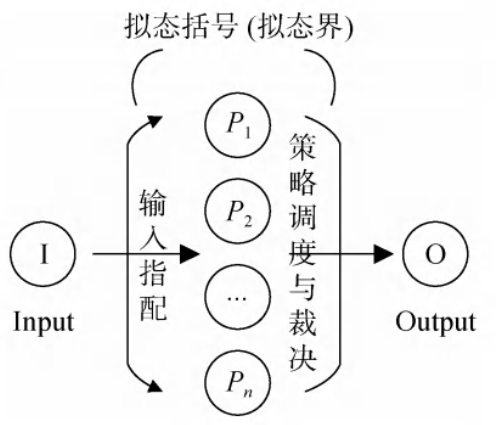


图3 IPO模型图[5]

输入代理具有输入指配功能，可以将输入转发给执行体集A_1，A_2，...，A_n。其中A_1，A_2，...，A_n等同于图3中的P_1，P_2，...，P_n。当在输入代理中面临DoS攻击时，可通过量子计算机的天然并行性及强大的算力，有效地对所有合理化输入分配对应的执行体进行请求处理。量子计算机之所以相较于经典计算机展现出其特有的“量子优越性”，正是由于其具有潜在的强大并行处理能力。例如，在进行大规模检索时，能够将经典计算机中搜索的时间复杂度O（N）缩短至O（$\sqrt{N}$）。因此，在面临DoS等攻击时，量子计算机能够极为容易地对经典计算机的DoS请求给出相应。

此外，将拟态防御技术与量子计算技术结合，能够更快地扫描输入代理节点，排查可能存在的安全漏洞，有助于对安全漏洞进行及时清理。同时，对所有输入，量子计算机都能给出有效地合理输出。

5.3 表决器

作为输出消息的必经通道，表决器是将执行体提交的输出矢量进行表决，从而得到系统的输出。在拟态防御中，表决器采用多模裁决算法，其结果具有叠加性。由于处理空间和算法能否完全独立或绝对相异是无法判定的，所以，当多模输出矢量比对出现不合规状况时，就表明存在攻击逃逸的可能。

工程实现上，为了达到攻防对抗复杂性的要求，拟态防御需要依照不同的情况选择对应的输出表决策略[27]。而量子计算机利用量子算法可以穷举所有可能出现的情况，有利于更快的依据出现的情况选择对应的输出表决策略，缩短判决时间。同时，在多模裁决算法上，量子计算也能发挥强大的效力。例如多模裁决方式中的带权重算法，量子计算机可以针对不同情况进行分析，在极短时间内分配对应的权重值，从而增强拟态防御的性能。

6 结束语

本文就现今量子时代的网络空间安全问题，简要介绍拟态防御技术和量子计算机的原理。从量子计算机发展现状中，分析了量子计算机芯片现今的发展形势；对2019年Google验证的量子霸权工作中进行简要介绍，分析了在量子霸权工作中的误差数值化评估方法；同时介绍量子计算机发展中存在的问题。

本文在量子计算机快速发展的态势下，对拟态防御技术所面临的巨大挑战进行了分析，最后对量子计算技术在拟态防御技术的应用的潜在应用价值进行论述。量子计算技术与拟态防御技术，都是当下两种对传统技术具有颠覆性的意义的重

要技术体系。如果在未来能够有效结合，有可能会彼此促进其发展并发挥巨大效力。

参考文献:

[1] 罗兴国,仝青,张铮,邬江兴. 拟态防御技术[J]. 中国工程科学，2016,18(06):69-73.

[2] octopus""Mimic. Wikipedia, the Free Encyclopedia. https://en. wikipedia. org/wiki/Mimic_ octopus.

[3] Voas J, Ghosh A, Charron F, et al. "Reducing uncertainty about common-mode failures."In Proc. IEEE Symp. Software Reliability Engineering (SRE'97)，pp. 308-319，1997.

[4] 张杰鑫,庞建民,张铮. 拟态构造的Web服务器异构性量化方法[J]. 软件学报,2020,31(02):564-577.

[5] 邬江兴. 网络空间拟态防御研究[J]. 信息安全学报，2016,1(4):1-10

[6] BenioffPaul. Models of Quantum Turing Machines. 1998，46(4-5):423-441.

[7] P. W. Shor，“Algorithms for quantum computation: discrete logarithms and factoring, " in Proceedings of the 35th Annual Symposium on Foundations of Computer Science，pp. 124-134，IEEE Press，Los Alamitos，CA，1994.

[8] GroverL. K. “A fast quantum mechanical algorithm for database search，”in Proceedings of the 28th Annual ACM Symposium on the Theory of Computation，pp. 212-219，ACM Press，New York，1996.

[9] 李盼池,李士勇. 一种Grover量子搜索算法的改进策略[J]. 智能系统学报,2007(01):35-39.

[10] D'ARIANO, MauroGiacomo; PRESTI, LoPaoloplacido. 8 Characterization of Quantum Devices. In: Quantum State Estimation. Springer, Berlin, Heidelberg，2004.
p. 297-332.

[11] 王腾辉. 超导量子比特与绝热快速捷径在量子模拟和量子门中的应用[D]. 浙江大学,2018.

[12] 孔飞. 基于金刚石固态自旋体系的量子模拟实验研究[D]. 中国科学技术大学,2018.

[13] Mark D. Bowdrey, Daniel K. L. Oi，Anthony J. Short，et al. Fidelity of single qubit maps. 2002，294(5):258-260.

[14] HuangW. , YangC. H. , ChanK. W. , et al. Fidelity benchmarks for two-qubit gates in silicon. 2019，569(7757):532-536.

[15] https://quantumcomputingreport. com/scorecards/.

[16] https://physics. aps. org/articles/v7/119.

[17] "5-qubit backend: IBM Q team, "IBM Q 5 Tenerife backend specification v1. 3. 0，(2018). ”https://ibm. biz/ qiskit-tenerife，Last Accessed: 2018-11.

[18] "16-qubit backend: IBM Q team, "IBM Q 16 Melbourne backend specification v1. 0. 0，(2018). ”https://ibm. biz/ qiskit-melbourne，Last Accessed: 2018-11.

[19] CROSS, Andrew W. , et al. Validating quantum computers using randomized model circuits. Physical Review A，2019，100. 3:032328.

[20] Arute, Frank. Et al. , Supplementary information for "Quantum supremacy using a programmable superconducting processor" arXiv: 1910. 11333. Nature, Vol 574，505(2019).

[21] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing，STOC '11，pages 333 – 342，New York，NY，USA，2011. ACM.

[22] PednaultEdwin; John A. Gunnels；Giacomo Nannicini；Lior Horesh；Thomas Magerlein；Edgar Solomonik；Robert Wisnieff "Breaking the 49-Qubit Barrier in the Simulation of Quantum Circuits". (2017).

[23] Harrow Aram W, AvinatanHassidim, SethLloyd. Quantum algorithm for linear systems of equations. . 2009，103(15):150502.

[24] 韩万军,王震宇. Windows平台下地址空间分布随机化技术研究及实现[J]. 计算机应用与软件,2011,28(04):117-119+158.

[25] 魏玉人,徐育军. DDoS攻击及防御技术综述[J]. 软件导刊,2017,16(03):173-175.

[26] 庞建民,张宇嘉,张铮,邬江兴. 拟态防御技术结合软件多样化在软件安全产业中的应用[J]. 中国工程科学,2016,18(06):74-78.

[27] 邬江兴,网络空间拟态防御导论(上册)[M]. 北京:科学出版社，2017. 12.

基于脑电信号的情感识别

李文强[1]，高彦钊[2]，陶常勇[1]

[1]天津市滨海新区信息技术创新中心， 天津， 300457；

[2]战略支援部队信息工程大学， 郑州， 450001

摘　要：针对脑电情感信号分类准确率低、情感分类类别少的问题，利用Deap情感数据库，根据心理效价和唤醒度的等级对不同情感进行划分，对沮丧和压力等五种情感状态进行研究。利用小波变换对脑电情感信号进行分解与重构，提取了脑电信号子频带中的线性特征（波动指数）和非线性动力学特征（样本熵，近似熵，排列熵和Hurst指数），提出基于反馈原理的加权情感特征融合方式来构建特征工程，并引入计算效率更快的孪生支持向量机（TWSVM），设计出OVO-TWSVM多分类模型进行情感分类，平均分类识别率达到了88.2%，证明了所提方法的有效性。

关键词：脑电信号、情感识别、特征提取、孪生支持向量机

Emotion recognition based on EEG signals

LI Wenqiang[1], GAO Yanzhao[2], TAO Changyong[1]

1.Information Technology Innovation Center of Tianjin Binhai New Area, Tianjin, 300457, China;

2.PLA Strategic Support Force Information Engineering University, Zhengzhou, 450001, China

Abstract: Aiming at the problems of low accuracy of EEG emotion signal classification and few emotion classification categories, the Deap emotion database is used to classify different emotions according to the level of psychological valence and arousal, and five emotional states such as depression and stress are studied. Using wavelet transform to decompose and reconstruct the EEG emotional signal, extract the linear features (fluctuation index) and nonlinear dynamic features (sample entropy, approximate entropy, permutation entropy and Hurst exponent) in the sub-bands of the EEG signal. The weighted emotional feature fusion method based on the feedback principle is used to construct feature engineering, and the faster computing twin support vector machine (TWSVM) is introduced, and the OVO-TWSVM multi-classification model is designed for emotional classification. The average classification recognition rate reaches 88. 2%. Proved the effectiveness of the proposed method

Key words: EEG; Emotion recognition; Feature extraction; TWSVM

1　引言

21世纪，被人们称为“脑科学时代”，脑科学带来的研究热潮正遍布世界各地。而情感作为人脑的高级功能，在人们的理性行为和日常生活中起着重要作用，与我们息息相关[1]。随着现代医学、心理学、计算机领域以及人工智能领域的发展和崛起，使得智能分析识别不同的情感状态变成了可能，这不仅对基础的科学研究具有非常重要的意义，也具有广阔的应用前景和巨大的经济价值[2]。

目前，对于情感分析的分类方法已有一定的规模，一般我们在研究过程中经常使用的方式就是在研究生理特点的基础之上对人的各种情绪差异进行对比分析，例如通过人的面部表情、行为动作和语音语调等外部生理特征来进行情感分析，但是这种研究方法易导致情绪识别的精度有所下降[3]。因为人的面部表情、声音等又容易受到自己的主观控制，与这些外在表现出来的生理信号特征相比，脑电信号存在于人们的中枢神经系统

基金项目：国家核高基重大专项基金资助项目（No.2017ZX01030301）

中，能够体现出不同时刻的差异，因此，与情感的关联性远远超过其他信号。基于脑电信号的情感识别有着非常广阔的应用前景和研究价值，越来越多的研究学者投入到了这项工作中。

Rizon等人利用小波变换中三种不同的"db8"，"sym8"和"coif5"小波基函数来提取脑电信号的统计特征，对常见的恐惧、惊讶、快乐和中性情感进行分类识别，最高的分类准确率达到了83.04%[4]。Duan等人对脑电信号进行了傅里叶变换，提取了微分熵和能量谱特征来对常见的正向情绪和负向情绪进行分类，结果表明，微分熵作为情感特征比利用能量谱特征所得到的分类准确率更高，也证明了其可以用于情感分类识别的研究中[5]。Li等人通过对EEG进行分帧处理，在频带能量的基础上计算了比值不对称性和差分不对称性作为大脑的左右不对称脑电特征，在愉悦度上进行二分类，平均准确率分别为76.67%和78.44%，证明了能量不对称性可以用来区分不同的脑电情感状态[6]。Chen等人采用经验模态分解对脑电信号处理，计算了其中前四个本征模式函数的近似熵，将其用于脑电信号的学习和识别。对常见的四种情感的平均分类准确率达到了83.34%[7]。杨默涵等人采用适应性更强的总体经验模态分解对脑电情感信号进行处理，得到了其中最重要的本征模态函数，再结合希尔伯特变换提取了脑电信号中的边际谱和瞬时能谱这两个特征指标，在9名被试人员中，这两个特征下的平均准确率达到了82.74%[8]。Mert等人将多元经验模式分解应用到脑电情感信号的分析中，在得到的多通道固有模态函数上提取多种时域和频域的情感特征，采用K近邻算法在唤醒度和愉悦度的分类准确率分别为51.01%和67%[9]。Pane等人在γ和β频带的时域和频域脑电特征基础上，分别比较了基于规则分类的RIPPER算法和决策树，在情感分类方面，基于规则的分类模型表现更好，通过RIPER算法对放松的情感状态达到81.64%的分类准确率[10]。汤明宏在对脑电信号分解与重构时引入了双树复小波变换，采用非线性动力学中的分形维数来表示不同的脑电情感状态，支持向量机被用来作为分类器，对平静、悲伤和高兴三种情感状态的平均识别率达到了82.5%[11]。闫梦梦等人利用共同空间模式（Common spatial pattern， CSP）对脑电情感信号进行线性投影，以提取脑电信号的空域情感特征，对积极、中性和消极3中情感类型的分类准确率为87.54%[12]。

综上所述，基于脑电信号的情感识别问题很大程度上依赖于传统的模式分类方法，识别效果的好坏关键在于我们所设计的脑电情感特征，以及如何量化不同情感特征与情感状态的相关性来构建情感特征工程，这仍需要进一步的探究。基于上述问题，本文提出采用基于反馈原理来进行加权脑电情感特征融合。采用小波变换对脑电情感信号进行四级分解，提取了脑电信号中子频带中的线性特征（波动指数）和非线性动力学特征（样本熵，近似熵，排列熵和Hurst指数），单一的情感特征仅能体现出部分情感信息，并且不同属性特征对于情感分类识别的贡献不完全相同。因此，利用基于反馈原理进行加权特征融合，和简单的组合方式相比，加权特征融合这种方式更能发挥出不同脑电特征所具有的情感信息特性，使相关性越强的特征得到更大的权重，并引入计算效率更快的孪生支持向量机，设计出多分类孪生支持向量机分类器，在β频带上的平均情感识别率达到了88.2%，证明了所提方法的有效性。

2 数据预处理

2.1 实验数据

Deap[13]数据库中主要采集了三十二名身体健康参与人员的脑电数据，在收集样本数据的过程中每个被试人员采用了32导AgCl电极。此外，还有八个电极用于收集外周的其它人体生理信号。在Deap数据库中，每位受试者每次实验信号采集过程如表1所示。

相关研究发现在情绪处理中大脑的额区和中央区的作用很重要，本文研究时采集的脑电信号数据主要来源于此区域的多个电极，分别是FP1、FP2、F3、Fz、F4、F7、F8、C3、C4。Deap数据库主要运用了效价和唤醒度所构成的二维情感模型，该模型可以把人类的情感划分为两种对应极端，一端为正极，还有一端为负极，当情感处于正极就意味着是正性情感，能够为人带来良好的比较愉悦的内心感受；而情感处于负极也就意味着是负性情感，也就是给人带来不好的情感感受，比如说：焦虑、暴躁等。本文选择五种常见的情

表1　Deap数据收集表

顺序	内容
1	把电极放在适当的实验位置,通过铃声来通知开始实验。
2	接着调整基线约2min,直到符合要求(主要的目的就是为了使受试人员能够保持放松的心情)。
3	显示两秒的音乐来提示实验具体的进程。
4	收集三秒钟的基线数据信息。
5	播放一分钟的音乐,并且详细的记录这个时间段的数据信息。
6	对收集到的数据信息进行统计整理,并且对各个相应的指标进行全面的评估。

感：轻松、沮丧、快乐、压力和平静。考虑到被试人员在观看视频时需要一定的反应时间，而且在最后阶段难免会出现倦怠感，因此，除去了数据中前23秒和后20秒的时间节点，仅保留了中间20秒的数据，共2560个时间节点。

2.2　EEG预处理

在情感识别研究中所用到的脑电数据，它是一种非平稳、无规律的信号。我们要对情感进行分类识别，不仅要分析整体的频率特性，对于局部时间内的特性也要进行了解，此时，通过传统的傅里叶变换和短时傅里叶变换方式并不能达到这种目的。小波变换克服了上述两种变换在信号处理中的缺点，有“数学显微镜”之称，是对脑电情感信号进行变换的理想工具，具备很强的自适应性特点，它能够提供一种变化的“时间-频率”窗口。脑电情感信号经过每一次的小波变换后可以获得脑电信号中的低频率和高频率两部分，也被称为“近似分量”和“细节分量”。在该层分解中得到的“近似分量”又可以分为低频和高频两部分，以这种方式进行多层分解，这样就达到了对原始的脑电情感信号多分辨率分析。如图1所示：

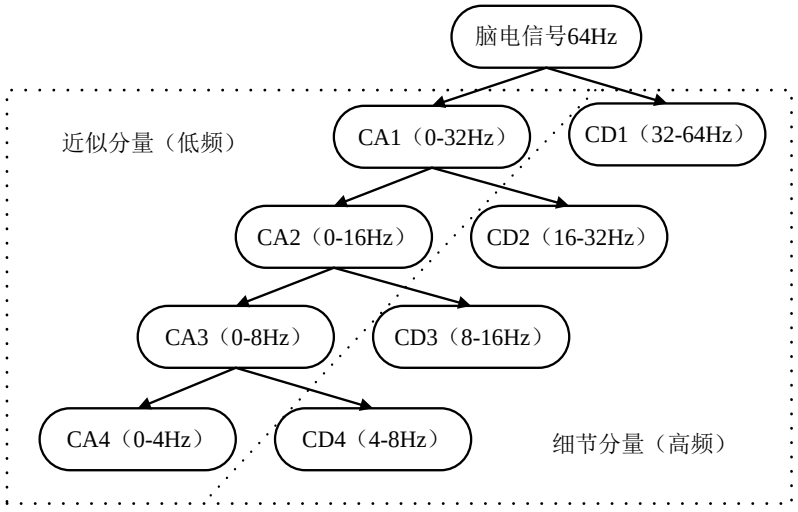


图1小波分解过程

本文的实验所用的是Deap脑电情感数据库，该数据库中的脑电情感信号是经过预处理之后得到的，其采样频率为128Hz，由奈奎斯特抽样定理，可检测0-64Hz范围内的脑电情感信号。脑电情感信号经过4层小波分解后，可近似得到脑电信号中的5个节律波的信号：δ节律（0.5-3Hz）、θ节律（4-8Hz）、α节律（9-13Hz）、β节律（14-30Hz）、γ节律（31Hz以上），分别对应图1中的近似分量（CA4）、细节分量（CD4）、细节分量（CD3）、细节分量（CD2）和细节分量（CD1）上。对Deap数据库中的第一个被试人员的FP1导联在第一个刺激素材下的前15秒的脑电情感信号进行4级分解，如图2所示：

3　脑电情感特征提取

脑电情感特征的提取在基于脑电信号的情感分类实验中起着重要作用，不同的脑电特征携带

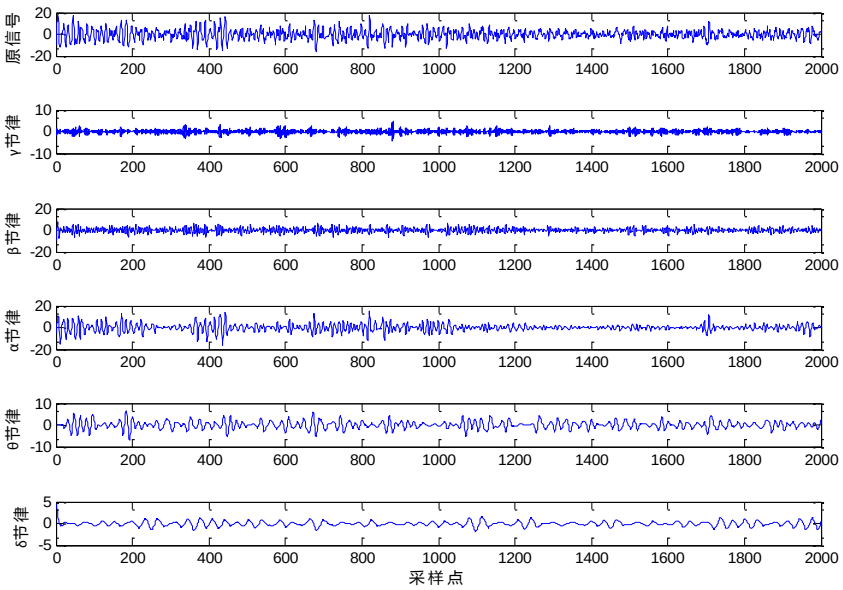


图2　5个子频带的分解图

着不同属性的情感信息，而有效的特征融合方式和选择合适的分类方法往往能够提高情感识别率。

3.1　波动指数

脑电情感信号作为一种非平稳的信号，早期的研究学者们发现虽然其中包含了随机信息，但也会存在确定性的线性成分，不同的情感状态下的脑电信号的振幅会产生不同的波动变化，为此，可以计算波动指数这一特征来体现脑电信号的强度变化，如公式（1）所示：

$$F_i(n)=\frac{1}{M}\sum_{j=1}^{M-1}\left|a_i(j+1)-a_i(j)\right| \tag{1}$$

式中，a_i为脑电情感信号小波变换后第i层的幅值，M为脑电序列的长度，取2560，通过计算该时间段内的幅值的差值总和的平均值作为一个脑电情感特征。

3.2　近似熵

为了度量复杂时间系统的不规则性及不可预料性，可以通过计算近似熵（Approximate Entropy，ApEn）这一非线性动力学指标的值来反映不同情感状态脑电信号的复杂程度。ApEn的值越大，则该时间段内的脑电信号越复杂，反之亦然。ApEn可以理解为当数据的维度变化时，其表示产生新模式的概率大小，在实际的计算中，给定脑电情感信号的序列长度为N，m和r是计算过程中的两个参数，代表的是嵌入维数和相似容限值。统计新生成的新向量$X(i)$与$X(j)$中各对应数据点的差值，取其中的最大值为d_{ij}，并统计距离d_{ij}小于或等于r的个数记为num，并计算num与序列总数的比值$C_i^m(r)$，将得到的$C_i^m(r)$取对数，并计算其对于所有i的均值$\phi^m(r)$：

$$\phi^m(r)=\frac{1}{N-m+1}\sum_{i=1}^{N-m+1}\ln C_i^m(r) \tag{2}$$

将m值增加1，对上述过程再次进行计算，得到$C_i^{m+1}(r)$和$\phi^{m+1}(r)$：

$$C_i^{m+1}(r)=\frac{num}{N-m}\quad,\mathrm{i}=1,2,\cdots,N-m \tag{3}$$

$$\phi^{m+1}(r)=\frac{1}{N-m}\sum_{i=1}^{N-m}\ln C_i^{m+1}(r) \tag{4}$$

根据$\phi^m(r)$和$\phi^{m+1}(r)$，可得近似熵为：

$$ApEn(m,r)=\phi^m(r)-\phi^{m+1}(r) \tag{5}$$

在本文中的脑电序列的数据点N=2560，通常将m的值定为2，r的值取脑电情感信号的标准差的0.2倍左右。将近似熵作为一个脑电情感特征，主要是因为其具有较强的抗干扰能力，所呈现出的分析结果要好于基本的方差、标准差等一般统计特征，另外，在计算过程中不需要过多的数据点就能得到对脑电情感信号的估计值。

3.3　样本熵

样本熵（Sample Entropy，SampEn）与近似熵二者的主要思想和实际意义在很大程度基本一致，但是从后者的计算过程中可以发现其在产生新序列信息的概率的同时，也对自身的数据进行了比较，这必然会产生误差。样本熵对其计算过程中

的一些步骤做了必要的改进，其改进计算过程如下：

Step1. 设定参数相似容限 r，统计所有 $d_{ij}<r$ 的总数记为 num，然后计算 num 与距离总数 N-m-1 的比值，排除自身的比较，记为 $B_i^m(r)$：

$$B_i^m(r)=\frac{num}{(N-m-1)+1}=\frac{num}{N-m} \tag{6}$$

Step2. 所有的 i 及对应的 $B_i^m(r)$ 求平均值 $B^m(r)$：

$$B^m(r)=\frac{1}{N-m}\sum_{i=1}^{N-m}\ln\left(B_i^m(r)\right) \tag{7}$$

Step3. 将时间序列的维度变为 m+1 维，重复上述计算的过程，可得 $B^{m+1}(r)$：

Step4. 根据 $B^m(r)$ 和 $B^{m+1}(r)$，可得样本熵 $SampEn$，即：

$$SampEn(m,r)=-\ln\left[B^{m+1}(r)/B^m(r)\right] \tag{8}$$

在本研究的实验中，脑电情感信号的时间点数 N=2560，m 的值取 2，r 选定为 0.2std（std 为脑电情感信号的标准差）。相较于 ApEn，SampEn 在计算过程中消除了与自身数据的比较的缺点，另外，改变两个参数 m 和 r，脑电情感信号的样本熵值的大小关系并不会改变，体现出良好的一致性。

3.4 排列熵

排列熵（Permutation entropy， PE）也是一种表征时间序列复杂性以及规律性的非线性动力学指标，对重构的子序列之间的复杂度进行求解时，将排列的计算思想融入其中，排列熵的计算流程如图 3 所示：

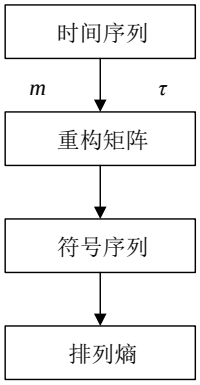


图 3　排列熵计算过程

设脑电情感信号的时间序列为 $X(i)=\{x(1),x(2),\cdots,x(n)\}$，对 X 中的每个元素 $x(i)$ 利用延迟坐标的方式进行相空间重构，将每个重构分量按照升序的方式进行排序，此时，各个重构后分量皆可以获得相应的新的符号序列，计算得到每种符号序列出现的概率为：$P_1,P_2,\cdots,P_k$，则由 Shannon 熵定义，时间序列 $X(i)$ 的排列熵为：

$$H_p(m)=-\sum_{j=1}^{k}P_j\ln P_j \tag{9}$$

当 $P_j=1/m!$ 时，排列熵取最大值 $\ln(m!)$。为了方便表示，往往都将 $H_p(m)$ 作归一化处理：

$$0<H_p(m)=H_p(m)/\ln(m!)\leqslant 1 \tag{10}$$

本文在计算排列熵时，m 取 5，τ 为 10，排列熵作为一个表征时间序列复杂程度的指标，在脑电情感识别研究中，其值越小，该时间段内的信号越规则，反之亦然。通过排列熵可以有效的反映出脑电情感信号中的细微变化。

3.5 Hurst 指数

Hurst 指数可以很好的对某一时间序列进行趋势判断，经常用于对时间序列中的信息的长程关联性进行揭示，其计算方法有很多，最常用的是 R/S 方法。在该方法中，为了使所有的数据都具有可比性，将值的范围进行了调整，Hurst 指数定义如下：

$$\frac{R(n)}{S(n)}=a\times n^H,\ \mathrm{n}=1,2,\cdots,\mathrm{N} \tag{11}$$

式中，S（n）为标准差，R（n）为极差，Hurst 发现 R（n）/S（n）与 n 之间有指数关系，对上式两边取对数，即可得到 Hurst 指数 H：

$$H(n)=\log\left[R(N)/S(N)\right]/\log(n),\ \mathrm{n}=1,2,\cdots\mathrm{N} \tag{12}$$

Hurst 指数的值在 0-1 之间，在实际的脑电情感研究中，不同的指标值具有不同的含义。当 H=0.5 时，表明脑电情感信号是随机和不相关的，现在的趋势并不会影响未来发展的趋势；当 0.5<H<1 时，意味着脑电情感信号是长期相关的，也就是说，未来的变化会和过去趋势有一定关联；当 0<H<0.5 时，表示脑电情感信号是反持久性的，即未来的总体趋势与过去相反。

3.6 加权特征融合

在脑电情感识别的过程中，情感特征起着关键作用，对相关情感特征进行探究的主要思想是计算对应的量化指标，可以体现出所得到的情感特征与给定类别的关联程度强弱性，不同的情感特征对于关键信息贡献又不同，因此，本文提出了加权特征融合方式，即按照一定的标准对计算

得到的各个脑电情感特征分配相应的权重系数来进行组合。采用这种方法对脑电情感特征进行融合，关键是得到每个特征所对应的权重，计算特征的权重来对脑电情感信号进行重要分析。下面介绍基于反馈的特征相关分析与权重确定方法，步骤如下：

步骤1：基于小波变换分析，选择与情感相关的5个脑电特征。

步骤2：计算第i个脑电特征的脑电情感分类的识别率为a_i：

$$a_i, 1 \leqslant i \leqslant 5 \tag{13}$$

步骤3：基于反馈的原理，利用公式（14）得到每个特征的权重，

$$w_i = \frac{a_i}{a_1 + a_2 + a_3 + a_4 + a_5},\ 1 \leqslant i \leqslant 5 \tag{14}$$

其中，$w_1+w_2+w_3+w_4+w_5=1$。

采用加权特征融合这种方式，可以计算出脑电情感信号的每个特征所对应权重，具有最高权重的情感特征就是所求的特征集合中鉴别性最高的特征，也是对情感分类识别贡献程度最大的。所以可以使用对应的权重来衡量各个脑电情感特征对于情感分类的关联性：权重越大，相关性越强。

4 分类算法

4.1 孪生支持向量机

相较于传统的支持向量机（Support Vector Machine，SVM），孪生支持向量机（Twin Support Vector Machine，TWSVM）的核心思想是构造一对不平行的正、负超平面[14]，如图5所示，通过这种改进，计算时间有了大幅度的缩减，大概为传统支持向量机的四分之一。

设训练样本集是：(x_j^i, y_j)，$i = 1, 2$；$j = 1, 2 \cdots l$，$x_j^i \in R^n\ y_j \in \{1, -1\}$，其中，$x_j^i$为输入的样本数据，$y_j$代表样本的分类标签。TWSVM算法寻找了两个不平行超平面，让满足特定某一类别的样本只能接近其中的一个超平面，两个超平面可表示为：

$$\begin{aligned} f_+(x) &= w_+^T x + b_+ = 0 \\ f_-(x) &= w_-^T x + b_- = 0 \end{aligned} \tag{15}$$

为了得到这对超平面，可以将其转化为一对优化问题的求解来完成，即：

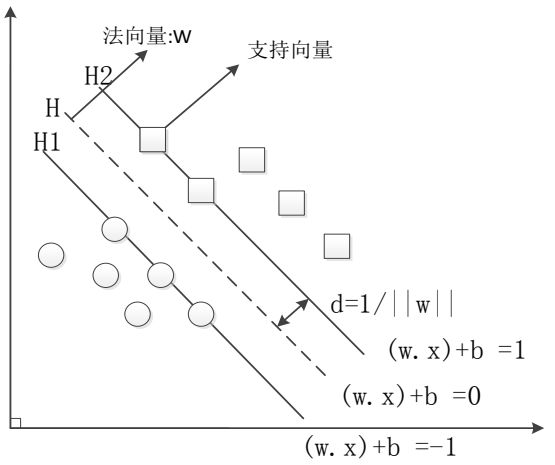


图4 SVM分类示意图

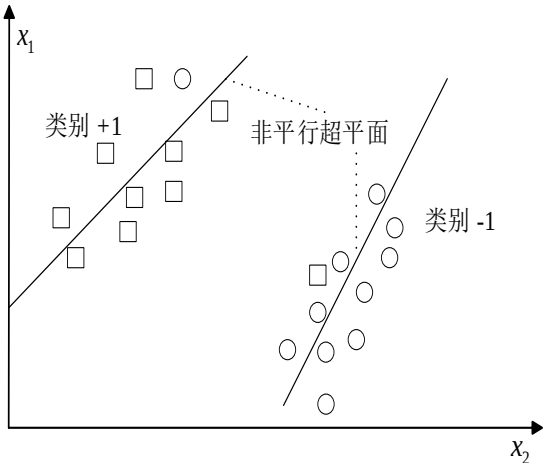


图5 TWSVM分类示意图

$$(TWSVM2)\ \min_{w_+,b_+,q} \frac{1}{2}\left\|X_+ w_+ + e_1 b_+\right\|^2 \quad s.t\ -(X_- w_- + e_2 b_-) + q \geqslant e_2, q \geqslant 0 \tag{16}$$

$$(TWSVM2)\ \min_{w_-,b_-,q} \frac{1}{2}\left\|X_- w_- + e_2 b_-\right\|^2 \quad s.t\ -(X_+ w_+ + e_1 b_+) + q \geqslant e_1, q \geqslant 0 \tag{17}$$

其中，X_+和X_-表示输入数据，e_1和e_2是全1向量。

与传统的SVM类似，引入了拉格朗日乘法进行求解，则公式（16）和（17）的对偶问题为：

$$\max_{\alpha} e_1^T \alpha - \frac{1}{2}\alpha^T \hat{X}_+ (\hat{X}_-^T \hat{X}_-)^{-1} \hat{X}_+^T \alpha \quad s.t 0 \leqslant \alpha \leqslant c_1 e_1 \tag{18}$$

$$\max_{\gamma} e_2^T \gamma - \frac{1}{2}\gamma^T \hat{X}_+ (\hat{X}_-^T \hat{X}_-)^{-1} \hat{X}_+^T \gamma \quad s.t 0 \leqslant \gamma \leqslant c_2 e_2 \tag{19}$$

式中：$\hat{X}_+ = [X_+ e_1]$，$\hat{X}_- = [X_- e_2]$，α和γ分别是拉格朗日乘数。

公式（18）和（19）需要求解矩阵的逆，因

此，在孪生支持向量机中加入了正则因子μ，来克服矩阵的“奇异性”，计算公式（18）和（19）可以得到公式（15）中定义的两个超平面。

$$z_{+} = -(\hat{X}_{+}^{T}\hat{X}_{+} + \mu)\hat{X}_{-\alpha}^{T}$$
$$z_{-} = -(\hat{X}_{-}^{T}\hat{X}_{-} + \mu)\hat{X}_{+\gamma}^{T} \quad (20)$$

其中$z_{+}=[w_{+}, b_{+}]^{T}$，$z_{-}=[w_{-}, b_{-}]^{T}$，这样样本数据的类别由距离两个超平面之间的距离决定，即：

$$class = \arg\min_{k=+,-} \frac{\left|w_k^T x + b_k\right|}{\left\|w_k\right\|} \quad (21)$$

4.2 OVO-TWSVM分类器设计

TWSVM继承和发展了传统的SVM，也是一种进行二分类的分类器，而本文对脑电情感进行了多分类，这就需要构造出多分类器来进行实验。在一般情况下，多分类的SVM模型常利用“一对一”（One-Versus-One，OVO）这种方法来实现，本文将这种方法与TWSVM相结合，构造出多分类的TWSVM模型。设样本类别数为n，OVO策略首先需要在任意的两类情感样本中进行训练得到一个二分类的训练模型，则n分类的问题就需要构造n（n-1）/2个分类器。当对测试集的样本进行分类时，分别采用这n（n-1）/2个模型进行分类，并统计结果中出现各个类别的次数，累计次数最多的类别就是该测试样本的类别。

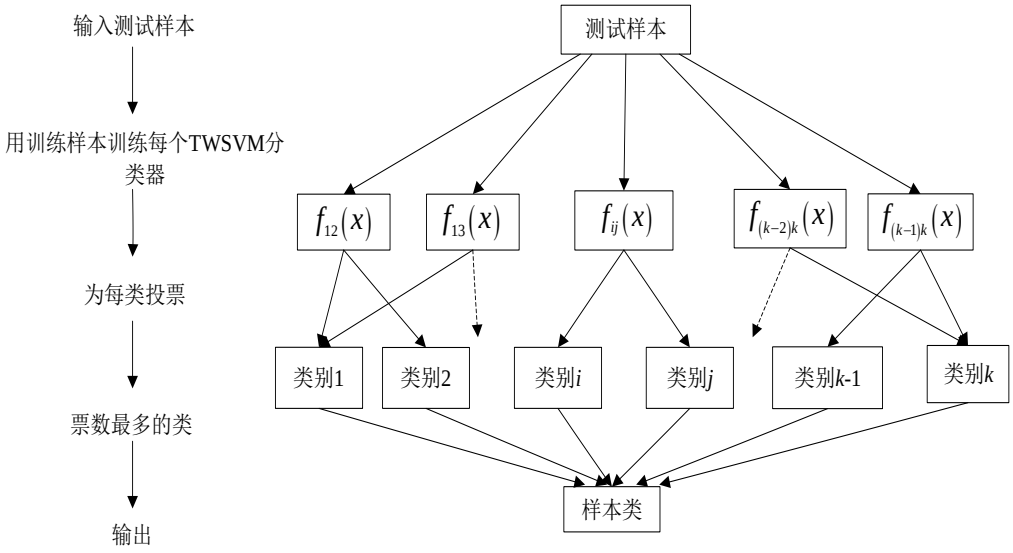


图6　多分类TWSVM原理图

5　实验结果分析

将脑电情感信号的子频带中提取的波动指数、近似熵、样本熵、排列熵和Hurst指数特征依次记为FI、AE、SE、PE和$Hurst$。传统的线性脑电特征组合向量时，默认每个特征前面的系数为1，由公式（22）所示：

$$Feature = [FI, AE, SE, PE, Hurst] \quad (22)$$

利用支持向量机和孪生支持向量机分别以波动指数、近似熵、样本熵、排列熵和Hurst指数为情感特征，轻松，沮丧，快乐，压力和平静五种情感的样本数分别为500个，分别采用SVM和TWSVM进行5折交叉验证实验，在各个子频带上得到的平均总体分类准确率如图7和图8所示：

由图7和图8可以看出，在单个特征下，采用TWSVM在单个特征下的情感分类准确率较SVM的分类准确率都有一定程度的提高。因此，本研究采用TWSVM来对脑电进行情感分类。根据每个特征在TWSVM下的分类识别率，由公式（14）可求得各个脑电特征的权重，如表2所示：

所以，由表2中各个特征的权重系数可得各个频带下的加权特征融合的组合特征，如公式（23）-（27）：

$$Feature_{\delta} = [0.248FI,\ 0.148AE,\ 0.192SE,\ 0.198PE,\ 0.214Hurst] \quad (23)$$

$$Feature_{\theta} = [0.249FI,\ 0.142AE,\ 0.184SE,$$

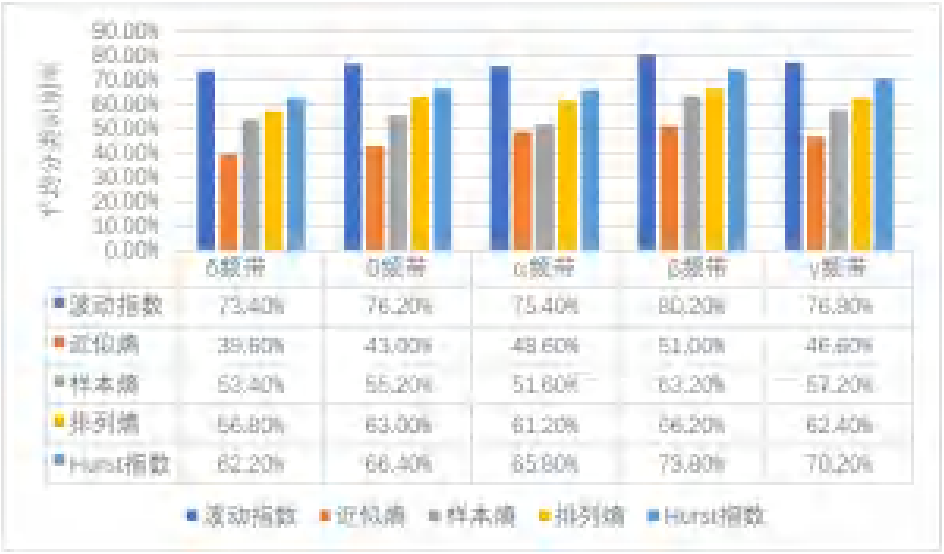


图7　采用SVM各个特征下的平均准确率

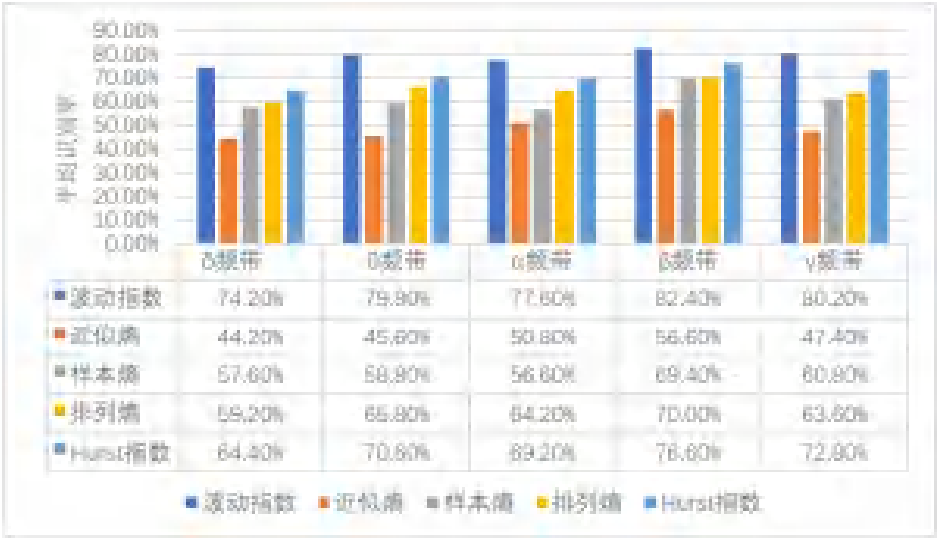


图8　采用TWSVM各个特征下的平均准确率

表2　权重系数分配表

特征	各个频带下的权重系数				
	δ频带	θ频带	α频带	β频带	γ频带
波动指数	0.248	0.249	0.244	0.233	0.247
近似熵	0.148	0.142	0.160	0.159	0.146
样本熵	0.192	0.184	0.178	0.195	0.187
排列熵	0.198	0.205	0.201	0.197	0.196
Hurst指数	0.214	0.220	0.217	0.216	0.224

0.205*PE*， 0.220*Hurst*］（24）

$Feature_{\alpha}$ = ［0.244*FI*， 0.160*AE*， 0.178*SE*，0.201*PE*， 0.217*Hurst*］（25）

$Feature_{\beta}$ = ［0.233*FI*， 0.159*AE*， 0.195*SE*，0.197*PE*， 0.216*Hurst*］（26）

$Feature_{\gamma}$ = ［0.247*FI*， 0.146*AE*， 0.187*SE*，0.196*PE*， 0.224*Hurst*］（27）

利用TWSVM分别对传统的特征组合和加权特征组合两种方式进行情感分类。分类问题中，最常见的评价指标是准确率（*acc*），它能够直接反映出分正确的比例，同时计算非常简单。但是实际的分类问题中，有时整体的准确率（*acc*）较高，但是部分类别的分类效果较差，这时单纯的准确率（*acc*）不能完全作为模型的评价标准。为此，计算每种情感的召回率（*recall*）来衡量分类模型的好坏。

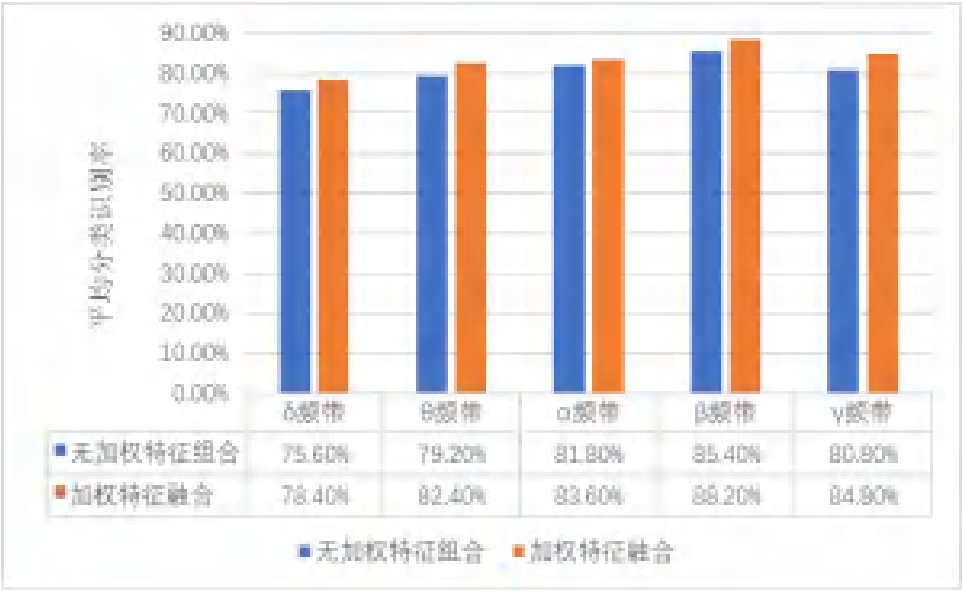


图9　两种特征组合方式在不同频带下的平均准确率

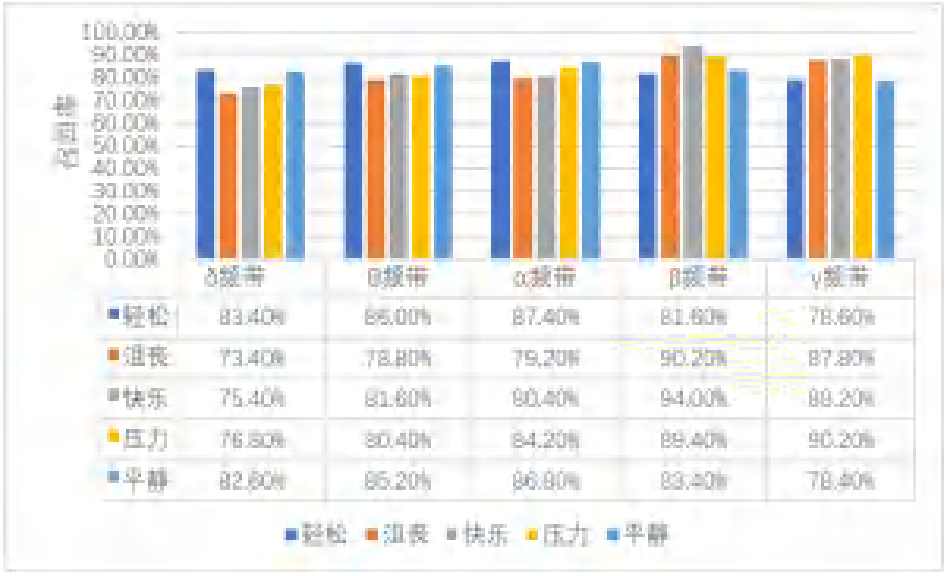


图10　不同频带下的每种情感召回率

由图9和图10可以看出，TWSVM算法下，采用加权特征融合较传统的线性特征组合的平均分类准确率都有一定的提高，并且在β频带下的达到了最高的88.2%的平均准确率。不同的情感状态在不同的频段下的准确率不同，采用加权特征融合的方式，在β频带下，“快乐”情感的识别率最高达到了94.0%，θ频段和α频段中轻松和平静的识别率较高。这是因为在不同情感状态下的脑部活动及反应并不是一种简单的线性系统，若把不同的脑电情感特征进行简单的线性组合时，容易破坏脑电情感特征的原本的非线性结构。使不同的情感特征之间出现负面作用，进而了导致了特征冗余的增加，这样就背离了我们想要充分利用多种脑电情感特征优点的原意。所以，我们需要在研究中创造一种来反应每个情感特征作用的方法。因此，本文提出了基于反馈的加权特征组合方式。通过这种有效方法，使多种脑电特征能够有效的融合在一起，以提升脑电信号情绪识别的分类性能，结果证明了所提出的基于反馈的加权特征组合方式能够充分的挖掘脑电情感信息。

将本文的实验结果同已有的研究成果进行对比，如表3所示，文献［15］中提取了脑电信号中

的组合熵特征，采用KNN进行情感分类，其准确率达到了77.8%，文献［16］提取功率谱密度为脑电情感特征，利用SVM进行二分类，得到的情感识别率为70.1%，文献［17］采用了自编码器SAE和循环神经网络的组合方式，情感分类准确率为79.26%。文献［18］中将卷积神经网络和循环神经网络进行组合，整体分类识别率为73.09%。

表3　已有研究成果与本文的对比

方法	情感类别数	准确率
KNN	2	77.8%
SVM	2	70.1%
SAE+RNN	4	79.26%
CNN+LSTM	2	73.09%
本文方法	5	88.2%

6　结束语

本文在研究的过程中主要使用了 DEAP 数据库中的数据，对沮丧、轻松、快乐、压力和平静五种情感脑电信号进行研究分析。针对目前基于脑电信号情感分类准确率较低且分类类别少，以及如何进行有效的脑电特征融合构建特征工程，提出了基于反馈原理的加权特征融合方式，采用小波变换对脑电情感信号进行四级分解，提取了脑电信号中子频带中的线性特征（波动指数）和非线性动力学特征（样本熵，近似熵，排列熵和Hurst指数），单一的情感特征仅能体现出部分情感信息，并且不同属性特征对于情感分类识别的贡献不完全相同。因此，利用基于反馈原理进行加权特征融合，和传统的线性组合方式相比，加权特征融合这种方式更能发挥出不同脑电特征所具有的情感信息特性，使相关性越强的特征得到更大的权重，并引入计算效率更快的孪生支持向量机，设计出OVO-TWSVM多分类器进行脑电情感识别，在β频带上平均情感识别率达到了88.2%，结果表明，本文所提方法可以帮助人们有效的区分不同脑电情感信号，为早日得到性能完善的脑-机接口提供了重要的参考价值。

参考文献:

[1] Chen G , Zhang X , Sun Y , et al. Emotion feature analysis and recognition based on reconstructed EEG sources［J］. IEEE Access，2020，8：11907-11916.

[2] Yadava M, Kumar P, Saini R, et al. Analysis of EEG signals and its application to neuromarketing ［J］. Multimedia Tools and Applications，2017，76(18)：19087-19111.

[3] Kim K, Kim M, Oh E, et al. A review on the computational methods for emotional state estimation from the human EEG［J］. Computational and mathematical methods in medicine，2013，2013(2)：573-734.

[4] Murugappan M, Rizon M, Nagarajan R, et al. Combining Spatial Filtering and Wavelet Transform for Classifying Human Emotions Using EEG Signals［J］. Journal of Medical and Biological Engineering，2011，31(1)：45-51.

[5] Duan R N, Zhu J Y, Lu B L. Differential entropy feature for EEG-based emotion classification［C］. International IEEE/EMBS Conference on Neural Engineering，SanDiego，California，USA，2013：81-84.

[6] Li Z , Tian X, Shu L, et al. Emotion Recognition from EEG Using RASM and LSTM［C］. International Coference on Internet Multimedia Computing and Service， Nanjing， China， 2018：310-318.

[7] Chen T, Ju S, Yuan X, et al. Emotion recognition using empirical mode decomposition and approximation entropy［J］. Computers & Electrical Engineering，2018，72：383-392.

[8] 杨默涵, 陈万忠, 李明阳. 基于总体经验模态分解的多类特征的运动想象脑电识别方法研究[J]. 自动化学报，2017，43(05)：743-752.
YANG Mohan, CHEN Wanzhong, LI Mingyang. Multiple Feature Extraction Based on Ensemble Empirical ModeDecomposition for Motor Imagery EEG Recognition Tasks［J］. Acta Automatica Sinica，2017，43(05)：743-752

[9] Mert A, Akan A. Emotion Recognition from EEG Signals by Using Multivariate Empirical Mode Decomposition［J］. Formal Pattern Analysis and Applications. 2018，21(1)：81-89.

[10] Pane E S, Hendrawan M A, Wibawa A D,et al. Identifying Rules for Electroencephalograph (EEG) Emotion Recognition and Classification［C］. 2017 5th International Conference on Instrumentation，Communications， Information Technology， and Biomedical Engineering，Bandung，Indonesi，2017：167-172.

[11] 汤明宏. 基于脑电信号的情感识别研究[D]. [硕士论文]. 南京邮电大学，2018.
TANG Minghong. Emotin recognition research based on EEG［D］. [master thesis]. Nanjing University of Posts and Telecommunications，2018.

[12] 闫梦梦, 吕钊, 孙文慧. 基于共同空间模式的情感脑电信号的空域特征提取[J]. 图学学报，2020，41(3)：424-429.
YAN Mengmeng, LV Zhao, SUN Wenhui. Extraction of spatial features of emotional EEG signals based on common spatial pattern［J］. Journal of Graphics，2020，41(3)：424-429.

[13] Theresia DiahK. , FaqihA. , KusumoputroB. . Exploring the Feature Selection of the EEG Signal Time and Frequency Domain Features for k - NN and Weighted k-NN［C］. 2019 IEEE R10 Humanitarian Technology Conference (R10-HTC) (47129)，Depok，West Java，

Indonesia，2019，pp. 196-199.

[14] MiaoY. , LiuF. , LiuY. , et al. Feature Selection Algorithm based on Double-hook Function and TWSVM［C］. 2019 Chinese Automation Congress (CAC)，Hangzhou，China，2019，pp. 4340-4344.

[15] 苏建新. 基于脑电信号的情绪识别研究［D］. [硕士论文]. 南京邮电大学，2015.
SU Jianxin. Emotion recognition research based on EEG signal［D］. [master thesis] Nanjing University of Posts and Telecommunications，2015

[16] FreireBastos-FilhoTeodiano, FerreiaAndre,et al. Evaluation of Feature Extraction Techniques in Emotional State Recognition. IEEE Proceedings of 4th International Conference on Intelligent Human Computer Interaction，Kharagpur，India，2012：27:29

[17] 李幼军, 黄佳进, 王海渊, 等. 基于SAE和LSTM RNN的多模态生理信号融合和情感识别研究［J］. 通信学报，2017，38(12)：109-120
LI Youjun, HUANG Jiajin, WANG Haiyuan, et al. Study of emotion recognition based on fusion multi-modal bio-signal with SAE and LSTM recurrent neural network［J］. Journal on Communications，2017，38(12)：109-120.

[18] Li X, Song D, Zhang P, et al. Emotion recognition from multi-channel EEG data through Convolutional Recurrent Neural Network［C］. IEEE International Conference on Bioinformatics and Biomedicine. Shenzhen，China，2016：352-359.

[作者简介]

李文强（1994-），男，硕士，天津市滨海新区信息技术创新中心算法工程师，主要研究方向为机器学习、深度学习、类脑计算。

高彦钊（1984-），男，博士，战略支援部队信息工程大学助理研究员，主要研究方向为高效能计算、类脑计算。

陶常勇（1982-），男，硕士，天津市滨海新区信息技术创新中心高级工程师，研究方向为拟态计算、高性能计算等。

光纤通信物理层中的内生经典密钥分发技术

张柳明，Adnan　Hajomer，杨学林

上海交通大学 区域光纤通信网与新型光通信系统国家重点实验室，上海 200240

摘　要：本文针对日益重要的光纤通信系统中信息安全的重大需求，提出并验证了光物理层中基于光特征参数波动的内生、经典密钥分发方案，并评估了光物理层中的密钥后处理协议。基于光纤信道的互易性，合法通信双方可共享高度相关的光特征参数波动，并从中提取密钥。另一方面，光纤信道的唯一性可确保密钥的高安全性。理论与实验结果表明，该类密钥分发技术具备高速率、高一致性等特征，对于高速、大容量光纤通信系统中信息的保密传输具有重要意义。

关键词：光物理层、密钥分发、光特征参数波动

Endogenous Classical Secure Key Generation and Distributionin Physical-Layer of Optical Networks

ZHANG Liuming, HAJOMER Adnan, YANG Xuelin

State Key Laboratory of Advanced Optical Communication Systems and Networks,Shanghai Jiao Tong University, Shanghai 200240

Abstract: In this paper, secure key generation and distribution (SKGD) schemes based on the fluctuations of optical features in classical optical physical-layer are proposed and demonstrated, which aims to enhance the data security level in optical fiber communication networks. Moreover, the specific post-processing protocol is evaluated. Due to the channel reciprocity, highly-correlated optical feature fluctuations can be shared between two legal parties, from where secret keys can be extracted. The channel uniqueness guarantees the secrecy of the generated key sequence. Theoretical and experimental results show that such kind of SKGD schemes can provide high key generation rate and high key agreement rate, which is of great significance for secret data transmission in high-speed and large-capacity optical fiber communication systems.

Key words: optical physical layer; secure key generation and distribution; optical feature fluctuations

1　引言

光纤通信具有高速率、大容量、传输距离长等优越性能，在当今信息化社会中的重要性日益显现。目前，全球超过95%的通信业务都由光网络承担。然而，光信息在传输过程中所面临的安全问题也非常严峻，攻击者可通过光纤弯曲耦合法、光束分离法等方式截获光信息[1]。因此，亟需对高速光信息进行加密传输。

根据密码学中的Kerckhoffs准则，一个加密系统的安全性仅取决于密钥的安全性[2]。在当前的光网络安全体系中，密钥分发一般采用公钥算法，如RSA及Diffie-Hellman等。但是，随着近年来量子计算机的飞速发展，基于计算安全机制的公钥算法，面临着被破解的威胁[3]，探索光网络中的全新密钥分发技术迫在眉睫。

另一方面，量子密钥分发（QKD， quantum key distribution）技术基于量子力学原理，能实现无条件安全的密钥分发，得到了广泛的研究。但由于光量子态传输对信道噪声及损耗较为敏感，现阶段的QKD技术在密钥速率及光纤传输距离等关键指标上受到了限制[4]。针对高速、长距离光纤保密通信的客观需求，以下讨论经典光物理层中基于光特征参数波动的内生密钥分发技术。

基金项目：国家自然科学基金（61431009, 61433009, 61571291）；国家重点研发计划（2016YFE0104500）。

2 基于光特征参数波动密钥分发技术的研究现状

近年来，国内外报道了一种经典光物理层中的新型密钥分发技术。该类技术是基于光纤信道特征，在经典光物理层中产生动态的光特征参数波动，由合法通信双方Alice和Bob共享，并从中提取密钥。经密钥后处理后，可在光物理层中直接用于对传输的数据进行加密。由于非法方Eve与合法双方之间关于信道状态的不对等性，她将无法获取光物理层中产生的正确密钥。经理论分析，经典光物理层中的该类密钥分发方案，不受攻击者强大计算能力的威胁，具备信息论层面的高安全性[5-6]。另外，该类密钥分发方案还具有结构简单、兼容于现有光纤通信系统的优势。

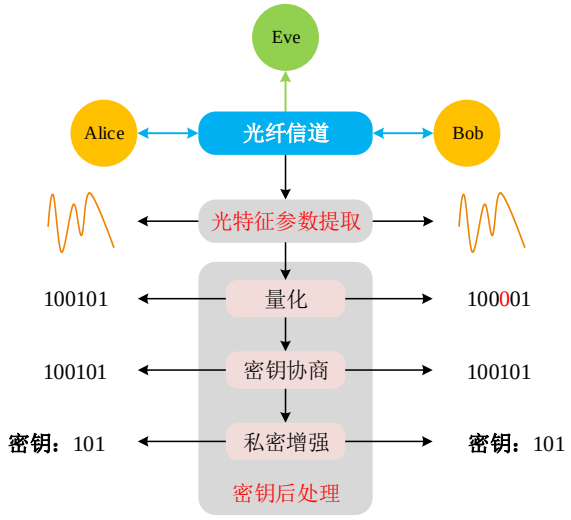


图1 经典光物理层中基于特征参数波动的密钥分发流程图

该类密钥分发技术方案主要包括光特征参数提取和密钥后处理两大步骤，如图1所示。在光特征参数提取阶段，基于光纤信道的互易性，合法通信双方将获得高度相关的光特征参数波动。密钥后处理阶段分为以下三个子步骤：量化、密钥协商以及私密增强。量化的目的是将所接收的模拟特征参数波动转化为初始二进制密钥序列。由于实际光纤信道中的非理想互易性，Alice和Bob量化后的初始密钥比特序列间具有一定的不一致率，可通过密钥协商技术将其降低至0。在密钥协商过程中，合法双方需要进行信息交互，这可能会向攻击者泄露部分密钥信息。因此，私密增强技术用于消除密钥协商阶段泄露的部分密钥信息，以进一步增强密钥的安全性及随机性。

目前针对经典光物理层中基于特征参数波动的密钥分发的研究主要有：

国际上，2013年，俄罗斯科学院的Kravtsov等人提出了基于大尺寸马赫-曾德尔干涉仪（MZI， Mach–Zehnder interferometer）的密钥分发方案，将光相位波动转换为光强波动，并从中提取密钥。该方案中，密钥分发速率为160bps，密钥不一致率为4%[7]。2018年，美国加州大学欧文分校的Zaman等人提出采用偏振模色散造成的光强失真作为密钥熵源，仿真中产生并分发了128个密钥比特，密钥不一致率<10%[8]。2019年，美国普林斯顿大学的Prucnal团队在通信双方间各部署一个完全相同的小型分布式MZI，实现了速率为502bps的密钥分发，密钥不一致率为0.3%[9]。2019年，美国耶鲁大学的Yaron等人利用多模光纤中的模式混合效应，实现了速率为20bps的密钥分发，密钥不一致率为5%[10]。

国内方面，2018年，北京邮电大学的张杰教授团队利用光通信中的误码率随机波动这一特征，实现了速率为400kbps的密钥分发[11]，并于2019年将密钥分发速率提升至了2Mbps，密钥不一致率为2%[12]。2020年，该课题组报道了速率为277 kbps的密钥分发，密钥不一致率为0[13]。

本文利用单模光纤中光波的两个正交偏振分量间相位差的随机波动，实验验证了基于偏振模干涉[14]和基于偏振态斯托克斯参数[15]的密钥分发方案，所得到的密钥分发速率分别为220bps和222bps，密钥不一致率分别为5%和4.5%。进一步通过在光纤信道中部署一高速偏振扰动仪[16]，实现了速率为200kbps的高速密钥分发，密钥不一致率为0，验证了通过对光特征参数施加主动扰动，进而提高密钥分发速率的可行性。最后，评估了光网络中的完整密钥后处理协议[17]。

3 基于偏振模干涉的密钥分发

基于偏振模干涉的密钥分发原理图如图2所示。两个偏振角度分别为α和β的线偏振器通过一段保偏光纤连接，构成一延时干涉仪。通信双方Alice和Bob分别位于该延时干涉仪的两端，相向注入波长相同的连续激光。由于保偏光纤的双折射效应，光波的两个偏振模分量之间的相位差为

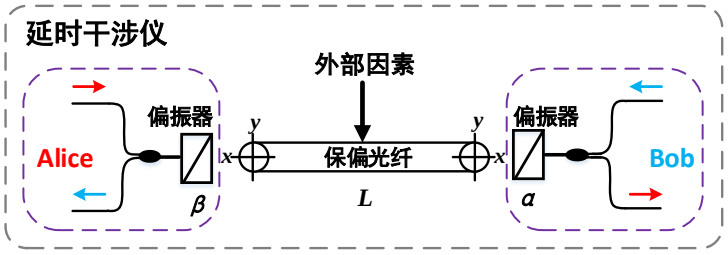


图2　基于偏振模干涉的密钥分发原理图

$$\Delta\varphi = (2\pi/\lambda)\,\Delta nL \tag{1}$$

其中，λ为光波波长，Δn为光纤双折射，L为光纤长度。由于此相位差对外界环境因素的敏感性，包括温度变化、光纤振动等，它将随时间的变化而变化。延时干涉仪可将此相位差波动转换为强度波动，如式（2）所示[18]：

$$\Delta I = \frac{1}{2}\,(1 + \cos 2\alpha\cos 2\beta + |\gamma|\sin 2\alpha\sin 2\beta\cos\Delta\varphi) \tag{2}$$

其中，γ为一互相关系数。Alice和Bob发射的两束激光将经历相同的光纤信道和外界因素，基于信道的互易性[19]，在输出端，两束光波偏振模分量的相位差$\Delta\varphi$将相同。因此，根据式（2），Alice和Bob将共享相同的强度波动，可作为光物理层中的密钥熵源。

图3（a）展示了该密钥分发方案的实验装置图。分别在Alice和Bob端部署一段不等长的保偏光纤，并通过25km长的单模光纤信道连接。延时干涉仪输出的强度波动经光电探测器后，由高速实时示波器记录，并用于密钥后处理。Alice与Bob测得的强度波形如图3（b）所示，可以看出，两个波形之间非常相似，二者的互相关系数超过了0.99，证明了从此强度波动中提取密钥的可行性。此外，密钥从光网络物理层中内生而来，兼容于目前已广泛部署的光通信基础设施。通过波分复用（WDM，Wavelength Division Multiplexing）技术，可在同一光纤信道中同时实现密钥分发与数据传输两大功能，如图3（a）所示。

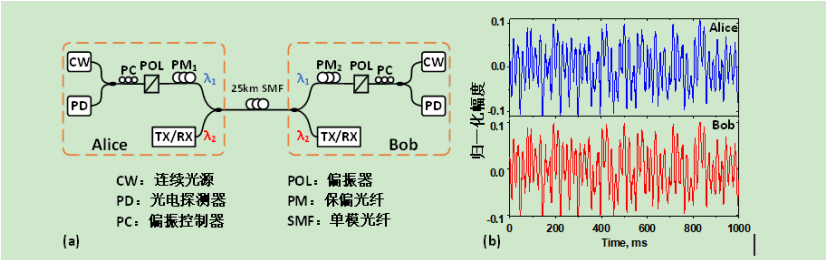


图3　（a）基于偏振模干涉的密钥分发实验装置图；（b）Alice与Bob测得的高度相关的强度波动

在密钥的安全性方面，由于光波的两个偏振模之间的相位差$\Delta\varphi$与光纤的长度L有关，如式（1）所示。因此，当攻击者无法获取两段保偏光纤的精准长度时，她将获得截然不同的$\Delta\varphi$。此外，基于信道的唯一性，攻击者无法复现合法双方间的外界环境因素，也将导致其获得不同的$\Delta\varphi$。因此，她将不能获得与合法方相同的强度波动，从而保障了密钥的安全性。

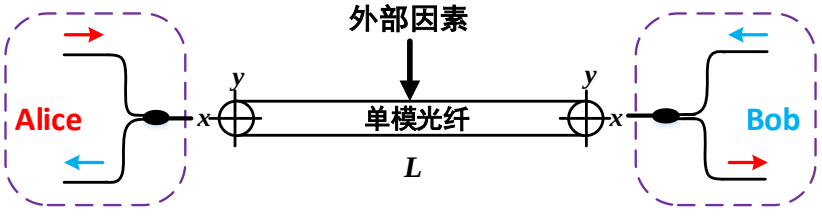


图4　基于偏振态斯托克斯参数的密钥分发原理图

4　基于偏振态斯托克斯参数的密钥分发

与基于偏振模干涉的密钥分发方案不同，基于偏振态斯托克斯参数的密钥分发方案不需要部署额外的光纤，其原理如图4所示。合法双方双向注入波长相同的连续激光至单模光纤信道中，对于任一单模光纤，其内部的光纤双折射效应将导致光波偏振态沿光纤长度的随机演变，如图5所示。由于光纤的双折射分布容易受到外界环境因素（包括温度变化、光纤振动）的随机影响，在输出端，光波的偏振态也将随机波动。根据光纤信道的互易性[19]，合法通信双方将共享此偏振态波动。

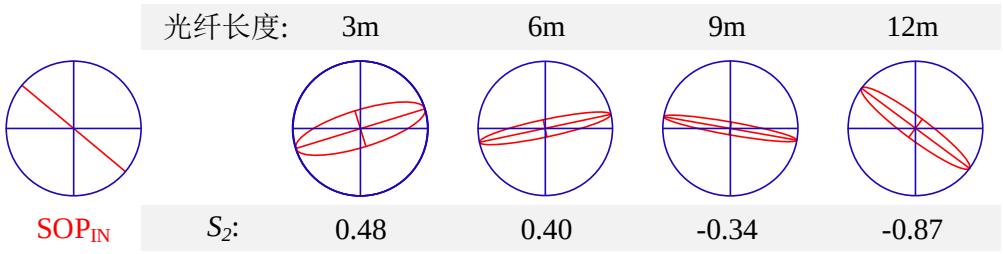


图5　光偏振态沿光纤长度的随机演变

任何一束偏振光都可以用一组斯托克斯参数（S_0，S_1，S_2，S_3）唯一表示：

$$\begin{cases} S_0 = E_{x0}^{\ 2} + E_{y0}^{\ 2} \\ S_1 = E_{x0}^{\ 2} - E_{y0}^{\ 2} \\ S_2 = 2E_{x0}E_{y0}\cos\Delta\varphi \\ S_3 = 2E_{x0}E_{y0}\sin\Delta\varphi \end{cases} \tag{3}$$

其中E_{x0}、E_{y0}分别为光波沿x轴偏振分量、y轴偏振分量的幅值，$\Delta\varphi$仍为两个偏振分量之间的相位差，如式（1）所示。可以看出，S_0、S_1仅与光波两个偏振分量的幅值有关，而S_2、S_3不仅与两个偏振分量的幅值有关，还与它们之间的相位差$\Delta\varphi$有关，且S_2与S_3之间存在着90°的相位差。因此，合法双方可从高度相关的S_2的波动中提取密钥。

图6（a）为基于偏振态斯托克斯参数的密钥分发方案的实验装置图。其中，两个偏振控制器用于调节合法双方间的初始偏振态，两个偏振分析仪用于记录对应的斯托克斯参数，并用于后续的密钥后处理。Alice和Bob测得的S_2波形如图6（b）所示，两个波形之间的互相关系数超过了0.98，验证了该密钥分发方案的可行性。

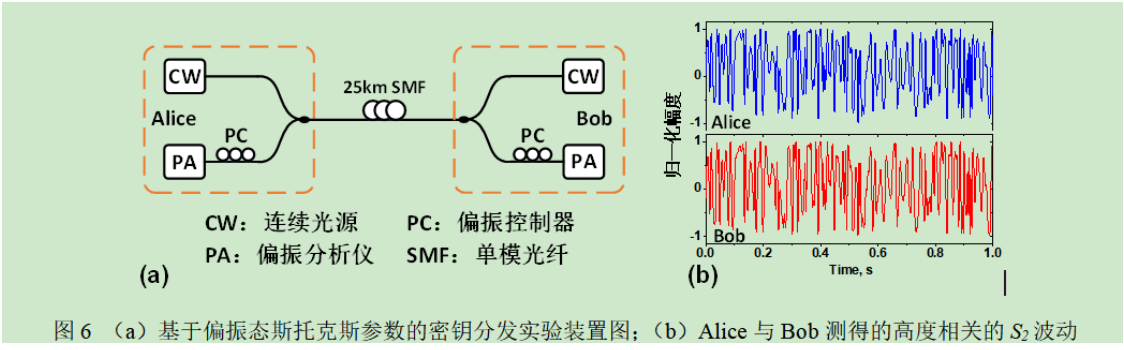


图6　（a）基于偏振态斯托克斯参数的密钥分发实验装置图；（b）Alice与Bob测得的高度相关的S_2波动

在密钥安全性方面，根据式（1）和式（3），光偏振态及其对应的斯托克斯参数S_2对光纤长度高度敏感。在图5所示的实验结果中，3m长的单模光纤足以影响光偏振态。实际上，这一敏感度可进一步降低至1m[15]。因此，可分别在合法通信双方两端额外部署一段具有一定长度的单模光纤，当攻击者无法获取合法双方间光纤信道的精确长度时，她将窃听到完全不同的S_2波动。另外，光偏振态对外界环境因素的敏感性也可进一步确保攻击者无法获取与合法双方相同的密钥熵源。

5　高速密钥分发的可行机制

在上述两种密钥分发方案中，偏振模相位差波动和偏振态斯托克斯参数波动仅来源于外界环

境因素。由于光纤信道的准静态性，所得到的密钥分发速率仅为220bps [14] 和222bps [15]，与目前的主流光通信速率（~10Gbps）相比，仍有不少差距。为此，我们提出对光偏振态施以主动扰动，以提高其波动带宽，进而提高密钥分发速率。

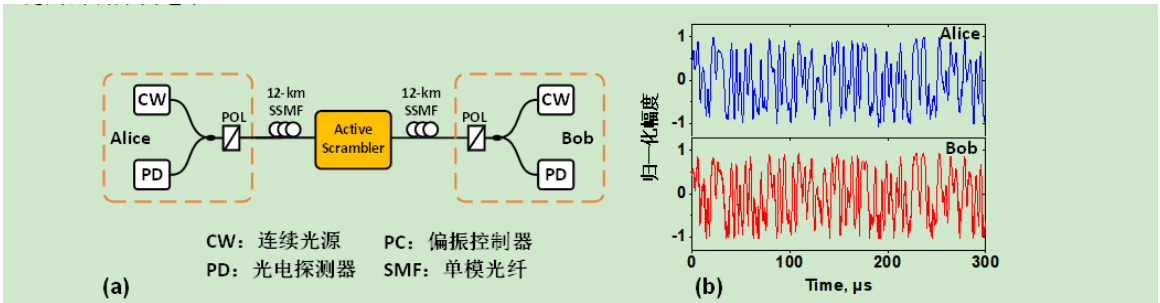


图 7 （a）基于偏振态扰动的密钥分发原理及实验装置图；（b）Alice 与 Bob 测得的高度相关的强度波动

基于偏振态主动扰动的密钥分发方案的原理与实验装置图如图7（a）所示，在24km长的光纤信道中间部署一个具有高带宽的偏振扰动仪，以高速改变光波的偏振态。两个偏振器构成的偏振干涉仪可将此偏振态波动转换为强度波动：

$$\Delta I = \frac{1}{2}\left[1+\cos 2\alpha\cos 2\beta+|\gamma|\sin 2\alpha\sin 2\beta\cos\left(\Delta\varphi_f+\Delta\varphi_s\right)\right] \quad (4)$$

其中，$\Delta\varphi_f$和$\Delta\varphi_s$分别为光纤信道和高速扰偏仪导致的光波沿x轴分量、y轴分量间的相位差。实验中，采用美国通用光电公司（General Photonics）的扰偏仪PCD-104，合法双方Alice与Bob获得的高度相关的强度波形如图7（b）所示，两个波形的互相关系数超过了0.97，可作为光物理层中的密钥熵源。同样，基于信道唯一性，攻击者无法获得合法双方间的全部光信道特征保证了密钥的安全性。

需要指出的是，在该密钥分发方案中，最终的密钥分发速率取决于扰偏仪的带宽，即式（4）中$\Delta\varphi_s$的带宽。实验中商用扰偏仪的带宽为750kHz，最终获得了200kbps的密钥速率 [16]，将基于偏振模干涉和基于偏振态斯托克斯参数的密钥分发方案中的速率指标提高了3个数量级，并通过密钥协商技术将密钥不一致率降低为0 [16]。最终的密钥序列通过了全部15个NIST随机性测试 [20]，验证了密钥的真随机性。

表1总结了目前已报道的光物理层中基于特征参数波动的密钥分发结果。可以看出，本文所验证的基于偏振态主动扰动的密钥分发方案，在密钥分发速率以及密钥不一致率这两个重要指标上都处于国际前列。通过设计更高带宽的扰偏仪，预计将进一步提高密钥分发速率。

6 密钥后处理协议

6.1 量化

在合法双方Alice和Bob获得了高度相关的特征参数波动之后，需将其量化为初始密钥序列。常见的量化法为双阈值量化：

表1 近年来基于光特征参数波动的密钥分发技术的报道结果

科研单位	物理机制	密钥分发速率	密钥不一致率
俄罗斯科学院[7]	MZI中的相位波动	160bps	4%
美国加州大学欧文分校[8]	偏振模色散引起的强度失真	–	<10%
美国普林斯顿大学[9]	分布式MZI中的相位波动	502bps	0.3%
美国耶鲁大学[10]	多模光纤中的模式混合效应	20bps	5%
北京邮电大学[11-13]	光通信中的误码率波动	400kbps	2%
		2Mbps	2%
		277kbps	0
上海交通大学（本文）	偏振模干涉	220bps	5%
	偏振态斯托克斯参数波动	222bps	4.5%
	偏振态主动扰动	200kbps	0

$$Q(y)=\begin{cases}1 & \text{if } f(y)>q+\\ 0 & \text{if } f(y)<q-\end{cases} \qquad q\pm=m\pm\varepsilon\times\sigma \quad (5)$$

其中，Q为量化后的密钥序列，f为接收端记录下的特征参数，$q\pm$为量化器的上、下阈值，m和σ分

别为特征参数f的均值与方差，ε是一个标量因子。在该双阈值量化器中，大于（小于）上（下）阈值的采样点将会被判决为比特“1”（“0”），而位于两个阈值中间的采样点将会被丢弃。

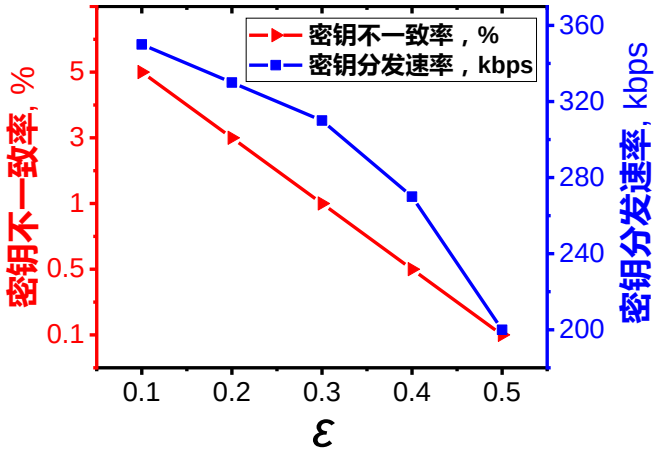


图8　标量因子ε对密钥分发速率和密钥不一致率的影响

图8给出了基于偏振态主动扰动的密钥分发方案中，密钥分发速率与密钥不一致率随标量因子ε的变化情况。随着的ε增大，两个量化阈值q-、q+之间的保护间隔增大，密钥不一致率降低。但与此同时，所舍弃的采样点也随之增加，导致密钥分发速率降低。因此，需要考虑密钥分发速率与密钥不一致率这两个关键密钥性能指标之间的矛盾。

6.2　密钥协商

受到实际光纤信道中非对称噪声的影响，合法双方量化后的初始密钥序列之间存在着一定的不一致率，可通过密钥协商技术[21]来消除该密钥不一致率。现有的密钥协商机制主要基于以下两类机制：二分查找和纠错编码。

基于二分查找的密钥协商机制如BBBSS协议[22]和Cascade协议[23]，其核心是利用校验码来发现不一致的密钥块，并通过二分法查找不一致密钥比特的位置。该方法的效果取决于分组的大小，分组越大，效果越好，但同时泄露给窃听者的信息越多，并且信息交互的次数过多，故效率不高。

另一种是基于纠错编码的密钥协商机制。由于密钥协商本质上是一个纠错过程，因此基于纠错编码的密钥协商方式具有更高的效率。其基本原理是将k比特的密钥编码成n比特，通过公开额外n-k比特的校验信息，实现协商纠错。常见的纠错编码包括LDPC码、Turbo码、BCH码、RS码等。

6.3　私密增强

在密钥协商阶段，由于合法双方在公共信道上进行了信息交互，不可避免地向攻击者泄露了关于密钥的部分信息。私密增强技术[24]用于消除此安全隐患，以得到绝对安全的密钥。同时，私密增强技术还能进一步增强密钥的随机性。常用的私密增强机制，包括：通用哈希函数法和提取器协议。

通用哈希函数法的思想是对密钥进行压缩映射，而提取器协议则是提取出部分密钥。由于哈希函数具有单向性和散列性，基于哈希函数的私密增强方案比较简单，同时安全性也很高，适用于任意长的输入序列。而对于输出更长的密钥序列，采用基于提取器协议的私密增强方案具有更高的效率。

7　结束语

经典光物理层中基于光特征参数波动的密钥分发技术基于光纤信道的互易性，具备结构简单、成本低、适用于远距离光网络等优点，并兼容于现有的光网络架构。本文介绍了课题组近年来在基于偏振模干涉、基于偏振态斯托克斯参数以及基于偏振态主动扰动的密钥分发领域中的研究成果，验证了通过对光特征参数施以主动扰动，以克服光纤信道的准静态特征，进而提高密钥分发速率的可行性。与传统的基于公钥算法的密钥分

发不同，光物理层中的密钥分发无法被计算破解，有望为光纤中高速、大容量数据传输的安全性带来大幅提升。

参考文献:

[1] SHANEMAN K, GRAY S. Optical network security: technical analysis of fiber tapping mechanisms and methods for detection & prevention［C］//IEEE MILCOM 2004. Military Communications Conference，2004.
IEEE，2004，2：711-716.

[2] KATZ J, LINDELL Y. Introduction to modern cryptography［M］. CRC press，2014.

[3] ARUTE F, ARYA K, BABBUSH R, et al. Quantum supremacy using a programmable superconducting processor［J］. Nature，2019，574（7779）：505-510.

[4] XU F, MA X, ZHANG Q, et al. Secure quantum key distribution with realistic devices［J］. Reviews of Modern Physics，2020，92（2）：025002.

[5] MAURER U M. Secret key agreement by public discussion from common information［J］. IEEE transactions on information theory，1993，39（3）：733-742.

[6] AHLSWEDE R, CSISZÁR I. Common randomness in information theory and cryptography. I. Secret sharing［J］. IEEE Transactions on Information Theory，1993，39（4）：1121-1132.

[7] KRAVTSOV K, WANG Z, TRAPPE W, et al. Physical layer secret key generation for fiber-optical networks［J］. Optics express，2013，21（20）：23756-23771.

[8] ZAMAN I U, LOPEZ A B, FARUQUE M AAL, et al. Physical layer cryptographic key generation by exploiting PMD of an optical fiber link［J］. Journal of Lightwave Technology，2018，36（24）：5903-5911.

[9] HUANG C, MA P Y, BLOW E C, et al. Accelerated secure key distribution based on localized and asymmetric fiber interferometers［J］. Optics express，2019，27（22）：32096-32110.

[10] BROMBERG Y, REDDING B, POPOFF S M, et al. Remote key establishment by random mode mixing in multimode fibers and optical reciprocity［J］. Optical Engineering，2019，58（1）：016105.

[11] YANG X, LI Y, GAO G, et al. Demonstration of key generation scheme based on feature extraction of optical fiber channel［C］//2018 Asia Communications and Photonics Conference (ACP). IEEE，2018：1-3.

[12] WANG X, ZHANG J, LI Y, et al. Secure key distribution system based on optical channel physical features［J］. IEEE Photonics Journal，2019，11（6）：1-11.

[13] LEI C, ZHANG J, LI Y, et al. Long-haul and high-speed key distribution based on one-way non-dual arbitrary basis transformation in optical fiber link［C］//2020 Optical Fiber Communications Conference and Exhibition (OFC). IEEE，2020：1-3.

[14] HAJOMER A A E, YANG X, SULTAN A, et al. Key distribution based on phase fluctuation between polarization modes in optical channel［J］. IEEE Photonics Technology Letters，2018，30（8）：704-707.

[15] ZHANG L, HAJOMER A A E, YANG X, et al. Error-free secure key generation and distribution using dynamic Stokes parameters［J］. Optics express，2019，27（20）：29207-29216.

[16] HAJOMER A A E, ZHANG L, YANG X, et al. Accelerated key generation and distribution using polarization scrambling in optical fiber［J］. Optics Express，2019，27（24）：35761-35773.

[17] HAJOMER A A E, ZHANG L, YANG X, et al. Post-processing protocol for physical-layer key generation and distribution in fiber networks［J］. IEEE Photonics Technology Letters，2020，32（15）：901-904.

[18] WOLIŃSKI T R. Polarization in optical fibers［J］. Acta Phys. Polonica A，1999，95（5）：749-760.

[19] POTTON R J. Reciprocity in optics［J］. Reports on Progress in Physics，2004，67（5）：717-754.

[20] BASSHAM III L E, RUKHIN A L, SOTO J, et al. Sp 800-22 Rev. 1a. A statistical test suite for random and pseudorandom number generators for cryptographic applications［M］. National Institute of Standards & Technology，2010.

[21] HUTH C, GUILLAUME R, STROHM T, et al. Information reconciliation schemes in physical-layer security: A survey［J］. Computer Networks，2016，109：84-104.

[22] BENNETT C H, BESSETTE F, BRASSARD G, et al. Experimental quantum cryptography［J］. Journal of cryptology，1992，5（1）：3-28.

[23] BRASSARD G, SALVAIL L. Secret-key reconciliation by public discussion［C］//Workshop on the Theory and Application of of Cryptographic Techniques. Springer，Berlin，Heidelberg，1993：410-423.

[24] BENNETT C H, BRASSARD G, CRÉPEAU C, et al. Generalized privacy amplification［J］. IEEE Transactions on Information Theory，1995，41（6）：1915-1923.

[作者简介]

张柳明（1997—），男，学士学位，博士研究生，主要研究方向为光网络物理层安全。

Adnan Hajomer（1993—），男，硕士学位，博士研究生，主要研究方向为光网络物理层安全及高速光信号处理。

杨学林（1966—），男，博士学位，教授，主要研究方向为高速光信号处理、光接入网、光网络安全等。

大尺度衰落环境下隐蔽通信的有效隐蔽区域定义与分析

林孟涵，洪玺，王文杰
西安交通大学信息与通信工程学院，陕西西安 710049

摘　要：现有隐蔽通信研究中常用的评价指标，隐蔽信道容量或隐蔽通吞吐速率，都难以在实际的无线隐蔽通信系统设计中使用。针对这个问题，本文在监测者对噪声的测量包含有界误差的假设下，在大尺度衰落环境中分析隐蔽通信的有效隐蔽区域，并提出以检测距离作为衡量通信隐蔽性能的指标进行系统设计。通过对自由空间传播和阴影衰落两种信号传输模型中检测距离的比较，本文讨论了各种因素如何影响检测距离的大小。本文证明，检测距离和发信机发射功率和监测者天线增益成对数线性关系，且两者的比例系数都为路径衰减指数的倒数。这些结果可为实际的隐蔽通信系统设计提供参考。

关键词：隐蔽通信、有效安全范围、点对点通信

Definition and Analysis on Effective Covert Area for Covert Communication in Large Scale Fading Environment

LIN Meng-han, HONG Xi, WANG Wen-jie
Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China

Abstract: The widely adopted criteria in the covert communications research, covert capacity or covert throughput rate, are difficult to use in practical design of wireless covert communication systems. To address this problem, the effective covert area in large scale fading environment is analyzed with the assuming that warden's measurement on noise is with bounded error. Additionally, the detection distance is proposed to be used as the criterion of covert communication ability for system designing in this paper. From the comparison of detection distances in two propagation models, free space propagation and shadow fading, the factors' effects on detection distance is discussed in this paper. It is proved in this paper that the detection distance is in log-linear relationship with transmitter's power and warden's antenna gain, meanwhile both of these scale factors are the reciprocal of path-loss exponent. These results can be used as references for the practical covert communication system designing.

Key words: covert communication; effective covert area; point-to-point communication

1　引言

近年来，无线通信的信息安全问题备受关注，促进了无线物理层安全的研究。物理层安全通过在物理层降低监听者接收信道的信道容量，从而实现通信的信息安全。但是，某些民用和军用通信中，对通信安全有着更苛刻的要求：它们不仅仅需要保证信息不被窃听，还要保证通信行为不被敌方发现。这样，只要不被发现，就不可能被敌方破译，通信信息的安全也因此得以保证。

隐蔽通信的研究最早集中在调制方面，通过扩频技术降低信号的功率谱密度［1］，或通过分

数阶傅里叶变换（Fractional Fourier Transform，FRFT）伪装信号的方式［2］实现通信信号的隐蔽。2012年，美国麻省大学的B. A. Bash等从信息论角度出发，假设监测者可以实现最优检测（最大后验概率检测），推导出了低检测概率（Low Probability of Detection， LPD）通信中对于隐蔽信道容量的平方根律限制［3］。之后，隐蔽通信的研究重点转向隐蔽通信容量和速率的分析和优化，考虑在检测者采用最优检测的条件下，如何提高隐蔽通信的吞吐速率。文献［4］和［5］考虑利用外界干扰或者己方的人工噪声实现提高隐蔽吞吐速率的方法，文献［6］则考虑利用监测者自身对噪声功率检测的不确定性实现隐蔽通信。为了降低通信信号的功率从而降低被发现的概率，文献［7］考虑在距离较远的合法通信方之间布设中继辅助提升隐蔽吞吐速率，文献［8］则进一步将信源发送和中继转发拆分成两个时隙完成，并让信源和中继互相在对方发送通信信号时产生人工噪声，干扰监测者的检测过程。

以上工作都考虑通信方采用无限长的信号进行通信，但实际信号长度有限，信号总能量也有随机性，这提升了通信的隐蔽性能。文献［9］和［10］就分别分析了单天线和多天线检测情景下的有限码长隐蔽通信问题，并刻画了相应的隐蔽吞吐速率。

这些研究大都以隐蔽容量或隐蔽吞吐速率作为性能指标进行分析和优化。但是在工程应用方面，吞吐速率往往因调制方式的确定而固定，隐蔽容量或隐蔽吞吐速率这样的信息论指标很难与实际通信系统的性能指标直接对应。最近也有使用其他指标来衡量隐蔽通信性能的研究，比如文献［11］考虑定位网络中的隐蔽问题，就以满足隐蔽要求下的定位误差作为隐蔽通信性能指标进行分析。对于不同的应用，有必要根据应用场景分析和评估相应的指标。

本文考虑室外大尺度衰落、速率恒定的点对点通信的场景，通信双方在通信的同时，这样需要与监测者（Willie）保持足够远，才能避免被Willie成功检测。有效隐蔽区域是Willie无法正确检出通信存在的区域，检测距离为Alice能被Willie检出的最远距离。本文假设各节点相对静止，监测者Willie对噪声有测量误差，Alice通过将信号隐藏于此实现隐蔽，与文献［6］类似。而合法接收者Bob只解调信号，不考虑噪声测量误差对其性能造成的影响。通过理论推导和数值实验，本文分析了自由空间传播和阴影衰落模型下，检测距离的大小，尤其是其与路径损耗指数和接收机灵敏度的关系，为实际隐蔽通信系统的设计提供参考。

本文余下部分安排如下：第2节介绍远距离点对点通信的系统模型，包括通信方Alice和Bob之间的通信服务质量（Quality of Service， QoS）要求和Willie的假设检验过程；第3节通过理论推导分析检测距离的大小，第4节对不同情景下的检测距离进行比较；最后第5节总结全文。

2　系统模型

本文考虑距离较远的两个单天线通信节点进行通信，其中Alice发送信息，Bob接收信息，另有监测者Willie使用功率计对Alice发送的信号进行检测，他们的相对位置关系如图1所示。其中，d_{AB}是Alice与Bob之间的距离，d_{AW}是Alice与Willie之间的距离，图中阴影区域外为有效隐蔽区域，Willie在有效隐蔽区域中，则无法正确检测。有效检测区域的大小和Alice发射功率大小以及Willie的检测能力有关。d_{det}是Willie能够正确检测的最大检测距离，$d_{AW} > d_{\mathrm{det}}$时认为Willie无法正确检测信号的存在，则可以实现隐蔽通信。需要指出的是，如果Alice配备有向天线或多天线，并向Bob定向发送信号，则有效隐蔽区域的大小和形状将发生变化。对于双向通信系统，只需将本文分析中的Alice替换为Bob即可。

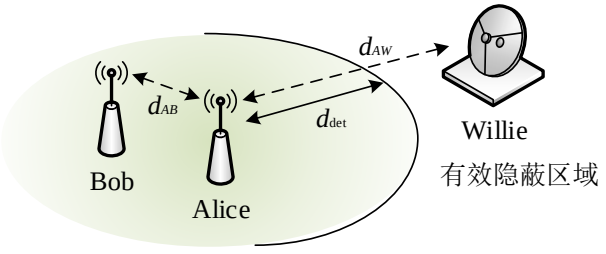


图1　点对点隐蔽通信系统模型

本文只考虑无线信号传输的大尺度衰落情景，其中包括路径损耗和阴影衰落。这样，信号从Alice到Bob和Willie的传输损耗$P_{L,\,AB}$和$P_{L,\,AW}$分别为［12］：

$$P_{L,AB}(\mathrm{dB}) = -10\lg K + 10\alpha\lg\left(d_{AB}/d_0\right) - \psi_{\mathrm{dB}}, \quad (1)$$

$$P_{L,AW}(\mathrm{dB}) = -10\lg K + 10\alpha\lg\left(d_{AW}/d_0\right) - \psi_{\mathrm{dB}}. \quad (2)$$

其中，lg表示以10为底的对数；$\alpha = 2\sim4$是路径损耗指数；d_0是参考距离，本文中设$d_0 = 1\mathrm{km}$；K为d_0处的路径增益；ψ_{dB}为阴影衰落系数，服从正态分布$N(0,\sigma^2_{\psi_{\mathrm{dB}}})$，在不考虑阴影衰落的情景下，即取$\psi_{\mathrm{dB}} = 0$。

2.1 通信QoS约束

首先考虑Alice与Bob之间通信的QoS约束。设Alice发送信号的功率为P_t，则Bob接收到的信号功率大小为：

$$P_{r,AB}(\mathrm{dBm}) = P_t(\mathrm{dBm}) - P_{L,AB}(\mathrm{dB}). \quad (3)$$

考虑接收方Bob的接收机灵敏度（即满足QoS要求的最小输入信号强度）为P_{rs}（dBm），则要能正常通信，Alice发射功率P_t应满足：

$$P_t(\mathrm{dBm}) - P_{L,AB}(\mathrm{dB}) \geqslant P_{rs}(\mathrm{dBm}). \quad (4)$$

对于自由空间没有阴影衰落的场景，满足（4）式即可正常通信。但是，对于服从对数正态分布的阴影衰落场景，无法保证接收功率总是大于接收机灵敏度。为此，需要把通信QoS约束调整为接收功率有η的概率大等于接收机灵敏度，即

$$\Pr\left\{P_{r,AB}(\mathrm{dBm}) \geqslant P_{rs}(\mathrm{dBm})\right\} \geqslant \eta. \quad (5)$$

这里η应取一个较大的值，根据衰落服从对数正态分布，将（5）式展开得到：

$$P_t(\mathrm{dBm}) - P_{L,AB}(\mathrm{dB}) - P_{rs}(\mathrm{dBm}) \geqslant -\sigma_{\psi_{\mathrm{dB}}}Q^{-1}(\eta), \quad (6)$$

其中$Q^{-1}(\cdot)$是标准正态分布的Q函数。以$\eta = 90\%$、$\sigma_{\psi_{\mathrm{dB}}} = 8\mathrm{dB}$为例，可得$-\sigma_{\psi_{\mathrm{dB}}}Q^{-1}(\eta) = 10.25\mathrm{dB}$，即在阴影衰落下，为保证90%的情况仍可正常通信，需要保留至少10.25dB的功率裕量。

2.2 Willie的监测过程

本文考虑Willie采用单天线功率计针对接收到的信号功率进行检测。在Alice不发送信号（H_0）和发送信号（H_1）两种情况下，Willie检测到的功率P_W为

$$\begin{cases} H_0: P_W = \hat{\sigma}_w^2, \\ H_1: P_W = G_w P_{r,AW} + \hat{\sigma}_w^2. \end{cases} \quad (7)$$

其中，$P_{r,AW} = P_t / P_{L,AW}$是Alice发射的信号传输到Willie处的功率，$\hat{\sigma}_w^2$为Willie检测到的噪声功率，G_w为Willie的天线增益，体现Willie的检测能力，假设Willie可以通过搜峰的方式把天线主瓣转向Alice。对于隐蔽通信而言，通常$G_w P_{r,AW} \ll \hat{\sigma}_w^2$，因此对接收信号功率$P_W$的测量误差就近似为对真实噪声功率$\sigma_w^2$的测量误差。通常可以将$\hat{\sigma}_w^2$建模为服从对数均匀分布的随机变量［11］［13］，即$\lg\hat{\sigma}_w^2 \sim U(-B\lg\sigma_w^2, B\lg\sigma_w^2)$，其中$B$为误差的对数上界，这样$\hat{\sigma}_w^2$的概率密度函数（PDF）为：

$$f_{\hat{\sigma}_w^2}(x) = \begin{cases} \dfrac{1}{2Bx\ln 10}, & 10^{-B}\sigma_w^2 < x < 10^{B}\sigma_w^2 \\ 0, & \text{otherwise} \end{cases}. \quad (8)$$

其中ln表示自然对数。

设Willie在H_0和H_1两种情况下接收功率的概率密度函数分别为$f_{P_W|H_0}(x)$和$f_{P_W|H_1}(x)$，并假设两种假设先验概率相等Pr（H_0）= Pr（H_1），则Willie的最优检测即为最大似然检测，根据似然函数L（x）判决：

$$\ln L(x) = \ln f_{P_W|H_1}(x) - \ln f_{P_W|H_0}(x) \begin{cases} > 0, D_1 \\ < 0, D_0 \end{cases}, \quad (9)$$

其中D_1表示判决为假设H_1，D_0表示判决为假设H_0。Willie的检测错误概率ξ为虚警概率P_{FA}和漏报概率P_{MD}的和：

$$\xi = P_{FA} + P_{MD}, \quad (10)$$

其中：

$$P_{FA} = \int_{D_1} f_{P_W|H_0}(x)\,dx, \quad (11)$$

$$P_{MD} = \int_{D_0} f_{P_W|H_1}(x)\,dx. \quad (12)$$

在隐蔽通信中，通信隐蔽性要求通常为保证Willie的最优检测的错误概率依然很大，即$\xi \geqslant 1 - \varepsilon$。

3 检测距离

3.1 自由空间的检测距离

我们首先考虑自由空间中检测距离的计算。自由空间中没有阴影衰落，在Alice、Bob和Willie相对位置固定的情况下$P_{r,AW}$也是定值，这时两种假设下Willie接收到信号功率的概率密度函数分别为：

$$\begin{cases} f_{P_W|H_0}(x) = f_{\hat{\sigma}_w^2}(x), \\ f_{P_W|H_1}(x) = f_{\hat{\sigma}_w^2}(x - P_{r,AW}). \end{cases} \quad (13)$$

显然$f_{P_W|H_1}(x)$是$f_{P_W|H_0}(x)$向右平移$P_{r,AW}$的结果，以$B = 0.1$、$\sigma_w^2 = 1$、$G_w = 1$、$P_{r,AW} = 0.1$为例，它们

的概率密度函数如图2所示。由（9）式可知，对Willie接收到的功率x，当$f_{P_W|H_1}(x) > f_{P_W|H_0}(x)$时认为Alice在发送信号，反之，当$f_{P_W|H_1}(x) < f_{P_W|H_0}(x)$时判定为只有噪声。判决门限和判决域（$D_0$和$D_1$）也在图2中示出。考虑它们都是单调递减函数，可知最优检测门限为其相交点$P_{Th}^{*} = 10^{-B}\sigma_w^2 + G_w P_{r,AW}$。该结论与文献［11］中自由空间噪声不确定性有上界时的结果一致。

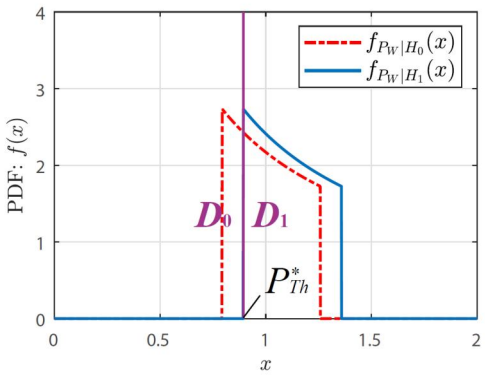


图2　自由空间条件两种假设下Willie接收功率大小的概率密度函数

这样，可以得到Willie的检测错误概率以及对应的隐蔽性能要求为：

$$\xi = P_{FA} = \int_{10^{-B}\sigma_w^2 + P_{r,AW}}^{10^{B}\sigma_w^2} f_{\hat{\sigma}_w^2}(x)\,dx \geqslant 1 - \varepsilon. \tag{14}$$

解该不等式方程得到对$P_{r,\,AW}$的约束，为

$$P_{r,AW} \leqslant (10^{2\varepsilon B} - 1)\cdot 10^{-B}\sigma_w^2 / G_w. \tag{15}$$

又由于$P_{r,\,AW}$（dBm）$= P_t$（dBm）$- P_{L,\,AW}$（dB），由（2）式和（15）式得到满足隐蔽条件的Alice到Willie距离为

$$d_{AW} \geqslant d_{\det} = d_0\left[\frac{G_w P_t K}{(10^{2\varepsilon B} - 1)\cdot 10^{-B}\sigma_w^2}\right]^{1/2}, \tag{16}$$

其中，路径损耗指数$\alpha = 2$，最大检测距离$d_{\det}$为满足隐蔽条件的d_{AW}的最小值。对（16）式等号两边取对数，可以得到

$$\lg d_{\det} = \frac{1}{2}P_t(\text{dBm}) + \frac{1}{2}\lg G_w + \frac{1}{2}\lg\left(\frac{K}{(10^{2\varepsilon B} - 1)\cdot 10^{-B}\sigma_w^2}\right) + \lg d_0 \tag{17}$$

从（17）式可知在其他条件都确定的情况下，$\lg d_{\det}$随着Alice发射功率或Willie天线增益的增加，以1/2的斜率线性增加。又由（4）式发射功率和Bob接收机灵敏度的关系，可知$\lg d_{\det}$随接收机灵敏度提升以1/2斜率线性增加。

3.2　阴影衰落环境中的检测距离

在阴影衰落的环境中，信号的路径损耗更大，这既干扰了Willie的检测，也降低了合法通信成功的概率。其中，Alice信号传到Willie处功率$P_{r,\,AW}$服从对数正态分布，使得尽管检测问题仍和（7）式一样，但最优检测门限发生了变化。阴影衰落中，$P_{r,\,AW}$的概率分布函数$F_{P_{r,AW}}(x)$和概率密度函数$f_{P_{r,AW}}(x)$分别为：

$$F_{P_{r,AW}}(x) = \begin{cases} \Phi\left(\dfrac{10\lg x - \mu_{r,AW}}{\sigma_{\psi_{dB}}}\right), & y > 0 \\ 0, & y \leqslant 0 \end{cases} \tag{18}$$

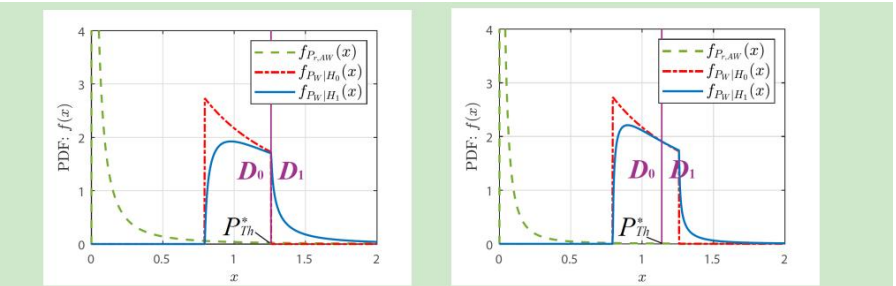
$$f_{P_{r,AW}}(x) = \begin{cases} \dfrac{10}{x\sigma_{\psi_{dB}}\sqrt{2\pi}\ln 10}\exp\left[-\dfrac{\left(10\lg x - \mu_{r,AW}\right)^2}{2\sigma_{\psi_{dB}}^2}\right], & y > 0 \\ 0, & y \leqslant 0 \end{cases} \tag{19}$$

其中$\Phi(\cdot)$是标准正态分布的概率分布函数，$\mu_{r,AW} = P_t(\text{dBm}) + G_w(\text{dB}) + 10\lg K - 10\alpha\lg\left(d_{AW}/d_0\right)$，是Willie接收到Alice信号的平均功率。由噪声和Alice发射信号相互独立，可知在阴影衰落场景中两种假设检验Willie检测到功率的概率密度函数为：

$$\begin{cases} f_{P_W|H_0}(x) = f_{\hat{\sigma}_w^2}(x), \\ f_{P_W|H_1}(x) = \displaystyle\int_{-\infty}^{+\infty} f_{\hat{\sigma}_w^2}(x - t)\, f_{P_{r,AW}}(t)\,dt. \end{cases} \tag{20}$$

其中$f_{P_W|H_1}(x)$是$f_{P_{r,AW}}(x)$和$f_{\hat{\sigma}_w^2}(x)$的卷积，闭式解难以求得。但在隐蔽通信场景中，$P_{r,\,AW}$的期望应当远小于噪声功率。以$\sigma_{\psi_{dB}} = 8$dB，$\mu_{r,AW} = -15$或-20为例，$f_{P_W|H_0}(x)$和$f_{P_W|H_1}(x)$的函数曲线如图3所示。图中的$f_{P_{r,AW}}(x)$较集中于$x = 0$附近，随着信号衰减加大，$P_{r,\,AW}$缩小，$f_{P_{r,AW}}(x)$趋近于单位冲激函数，使得$f_{P_W|H_1}(x)$更加接近$f_{P_W|H_0}(x)$，即从Willie来看，Alice发送信号与不发的差别更小。

从图3（a）中可以看出，信号衰减不够大的时候，在区间$[10^{-B}\sigma_w^2, 10^{B}\sigma_w^2]$内$f_{P_W|H_1}(x) \leqslant f_{P_W|H_0}(x)$，根据最大似然准则应判定为Alice没有发射信号；而在$x > 10^{B}\sigma_w^2$时，$f_{P_W|H_1}(x) > f_{P_W|H_0}(x) = f_{\hat{\sigma}_w^2}(x) = 0$，则判定为Alice发



(a) $\mu_{r,AW}=-15$　　(b) $\mu_{r,AW}=-20$

图 3　阴影衰落条件两种假设下 Willie 接收功率大小的概率密度函数、判决门限和判决域

射了信号。这样，最优的检测门限为$P^*_{Th}=10^B\sigma_w^2$。不过，以上分析并不总成立，能使该最优检测门限的成立的充分条件由如下定理给出。

定理 1：

最优的检测门限为$P^*_{Th}=10^B\sigma_w^2$的充分条件是：

$$F_{P_{r,AW}}(10^B\sigma_w^2 - 10^{-B}\sigma_w^2) \leqslant \frac{f_{\hat{\sigma}_w^2}(10^B\sigma_w^2)}{f_{\hat{\sigma}_w^2}(10^{-B}\sigma_w^2)} \tag{21}$$

在附录 1 中证明。

这样，根据隐蔽性能要求，有

$$\xi = P_{MD} = F_{P_{r,AW}}(10^B\sigma_w^2) \geqslant 1-\varepsilon. \tag{22}$$

若定理 1 不能满足，由于未能求出概率密度函数的闭式，最优门限P^*_{Th}的闭式解也无法求得。但是从图 3（b）中可以看出，由于此时$P_{r,\,AW}$很小，在区间$[P^*_{Th}, 10^B\sigma_w^2]$内$f_{P_W|H_1}(x)\approx f_{P_W|H_0}(x)$，故有近似结果：

$$\begin{aligned}\xi &= P_{FA}+P_{MD}=\int_{P^*_{Th}}^{10^B\sigma_w^2} f_{P_W|H_0}(x)\,dx+\int_{10^{-B}\sigma_w^2}^{P^*_{Th}} f_{P_W|H_1}(x)\,dx\\ &\approx \int_{10^{-B}\sigma_w^2}^{10^B\sigma_w^2} f_{P_W|H_1}(x)\,dx = F_{P_{r,AW}}(10^B\sigma_w^2)=F_{AW}\geqslant 1-\varepsilon,\end{aligned} \tag{23}$$

其中，令$F_{AW}=F_{P_{r,AW}}(10^B\sigma_w^2)$。此时我们可以得到近似的检测错误概率和 Willie 接收到的 Alice 信号功率的关系，由如下定理阐明：

定理 2：

Willie 的近似检测错误概率F_{AW}关于 Alice 到达 Willie 处信号的功率$\mu_{r,AW}$单调递减。

在附录 2 中证明。

其实该定理也非常容易被直观理解：Willie 接收到的 Alice 的信号功率越小，检测错误概率通常就越大。但由此单调性可知，在其他条件都一致的情况下，就必然存在唯一的$\mu^*_{r,AW}$使得$F_{AW}(\mu^*_{r,AW})=1-\varepsilon$。于是，隐蔽性要求可以转化为：

$$P_t(\mathrm{dBm})+G_w(\mathrm{dB})+10\lg K-10\alpha\lg\left(d_{AW}/d_0\right)\leqslant\mu^*_{r,AW} \tag{24}$$

得到检测距离为：

$$d_{AW}\geqslant d_{\mathrm{det}}=d_0\left(\frac{G_wP_tK}{10^{\mu^*_{r,AW}/10}}\right)^{1/\alpha} \tag{25}$$

转化为对数形式为：

$$\lg d_{\mathrm{det}}=\frac{1}{\alpha}P_t(\mathrm{dBm})+\frac{1}{\alpha}\lg G_w+\frac{1}{\alpha}\lg K+\lg d_0-\frac{1}{\alpha}\mu^*_{r,AW} \tag{26}$$

从（26）看出，$\lg d_{\mathrm{det}}$随发射机发射功率和监测者天线增益提升以$1/\alpha$斜率线性增加，这个结论与自由空间中的情况类似，只是自由空间中$\alpha=2$。因此我们得到如下有关 Willie 的最大检测距离的定理：

定理 3：

Willie 的最大检测距离d_{det}的对数与发射机发射功率和监测者天线增益都成线性关系，比例因子都为路径损耗指数α的倒数。

4　不同情景下检测距离的比较

从第 3 节的理论分析可知，自由空间中的检测距离有闭式解，在阴影衰落场景中则可以通过计算机搜索求解。

我们考虑的场景为：Alice 和 Bob 距离 1km，通信载波频率为 1900MHz，通信带宽 200kHz，Willie 的监测带宽也是 200kHz。考虑噪声电平为-174dBm/Hz，则 Willie 的噪声功率为σ_w^2=-121dBm，测量误差取 B=0.1，Willie 的检测天线增益取G_w=1 或 2。对于自由空间传播，可以计算得d_0 = 1km 处的衰减-lg$K$ = -98dB；对于阴影衰落信道，则采用 SCM 信道（3GPP TR 25.996）中郊区宏小区的参数，d_0 = 1km 处的衰减-lg$K$ = -136.5dB，α=3.5，

$\sigma_{\psi_{\mathrm{dB}}}$ = 8dB［14］，通信 QoS 要求 Bob 接收 Alice 信号功率大于灵敏度的概率 $\eta = 90\%$。为了考察阴影衰落对数正态分布方差对检测距离的影响，我们另取 $\sigma_{\psi_{\mathrm{dB}}}$ = 4dB 的场景进行比较。隐蔽性能要求设为 $\varepsilon = 0.1$。通过计算机数值计算得到检测距离与 Bob 的接收机灵敏度关系如图4所示。

从图4中可以看出，检测距离在对数坐标下，确实和 dB 单位表示的 Bob 接收机灵敏度成线性关系，灵敏度数值越高（要求更高功率的输入），则检测距离越远；又由于 Alice 的最低发射功率和 Bob 接收机灵敏度也为线性关系，故该结果与第3节的推导结论一致。衰落环境中的多径和阴影虽然妨碍了通信，但也对 Willie 的监测产生影响，在接收机灵敏度比较高的时候，阴影衰落的作用使得检测距离更短，有效隐蔽区域更大，更有利于隐蔽通信。阴影衰落对数正态分布的标准差 $\sigma_{\psi_{\mathrm{dB}}}$ 刻画了 Alice 信号传到 Willie 处分布的随机性，$\sigma_{\psi_{\mathrm{dB}}}$ 缩小，则 $P_{r,\,AW}$ 的概率密度函数更接近单位冲激函数，使得两种假设下 Willie 收到的信号功率分布更接近，妨碍了 Willie 的检测，于是缩短了检测距离。

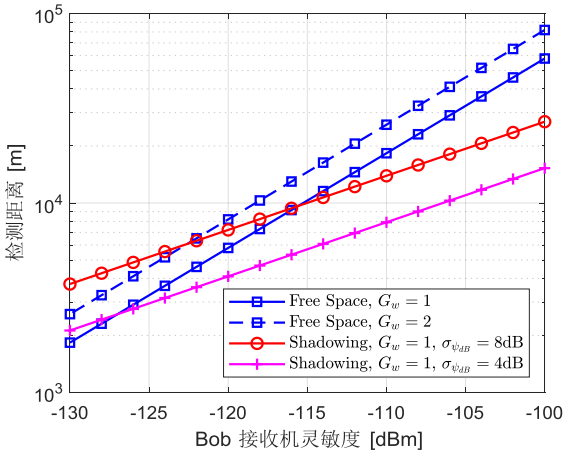


图4　不同场景下检测距离比较

在实际的隐蔽通信系统设计中，在多径更为复杂的环境中通信，使得路径损耗指数 α 更大，可以在需要较高接收机灵敏度的场合提升通信隐蔽性。但通常信道客观存在难以改变，此时可以通过改进通信的调制编码方式，或者提升接收机性能，降低接收机灵敏度系数，从而缩短检测距离，扩大有效隐蔽区域，提升通信的隐蔽性能。

5　结束语

本文从大尺度衰落环境点对点的隐蔽通信模型出发，以工程实际中常用的灵敏度为通信 QoS 要求，提出以检测距离和有效隐蔽区域为隐蔽通信的性能指标，并在自由空间和阴影衰落两种环境中，假设 Willie 对噪声的功率测量具有误差，基于监测者 Willie 的假设检验问题分析影响检测距离和有效检测范围的因素。本文证明了检测距离和 Alice 发射功率与 Willie 天线增益成对数线性关系，比例系数为路径损耗指数 α 的倒数。本文亦通过数值计算结果对比了不同环境条件下检测距离变化的异同。

本文提出的检测距离为隐蔽通信系统设计提供了更接近工程实际的性能评价指标，并在自由空间和阴影衰落两种环境中分析了影响有效隐蔽区域的因素，其既与通信方发射的信号功率有关，也和检测者的检测能力相关。结论适用于远距离大尺度衰落信道，对实际隐蔽通信系统的设计提供了参考。

参考文献:

[1] BASH B A, GOECKEL D, TOWSLEY D, et al. Hiding information in noise: fundamental limits of covert wireless communication［J］. IEEE Communications Magazine，2015，53(12)：26-31.

[2] 梅林, 沙学军, 冉启文, 等. 四项加权分数Fourier变换在通信系统中的应用研究[J]. 中国科学: 信息科学，2010，40(5)：732-741.
MEI L, SHA X J, RAN Q W, et al. The research on the application of 4-WFRFT in communication system［J］. Scientia Sinica(Informationis)，2010，40(5)：732-741.

[3] BASH B A, GOECKEL D, TOWSLEY D. Limits of reliable communication with low probability of detection on AWGN channels［J］. IEEE Journal on Selected Areas in Communications，2015，31(9)：1921-1930.

[4] SOBERS T V, BASH B A, GUHA S, et al. Covert communication in the presence of an uninformed jammer［J］. IEEE Transactions on Wireless Communications，2017，16(9)：6193-6206.

[5] SHAHZAD K, ZHOU X, YAN S, et al. Achieving covert wireless communications using a full-duplex receiver［J］. IEEE Transactions on Wireless Communications，2018，17(12)：8517-8530.

[6] LEE S, BAXLEY R J, WEITNAUER M A, et al. Achieving undetectable communication［J］. IEEE Journal of Selected Topics in Signal Processing，2015，9(7)：1195-1205.

[7] WANG J, TANG W, ZHU Q, et al. Covert communication with the help of relay and channel uncertainty［J］. IEEE Wireless Communications Letters，2019，8(1)：317-320.

[8] FOROUZESH M, AZMI P, KUHESTANI A, et al. Covert communication and secure transmission over untrusted relaying networks in the presence of multiple wardens[J]. IEEE Transactions on Communications，2020，68(6)：3737-3749.

[9] YAN S, HE B, ZHOU X, et al. Delay-intolerant covert communications with either fixed or random transmit power[J]. IEEE Transactions on Information Forensics and Security，2019，14(1)：129-140.

[10] SHAHZAD K, ZHOU X, YAN S. Covert wireless communication in presence of a multi-antenna adversary and delay constraints[J]. IEEE Transactions on Vehicular Technology，2019，68(12)：12432-12436.

[11] ZHAO Y, LI Z, CHENG N, et al. Covert localization in wireless networks: feasibility and performance analysis [J]. IEEE Transactions on Wireless Communications，2020，19(10)：6549-6563.

[12] GOLDSMITH A. Wireless communications[M]. Cambridge，UK: Cambridge University Press，2005.

[13] ZHANG R, LIM T J, LIANG Y, et al. Multi-antenna based spectrum sensing for cognitive radios: A GLRT approach [J]. IEEE Transactions on Communications，2010，58(1)：84-88.

[14] 3rd Generation Partnership Project. TR 25.996 V2.0.0, Spatial Channel Model for Multiple-Input Multiple Output Simulations (Release 6)[R]. Valbonne，France：3GPP，2003.

[作者简介]

林孟涵（1996年生），男，工学学士，现为西安交通大学博士研究生，主要研究方向为认知无线电和隐蔽通信。E-mail：linmh0130@stu.xjtu.edu.cn

洪玺（1990年生），男，工学学士，现为西安交通大学博士研究生，主要研究方向为阵列信号处理和卫星导航信号处理。E-mail：harryhong@stu.xjtu.edu.cn

王文杰（1971年生），男，工学博士，现为西安交通大学教授，博士生导师，主要研究方向为无线通信信号处理、阵列信号处理和卫星导航信号处理。E-mail：wjwang@xjtu.edu.cn

A Quantification Method for the Heterogeneity of Mimic Control Plane in SDN

ZHANG Wenjian, LIU Qinrang, SONG Ke, WEI Shuai, MU Qing

Information Engineering University, Zhengzhou 45002, China

Key words: SDN; mimic control plane; heterogeneity quantification; high order symbiosis

Abstract. SDN （Software-Defined Networking） makes network management more centralized and flexible， however， its control plane more vulnerable to attacks. By introducing the idea of dynamic heterogeneous redundancy， i. e.， mimic control plane， the problems of single control points of failure， unknown backdoor vulnerabilities， and static configuration can be effectively solved. Heterogeneity is an important indicator for evaluating the security of the system. Generally speaking， the greater the heterogeneity， the higher the security. Currently， there are few researches to evaluate the security of the mimic control plane in SDN by quantifying the Heterogeneity. To this end， this paper proposes two methods for measuring heterogeneity based on the biological population diversity assessment method. First， heterogeneity quantification method， i. e.， method 1， based on system complexity and difference is proposed， and on this basis， another heterogeneous quantification method， i.e.， method 2， is established for the mimic system using high order symbiosis of vulnerabilities. The experimental results show that， compared to the known method， the heterogeneous quantification method 1 considers more comprehensive factors and is closer to reality. The heterogeneity quantification method 2 is more instructive to quantify the security of the system.

1 Introduction

The core idea of SDN （Software-Defined Networking） [1] is logical centralization， separation of control plane and data forwarding plane. This centralized and flexible design concept helps to flat network management and makes network operation and maintenance more flexible and efficient. With the widespread application of SDN， attackers gradually increase their attack methods on control plane， resulting in poor scalability [2]， single point of failure [3] and other shortcomings brought by centralized management. Therefore， the security development and design of the SDN control plane gradually attract widespread attention from academia and industry.

In the field of SDN controller security research， there are two main ideas： one is to carry out incremental security development based on the characteristics of controllers； the other is to enhance the security of the control plane by introducing multi-controller technology. Research on using multi-controller technology to enhance the security of the control plane can be mainly divided into three categories： fault tolerance [4]， moving target defense [5]， and mimic defense [6] .

By adopting diversified technologies to enhance

the uncertainty and polymorphism of the system, moving target defense can reduce the exposure time of system vulnerability as much as possible and increase the difficulty of attack, thus ensuring the security of SDN [7-9]. Fault tolerance takes advantage of multi-controller technology to resist failures of some controllers, thereby improving the reliability of SDN systems [10-12]. Based on diversity technologies, mimic defense utilizes the dynamic, heterogeneous, redundant and negative feedback characteristics of the "Dynamic heterogeneous redundancy (DHR)" architecture to make the system have endogenous security attributes. Compared to moving target defense and fault tolerance, mimic defense significantly improves the security of the system [13-15].

In order to solve the security problem of SDN control plane, mimic defense is introduced into SDN control plane. A set of heterogeneous executors is constructed by applications, controllers, and operating systems, naming mimic control plane. The endogenous security attributes of control plane are formed by dynamic scheduling of these executors, multi-executor arbitration, and feedback of abnormal executor information. The present studies on the security of mimic control plane focus on the architecture, scheduling space, timing, and algorithms based on negative feedback. Hu proposes the basic evaluation model of MNOS (Mimic Network Operating System), and summarizes the bottleneck of mimic defense in the field of SDN. Aiming at how to evaluate security, MNOS offers a security assessment method [6]. Based on the diversity of executors, Qi et al. proposes a multi-controller security architecture, Mcad-SA [16], [17], to prevent flow rule modification attack in SDN. A dynamic scheduling algorithm is also proposed to evaluate the security of mimic SDN controllers based on security policies [18]. Regarding the reliability and availability of the SDN controller as the optimization goal, a heuristic genetic algorithm suitable for multiple controllers is proposed in [19]. In order to solve reliability and scalability problems bringing in by mimic defense, DSL scheduling algorithm is put forward in [20]. Compared with the random scheduling algorithm, this algorithm improves the reliability of multiple controllers. In the perspective of SDN service deployment, Li [21] introduced mimic defense technology to increase the degree of heterogeneity, and on this basis, improved the scheduling algorithm and decision mechanism. However, there are few researches on the heterogeneous quantification of SDN control plane. Because it is difficult to evaluate the heterogeneous quantification of multi-controller failures, and the mimic architecture has problems that are difficult to evaluate in terms of security gain quantification. However, heterogeneity is one of the foundations of mimic defense. The greater the heterogeneity between executors, the higher the security. The rationality of quantitative evaluation of heterogeneity is closely related to the rationality of mimic system security. Therefore, it is of great significance to study the quantitative evaluation of heterogeneity suitable to mimic control plane in SDN.

Heterogeneity means diversity. There are currently many system diversity assessment methods, and their ideas are basically similar: quantifying system similarity and complexity. In traditional quantification methods, Euclidean distance, Mahalanobis distance, and cosine quantification are used in low-dimensional systems. The attribute dimension of executors for mimic control plane in SDN is relatively high. In high-dimensional application scenarios, the effect of traditional methods is relatively poor [22]. Mitra et al. [23] uses the method of maximum information compression index to calculate the similarity between feature values, and the definition of the relationship between its attribute values will lead to inaccurate quantification of similarity. Pincus [24] proposes to use approximate entropy to determine system complexity, but the relative consistency is low and it is impossible to quantify the difference between executors in mimic control plane. A fuzzy probability model is proposed in [25], formalizing the de-

scription of hypotheses and problems and drawing conclusions for diversity validity by deduction. Although, in a certain extent, the conclusions support that mimic defense technology can ensure the reliability of the system by diversity of executors in a certain way, it does not give a specific quantitative evaluation method.

In the present field of mimic defense technology, some studies are trying to use heterogeneity to quantitatively evaluate the security of mimic systems. Liu et al. [26] determines the final scheduling scheme according to the similarity index, and weighs the dynamics and reliability, but the quantification of the similarity uses an exponential function for fitting. Zhang et al. [27] takes known symbiotic vulnerabilities as the evaluation parameter, and constructs a Heterogeneity based on Common Vulnerability (HBCV) method of mimic construction for Web servers. Compared with Shannon entropy and quadratic entropy, this method can distinguish the heterogeneity of heterogeneous systems more finely. However, there are several shortcomings: 1) Heterogeneity is assessed by known vulnerabilities, without considering the impact of potential vulnerabilities brought by code reuse on the system; 2) The quantification of heterogeneity is based on the dissimilarity of each pair executors in executor set. As measured by the heterogeneity of the mimic system, the heterogeneity of each pair executors of the mimic system cannot completely determine its security, and the heterogeneity model to evaluate the security of mimic system is limited.

Based on the existing heterogeneity research in the field of mimic defense technology, combined with the characteristics of the SDN control plane, this work proposes a method to quantify the heterogeneity of the mimic control plane in SDN. The main contributions and innovations are as follows:

• Based on Shannon entropy and quadratic entropy, the mimic control plane in SDN is modeled, by incorporating parameters such as symbiotic vulnerabilities and code reuse rate into the heterogeneous quantitative model 1. Compared with the HBCV quantitative analysis method [27], model 1 can better describe the code reuse rate and its impact on heterogeneity;

• Define the high-order symbiosis of vulnerabilities, combining the high-order symbiosis of vulnerabilities and the security of the mimic system to construct a heterogeneous quantitative model 2. Compared with the quantitative analysis method of HBCV, this model can better describe the impact of higher-order vulnerabilities on the heterogeneity in scenarios with more than 3 redundancy.

• The simulation of the quantified model 2, which uses the vulnerability database data and the controller vulnerabilities in [32], confirms the influence of parameters such as code reuse rate and common vulnerability ratio on heterogeneity measurement, verifies that the proposed method is more accurate and effective compared with the scenario based on [27] by taking the vulnerability distribution of SDN control plane application scenarios into account.

The organizational structure of the remaining part is as follows. Section 2 describes the mimic structure of the SDN control plane, and construct a mimic model of SDN. In section 3, the heterogeneity of the quantitative method for mimic control plane is proposed based on the biodiversity assessment model. Section 4 verifies the effectiveness and advancement of the method proposed in this paper through simulation and experiment. Finally, our work is summarized and future work is discussed in section 5.

2 Mimic model in SDN

2.1 Mimic architecture

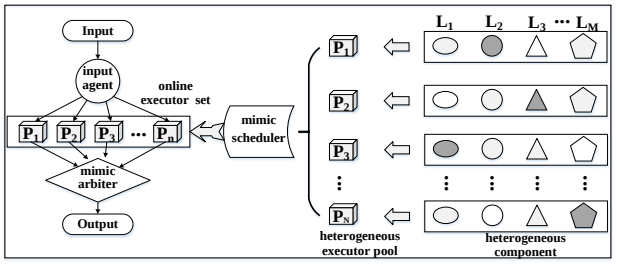


Figure 1 The Schematic diagram of DHR structure

Cyberspace mimic defense （CMD） is a defense technology against unknown attacks， proposed by Jiangxing Wu， an academician of the Chinese Academy of Engineering ［28］. The theory of CMD introduces dynamics， heterogeneity and redundancy mechanisms into cyberspace to defend against cyberspace attacks. The basic architecture of DHR is shown in Figure 1， which mainly includes： input agent， heterogeneous executor pool， heterogeneous component， mimic scheduler， online executor set and mimic arbiter. Among them， the input agent is responsible for the distribution of input data to the system. The distribution principle is duplicated， that is， replicate the input data into n copies and distribute them to n heterogeneous executors with different structures and functionally equivalents. Each executor is independent of each other and manipulates the input data in parallel. And the respective results are summarized to the mimic arbiter， which is a voting module that employs a certain voting algorithm to generate voting results. The results indicate the status of every executor， whether the output data of the heterogeneous executors are consistent. In addition， the mimic arbiter feeds back the status to the scheduler. If the status show that the behavior of some online executors is different from the others， these online executors must be replaced by offline executors from the heterogeneous executor pool. The selection of online executors to be replaced and offline executors to be online is performed by a specific scheduling algorithm according to the current situation. And the executors to be replaced should be cleaned and perform recovery operations. Each executor in the heterogeneous executor pool is composed of elements that belong to different component sets. The different distribution of these elements in each executor results in heterogeneous executor pool with different structures and functionally equivalent executors. The heterogeneity， dynamics， and redundancy of DHR provide the system with uncertainty in time and space， making it difficult for attackers to exploit the vulnerabilities of the system， which in turn makes the system possess endogenous defense characteristics and natural immunity.

2.2 Mimic architecture in SDN

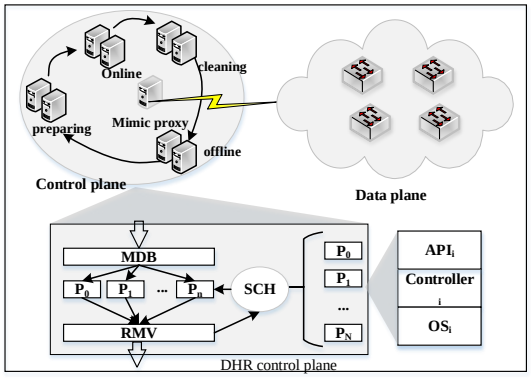


Figure 2 The DHR architecture of control plane in SDN

In order to solve the SDN control plane security problem， the DHR structure is introduced into the SDN control plane， which is called mimic control plane. The architecture is shown in Figure 2. Mimic proxy is introduced to build the DHR architecture. The functions of mimic proxy are message distribution and balancing， executor resource scheduling， and redundant message voting. The heterogeneous executor pool includes all control plane resources， and each executor is an entity with control plane functions， including APIs， controllers， and operating systems. The heterogeneity of each executor is mainly reflected in the diversity of components such as APIs， controllers， and operating systems.

Message dispatching and balancing module （MDB） is the entry point for the data plane to send

messages to the control plane， copying and distributing request messages for equivalent heterogeneous executors. Heterogeneous and redundant controlling pool （HRCP） is the function service provider of the control and management plane， and each executor has a corresponding initial state. Redundant message voter （RMV） provides the data message channel from the control plane to the data plane and executor status information. On the one hand， RMV judges the output results of heterogeneous executors with equivalent functions， and sends data to the data plane according to the voting algorithm； on the other hand， RMV feeds back the inconsistent state of the output data of executors to the scheduler （SCH）.The SCH detects the running status of the executor and the feedback information of the RMV， and performs operations such as offline cleaning， synchronizing the executor， and scheduling to be online according to the specific scheduling algorithm. Mimic architecture in SDN has significantly improved the security of the control plane.

2.3 Mimic model in SDN and heterogeneity analysis

As shown in Figure 2， we model the mimic control plane in SDN and analyze the working mechanism of the mimic architecture.

Definition 1 (Equivalent executor)

Equivalent executor （executor for short） refers to an entity that can independently provide control plane function services， denoted as P_i.

$\exists i$， $j \in$ {1， 2， …}， if P_i and P_j are exactly the same， that is， each executor not only has the same function but also the same structure， then $P_i = P_j$. On the contrary， if they have different structure， it is recorded as $P_i \neq P_j$. The set of mimic control plane executors is denoted as $\mathbf{P} = \{P_i | P_i$ is an executor， and $i = 1, 2, ..., n\}$.

When a set of heterogeneous executors encounter legal or illegal service requests from the data plane， the heterogeneity among executors can improve the reliability and security of the mimic control plane. This section focuses on the impact of the heterogeneity of executor sets on the reliability and security of the entire control plane.

Definition 2 (Vulnerability set)

The set of all vulnerabilities in the executor P_i， denoted as $\mathrm{VUL}_i = \{\mathrm{VUL}_{ij} | \mathrm{VUL}_{ij}$ is a vulnerability in the component j on the executor P_i， and $j = 1, 2, \cdots, m\}$.

If the VUL_i of P_i is the same as the VUL_q of P_q， then $\mathrm{VUL}_i = \mathrm{VUL}_q$； if vulnerabilities in VUL_i and VUL_q are completely different， then $\mathrm{VUL}_i \neq \mathrm{VUL}_q$. Otherwise， there is one or more groups that can be exploited in the same pattern， namely $\mathrm{VUL}_i \wedge \mathrm{VUL}_q \neq \square$， which is recorded as $\mathrm{VUL}_i \backsimeq \mathrm{VUL}_q$.

Definition 3 (Structure function of executor)

The output of the executor P_i is determined by the input excitation and the executor’ s structure. The executor structure function is set as f （*）.

Definition 4 (Attack process)

The attack process refers to the process in which the attacker exploits the vulnerability VUL_{ij} to attack the executor P_i and generates the abnormal output of f （*）， that is， $f(\mathrm{VUL}_{ij} \rightarrow P_i) = \mathrm{AO}_{ij}$.

In order to describe the attack situation of the mimic control plane， only the output results obtained by attacking the executor P_i by the attacker using the vulnerability VUL_i in the executor P_i vulnerability set VUL_{ij} when the executor is attacked， including abnormal output and data for tampering， downtime， etc.， the attack effect is recorded as AO_{ij}. If the attack is unsuccessful， the attack effect is recorded as Ø. If the attacker uses the vulnerability VUL_{ij} to attack the executors P_i and P_q， where $i \neq q$， the attack effect obtained is exactly the same， it is recorded as $\mathrm{AO}_{ij} = \mathrm{AO}qj$； if the attack effect obtained is different， it is recorded as $\mathrm{AO}_{ij} \neq \mathrm{AO}_{qj}$.

Definition 5 (Voting)

The voting process of RMV on the attack effect obtained by the attacker using the vulnerability VUL_{ij} to attack the executor set P is recorded as f_{arb}（AO_{1j}， AO_{2j}， …， AO_{nj}）， and the voting result is recorded as RES， where j=1， 2， …， m， f_{arb}（*） characterizes voting algorithms， such as majority-rule

voting, maximum likelihood voting, voting by consensus, large number voting based on historical information, weighted voting, mask voting, etc. This article discusses voting algorithms based on consensus.

1. In a homogeneous redundant system (redundant backup system), if all the executors in the executor set are exactly the same, then all the vulnerability sets are the same, that is, $VUL_1 = VUL_2 = \cdots = VUL_n$, attacks can be launched successfully when exploiting arbitrary vulnerability of the executor, and achieve the same attack effect, that is, $f(VUL_{1j} \rightarrow P_1) = f(VUL_{2j} \rightarrow P_2) = \cdots = f(VUL_{nj} \rightarrow P_n)$, $AO_{1j} = AO_{2j} = \cdots = AO_{nj}$, $j = 1, 2, \cdots, m$, RMV vote on the output of the executor, the voting result is expressed as follows:

$RES = f_{arb}(AO_{1j}, AO_{2j}, \cdots, AO_{nj}) = AO_{ij}$, $i = 1, 2, \cdots, n$.

Property 1 When an attacker launches an attack on a homogeneous redundant system, the attack target is any vulnerability in the vulnerability set, which will cause the consensus voting RMV to fail, and then cause the failure or error of the entire system.

2. Assuming an ideal heterogeneous redundant system, all the executors in the executor set P are completely orthogonal, expressed as $P_1 \perp P_2 \perp \cdots \perp P_n$. The so-called orthogonal means that there is no common vulnerability among the executors, that is $\{VUL_i \wedge VUL_j = \square \mid \exists i, j \in 1, 2, \cdots, N, \text{ and } i \neq j\}$. Therefore, an attacker can use the vulnerability to successfully attack the executor P_i, but cannot use the vulnerability to successfully attack the executor other than P_i. This case can be expressed as follows:

$f(VUL_{ij} \rightarrow P_i) = AO_{ij}$, $f(VUL_{ij} \rightarrow P_1) = NULL$, $(l \in 1, 2, \cdots, N \text{ and } l \neq i)$

RMV votes based on the attack effect, and the voting result RES can be shown as:

$RES = f_{arb}(AO_{1j}, AO_{2j}, \cdots, AO_{nj}) = NULL$, $i = 1, 2, \cdots, n$

Property 2 When an attacker launches an attack on an ideal heterogeneous redundant system, the attack target is arbitrary vulnerabilities and will not cause consensus voting failure. RMV can successfully block the attack against a single executor and ensure the normal operation of the system.

3. The ideal heterogeneous redundant system does not exist in the real environment, and the non-ideal redundant system has the symbiotic vulnerability VUL_{cov}. Assuming that VUL_{cov} exists in $\{P_1, P_2, \cdots, P_m, \text{ and } m<n\}$, the attacker uses the vulnerability VUL_{ip} to executor set **P** to attack, then all executors can be attacked successfully, and the attacker gets the same attack effect. We have

$f(VUL_{cov} \rightarrow P_1) = f(VUL_{cov} \rightarrow P_2) = \cdots = f(VUL_{cov} \rightarrow P_m) = AO_{cov}$;

$f(VUL_{cov} \rightarrow P_{m+1}) = f(VUL_{cov} \rightarrow P_{m+2}) = \cdots = f(VUL_{cov} \rightarrow P_n) = NULL$.

When $m > \left\lfloor \frac{n+1}{2} \right\rfloor$, according to the consensus voting algorithm, the RMV voting result is $RES = f_{arb}(AO_{1j}, AO_{2j}, \cdots, AO_{nj}) = AO_{cov}$, and the attack is successful;

When $m < \left\lfloor \frac{n+1}{2} \right\rfloor$, according to the consensus voting algorithm, the RMV voting result is $RES = f_{arb}(AO_{1j}, AO_{2j}, \cdots, AO_{nj}) = NULL$, and the attack is not successful;

In addition, in a real environment, when the attack vulnerability is a non-symbiotic vulnerability, the attack effect is the same as that of an ideal heterogeneous executor of Property 2, and cannot be successful.

Property 3 The attacker launches an attack on a real heterogeneous redundant system. When the attack target is a symbiotic vulnerability, if the number of executors of the symbiotic vulnerability is greater than the number of other executors in the heterogeneous system, the consensus voting will fail, leading to RMV Failure and causing failure or error of the entire system; when the target of the attack is symbiotic vulnerabilities and the number of executors with symbiotic vulnerabilities is less than the number of other executors, or when the target is non-symbiotic vulnerabilities, it will not cause consensus vot-

ing failure， RMV can successfully block attacks against a single or multiple executor with symbiotic vulnerabilities and ensure the normal operation of the system.

When constructing a mimic control plane in SDN， there is no guarantee that the controller and operating system must not have symbiotic vulnerabilities， that is， the ideal heterogeneous redundant system does not exist. Therefore， only property 3 conforms to the real application scenario of the mimic control plane. The greater the heterogeneity among the executors in the system， the less likely the existence of symbiotic vulnerabilities， and the lower the probability of successfully attacking the system， that is， the higher the security of the system. Therefore， the heterogeneity can be used as one of the indicators for evaluating the safety of mimic defense systems.

The mimic control plane selects functionally equivalent heterogeneous components at the application layer， controller layer， and operating system layer to form functionally equivalent heterogeneous executors. in principle， the same executors will not be selected in engineering practice.

3 Quantitative analysis of heterogeneity of mimic control plane

In the preceding section， we have analyzed the reasons why heterogeneity can improve the security of the mimic control， and the heterogeneity can be regarded as an indicator to quantify its security. In this section， the definition of heterogeneity is given， and two quantitative methods of heterogeneity are proposed on the basis of analyzing the existing quantitative methods.

3.1 Definition of heterogeneity

Literature ［29］ describes heterogeneity as complexity and difference based on biodiversity， proves and deduces about this definition. Intuitively， as shown in Figure 3， there are four typical scenarios for the heterogeneity of the population. The scenario A has a single type without complexity and difference， and the scenario B has many types， but the difference is small； the difference is huge in Scenario C， but the complexity is small； scene D has both complexity and difference.

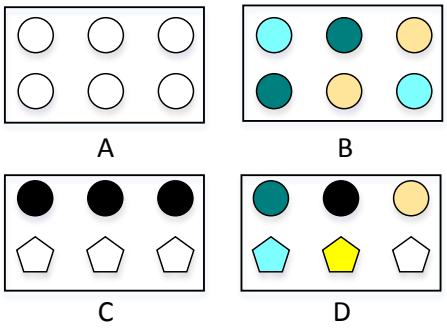


Figure 3 Schematic diagram of heterogeneity

Through the analysis of the mechanism of the mimic control plane in the previous section， it can be seen that the heterogeneity increases the difficulty of attacking in the mimic system and improves the security of the mimic system. However， the current mimic control plane and even all mimic defense systems lack standardized and unified conventions and definitions for heterogeneity. Here， we learn from the research methods of biological population diversity to study the heterogeneity of the heterogeneous executors of the mimic system. On the one hand， the mimic control plane executor set is heterogeneous and redundant， and the type of executor is not unique. Therefore， when defining the heterogeneity of the mimic control plane， the complexity of the executor type must be reflected； on the other hand， the complexity of the types of executors cannot explain whether there are same vulnerabilities among the executors， so it is necessary to further reflect the difference in the threat of known and unknown vulnerabilities between two heterogeneous executors.

In addition， when there are no less than 5 executors online for operation， the difference between two executors cannot fully reflect the impact of heterogeneity on the security of the mimic system ［6］. the high-order symbiosis of vulnerabilities can be used to define this feature. Therefore， to measure the heterogeneity of a mimic system， two indicators are needed. heterogeneity 1 can be expressed based

on the literature [29] using complexity and difference. The heterogeneity 1 of the mimic control plane is denoted as *HET1*, the complexity of the mimic control plane is denoted as *C*, and the difference of the mimic control plane is denoted as *FD*. Its heterogeneity, complexity, and difference have the following relationship $HET1 = C \times FD$. Because heterogeneity 1 cannot reflect the harm of multi-executor symbiotic vulnerabilities to the system, heterogeneity 2 (*HET2*) can be used to measure the heterogeneity of the system through the high-order symbiosis (*HOS*) of system vulnerabilities.

3.2 Formal description of executor

As the main carrier of the entire control plane, the executor set of the mimic control plane is composed of multiple heterogeneous executors. Each executor is composed of many components. The following is a further formal description of the executor set.

Definition 6 (Component set)

A set of components that implement a certain type of function of the SDN control plane system. This set must cover all functional components of this type of all executors of the mimic control plane, denoted as $L_k = \{L_{kj} | j = 1, 2, \cdots, m\}$.

All elements in the component set must meet the requirements of functional equivalence. If the controller is regarded as a type of functional equivalent, this type of component set can be defined as {ONOS, NOX, OpenDaylight, Beacon, Floodlight, Ryu, ...}; if the operating system is regarded as a functional equivalent, this type of component set can be defined as { {Centos x}, {Debian x}, {Ubuntu x}, {Red Hat x}, {Windows x}, {MAC OS x}, ⋯}, where x represents the system version.

Definition 7 (Feature matrix of executors)

The mimic control plane describes the feature matrix of executors and component set, which can be denoted as

$$\mathbf{C} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{1n} \\ \cdots & \cdots & \cdots & \cdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{pmatrix}$$

Among them, c_{i1}, c_{i2}, ⋯, c_{in} represents the same type of components with equivalent functions, and group c_{1j}, c_{2j}, ⋯, c_{mj} represents a collection of different types of components that form a single executor. Here, each eigenvalue has no mathematical meaning, and only represents whether each component of each executor is the same. Assume that the mimic control plane consists of three heterogeneous executors. The components of executor 1 are {OpenDaylight, ubuntu 12}^{T}, the components of executor 2 include {Ryu, Red Hat 7}^{T}, and the components of executor 3 include {Floodlight, ubuntu 12}^{T}. Then the feature matrix is

$$\mathbf{C} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 1 \end{pmatrix}$$

The feature matrix has the following two properties:

Property 4 The eigenvalues of the feature matrix C of the mimic control plane executor set are only for characterization. So arithmetic operations or matrix operations cannot be performed.

Property 5 The feature matrix of the mimic control plane can be interchanged in any column, but row interchange is not allowed.

Definition 8 (Abundance eigenvector)

The feature vector formed by the proportion of each component of the component set in the heterogeneous executor, denoted as $\mathbf{p_k} = (p_{k1}, p_{k2}, ..., p_{ks})^T$, we call p_{ki} the abundance eigenvector of the component set L_k, which represents all the same component abundances as c_{ki} [22], the eigenvalues of the abundance eigenvectors have the following relationship

$$\sum_{i=1}^{S} p_{ki} = 1$$

Suppose the feature matrix of the executor set is

$$\mathbf{PM} = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 1 & 3 \\ 2 & 2 & 1 \end{pmatrix}$$

The abundance feature vectors of its components are $\mathbf{p_1} = \left(\frac{1}{3} \quad \frac{2}{3}\right)^T$, $\mathbf{p_2} = \left(\frac{1}{3} \quad \frac{1}{3} \quad \frac{1}{3}\right)^T$, $\mathbf{p_3} =$

$\left(\frac{1}{3} \quad \frac{2}{3}\right)^T$, respectively. The abundance feature vector represents the diversity of the components in the executor concentration and is the basis for evaluating the complexity of the system.

3.3 Complexity description

Based on the research method of biological population diversity [30], Shannon entropy can be used to estimate the complexity of the execution volume. Here, regardless of the attacker's factor, the complexity of the component set L_k can be expressed as

$$Comp_k = -\sum_{i=1}^{S} p_{ki} ln p_{ki} \tag{1}$$

The complexity of the executor set can be expressed by the feature vector **COMP** = $(\mathrm{Comp}_1, \mathrm{Comp}_2, ..., \mathrm{Comp}_s)^{\mathrm{T}}$.

In the mimic control plane, s represents the number of component types, and p_{ki} is the proportion of component i in component set L_k. For the mimic control plane, in extreme cases, when there is only one component, the Shannon entropy is 0. When each component uniquely exists in each executor in the executor set, the Shannon entropy reaches the maximum value lnS. Therefore, t the complexity is the highest for a certain type of component of the mimic control plane when the number of types of components is equal to the number of executors.

3.4 Difference description

In order to evaluate the security of the mimic control plane in SDN, we describe the difference of executors in executor set. In terms of the description for the difference of executors, the quantification method of quadratic entropy can be used for measurement [31]. The premise of quantifying the quadratic entropy is to calculate the square distance between each member of the population. Current research introduces quadratic entropy into the evaluation of system diversity. For the mimic control plane, the difference of the component set L_k can be expressed as:

$$FD_k = \sum_{j=1}^{S} \sum_{i=1}^{S} d_{kij}^2 p_{ki} p_{kj} \tag{2}$$

Among them, d_{kij} represents the difference between components, which must meet the constraints: $d_{kij} = d_{kji}$ and $d_{kii} = 0$. This parameter plays a key role in evaluating the difference, so d_{kij} needs to be defined according to the application scenario of mimic SDN.

Starting from the original intention of evaluating the mimic control plane, the defined heterogeneity must be able to quantify the difficulty of resisting attackers to exploit vulnerabilities, and d_{kij} must be able to reflect the difference in vulnerabilities between components. The vulnerabilities mentioned here include known and potentially unknown vulnerabilities.

$$d_{kij} = 1 - (\alpha \times \xi_{kij} + (1 - \alpha) \times \zeta_{kij}) \tag{3}$$

Among them, α is the ratio of known vulnerabilities to all vulnerabilities ($0 \leqslant \alpha \leqslant 1$), which is called the vulnerability discovery coefficient, ξ_{kij} represents the characteristic value of known symbiotic vulnerabilities, and ξ_{kij} represents the feature value of unknown symbiotic vulnerabilities. The symbiotic vulnerabilities have been discussed in the literature [32]. Most of the symbiotic vulnerabilities are caused by code reuse, usually characterized by code reuse rate.

In the initial stage, α changes with the degree of recognition of the system, starting from 0 and increasing with time; with the recognition of components, it gradually increases, and finally approaches to 1. This change trend is similar to exponential distribution, so exponential function can be used to characterize this parameter.

$$\alpha = 1 - e^{-\lambda(t)} \tag{4}$$

$\lambda(t)$ represents the degree of people's awareness of the component over time, and this function changes slowly over time.

The known symbiotic vulnerability feature value ξ_{kij}, can be evaluated according to the degree of vulnerability based on NVD database and CVSS standard [33]. CVSS establishes a standard for measuring the severity of vulnerabilities based on a series of dimensions. The scoring range of vulnerabilities is 0-10. The higher the score, the greater the damage of the vulnerability.

$$\xi_{kij} = \frac{\sum_{t} vul'_{kij}(t)}{\sum_{t} vul_{ki}(t) + \sum_{t} vul_{kj}(t) - \sum_{t} vul'_{kij}(t)} \quad (5)$$

Among them, t represents the vulnerability of a single component, vul'_{kij} uses the CVSS score of each symbiotic vulnerability of component i and j for quantification, $vul_{ki}(t)$ and $vul_{kj}(t)$ are quantized respectively by the CVSS score of vulnerabilities of the component i and j. The component differences are characterized by ξ_{kij}according to the CVSS score of each known vulnerability. If components i and j have the same vulnerabilities, ξ_{kij}=1; otherwise the components i and j are completely heterogeneous, ξ_{kij}=0.

Executor difference is used to characterize the difference between executors by quantifying the characterization of unknown vulnerabilities and symbiotic vulnerabilities. When there is only one component set of a certain type of executor, d_{kij}=0, and FD_{Qk} takes the minimum value; with the increase of d_{kij}, the difference of executor set increases gradually. When there is only one type of multiple types of components in the executor, the quadratic entropy degenerates to the Simpson index, and FD_{Qk} takes the maximum value (1-1/S).

3.5 Quantification of heterogeneity 1

Considering the complexity and difference, the heterogeneity 1 (*HET1*) of the executor set of the mimic control plane can be calculated by the heterogeneity of the component set L_k of the M-type executor set, the formula is as follows:

$$HET1 = \sum_{k=1}^{M} COMP_k \times FD_{Qk} \quad (6)$$

Property 6 If the executors of the executor set $P_1 = P_2 = \cdots = P_n$, the heterogeneity of the executor set reaches the minimum value $HET1_{min}$=0; if $P_1 \neq P_2 \neq \cdots \neq P_n$ are satisfied, any two components of the same kind of the executor set are set. If $L_{ki} \neq L_{kj}, i \neq j$ is satisfied, the heterogeneity of the executor set reaches the maximum value, the formula is as follows:

$$HET1_{\max} = M(1 - \frac{1}{S})\ln S \quad (7)$$

In this paper, quantification of heterogeneity 1 is based on quadratic entropy, which is similar to the application scenario of reference [29], so the proof of this formula will not be repeated here.

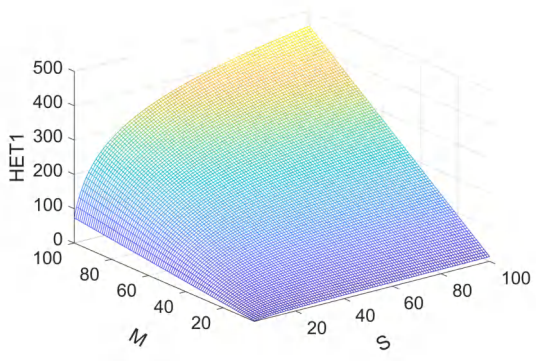


Figure 4 $HET1_{max}$ of the mimic control plane

Heterogeneous redundant executors are selected based on the complexity and difference among the candidate executors and their components, so that the heterogeneity of the executors is maximized. Ideally, the relationship between the heterogeneity of the mimic control plane and the types of components (i.e., S) and the number of executors (i.e., S) is shown in Figure 4.

3.6 Quantification of heterogeneity 2

The quantification of heterogeneity 1 takes into account the complexity and differentiation between two executors, and to a certain extent reflects the degree of heterogeneity of the mimic system [27]. However, considering the DHR characteristics of mimic SDN, the difference between two executors cannot directly reflect the security situation of the system when more than two executors in the mimic control plane have symbiotic vulnerabilities. Therefore, here we define another quantitative indicator of system heterogeneity based on the high-order symbiosis of vulnerabilities: *HET2*.

Definition 9 (High-order Symbiosis of Vulnerabilities)

When there are vulnerabilities in different executors in the executor set that can achieve the same attack effect, and the number of executors that meet this situation is m, it is said that such vulnerabilities have m' th-order symbiosis. When $m \geqslant 3$, it is collec-

tively referred to as high-order symbiosis of vulnerabilities.

On the one hand， high-order symbiosis vulnerabilities depend on the common vulnerabilities of different components. On the other hand， they also depend on whether different executors contain the same components. If no less than three executors contain the same components， their corresponding vulnerabilities are transformed into high-order symbiotic vulnerabilities.

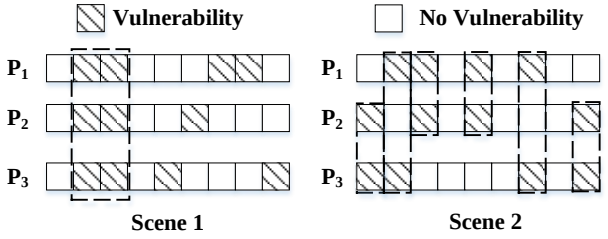


Figure 5 Schematic diagram of high-order symbiosis of vulnerabilities

For example， as shown in Figure 5， it is assumed that there are three equivalent heterogeneous executors P_1， P_2， P_3. The vulnerabilities in each executor are shown in Figure 5. The executor can be compared to a chromosome， and resources with vulnerabilities and those without vulnerabilities can be compared to two kinds of gene fragments. Assuming that P_1， P_2， and P_3 all have 4 vulnerabilities， the gene fragments corresponding to the vulnerabilities are different. There are two high-order symbiotic vulnerabilities in scenario 1， and there is no high-order symbiotic vulnerability in scenario 2.In these two situations， if the number of executors in the executor set is greater than 5， theoretically scenario 1 has a risk of being attacked ， while scenario 2 has no risk of being attacked. However， if measured according to the above-mentioned quantitative indicators of the complexity and difference of the executor， the heterogeneity of the two scenarios is equal.

The reason is that the quantification process of isomerism 1 only considers the performance of the heterogeneous executors in the isomer concentration. According to the voting properties of the mimic control plane， when the number of online executors is （$2f$+1）， it will cause the vote to fail if more than m （$m \geqslant f$） executors have the same output result due to symbiotic vulnerabilities. The attack will be successful. Therefore， it is necessary to consider the impact of high-order symbiotic vulnerabilities on heterogeneity. For m executors， the mathematical expression of the m’ th order symbiotic of the vulnerability， the heterogeneity quantification 2， is as follows：

$$hos_k^m = \frac{\sum_t vul_k^m(t)}{\sum_{i=1}^{m}\sum_t vul_{ki}(t) - (m-1)\sum_t vul_k^m(t)} \tag{8}$$

Among them， t represents the vulnerability of a single component； vul_k^m represents the total CVSS score for symbiotic vulnerabilities of component k in m executors； $vul_{ki}(t)$ represents the total CVSS score for the vulnerabilities of component k in executor i. In the extreme case， if component k in m executors are exactly the same， then$hos_k^m = 1$； otherwise， the m’th-order symbiotic vulnerability is 0， $hos_k^m = 0$. We can get the symbiosis evaluation expression of m high-order symbiotic vulnerabilities of n executors with component k：

$$HOS_k^m = \sum_{j=1}^{O}\frac{1}{O}hos_k^m|_j \tag{9}$$

We select m executors from n executors to form the executor groups. In formula （9）， j stands for the number of the executor groups. Orepresents the total number of executor groups， and $O = \binom{m}{n}$， $m \geqslant 2$.

The m’ th-order symbiosis of vulnerabilities in all the components of the executors， HOS， can be expressed as：

$$HOS = \sum_{i=1}^{S}\mu_i HOS_i^m \tag{10}$$

Among them， μ_i represents the proportion of the component i in the total number of vulnerabilities of this type of component in all executors.

According to formulas （8）-（10）：

$$HOS = \sum_{i=1}^{S}\sum_{j=1}^{O}\frac{1}{O}\frac{\mu_i\sum_t vul_k^m(t)}{\sum_{i=1}^{m}\sum_t vul_{ki}(t) - (m-1)\sum_t vul_k^m(t)} \tag{11}$$

When the overall vulnerabilities of various com-

ponents of the executors are fixed， the more high-order symbiotic vulnerabilities， the bigger the HOS. the more vulnerabilities the attacker can exploit， the lower the system security.

Therefore， the ratio of non-symbiotic vulnerabilities to total vulnerabilities can be used to characterize *HET2*， which can be obtained by formulas （8）-（11）：

$$HET2 = \sum_{i=1}^{S}\sum_{j=1}^{O}\frac{1}{O}\frac{\mu_i\left(\sum_{i=1}^{m}\left(\sum_{t} vul_{ki}(t) - \sum_{t} vul_{k}^{m}(t)\right)\right)}{\sum_{i=1}^{m}\sum_{t} vul_{ki}(t) - (m-1)\sum_{t} vul_{k}^{m}(t)} \tag{12}$$

In addition， the larger the *HET2* and the smaller the HOS， the less likely the system will be successfully attacked. Assuming that the probability of the system being successfully attacked is P_e and the system security is A， then $A \propto 1/P_e$. Therefore， *HET2* has following properties：

Property 7 When the overall vulnerability of the mimic control plane is certain， *HET2* is negatively related to the probability of the system being successfully attacked， and positively related to the security of the system， denoted as HET2$\propto 1/P_e$， HET2$\propto A$.

When the number of online executors is （$2f+1$）， the m’th-order （$m \geqslant f$） symbiosis of vulnerabilities becomes a key factor in the security of the mimic control plane. When the number of online executors on the mimic control plane changes， the security of the system needs to be re-evaluated according to the quantitative index *HET2* of the heterogeneity of the online executors.

4 Simulation and experimental evaluation analysis

MATLAB R2016a is taken as the simulation platform for evaluating heterogeneity indicators *HET1* and *HET2*. We evaluate the influence of the executor feature （for example， number of executors， vulnerability distribution， high-order symbiosis of vulnerabilities， etc.） on two quantitative indicators. In addition， an example of a mimic control plane is built in the virtual machine VMWare environment to evaluate heterogeneity. The vulnerability information of the operating system and controller is collected and obtained from literature ［34］ and literature ［35］.

4.1 Evaluation of *HET1*

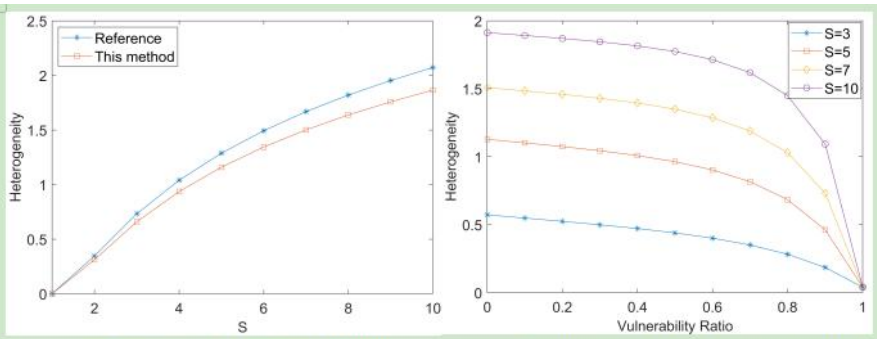
According to the analysis above， there are many factors that affect *HET1*， including the type and number of executor components， symbiotic vulnerability ratio， code reuse rate， and vulnerability discovery coefficient. It should be noted that every factor has a default value respectively in order to simplify the analysis. The following is a brief analysis of factors affecting heterogeneity in simulation：

1. The type and number of executor components are the functional entities of the mimic control plane， as described in this work. In the simulation， the component type and the number of executors is set to the same value in order to facilitate analysis. Less than 10 mimic control plane scenarios are evaluated in default.

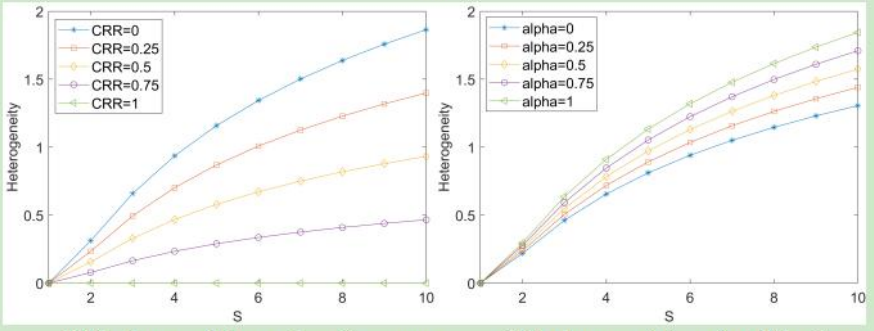
2. Symbiotic vulnerability ratio refers to the ratio of symbiotic vulnerabilities of different executors to all online executors’ vulnerabilities， which reflects the distribution of symbiotic vulnerabilities of executors. In the simulation， the symbiotic vulnerability ratio between each executor is set to the same value in default， which is set to 0.1.

3. The code reuse rate refers to the code reuse ratio of different executors， which reflects the distribution of symbiotic vulnerabilities of the executors. The evaluation method of code reuse rate for specific control plane can be referred to literature ［32］. In the simulation， the symbiotic vulnerability ratio between each executor is set to the same value， which is set to 0.2 in default.

4. The vulnerability discovery coefficient （α） reflects the ratio between the known vulnerabilities of the executor and all the vulnerabilities that may exist including unknown vulnerabilities. It is calculated according to formula （4）. As time goes on， the coefficient gradually becomes larger until approaches to 1. In the current simulation， the value is set to 0.6 in default.



(a)Comparison of *HET1* and *HBCV* (b)The impact of symbiotic vulnerability ratio



(c)The impact of the number of executors (d)The impact of the vulnerability discovery coefficient

Figure 6 *HET1* of the mimic control plane

As shown in Figure 6 （a）， in order to compare with the *HBCV* algorithm in ［27］， the existing common vulnerabilities is not considered. It can be seen from the figure that the reference literature evaluates the ideal heterogeneous executor， and its heterogeneity reaches the maximum. *HET1* is less than the heterogeneity of *HBCV*. The reason is that the evaluation criteria of this article consider the possibility of unknown vulnerabilities caused by code reuse. That is， when there are no symbiotic vulnerabilities in heterogeneous executors， unknown vulnerabilities may be introduced due to code reuse. *HET1* in this article can better reflect the true heterogeneity.

As shown in Figure 6 （b）， when the symbiotic vulnerability ratio is 0， the heterogeneity is the largest； when the symbiotic vulnerability ratio is 1， the heterogeneity is the smallest in the current cognitive situation.

As shown in Figure 6 （c）， the greater the degree of code reuse rate （CRR）， the lower the heterogeneity. When the code reuse rate is 1， it is equivalent to a homogeneous system， and the heterogeneity is 0； when the code reuse is 0， the heterogeneity is higher than the other situation.

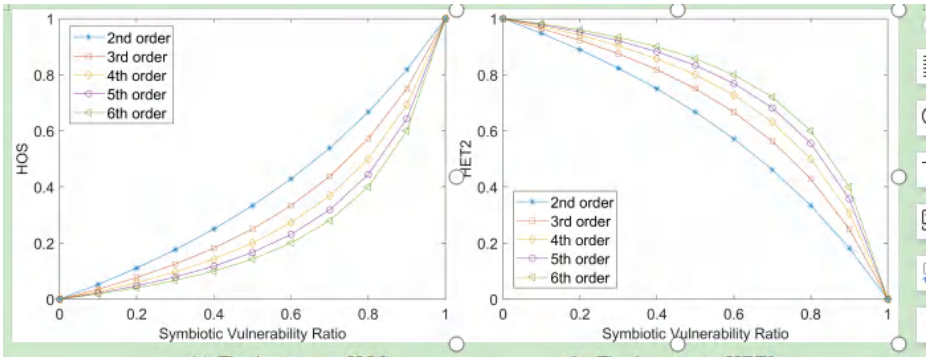
As shown in Figure 6 （d）， with the increase of heterogeneous executors， the heterogeneity of the mimic control plane gradually increases. when heterogeneity equals to 1， there is no symbiotic vulnerability. In this situation， the heterogeneity is maximized. In the case of equal executors， with the recognition of code reuse， heterogeneity is reduced. This is because code reuse may lead to symbiotic vulnerabilities， which reduce the heterogeneity of the system.

The above experimental results show that when the system is ideally heterogeneous， the method of literature ［27］ can be used for evaluation. However， in reality， the type and number of executor components of the mimic control plane， symbiotic vulnerability ratio， component code reuse rate， and vulnerability discovery coefficient all have a direct impact on the heterogeneity of the mimic control plane， so *HET1* is more intuitive and reasonable.

4.2 Evaluation of *HET2*

When evaluating heterogeneity 2 (*HET2*), in order to simplify the analysis, the symbiotic vulnerability ratio can be unified. Set the symbiotic vulnerability ratio of each executor to the same value, and evaluate the impact of different symbiotic vulnerability ratios on the *HOS* of different high-order symbiotic vulnerabilities.

As shown in Figure 7, as the vulnerability ratio increases, *HOS* gradually increases. The higher the order of symbiosis, the lower the *HOS*. When the vulnerability ratio is 0 and 1, *HOS* is 0 and 1, respectively. Moreover, as shown in Figure 8, as the vulnerability ratio increases, *HET2* gradually decreases. The higher the symbiosis order, the larger *HET2*. When the vulnerability ratio is 0 and 1, *HET2* are 1 and 0 respectively. The simulation results show that if the SDN control executor has more symbiotic vulnerabilities, higher HOS and larger the HET2 can be achieved. The above simulation results also conform to common sense.



(a) The impact on *HOS* (b) The impact on *HET2*

Figure 7 The impact of symbiotic vulnerability ratio

4.3 Mimic control plane instance and quantitative analysis of heterogeneity

In this experiment, five different mimic control plane executor sets are selected, each of which includes 3 executors. Different operating systems and SDN controllers are selected as the executors of the mimic control plane. The vulnerability information of the operating system comes from the NVD database [34], and the method of obtaining the vulnerability information of the controller comes from the literature [35]. Since its CVSS score is not given, the default unified setting is 5, that is, the level of all vulnerabilities is medium.

All components in executor set 1, 2 and 3 are not the same. The controller of executors in executor set 4 is exactly the same, and the executors of executor set 5 are completely homogeneous system. The experimental results are shown in Table 1.

Table 1 Comparison of heterogeneous quantification results of mimic control plane

No.	Executor sets	*HBCV*	*HET1*	*HET2*
1	Ubuntu12+ Ryu OpenBSD+floodlight Win7+opendaylight	1.4363	1.4363	0.9865
2	Ubuntu12+ Ryu Redhat7+floodlight Win7+opendaylight	1.4021	1.3794	0.9742
3	Ubuntu12+ ONOS Redhat7+floodlight Win7+opendaylight	1.3763	1.3562	0.9806
4	Ubuntu12+ Ryu Redhat7+ Ryu Win7+ Ryut	0.7364	0.7213	0.6647
5	Ubuntu12+ Ryu Ubuntu12+ Ryu Ubuntu12+ Ryu	0	0	0

(1) In executor set 1, *HBCV* has the same heterogeneity as *HET1*, mainly because the operating system used by the components in executor set is a non-homologous system, so the code reuse rate is 0. In the absence of code reuse rate, the *HET1* evalua-

tion index in this article is the same as *HBCV*.

（2） Compared with executor set 2 and 1， the second executor was replaced by Redhat7 from OpenBSD. Redhat7 and Ubuntu12 are Linux homologous systems and have code reuse. Therefore， the heterogeneity of HET1 is lower than *HBCV*. At the same time， due to the introduction of Redhat7， symbiotic vulnerabilities have increased， *HET2* has become larger， and the security of executor set 2 is lower than that of executor set 1.

（3） Compared with executor set 3 and 2， the controller of the first executor is changed from Ryu to ONOS. Compared with Ryu， the vulnerabilities of ONOS have increased. Both *HBCV* and *HET1* have decreased， while *HET2* has increased. The main reason is that the symbiotic vulnerabilities in executor set 3 have increased. Higher *HET2* indicates more security， and executor set 3 is more secure than executor set 2. Combined with property 7， this group's comparison shows that the reduction of *HBCV* and *HET1* does not necessarily make the system more secure， and the system security needs to be evaluated in conjunction with HET2.

（4） Compared with executor set 4 and 2， the controller is a unified Ryu controller， its *HBCV/HET1/HET2* is greatly reduced， and the system security is greatly reduced.

（5） Executor set 5 is a completely homogeneous system， *HBCV/HET1/HET2* are all 0， heterogeneity 2 is the maximum value 1， and the system has the lowest security.

It can be seen from the example of the mimic control plane constructed above that， on the one hand， the *HET1* is different from *HBCV*. When the same software or operating system exists， the *HET1* mentioned is lower than *HBCV*. On the other hand， *HBCV* and *HET1* cannot fully reflect the security of the mimic control plane， but *HET2* can more accurately reflect the security of the system. Therefore， using *HET1* and *HET2* to comprehensively evaluate the heterogeneity of the mimic control plane can more accurately evaluate its security.

5 Conclusions and future work

The mimic control plane significantly increases the security of the SDN control plane， but how to effectively evaluate the security of the mimic control plane is a key issue that needs to be resolved. Based on previous studies， this paper draws on biodiversity， based on Shannon entropy and quadratic entropy， introduces the vulnerability attributes of the SDN control plane into quadratic entropy， and incorporates it into the quantitative model of *HET1* as a parameter for evaluating differences. And based on the high-order symbiosis of vulnerabilities， a quantitative model of *HET2* is constructed to effectively evaluate the security of the mimic control plane. The experimental results show that， compared with the heterogeneity quantification method *HBCV* based on known vulnerabilities， the two quantification models proposed in this paper are more targeted for the heterogeneity quantification of the mimic control plane， and can better reflect system security.

This article conducts a quantitative analysis of the proposed heterogeneity metrics and high-order symbiosis of vulnerabilities， but does not involve the scheduling algorithm and the impact of voting strategies on scheduling. Therefore， evaluating the security of the system in combination with the scheduling algorithm and voting strategy based on the attack attributes will be considered as future work.

References:

[1] MckeownN.， AndersonT， BalakrishnanH.， et al.， “OpenFlow：Enabling innovation in campus networks，" Acm Sigcomm Computer Communication Review，vol. 38，no. 2，pp. 69-74，2008. DOI：10. 1145/1355734. 1355746

[2] ZuoQ. Y.， ChenM.， DingK.， et al.， “On Generality of the Data Plane and Scalability of the Control Plane in Software-Defined Networking，” China Communications，vol. 11，no. 2，pp. 55-64，2014. DOI: 10. 1109/cc. 2014. 6821737

[3] LiuY. F.， ZhaoB.， ZhaoP. Y.， et al.， “A Survey：Typical Security Issues of Software-Defined Networking，" China Communications，vol. 16，no. 7，pp. 13-31，2019. DOI：10. 23919/JCC. 2019. 07. 002

[4] RehmanA. U.， AguiarR. L.， and BarracaJ. P.， "Fault-Tolerance in

the Scope of Software-Defined Networking (SDN)，" Ieee Access，vol. 7， pp. 124474-124490， 2019. DOI：10. 1109/access. 2019. 2939115

[5] Yang, and L. M. ChengY. B. "An SDN-based MTD model，" Concurrency and Computation-Practice & Experience，vol. 31，no. 21，2019. DOI: 10. 1002/cpe. 4897

[6] HuH. C. , WangZ. P. , ChengG. Z. , et al. ，"MNOS：a mimic network operating system for software defined networks, " IET Information Security，vol. 11，no. 6，pp. 345-355，2017. DOI：10. 1002/cpe. 4897

[7] AydegerA. , SaputroN. , and AkkayaK. ，"A moving target defense and network forensics framework for ISP networks using SDN and NFV," Future Generation Computer Systems-the International Journal of Escience，vol. 94，pp. 496-509，2019. DOI：10. 1016/j. future. 2018. 11. 045

[8] ChoJ. H. , SharmaD. P. , AlavizadehH. , et al. ，"Toward Proactive，Adaptive Defense: A Survey on Moving Target Defense ，" Ieee Communications Surveys and Tutorials，vol. 22，no. 1，pp. 709-745，2020. DOI: 10. 1109/comst. 2019. 2963791

[9] ZhengJ. J. , and NaminA. S. ，"A Survey on the Moving Target Defense Strategies：An Architectural Perspective, " Journal of Computer Science and Technology，vol. 34，no. 1，pp. 207-233，2019. DOI：10. 1007/s11390-019-1906-z

[10] ChiangM. L. , HsiehH. C. , and WangC. W. "Improving the Fault-Tolerance Under Software-Defined Network Based on New Sight of Agreement Protocol," Ieee Access，vol. 6，pp. 40898-40908，2018. DOI：10. 1109/access. 2018. 2859023

[11] SakicE. , DericN. , and KellererW. ，"MORPH：An Adaptive Framework for Efficient and Byzantine Fault-Tolerant SDN Control Plane," Ieee Journal on Selected Areas in Communications，vol. 36，no. 10，pp. 2158-2174，2018. DOI：10. 1109/jsac. 2018. 2869938

[12] YuanB. , JinH. , ZouD. Q. , et al. ，"A Practical Byzantine-Based Approach for Faulty Switch Tolerance in Software-Defined Networks，" Ieee Transactions on Network and Service Management，vol. 15， no. 2， pp. 825-839， 2018. DOI: 10. 1109/tnsm. 2018. 2822668

[13] HuH. C. , WuJ. X. , WangZ. P. , et al. ，"Mimic defense：a designed-in cybersecurity defense framework," IET Information Security，vol. 12，no. 3，pp. 226-237，2017. DOI：10. 1049/iet-ifs. 2017. 0086

[14] WangZ. P. , HuH. C. , and ChengG. Z. ，"A DNS Architecture Based on Mimic Security Defense，" Acta Electronica Sinica，vol. 45，no. 11，pp. 2705-2714，2017.

[15] ZhangZ. , MaB. L. , and WuJ. X. "Testing and Analysis of Web Server Mimic Defense Principle Verification System，" Journal of Information Security，vol. 2，no. 1，pp. 13-28，2017.

[16] QiC. , WuJ. X. , HuH. C. , et al. ，"An intensive security architecture with multi-controller for SDN，" Proceedings of the IEEE INFOCOM 2016 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)，2016，pp. 401-402.

[17] QiC. , WuJ. X. , ChengG. Z. , et al. ，"An aware-scheduling security architecture with priority-equal multi-controller for SDN，" China Communications，vol. 14，no. 9，pp. 144-154，2017. DOI: 10. 1109/CC. 2017. 8068772

[18] QiC. , WuJ. X. , HuH. C. , et al. ，"Dynamic-scheduling mechanism of controllers based on security policy in software-defined network，" Electronics Letters，vol. 52，no. 23，pp. 1918-1920，2017. DOI: 10. 1049/el. 2016. 2670

[19] LiJ. F. , LanJ. L. , HuY. X. , et al. ，"Quantitative approach of multi-controller's consensus in SDN，"Journal of Communicaiton. vol. 37，no. 6，pp. 86-93，2016.

[20] LiJ. F. , WuJ. X. , HuY. X. , et al. ，"DSL：Dynamic and Self-Learning Schedule Method of Multiple Controllers in SDN, " Etri Journal，vol. 39，no. 3，pp. 364-372，2017. DOI：10. 4218/etrij. 17. 0116. 0460

[21] LiC. H. , RenY. F, TangZ. Y. , et al. ，"Mimic defense method for service deployment in SDN，" Journal of Communicaiton，vol. 39，no. 2，pp. 121-130，2018.

[22] ShaoC. S. , LouW. , and YanL. M. "Optimization of Algorithm of Similarity Measurement in High-Dimensional Data，" Computer Technology and Development，vol. 21，no. 4，pp. 1-4，2011.

[23] MitraP. , MurthyC. , and PalS. K. "Unsupervised feature selection using feature similarity，" IEEE transactions on pattern analysis machine intelligence，vol. 24，no. 3，pp. 301-312，2002. DOI: 10. 1109/34. 990133

[24] PincusS. M. "Approximate entropy as a measure of system complexity，" Proceedings of the National Academy of Sciences of the United States of America，vol. 88，no. 6，pp. 2297-2301. ，1991. DOI: 10. 1073/pnas. 88. 6. 2297

[25] SalakoK. , and StriginiL. ，"When Does" Diversity"'in Development Reduce Common Failures? Insights from Probabilistic Modeling，" IEEE Transactions on dependable secure computing，vol. 11, no. 2，pp. 193-206，2013. DOI: 10. 1109/TDSC. 2013. 32

[26] LiuQ. R. , LinS. J. , and GuZ. Y. "Heterogeneous redundancies scheduling algorithm for mimic security defense，" Journal of Communicaiton，vol. 39，no. 7，pp. 188-198，2018.

[27] ZhangJ. X. , PangJ. M. , and ZhangZ. ，"Quantification Method for Heterogeneity on Web Server with Mimic Construction，" Journal of Software，vol. 31，no. 2，pp. 564-577，2020.

[28] J. X. Wu，"Research on cyber mimic defense, " Journal of Cyber Security，vol. 1，no. 4，pp. 1-10，2016.

[29] Twu YP. . Mostofi, and M. Eerstedt，"A measure of heterogeneity in multi-agent systems，" proceedings of the 2014 American Control Conference， 2014， pp. 3972-3977. DOI: 10. 1109/ACC. 2014. 6858632

[30] M. O. Hill，"Diversity and evenness: a unifying notation and its consequences," Ecology，vol. 54，no. 2，pp. 427-432，1973.

[31] C. R. Rao，"Diversity and dissimilarity coefficients: a unified approach," Theoretical population biology，vol. 21，no. 1，pp. 24-43，1982. DOI: 10. 1016/0040-5809(82)90004-1

[32] ShahzadM. , ShafiqM. Z, and LiuA. X. "Large scale characterization of software vulnerability life cycles，" IEEE Transactions on Dependable Secure Computing，vol. pp. 1-14，2019. DOI: 10. 1109/TDSC. 2019. 2893950

[33] MellP. , ScarfoneK. , and RomanoskyS. ，"Common Vulnerability

Scoring System，" IEEE Security Privacy，vol. 4，no. 6，pp. 85-89，2006.

[34] GarciaM. , BessaniA. , GashiI. , et al. ，"Analysis of operating system diversity for intrusion tolerance，" Software-Practice & Experience，vol. 44，no. 6，pp. 735-770，2014. DOI: 10. 1002/spe. 2180

[35] LeeS. , KimJ. , WooS. , et al. ，"A comprehensive security assessment framework for software-defined networks，" Computers & Security，vol. 91，2020. DOI: 10. 1016/j. cose. 2020. 101720

About the authors

ZHANG Wenjian ［corresponding author］ was born in Shangqiu， Henan， China， in 1987. He received the M. S. degrees in Communication and Information System from Information Engineering University， Zhengzhou， Henan， China， in 2013， where he is currently pursuing the Ph. D. degree in Information and Communication Engineering from Information Engineering University. His research interests include information security， network programmable design， integrated circuit design （wenjian0509@163. com）.

LIU Qinrang was born in Shangqiu， Henan， China， in 1975. He received the Ph. D. degree in Information and Communication Engineering from Information Engineering University， Zhengzhou， Henan， China， in 2004. He is currently a Research Professor in Information Engineering University. Research interests include Integrated circuit design technology and information security VLSI/SoC， Network architecture and network security technology.

Ke Song was born in Zhengzhou， Henan， China， in 1976. He received the Ph. D. degree in Computer Science and Technology from Fifty-sixth Institute， Wuxi， Jiangsu， China， in 2020. He is currently an associate researcher in Information Engineering University， Zhengzhou， China. His research focuses on Integrated circuit design technology and information security VLSI/SoC， Network architecture and network security technology.

WEI Shuai was born in Dengzhou， Henan， China， in 1984. He received the Ph. D. degree in High performance computing and Parallel Compiling from Information Engineering University， Zhengzhou， Henan， China， in 2012. He is currently a Research Assistant in Information Engineering University. His research interests include high performance computing， cyber security， and machine learning.

MU Qing was born in Luoyang， Henan， China， in 1985. He received the M. S degrees in network and software engineering from Information Engineering University， Zhengzhou， Henan， China， in 2012. He is currently a Research Assistant in Information Engineering University. His research interests include information security， quantum computation， integrated circuit design.

A Fast Physical Layer Security-Based Location Privacy Parameter Recommendation Algorithm in 5G IoT

ZHAO Hua[1], XU Mingyan[2], HUANG Kaizhi[1], ZHONG Zhou[1]

1.PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, China;

2.National Digital Switching System Engineering & Technological Research Center, Zhengzhou 450002, China

Abstract: The 5G IoT (Internet of Things, IoT) is easier to implement in location privacy-preserving research. The terminals in distributed network architecture blur their accurate locations into a spatial cloaking region but most existing spatial cloaking algorithms cannot work well because of man-in-the-middle attacks, high communication overhead, time consumption, and the lower success rate. This paper proposes an algorithm that can recommend terminal's privacy requirements based on getting terminal distribution information in the neighborhood after cross-layer authentication and therefore help 5G IoT terminals find enough collaborative terminals safely and quickly. The approach shows it can avoid man-in-the-middle attacks and needs lower communication costs and less searching time than 520ms at the same time. It has a great anonymization success rate by 93% through extensive simulation experiments for a range of 5G IoT scenarios.

Key words: Cross-layer Authentication; Location Privacy Parameter Recommendation; 5G IoT

1 Introduction

Location-based services （LBS）， which provide the 5G IoT terminals with highly personalized content customized by 5G terminals’ current locations， had become a popular application for devices capable of location detection. Since the terminals’ locations are essential to obtain LBS while the service provider is not trusted， reporting private location may lead to privacy threats. o preserve the terminal privacy， most of the location privacy protection focuses on k-anonymity [1]， which was defined as if a terminal’s location is indistinguishable from the location information of at least K-1 other terminals. The spatial region which contains the terminal and another k-1 additional terminals’ location is usually called the k-anonymity spatial region （K-ASR）. To tackle this major privacy concern， the centralized framework and distributed framework are proposed for LBS. Recent research on LBS privacy focuses on a distributed architecture that is suitable for 5g IoT scenarios [2]-[12]. The distributed architecture in which mobile terminals communicate among each other via one-hop and/or multiple routing and constructs K-ASRs by considering terminals in the neighborhood of the querying terminal is popular. However， there are many unique limitations， e.g.， terminal legitimacy， terminal mobility， constrained transmission range， scarce communication resource， and limited energy of mobile devices. Terminal suffers long delay and huge communication overhead in searching terminals and forming K-ASRs. And the neighborhood terminal may be an illegitimate user who can intercept the sensitive location information. Moreover， owing to the lack of background information about the scenarios terminal is not able to choose ap-

propriate privacy parameters to best satisfy the requirements of privacy protection. It lowers the anonymization success rate and degrades the quality of service.

In this paper， we propose a fast physical layer security-based location privacy parameter recommendation algorithm （PBPR） which provides three key features to enhance the security and efficacy of location privacy-preserving in 5G IoT environment. In general， the contributions of this paper can be summarized as follows：

1） We propose a “physical layer security-based cross-layer authentication protocol” for 5G IoT terminals to find collaborative terminals safely with preventing a man in the middle attack. （Section 4.1）

2） We propose a neighborhood-distribution-perceptive scheme for our PBPR algorithm to enable terminals to update and share their neighborhood-distribution parameter periodically to nearby terminals. It helps them perceive the real-time terminal density in the neighborhood. （Section 4.2）

3） We introduce a “privacy parameter recommendation scheme” in 5G IoT to provide mobile terminals with the optimal matching privacy parameter for the scenarios in order to strengthen the location privacy security and enhance the anonymization success rate. （Section 4.3）

4） After an analysis of security capability and compared with the other two spatial cloaking algorithms， we experimentally evaluate our PBPR algorithm with different scenarios. It provides protection from man-in-the-middle attacks and performs high anonymization success rate with less communication overhead and searching time. （Section 5）

The rest of this paper is organized as follows. Section 2 summarizes related works. Section 3 depicts our system model. Section 6 concludes this paper.

2 Related work

There is a wide literature on preserving terminal location privacy during the use of LBS. Some concepts， such as False location [13] [14]， Space transformation [15-16] and Spatial cloaking [1] [2] [17] [18]， are used in these techniques. A large portion of location privacy mechanisms is based on spatial cloaking techniques which are borrowed from k-anonymity. The main idea of the spatial cloaking is to blur q terminal’s location into a cloaked area that satisfies the terminal’s privacy requirements， i.e.， the cloaked area contains at least K terminals and minimum area Amin. There exist two types of the decentralized system. （1） The distribute framework in which mobile terminals establish point-to-point communication through a common infrastructure such as a base station （i.e.， the two terminals do not need to be within the range of each other） is widely used. For example， PRIVE [19] cluster terminals in a hierarchical overlay network based on the Hilbert space-filling curve and resembles a distributed B+-tree to support K-ASR construction. MOBIHIDE [20] form a hierarchical distributed hash table which indexes the locations of all terminals and the K-ASR are constructed by choosing random groups of k terminals. However， each mobile terminal needs to maintain a relatively complex data structure and communication protocol as well as long-range communication among terminals. Therefore additional computation and communication costs may be posed to mobile terminals with limited capabilities. （2） The other distributed system is the mobile P2P system in which mobile terminals communicate with each other via one-hop and/or multiple routing.

Chow et al. [2] are the first to apply the group formation technique to the cloak single terminal’s location. In Chow’s method， the terminal U starts the terminal searching by broadcasting a request to her neighbor terminals （the hop distance is one）. If U does not have at least k-1 terminals replying to the request she increases hop by one and broadcasts the request to the terminals within two-hop distance. U repeats the process until U finds enough terminals. This P2PCloak algorithm is simple and it’s applied in

many research of location preserving algorithm in mobile p2p environments [5] - [12]. The drawback to this approach is that the terminal suffers from long searching time and a low success rate. In [3], Che describes the dual-active (DA) approach that the terminals not only actively collect but also disseminate location information to others. Che's approach increases the anonymization success rate at the price of the high communicating cost to maintain the gathered location records.

To the best of our knowledge, the existing LBS privacy protection proposals for k-anonymity spatial cloaking in 5G IoT have the drawbacks as long delay, low success rate, and high communication overhead. In this paper, we focus on the K-ASR construction with shorter delay and higher success rate by means of detecting the terminal distribution with lower communication overhead. Moreover, the existing LBS privacy protection proposals do not take into the account legitimacy of neighborhood terminals. It is vulnerable to man-in-the-middle attacks, where the attacker can observe the communication between terminals and collect the private data transmitted.

3 System Model

The system model of our PBPR algorithm can be defined in terms of network infrastructures and main entities.

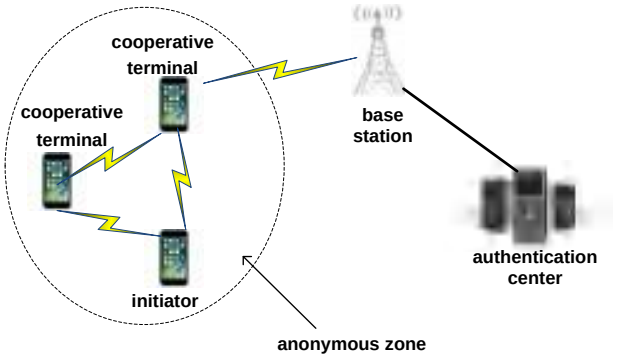


Fig.1 System model

As shown in Figure 1, there are two different types of network infrastructures in order to fulfill the location privacy-preserving in 5G IoT.

1) The wireless network via which terminal can communicate with the authentication center. The mobile communication network is most often used.

2) An Ad hoc wireless network. It is created on the fly when terminals need to exchange information among each other in order to find enough terminals.

The system consists of three types of entities.

1) A base station and an authentication center. With a cross-layer authentication processor embedded inside they have the ability to deal with physical layer random authentication parameters-based authentication.

2) 5G IoT terminals who need the LBS and want to be anonymized. They are equipped with two wireless network interface cards; one is used to connect to the mobile base station to communicate with the LBS provider, while the other one is dedicated to communicating with other terminals via multi-hop. Each terminal can also locate himself with a positioning device, e.g. GPS, or some other assortment of techniques such as propagation time, time difference of arrival, and angle of arrival.

3) A set of terminals work as cooperators. They share the same cloaked region as their location to get the information from the LBS provider.

4 Physical layer security-based location privacy parameter recommendation algorithm

This section presents the proposed physical layer

security-based location privacy parameter recommendation algorithm for location privacy-preserving in 5G IoT.

4.1 Cross-layer authentication protocol

All the neighborhood terminals involved in the traditional LBS privacy protection proposals are considered legitimate by default， and will not attack other terminals or systems by collecting the location information of cooperative terminals. In fact， there may be man-in-the-middle attackers in the 5G IoT. They are connected between legitimate terminals and base stations to intercept private information such as terminals' location data. The physical layer security and the existing 5G authentication mechanism can be combined to realize enhanced cross-layer authentication and ensure the legitimacy of terminals before the perception of neighborhood distribution.

The specific cross-layer authentication protocol is as follows： the authentication center obtains the terminal's identity information and group information during the terminal registration process. The terminal and the base station generate physical layer random authentication parameters through the physical layer key agreement mechanism. The base station reports these physical layer random authentication parameters to the authentication center. The authentication center uses the root key and physical layer random authentication parameters related to the terminal's identity to generate authentication data and sends it to the terminal. The terminal uses the root key， physical layer random authentication parameters， and authentication data to authenticate the base station and the authentication center， then generates terminal authentication data and sends it to the authentication center. The authentication center authenticates the terminal with the root key， physical layer random authentication parameters， and the terminal's authentication data. If the authentication is passed， it sends the group key to the terminal to encrypt subsequent neighborhood weighted density parameters and other information.

In the original high-level authentication， both the generated authentication message and the response message need to be transmitted in plaintext in the wireless channel， so that these messages can be obtained by a man-in-the-middle attacker. Since the security of high-level authentication is based on computational complexity， computers with high computational performance can further crack the authentication data. Therefore， as the frequency of authentication increases， the entropy of high-level key information will continue to decrease. Combining physical layer random authentication parameters with high-level authentication makes the physical layer channel information part of the authentication information， which essentially provides information entropy for authentication. Using the strong coupling and correlation between the physical layer random authentication parameters and the wireless link， combined with the identity of the terminal in the 5G IoT， the strong binding of the terminal identity and the wireless link can be realized， which can identify and suppress the "transparent forwarding" attack of the man-in-the-middle attacker in the 5G IoT， and ultimately ensure the security of the formation of the anonymous zone.

4.2 Neighborhood-Distribution-Perception Scheme

The terminal spatial distribution in 2 or 3-dimensional Euclidean space can be characterized by terminal density d which means the terminal number presented in a unit area. If D is defined as a continuous space， the mean user density expectationd^* can be estimated by maximum likelihood stated in （1）：

$$d^* = \overline{D} = \frac{1}{n}\sum_{i=1}^{n} D_i \tag{1}$$

whereD_i is the sample size of the terminal number in a unit area and$\{D_1, D_2, \cdots D_n\}$ is the set ofD_i. For a requiring terminal u in D， it is the terminal spatial distribution around terminal u that most influence the shape and size of the cloaking region， and hence， we propose the Neighborhood-distribution-perceptive scheme by formula （1）. To this end， we have the following definitions.

Definition. 1. The space around terminal u within

transmission range of h hop is defined as the *neighborhood zone* $D(u,h)$:

$$D(u,h)=\{y \mid y \in D, \|u-y\| \leqslant h\} \quad (2)$$

In particular, $D(u,1)$ is called the unit area where $h=1$.

Definition.2.

For a terminal u in D, the *terminal density in the neighborhood*, represented by d_u, is defined as the mean value of the terminal density in the $D(u,h)$. In particular, the terminal number in $D(u,1)$ is represented by D_u.

A Estimation of the terminal density in the neighborhood d_u

Every terminal in 5G IoT calculates and stores her d_u locally and sends it to terminals in $D(u,1)$ periodically. By this way terminal u can receive the d_i sent by the peer terminals in her $D(u,1)$ in which n_u is the number of d_i that u receives in a regular interval. This enables her to update d_u according to (1):

$$d_u=\frac{1}{n_u+1}\left(D_u+\sum_{i=1}^{n_u} d_i\right) \quad (3)$$

And every terminal u_i updates her new d_i periodically:

$$d_i=\frac{1}{n_i+1}\left(D_i+\sum_{j=1}^{n_i} d_j\right) \quad (4)$$

Where d_j is the terminal density in the neighborhood that u_j was received from her $D(u_j,1)$. Here we replace the d_i in (3) by (4) and d_j can be replaced in the same way based on the terminal densities in the neighborhood that u_j receives around her:

$$d_u=\frac{1}{n}\sum_{m=1}^{n}\rho_m D_m \quad (5)$$

Where n is the total terminal number in $D(u,h)$, D_m is $D(u_m,1)$ for terminal u_m and parameter ρ_m is defined as the distribution influence factor. The further the u_m is from u, the smaller the ρ_m is because D_m has to be iterated more times to arrive and be used by u. Through exchanging the d_u in the neighborhood zone, terminal u can collect the sample size D_m from remote terminals during the procedure of Neighborhood-Distribution-Perception. Compared with the terminals located far from u, terminals located closely with u are more likely to be the candidates of the cloaked area, and hence her ρ_m is much bigger in (5).

B Occasion and frequency of sending d_u

Sending d_u periodically enables terminals to get the real-time spatial terminal distribution at the price of a higher communication cost. In the case that spatial terminal distribution is relatively stable in D, the d_u of every terminal is most often steady. It is a waste of communication resources sending the d_u frequently in this situation. We suggest the strategy of updating and sending d_u. Every terminal keeps the latest d_i she received from the neighborhood in a local table. She also sets a regular timer for updating her d_u with the latest records periodically and broadcasts her d_u only when d_u changes or she detects there are new terminals that appeared in the neighborhood. In this way not only can terminals calculate the real-time d_u but also the communication cost is saved.

C Neighborhood-distribution-perceptive scheme design

Algorithm 1 depicts the main idea of the neighborhood-distribution-perceptive scheme.

For every terminal u we define the updating period as of t and local parameter table. During the updating period, if u receive the new d_i from direct neighbors or she detects the terminal change in the direct neighborhood, which results in the updating of the local parameter table, she will recalculate her d_u according to formula (3). At the end of every updating period if her d_u changes or there are new terminals coming up in the neighborhood she will broadcast her d_u encrypted by group key.

We define the broadcast message $m\_share=\{d_i\}$ where d_i is the neighborhood terminal density of the terminal u_j.

Every terminal in 5G IoT can get the real-time spatial terminal distribution information around her

End if If () then End While End if If () End if

Algorithm 1: Neighborhood-distribution-perceptive algorithm

through Algorithm1. The neighborhood-distribution-perceptive scheme is the foundation of our PBPR algorithm.

4.3 Recommendation of privacy parameters and searching region

It is known that the bigger the privacy parameter k is, the stronger the privacy protection is. However, a bigger k always results in the failure of finding enough cooperated terminals, particularly in the scenario where mobile terminals are few and scattered. How to choose a suitable k is important in k-anonymity. Ahamed mentions in the work [21] that k is the key factor in the success rate as well as the security level of k-anonymity. Based on our neighborhood-distribution-perceptive scheme terminal is able to get the information of spatial terminal density in her neighborhood and estimate the appropriate k and the searching size of K-ASR.

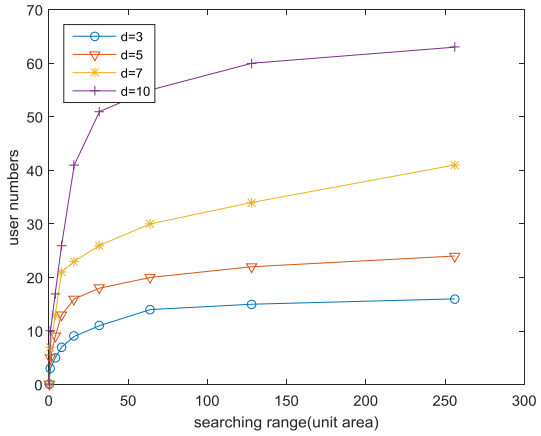


Fig.2 Relationship Between Searching Region and Cooperated Peers

Figure 2 gives the pictorial relationship between the searching region and cooperated terminals by changing d_u. Every point on the curve means the increase of hops by 1. We observe that with the expanding of searching range more cooperated terminals can be found. Considers the increasing terminal number between every adjacent hop to gain, N_h, while considering the additional searching region to the communication cost C_h, the searching gain is donated by the slope of the curve segment between every hop, N_h/C_h. We notice that the searching gain becomes smaller along with the expanding of searching range and it is no more than 1 when the searching range is over 4 hops, which means the searching process costs huge communication resources for finding more cooperated terminals. So for a given d_u, we recommend better performance of anonymization success rate can be achieved with lower communication resource where $k \leqslant 4*d_u$.

The size and shape of the K-ASR rely on the spatial terminal distribution in 5G IoT. If mobile terminals are independently and uniformly located in D, terminal u only needs $\sqrt{k/d_u}$ hops to find k peers. However, if mobile terminals are located in the line, u has to take k/d_u hops to find enough terminals in order to construct K-ASR. We recommend the $\sqrt{k/d_u}$ as the initial searching hops, $h_{initial}$, and $\mathrm{Max}(k/d_u, 8)$ as the end of searching hops, h_{end}.

5 Experimental results

In this section, we evaluate the performance of our PBPR algorithm with a set of key features in the analysis of security capability and comparison with two existing algorithms, Chow's P2PCloak [2] and Che's DA approach [3]. The simulated experiments are executed in a range of scenarios that are generated by the use of Generator of Network-based moving objects [22] and the realistic road network of the Germany city, Oldenburg. The experiments are implemented using JAVA and are run on Intel i5 2.3GHz desktop with 8GB of RAM. Table 1 summarizes the specification of our simulation.

According to the running environment, the to-

TABLE 1 THE SPECIFICATION OF EXPERIMENTS

Parameters	values
Terminal population	3000–9000
Transmission range	250m
Channel model	3GPP_38.901_UMa_NLOS
Carrier frequency	3.5 GHz
Channel estimation	Least square
k–anonymity	5–40
Message processing time	100ms

tal number of mobile terminals is from 3k to 9k. The values of k and transmission range are generated following the uniform distribution. Without losing generosity， we consider the message processing time to be 100ms， and the length of the message is 64 bytes.

5.1 Evaluation of authentication parameter consistency rate

Authentication parameter generation consistency rate： This refers to the consistent proportion of physical layer random authentication parameters generated by channel estimation and quantization of the terminals and man-in-the-middle attacker. The value of the authentication parameter generation consistency rate can directly reflect the defense degree of PBPR algorithm against man-in-the-middle attacks and embody the security increment.

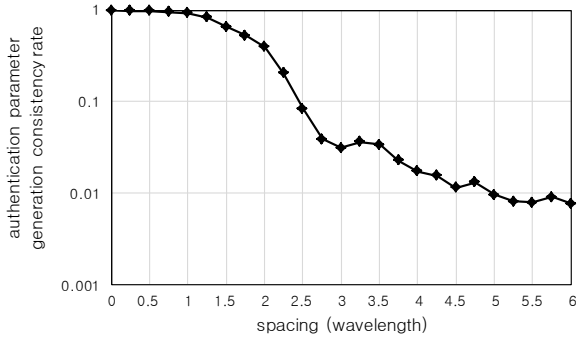


Fig.3 Authentication parameter consistency rate

It is assumed that the channel model between the 5G Base Station and the Mobile Station is 3GPP_38.901_UMa_NLOS and there is no noise at the receiving end of the terminal and the man-in-the-middle attacker. Figure 3 shows the authentication parameter generation consistency rate versus the distance between the terminal and the man-in-the-middle attacker （The unit is carrier wavelength）. We can see that the authentication parameter generation consistency rate of the proposed scheme is below 10^{-2} when the distance between the legitimate user and the illegal eavesdropper is 5 times the carrier wavelength. It shows that while ensuring normal communication， PBPR algorithm makes full use of the advantages of diverse spatial characteristics， rich channel characteristics， and high position sensitivity brought by physical layer technologies such as 5G large-scale antennas， large bandwidth and high-frequency bands to complete the generation of random authentication parameters. Based on the above， the algorithm can achieve good anti-eavesdropping performance with low cost and complexity.

5.2 Evaluation of average anonymizing time per query

Average anonymizing time per query： This is the mean time between the terminal’s initiating a query and generating a cloaked region. It indicates the response time of the anonymization procedure.

Figure 4 shows the average anonymizing time for each anonymization request. Compared with P2PCloak， the anonymizing time of DA and our algorithm is about 520ms and reduced by 50%. DA algorithm uses a little shorter time than our algorithm because it gathers terminal information before request launches with more communication overhead.

5.3 Evaluation of anonymization success rate

Anonymization success rate： This is a ratio of the number of times that the anonymization algorithm can find enough terminal location information to satisfy the terminal’s k-anonymity privacy requirement to the total number of queries.

Figure 5 shows the anonymization rate of the three algorithms in four scenarios. It shows that all algorithms can achieve a higher success rate as the terminal number grows. However， our algorithm presents a better success rate owing to the recommendation of privacy requirements and searching range， in particular in the scenario with fewer terminals. The average success rate of our algorithm is above 93%.

5.4 Evaluation of network bandwidth utilization

Network Bandwidth Utilization： It measures the network bandwidth utilization involved in a query’s whole anonymization procedure. Figure 6 presents the network bandwidth utilization comparison. In general， P2PCloak and our algorithm use little communication resources and exhibit a smooth curve. For

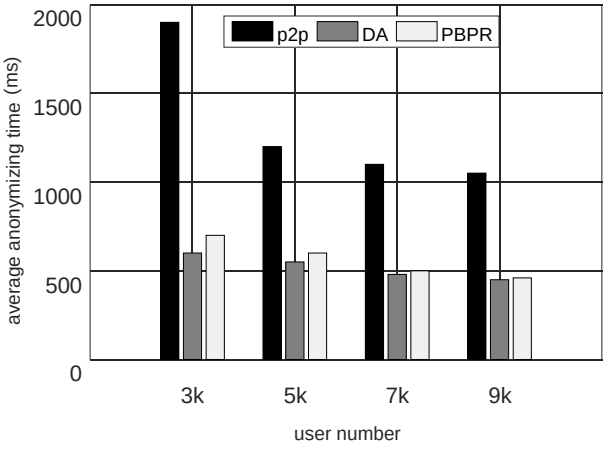


Fig.4 Average anonymizing time per query

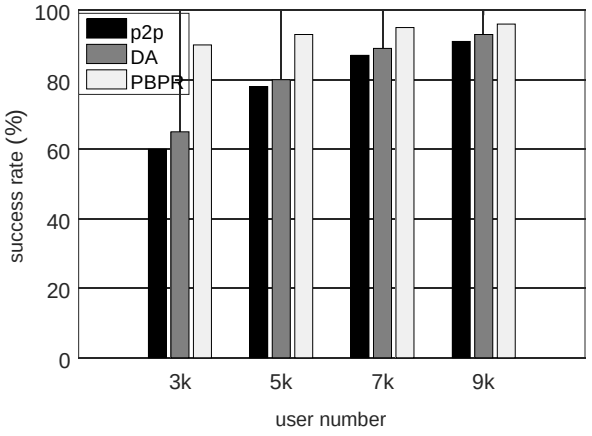


Fig.5 Anonymization success rate

P2PCloak， the curve rises slightly during the terminal searching. Terminals in our algorithm only broadcast message in one hop neighborhood and there are fewer messages exchange after d_u is stable. As a result， the curve of our algorithm just ascends a little in the neighborhood-distribution-perception stage. On the contrary， for the DA algorithm， the curve increases significantly from scenario 1 to 4 because of the exchange of the location records.

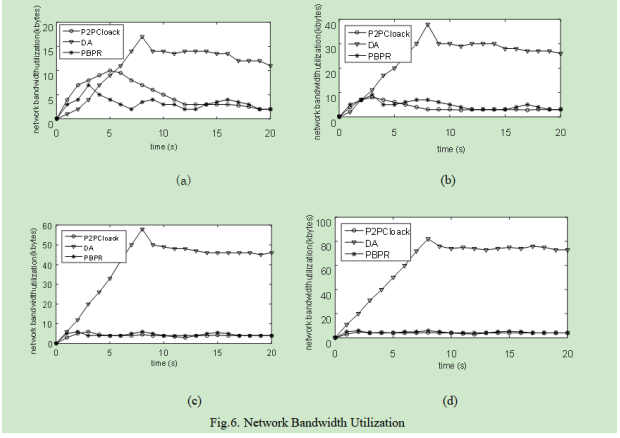


Fig.6. Network Bandwidth Utilization

6 Conclusion and future work

In this paper， we exploited the privacy issue of LBS and presented a fast physical layer security-based location privacy parameter recommendation algorithm for 5G IoT. By cross-layer authentication， exchanging neighborhood distribution information， and the recommendation of privacy requirements， our algorithm improves security capability and

achieves a higher anonymization success rate without additional communication cost. The extensive experiment demonstrates the PBPR algorithm can work safely in 5G IoT. We will work on saving computing resources and reducing signaling overhead by asymmetric implementation architecture to improve the terminals' energy efficiency in 5G IoT.

References:

[1] MOKBEL. M F. Privacy in location-based services: start-of-the-art and research directions [C]//Proceeding of 8th International Conference of Mobile Data Management，MDM（2007）

[2] Chow C Y, MOKBEL M F, LIU X. A peer-to-peer spatial cloaking algorithm for anonymous location-based service [C]//Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems，ACM（2006）

[3] CHE Y, YANG Q, HONG X. A dual-active spatial cloaking algorithm for location privacy preserving in mobile peer-to-peer networks [C]// 2012 IEEE Wireless Communications and Networking Conference，WCNC（2012）

[4] CHOW C Y, MOKBEL M F, LIU X. Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments [J]. GeoInformatica，2011，15（2）：351-380.

[5] HUANG Y, HUO Z, MENG X. CoPrivacy: A Collaborative Location Privacy-Preserving Method without Cloaking[J]. Chinese Journal of Computers，2011，34（10）：1976-1985.

[6] ZHANG C, HUANG Y. Cloaking locations for anonymous location based services: a hybrid approach [J]. GeoInformatica，2009，13（2）：159-182.

[7] YANG N, CAO Y, LIU Q, et al. A novel personalized TTP-free location privacy preserving method [J]. International Journal of Security and Its Applications，2014，8（2）：387-398.

[8] SOLANAS A, MARTINEZ-BALLESTE A. A TTP-free protocol for location privacy in location-based services [J]. Computer Communications，2008，31（6）：1181-1191.

[9] HASHEM T, KULIK L. "Don't trust anyone": Privacy protection for location-based services[J]. Pervasive and Mobile Computing，2011，7（1）：44-59.

[10] GAO S, Ma J, Yao Q, et al. Towards cooperation location privacy-preserving group nearest neighbor queries in LBS [J]. Journal on Communication，2015（3）:17-2015054.

[11] GHAFFARI M, GHADIRI N, MANSHAEI M H, et al. P4QS: A Peer to Peer Privacy Preserving Query Service for Location-Based Mobile Applications[J]. 2016，PP(99):1

[12] DARGAHI T, AMBROSIN M, CONTI M, et al. ABAKA: A novel attribute-based k-anonymous collaborative solution for LBSs [J]. Computer Communications，2016，85：1-13.

[13] Yiu M. L. , Jensen C. , Huang X. ，Lu, H. Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services [C]// Proceedings of the International Conference on Data Engineering，ICDE（2008）

[14] Kido H. , Yanagisawa Y. ，Satoh, T. An anonymous communication technique using dummies for location-based services [C]// Proceedings of IEEE International Conference on Pervasive Services，ICPS（2005）

[15] Ghinita G. , Kalnis P. , Khoshgozaran A. , Shahabi C. ，Tan, K. L. Private queries in location based services: Anonymizers are not necessary [C]//Proceedings of the ACM International Conference on Management of Data，SIGMOD（2008）

[16] Yiu M. L. , Ghinita G. , Jensen C. S. ，Kalnis, P. Outsourcing search services on private spatial data [C]//Proceedings of the International Conference on Data Engineering，ICDE（2009）

[17] Xu T. ，Cai, Y. Location anonymity in continuous location-based services [C]//Proceedings of the ACM Symposium on Advances in Geographic Information Systems，GIS（2007）

[18] Xu T. ，Cai, Y. Exploring historical location data for anonymity preservation in locationbased services [C]//Proceedings of the International Conference of the Computer and Communications Societies，INFOCOM（2008）

[19] Ghinita1，
G. ，Kalnis, P. , Skiadopoulos, S. Prive: Anonymous location based queries in distributed mobile systems[C]// Proceedings of 16th International World Wide Web Conference，WWW（2007）

[20] Ghinita1，
G. ，Kalnis, P. , Skiadopoulos, S. Mobihide : A mobile peer-to-peer system for anonymous location-based queries[C]//Proceedings of the International Symposium on Advances in Spatial and Temporal Databases，SSTD（2007）

[21] AHAMED S I, HAQUE M M, HASAN C S. A novel location privacy framework without trusted third party based on location anonymity prediction [J]. ACM SIGAPP Applied Computing Review，2012，12（1）：24-34.

[22] BRINKHOFF T. A framework for generating network-based moving objects[J]. GeoInformatica，2002，6（2）：153-180.

Research on the Security of the Edge Trusted Execution Environment of the Internet of Things Based on TrustZone

Abstract: . In the era of the Internet of Everything, terminal devices are connected and produce many sensitive data. The IoT architecture based on edge computing can effectively improve the real-time performance of sensitive data processing. However, this architecture easily exposes data to the vulnerable software stack at the edge, posing security threats to the edge. In recent years, researchers have used TrustZone to build a trusted execution environment at the edge to improve edge data security. To this end, this article will study the TrustZone-based edge trusted execution environment of the Internet of Things. Starting from TrustZone, we first introduce the trusted execution environment, the architecture and working principle of TrustZone, then summarize and analyze the challenges and specific solutions of the TrustZone-based edge trusted execution environment, and compare it with other edge trusted execution solutions, then look forward to the future research direction of IoT edge security based on TrustZone, and finally summarize the article.

Key words: Internet of Things; Edge Computing; Trusted Execution Environment; ARM TrustZone

1 Introduction

In recent years， with the continuous development of the Internet of Things （IoT）[1] and the popularization of wireless networks （such as 4G， 5G）， more and more terminal devices are connected to the edge of the IoT network. According to the prediction of the famous international network solution provider Cisco， there will be 50 billion terminal devices connected to the IoT network in 2020 [2] . At the same time， the amount of data generated by IoT applications （smart cities [3]， intelligent networked vehicles [4]， etc.） composed of these terminal devices is also increasing rapidly. IDC predicts [5] that the total amount of data generated by the IoT in the world will reach 40 ZB （ZettaByte）， and 45% of these data are processed on the network terminal. On the one hand， traditional cloud computing has poor real-time performance， insufficient bandwidth， and high energy consumption， making it difficult to ensure the security and privacy of IoT data [6] . On the other hand， the data generated by the IoT has high throughput and requires high latency [7] . Therefore， traditional cloud computing is difficult to efficiently process data generated by terminal devices.

To solve the above-mentioned problems， the edge computing model oriented to the calculation of the volume data of the terminal equipment was born [8] . It is a new type of computing model that performs computing at the edge of the network. It has the following advantages. On the one hand， it can process a large amount of temporary data at the edge of the network， and reduce network bandwidth pressure and system delay. On the other hand， it can save user data on network edge devices， and reduce the risk of data leakage. Therefore， edge computing has been widely used in IoT in recent years. However， compared with terminal and cloud computing， the edge has shorter development history and faces many security threats [9] . These threats can be caused by too many lines of code [10]， improper configuration of edge platforms [11] ， slow software up-

dates [12], etc.

Traditional security solutions can still be used for protection, such as password encryption, access control strategies, etc. But these solutions are only applicable to the edge of the IoT with certain modifications. In recent years, some emerging security technologies, such as hardware-assisted TEE [13] (Trusted Execution Environment), can be used in edge computing to enhance the security of edge computing. TEE refers to the use of hardware mechanisms to partition a trusted, isolated, and independent execution environment on an untrusted operating system, providing a safe and confidential space for users' private data and sensitive computing. Compared with traditional security technologies, TEE can actively defend against external security threats, thereby more effectively protecting the system's security. According to a survey of Eclipse in 2018 [14], edge platforms generally use ARM TrustZone to build edge security systems. The CPU cores in this platform are generally 2-8 and the DRAM is generally several GB.

The rest of this paper is structured as follows. Section 2 introduces the basic concepts of the TEE, the principle and architecture of ARM TrustZone, and a security platform similar to TrustZone. Section 3 introduces the security challenges faced by TrustZone-based edge trusted execution environment systems, security solutions, and related solutions based on other hardware platforms. Section 4 looks forward to the future of IoT edge security based on TrustZone. Section 5 summarizes the article.

2 TrustZone related technologies

Many hardware platforms of edge devices in IoT applications use embedded development boards based on ARM TrustZone, such as smart homes and smart transportation. ARM's unique TrustZone technology can provide a TEE environment, which is suitable for securely building a TEE at the edge of the Internet of Things. This section will first introduce the trusted execution environment. Secondly, introduce the working principle and architecture of TrustZone. Finally, compare TrustZone with other technologies that can be used to build edge TEEs.

2.1 Overview of the TEE

The design of the TEE can meet the requirements for the secure processing of IoT data. Ben Pfaff [15] gave a detailed definition of TEE: "a dedicated closed virtual machine that isolates the rest of the platform. Using hardware memory protection and storage password protection to protect its content from unauthorized person' observation and tampering.". As a result, TEE can be simply described as a safe, integrity-protected processing environment, and the code and data running and stored in the environment have high security and confidentiality.

According to the definition of the Global Platform International Standards Organization (Global Platform) [16], a TEE mainly includes a common execution environment, a trusted execution environment, client applications, and trusted applications, as shown in Figure 1.

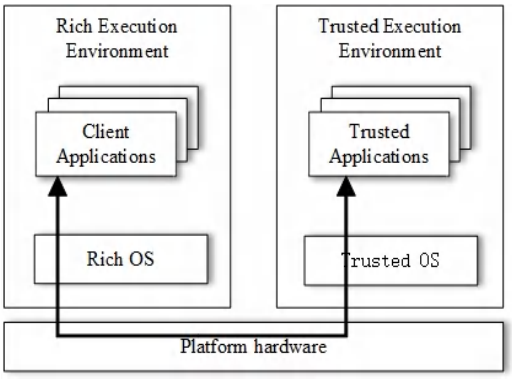


Figure 1 TEE architecture

There are many kinds of construction schemes for TEE architecture. Literature [17, 18] divides them into three categories: memory encryption-based TEE, co-processor based TEE, and processor-based TEE. TEE based on memory encryption directly builds a TEE at the application layer. Co-processor based TEE uses additional processors to handle security and integrity protection tasks, but the data transmission between the two processors will cause certain losses. The processor-based TEE [19] divides the en-

tire processor into multiple cores and uses a dedicated monitor to switch different states. The TrustZone technology introduced below uses the third method to build a TEE. Table 1 shows a summary of these three types of TEE construction methods.

Table 1 Classification of TEE construction methods

Types	Privilege	Example
Memory encryption-based TEE	Access application memory	AMD SME
Co-processor based TEE	Have a processor and memory. Exchange data with the main processor	Intel ME, AMD Platform Security Processor
Processor-based TEEE	Be able to access all hardware. Such as physical memory	Intel SGX, ARM TrustZone

2.2 Principles and Architecture of TrustZone

2.2.1 How TrustZone works

In the security environment established by TrustZone, a security configuration register is added to the core of the CPU, and an NS bit is defined in this register, which is used to identify the current security status. If the NS bit is 1, it means that it is currently a safe environment, otherwise, it is an unsafe state. Also, each environment has its hardware and software resources. In an unsafe environment, you can only access the corresponding resources; in a safe environment, you can access all the resources of the system.

To achieve system security, TrustZone has designed a memory isolation mechanism, a peripheral protection mechanism, and an interrupt isolation mechanism. The memory isolation mechanism is jointly completed by the MMU (Memory Management Unit) and TZASC (TrustZone Protection Controller). The peripheral protection mechanism is to protect the device, and the mechanism is mainly implemented with the help of TZPC (TrustZone Address Space Controller). The interrupt isolation mechanism of ARM TrustZone means that both the safe world and the unsafe world have their interrupt modes. The interrupt mode of the safe world is FIQ (Fast Interrupt Request), and the interrupt mode of the unsafe world is IRQ (interrupt request).

2.2.2 TrustZone Architecture Overview

TrustZone divides a single physical processor into two virtual processor cores, which are secure and insecure processors. At the same time, a special mechanism-monitoring mode is introduced to manage the state switching between safe and unsafe processors to ensure the safety of the system.

The whole of TrustZone can be divided into the software layer (application layer, an interface layer, system layer) and hardware layer (security kernel and ordinary kernel), as shown in Figure 2.

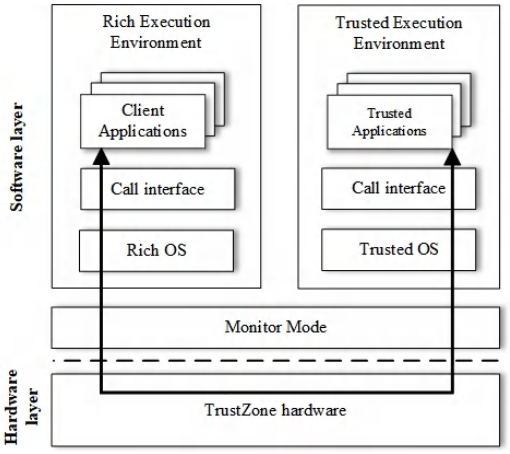


Figure 2 The overall architecture model of TrustZone

Firstly, the client application request is passed to the hardware layer through the application layer, client interface, and REE OS. Secondly, the current processor state is switched from normal mode to safe mode through the monitoring mode and passed to the trusted application. Finally, the trusted application sends the processing result back to the client application.

2.3 Comparative analysis of TrustZone and other similar technologies

As mentioned in the previous article, besides ARM TrustZone, the hardware platforms currently supporting TEE also include Texas Instruments M-Sheild [20], Intel SGX [21], AMD SVM [22], and

SoC based on RISC-V [23]. Similar to the ARM TrustZone, TI M-Shield is suitable for embedded systems. TI M-Shield provides a secure mobile framework while providing a secure isolation environment.

Table 2 Classification of platforms supporting TEE

Names	Field of application	Main idea	Defects
ARM TrustZone	Embedded	Dividing a physical processor into secure and insecure processors using virtualization techniques	No protection against physical attacks, side-channel attacks, TEE kernel attacks, etc.
TI M-Shield		Define secure ROM and RAM and embed a secure state machine	Over-reliance on hardware configuration and security control policies leads to narrow application areas
Intel TXT和 AMD SVM	PC	Extending the x86 instruction set	Not very widespread
Intel SGX		Reverse Sandbox	Older applications not easily migrated to SGX
Rocket, Hummingbird, VexRiscv	Embedded, PC	Custom build TEE security solution based on RISC-V instruction set architecture	Late in life and immature in development

However, the safety isolation technology of TI M-Shield is limited to two processors, OMAP and OMAP-Vox [24], so the application field is relatively narrow. Both Intel TXT and AMD SVM are developed based on TPM (Trusted Platform Module), and provide a complete chain of trust from the BIOS to the application layer, thereby improving the security of the system. However, Intel TXT and AMD SVM are unable to gain popularity due to their excessive reliance on hardware configuration and security control strategies. Besides, Intel also introduced a processor SGX that uses reverse sandboxing technology to improve software security. His main idea is to encapsulate legal software in a safe environment. No matter what privilege level the malware has, it cannot access the code in it. However, few CPUs support SGX on the market, and simulators are hard to find.

RISC-V is a brand-new instruction set architecture, which is a completely open-source and allows the freedom to manufacture and sell chips or software based on this instruction set. The project started in 2010 and was led by Professor David A. Patterson of the University of California, Berkeley. Compared with other open-source instruction sets, it is suitable for modern computing devices such as cloud computers, mobile phones, and embedded systems. The common SoC (system on chip) based on RISC-V include Rocket [25], Hummingbird [26], VexRiscv [27], etc. However, due to its short birth time, its related compilers, software development environments, development tools, and other ecological elements are still in the development stage. RISC-V based edge security flow processing schemes are also not common today. A summary of these different platforms that support TEE is shown in Table 2.

3 TrustZone-based TEE at the edge of the IoT

This Section will introduce the TrustZone-based edge TEE from three aspects: security challenges faced by the TrustZone-based edge TEE systems, TrustZone-based edge TEE security design solutions, and other edge TEE security solutions.

3.1 Security Challenges for IoT Edge TEE Based on TrustZone

Most of the data in IoT applications today is continuous, such as sensor data, human body temperature, blood pressure, and other health information. Therefore, it is critical to building a robust TEE at the IoT edge. This section introduces the security technology requirements for IoT edge stream processing. Besides, this section summarizes the security challenges of TrustZone-based TEE at the IoT edge from three aspects: architecture, software, and hardware.

3.1.1 Security Technology Requirements for IoT Edge Stream Processing

Before the advent of stream processing, data was usually stored in databases, file systems, or other forms of mass storage, with batch processing of the data taken at a later stage. However, for IoT applications that require a high degree of real-time performance, the drawbacks of still processing data in this manner would be disastrous.

There is a component in the stream processing architecture called a stream processor, which has two main types of functions: data functions for data movement and computation, control functions for resource management and computational processes (e.g. creating and scheduling tasks). The basic requirements of a stream processor are to ensure that the huge amount of data in the IoT can flow efficiently as well as be computed in a fast and timely manner and have some fault tolerance to ensure that each data is processed at least once.

The limited resources of terminal devices cannot guarantee the security and real-time requirements of data stream processing at the same time. Even if the cloud computing technology is borrowed, it still faces the limitation of bandwidth and energy consumption, which makes it difficult to guarantee the security and privacy of IoT stream data. Besides, the cloud is far away from the terminal, the transmission delay is high, so it is difficult to guarantee its real-time performance. Edge computing brings a new approach to solving this problem. A TEE built at the edge is equivalent to a stream processor, which can effectively ensure efficient and fast processing of stream data for IoT, so the security of the TEE is critical.

3.1.2 Architectural design flaws in TEE

TEE systems built with the assistance of TrustZone in the IoT edge often expose a broad attack surface that can be exploited to undermine overall security. These TEE systems require that drivers exist in the software for access to security-sensitive devices, e.g., fingerprint sensors for user authentication. These drivers are often complex and run in TEE kernel space, which can lead to attacks on the driver or TEE kernel and compromise the integrity and availability of the system. Besides, the interfaces between the components of the TEE system are too broad. the TEE also has a wide attack surface due to the large TCB of the TEE. An important reason why TEE systems can be secure is to separate the secure world from the troubled world while enabling effective communication between the two worlds. This isolation may be broken if the TEE system is designed with improper system calls. For example, trusted applications can map the physical memory of the ordinary world. If a trusted application is compromised, an attacker can take control of the OS by scanning the physical address space of the Linux kernel and patching it to introduce a backdoor. Also, the current memory protection mechanism in the TrustZone-based TEE built in the IoT edge is flawed. These deficiencies are, respectively, missing or weaker address randomization real and missing stack cookies to protect the page or enforce protection.

These are the architectural design flaws that lead to vulnerabilities in TEE systems built on TrustZone. Three main aspects are summarized. Firstly, too many lines of system design code, which leads to an oversized attack surface. Secondly, improperly designed system call patterns, which destroy the isolation between the secure and non-secure worlds. Thirdly, under-designed memory protection mechanisms, which leads to the exploitation of TEE.

3.1.3 Software errors in TEE

IoT edge TEE systems often fail to ensure the correct processing of input or output values, a common software error that can also be referred to as a validation error. For example, buffer overflows, incorrect parameter validation, integer overflows for incorrect processing, etc. There are currently four main categories of validation errors in IoT edge TEE systems built based on TrustZone: validation errors in the security monitor, the trusted application, the trusted kernel, and the TEE secure bootloader. Al-

so， when implementing business code in an IoT edge TEE system， unregulated coding habits of designers often lead to programming errors， such as mis-programming of cryptographic algorithms. The programming errors in IoT edge TEE systems built based on TrustZone fall into three main categories：memory protection programming errors， peripheral configuration programming errors， and programming errors in security mechanisms. Several other software errors can lead to IoT edge TEE system vulnerabilities. For example， concurrency errors due to multiple concurrent programs interfering with each other （e.g.， thread interleaving）， and these concurrency errors may introduce security vulnerabilities in the TEE system.

The above are the software errors that cause vulnerabilities in TEE systems built on TrustZone. They are summarized in three main areas： validation errors， programming errors， and other specific errors.

3.1.4 Hardware deficiencies in TEE

An IoT edge TEE built on TrustZone relies not only on the correct software architecture and software implementation but also on the correct trusted hardware components. Therefore， TEE developers must fully understand all hardware components （e.g.， FPGAs） inside and outside the system-on-chip （SoC） boundary， and configure them correctly. Otherwise， attackers can exploit hardware component misconfigurations to attack TEE systems. For example， CLKSCREW [28] exploits a malicious kernel driver that keeps frequency and voltage regulators running， and thus launches an attack. The attack is based on the principle that the frequency and voltage regulators will exceed their power limits if they are constantly operating， which causes false calculations.

In addition to the aforementioned hardware-based architecture security vulnerabilities， TEE security also depends on micro-architecture （e.g.caching）. There are three main types of threats to the IoT edge TEE microarchitecture， which are information leakage using cache information， information leakage using branch predictors， and information leakage using the Rowhammer attack. Rowhammer attack is a hardware failure caused by software that affects memory and allows the attacker to perform memory read operations [29].

These are the hardware errors that cause vulnerabilities in TEE systems built on TrustZone. The main two aspects are discussed in terms of the architecture of the hardware and other microarchitectures.

3.2 TrustZone Edge-based TEE Security Solution

This section provides a brief description of three edge TEE solutions for IoT built on TrustZone.

A TurstShadow

The commercial operating systems responsible for protecting sensitive data at the edge of the IoT are very large and complex and therefore often compromised. Therefore， an attacker can access this sensitive data. To solve this problem， traditional solutions such as processing data in a separate virtual machine [30]， exploiting hardware features [31]， or modifying a commercial operating system [32] protect applications from exploitable errors or operating systems affected by configuration errors. However， these solutions do not meet the high data throughput and low latency characteristics of IoT applications. A system， TrustShadow [33]， is developed for this literature. The system utilizes ARM TrustZone technology to isolate older versions of applications from the threatened OS and to partition resources into a secure and normal environment. In a secure world， TrustShadow builds a TEE for applications where security is critical. This trusted environment is maintained by a lightweight system that coordinates the application's communication with the common OS. Le Guan prototyped TrustShadow on real chipboard with ARM TrustZone and evaluated its performance using micro-benchmarks and real-world applications， with excellent results. But the system pulls the entire user application and library into the TCB. This can lead to an oversized TCB， making the stream analysis engine

and its libraries large， complex， and vulnerable to attack.

B StreamBox-TrustZone

The literature [34] proposes an edge stream processing security engine， StreamBox-TrustZone， or SBT for short， based on ARM TrustZone. The solution can solve the problem of oversized TCBs by reducing them to sections containing only protected functions， TEE， and secure hardware. As shown in Figure 3， the SBT is divided into two main parts， the top half is the trusted data plane and the bottom half is the untrusted control plane， the bottom part faces many security issues during stream processing. Also， the data stream transmitted by the endpoint only flows through the secure data plane via the ARM TrustZone's trusted IO channel， and all processing of the data stream occurs in the data plane. Whenever a data stream passes through the data plane， the data plane sends a specific credential to the control plane. The control plane will use credentials as needed， invoke system interfaces to arrange the data analysis pipeline， and the data plane will partition the operational pipeline into multiple trusted computational primitives to process the data stream. Once the data processing is complete， the data plane sends the data to the cloud for further validation and analysis to verify the correctness and freshness of the results of the data stream.

By creating a TEE and an efficient verification mechanism， SBT effectively improves the security of edge data flow processing. SBT also designs special memory management and parallel computing techniques to improve the throughput and latency of data flow analysis Heejin Park et al. developed an implementation of the system in Huawei's Hikey960. The SBT is proven to process input events up to 140 MB/s with sub-second latency on an octa-core ARMv8 platform， resulting in superior performance.

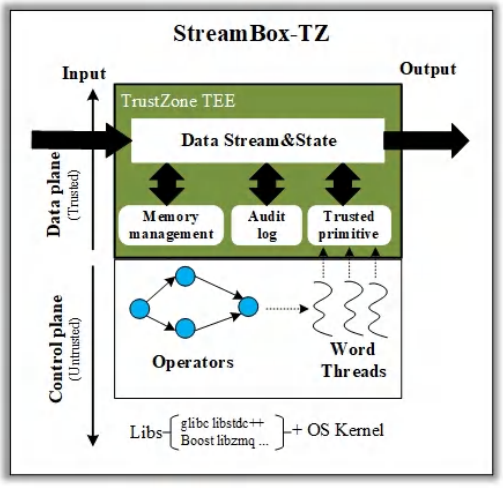


Figure 3 SBT framework diagram

C MQT-TZ

There is a constant need to transmit and process data between IoT devices. Effective communication between things is therefore essential. The traditional communication model for IoT is the publish-subscribe [35] model， and specific subject-based publish-subscribe protocols are MQTT [36] （Message Queuing Telemetry Transport）. Most MQTT implementations support TLS [37] for transport-level security， thereby preventing attacks on application data. However， there are still many holes in the method [38]. The literature [39] addresses the above problem by proposing MQT-TZ， a publish-subscribe agent for edge-based design. This system leverages MQTT and ARM TrustZone to protect IoT systems from other attacks. The main idea is to use ARM TrustZone to add a secure middle layer between publishers and subscribers when the communication mode in IoT applications is a publish-subscribe model. As shown in Figure 4， both the publisher and the subscriber have their decryption keys， while TrustZone has tamper-proof and secure storage technology and can securely store both publisher and subscriber keys. Firstly， the publisher sends the encrypted data to the normal world at the intermediate proxy level， then the monitor switches the current state to the secure world. The encrypted data is then decrypted in the secure world using the publisher's key and then encrypted using the subscriber's key. The monitoring mode then switches the state to the normal world and publishes the encrypted data to the subscriber.

Carlos Segarra's implementation of the system on the Raspberry Pi 3 Model B and experimental eval-

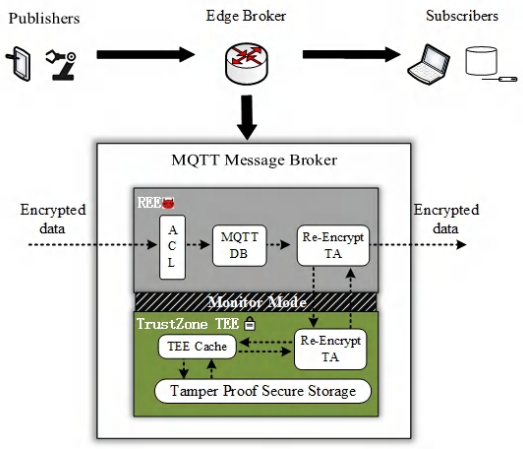


Figure 4 MQT-TZ frame diagram

uations of micro, macro benchmarks, and real industrial workloads in the MedTech use case have all yielded good results.

3.3 Other security solutions for edge TEE

In addition to TrustZone, many other hardware platforms or related technologies are capable of building TEE at the edge. Suman Jana [40] and others have proposed a privacy protection system, Darkly, in response to the problem that applications now have extremely high privileges over the system, leading to system compromise. Also, there exist some edge security schemes built on Intel SGX, such as SecureStream [41], Scone [42], VC3 [43], and Haven [44]. However, the aforementioned solutions often lead to system vulnerabilities due to the oversized TCB design and failure to utilize hardware to design a TEE. VC3 and Haven systems are built in the cloud and may not apply to the resource-constrained edge. The literature [45] has developed a secure virtual machine manager, called TrustVisor, to provide a TEE for security-sensitive code modules, plus the TCB contains only 5306 lines of code (more than half of which are used for encryption operations). The approach faces many challenges when it comes to supporting data-intensive computing with minimal TCB.

Security solutions based on ARM TrustZone designs have lower cost and power consumption and lower hardware requirements. Compared to other hardware-based designs, ARM TrustZone-based security solutions tend to have the best performance when costs are equal. As a result, ARM TrustZone is the number one choice for building trusted execution environments for edge devices.

4 Security Outlook

At present, the design of the edge TrustZone-assisted TEE of IoT is still in the early stage of development, and there are still many security challenges. The following content gives a security outlook in this direction.

Firstly, as mentioned in the previous article, there are generally architecture, software, and hardware problems in the TEE based on ARM TrustZone. Given the problem of architecture, we research from the following four aspects: building multiple isolation environments, enhancing the security of the two-world transition, designing memory encryption technology, and designing reliable computing primitives to ensure remote certification and sealed storage. For software problems, we research the following three aspects. Firstly, designing mechanisms such as memory checking and garbage collection, which can ensure the safety of the code when it is managed. Then, designing a safe programming language (such as RustZone [46]). Finally, designing software verification mechanisms (such as model checking and formal methods) to determine whether expectations match reality. Aiming at the hardware problem, we research from the secure hardware architecture and microarchitecture.

Besides, the trade-off between security and efficiency in edge data processing also needs to be considered. Most of the IoT application scenarios require edge nodes to perform complex computing operations, and these complex operations often consume a lot of time and computing resources. If the computing resources and time cost of the security mechanism adopted on the edge device are too high, it will aggravate the delay between the IoT device and the edge. Therefore, in the actual application scenario of the IoT, it is necessary to do a good job between safety and efficiency. The trade-off is to ensure that

the terminal device can respond in time while ensuring the security of edge data processing.

Then， ARM TrustZone's security issues. Due to the design flaws of ARM TrustZone， the current edge flows analysis engines built on ARM TrustZone are difficult to defend against TEE kernel defect attacks， side-channel attacks， and physical attacks（memory attacks）. Although there are methods that can protect against these attacks to some extent， these methods have not been applied to the ARM TrustZone platform based edge secure flow processing engine. In the future， these methods can be borrowed and utilized to design an edge secure stream processing engine based on the ARM TrustZone platform to solve the above problems and improve the security of edge stream processing.

Finally， data processing security issues across multiple edge devices are also worth studying. At the edge of the Internet of Things， there is often more than one device， and one or more edge devices may work together to perform stream processing operations. Besides， the credibility of network equipment at the edge still needs further research.

5 Conclusion

The development of the IoT brings convenience to people's life， learning， and work， but also brings many security issues. The emergence of edge computing-based IoT architecture can improve data processing speed and system security to a certain extent. However， due to the short development time of edge computing， there are huge challenges in application security， data security， network security， and node security. As a result， more and more researchers， are building trusted execution environments at the edge to improve the security of the edge. However， there are many exploitable vulnerabilities in the current TEE system built on TrustZone. In this article， we analyze the architecture， software， and hardware of the TrustZone-based IoT edge TEE system. We present three specific security scenarios and look ahead to future research directions， thus making TrustZone-based IoT edge TEE systems more secure.

References:

[1] Gubbi J, Buyya R, Marusic S, et al. Internet of Things (IoT): A vision, architectural elements, and future directions［J］. Future generation computer systems，2013，29（7）：1645-1660.

[2] EVANS D. The internet of things: How the next evolution of the internet is changing everything［J］. CISCO white paper，2011，1（2011）：1-11.

[3] PAN G, QI G, ZHANG W, et al. Trace analysis and mining for smart cities: issues, methods, and applications［J］. IEEE Co m munications Magazine，2013，51（6）：120-126.

[4] Gerla M, Lee E K, Pau G, et al. Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds［C］//2014 IEEE world forum on internet of things (WF-IoT). IEEE，2014：241-246.

[5] Turner V, Gantz J F, Reinsel D, et al. The digital universe of opportunities: Rich data and the increasing value of the I nternet of things［J］. IDC Analyze the Future，2014，16.

[6] Lu G, Zeng W H. Cloud computing survey［C］//Applied Mechanics and Materials. Trans Tech Publications Ltd，2014，530：650-661.

[7] Ding W, Jing X, Yan Z, et al. A survey on data fusion in internet of things: Towards secure and privacy-preserving fusion［J］. Information Fusion，2019，51：129-144.

[8] Shi W, Cao J, Zhang Q, et al. Edge computing: Vision and challenges［J］. IEEE internet of things journal，2016，3（5）：637-646.

[9] Roman R, Lopez J, Mambo M. Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges［J］. Future Generation Computer Systems，2018，78：680-698.

[10] ChenH. , MaoY. , WangX. , ZhouD. , ZeldovichN. , and KaashoekM. F. . Linux kernel vulnerabilities: State-of-the-art defenses and open problems. In Proceedings of the 2nd Asia-Pacifific Workshop on Systems (APSys)，2011.

[11] Symantec. Internet Security Threat Report. https://www. symantec. com/content/dam/symantec/docs/reports/istr-22-2017-en. pdf，2017.

[12] Vasisht D, Kapetanovic Z, Won J, et al. Farmbeats: An iot platform for data-driven agriculture［C］//14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17). 2017：515-529.

[13] Sabt M, Achemlal M, Bouabdallah A. Trusted execution environment: what it is, and what it is not［C］//2015 IEEE Trustcom/BigDataSE/ISPA. IEEE，2015，1：57-64.

[14] Eclipse IoT Working Group. IoT Developer Survey 2018. https:// https://blogs. eclipse. org/post/benjamin-cab% C3%A9/key-trends-iot developer-survey-2018，2018.

[15] GARFINKEL T, PFAFF B, CHOW J, et al. Terra: A virtual mac hine-based platform for trusted computing［C］//Proceedings of t he nineteenth ACM symposium on Operating systems prin ciples. 2003：193-206.

[16] Specification G P. TEE System Architecture, version 1. 0［J］. 2011.

[17] Fan Guannan, Dong Pan. Research on TrustZone based Trusted Execution Environment Building Techniques［J］. Information Network Security，2016，16(3)：21-27.

[18] Ning Zhenyu, Zhang Fenwei, Shi Weisong. Research on trusted execution environment based on edge computing［J］. Computer Research and Development，2019，56(7)：1441-1453.

[19] Cerdeira D, Santos N, Fonseca P, et al. SoK: Understanding the prevailing security vulnerabilities in TrustZone-assisted TEE systems［C］//Proceedings of the IEEE Symposium on Security and Privacy (S&P)，San Francisco，CA，USA. 2020：18-20.

[20] Azema J, Fayad G. M-Shield mobile security technology: making wireless secure［J］. Texas Instruments white paper，2008.

[21] IncAMD. AMD64 Virtualization Codenamed "Pacifica" Technology: Secure Virtualization Machine Archtecture Reference Manaul. White Paper，2005

[22] Intel R. Software guard extensions programming reference［J］. Intel Corporation，2014.

[23] Costan V, Lebedev I, Devadas S. Sanctum: Minimal hardware extensions for strong software isolation［C］//25th {USENIX} Security Symposium ({USENIX} Security 16). 2016：857-874.

[24] Cumming P. The TI OMAP ™Platform Approach to SoC［M］// Winning the SOC Revolution. Springer, Boston, MA，2003：97-118.

[25] Asanovic K, Avizienis R, Bachrach J, et al. The rocket chip generator［J］. EECS Department，University of California，Berkeley，Tech. Rep. UCB/EECS-2016-17，2016.

[26] HuB. , (Online). Available:， https：//github. com/SI-RISCV/e200_opensource/blob/master/doc

[27] "GitHub - SpinalHDL/VexRiscv: A FPGA friendly32 bit RISC-V CPU Implementation, " Last access date: 2019/06/21. [Online]. Available：https：//github. com/SpinalHDL/VexRiscv

[28] Tang A, Sethumadhavan S, Stolfo S. {CLKSCREW}: exposing the perils of security-oblivious energy management［C］//26th {USENIX} Security Symposium ({USENIX} Security 17). 2017：1057-1074.

[29] Van Der Veen V, Fratantonio Y, Lindorfer M, et al. Drammer: Deterministic rowhammer attacks on mobile platforms［C］// Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. 2016：1675-1689.

[30] Chen X, Garfinkel T, Lewis E C, et al. Overshadow: a virtualiz ation-based approach to retrofitting protection in commodity oper ating systems［J］. ACM SIGOPS Operating Systems Review，2008，42(2)：2-13.

[31] MCCUNE J M, PARNO B J, PERRIG A, et al. Flicker: An execution infrastructure for TCB minimization［C］//Proceedings of the 3rd ACM SIGOPS// EuroSys European Conference on Computer Systems 2008. 2008：315-328.

[32] Criswell J, Dautenhahn N, Adve V. Virtual ghost: Protecting app lications from hostile operating systems［J］. ACM SIGARCH Co mputer Architecture News，2014，42(1)：81-96.

[33] Guan L, Liu P, Xing X, et al. Trustshadow: Secure execution of unmodified applications with arm trustzone［C］//Proceedings of the 15th Annual International Conference on Mobile Systems，Applications，and Services. 2017：488-501.

[34] Park H, Zhai S, Lu L, et al. StreamBox-TZ: secure stream analytics at the edge with TrustZone［C］//2019 {USENIX} Annual Technical Conference ({USENIX}{ATC} 19). 2019：537-554.

[35] Baldoni R, Beraldi R, Quema V, et al. TERA: topic-based event routing for peer-to-peer architectures［C］//Proceedings of the 2007 inaugural international conference on Distributed event-based systems. 2007：2-13.

[36] Jaloudi S. MQTT for IoT-based applications in smart cities［J］. Palestinian Journal of Technology and Applied Sciences (PJTAS)，2019(2).

[37] Dierks T, Rescorla E. The transport layer security (TLS) protocol version 1. 2［J］. 2008.

[38] Teserakt AG. Is MQTT secure? (A report). https://blog. teserakt. io/2019/03/04/is-mqtt-secure/，2019.

[39] Segarra C, Delgado-Gonzalo R, Schiavoni V. MQT-TZ: Hardening IoT Brokers Using ARM TrustZone［J］. arXiv preprint arXiv: 2007. 12442，2020.

[40] JANA S, NARAYANAN A, SHMATIKOV V. A scanner darkly: Protecting user privacy from perceptual applications［C］//2013 IEEE symposium on security and privacy. IEEE，2013：349-363.

[41] HAVET A, PIRES R, FELBER P, et al. Securestreams: A reactive middleware framework for secure data stream processing［C］// Proceedings of the 11th ACM International Confer ence on Distributed andEvent-based Systems. 2017： 124-133.

[42] ARNAUTOV S, TRACH B, GREGOR F, et al. {SCONE} ：Secure linux containers with intel {SGX} ［C］//12th {USENIX} Sy mposium on Operating Systems Design and Implementation ({O SDI} 16). 2016：689-703.

[43] Schuster F, Costa M, Fournet C, et al. VC3：Trustworthy data analytics in the cloud using SGX［C］//2015 IEEE Symposium on Security and Privacy. IEEE，2015：38-54.

[44] BAUMANN A, PEINADO M, HUNT G. Shielding applications from an untrusted cloud with haven［J］. ACM Transactions on Computer Systems (TOCS)，2015，33(3)：1-26.

[45] MCCUNE J M, LI Y, QU N, et al. TrustVisor: Efficient TCB reduction and attestation［C］//2010 IEEE Symposium on Security and Privacy. IEEE，2010：143-158.

[46] Evenchick E. RustZone: Writing Trusted Applications in Rust［J］. 2018. Available: https：//i. blackhat. com/eu-18/Thu-Dec-6/eu-18-Evenchick-RustZone. pdf

Implementation of redundant heterogeneous multi-core architecture in Mimic SDON transmission system

ZHANG Yongzhuang, ZHANG Huibin, YANG Chenguang

School of Electronic Engineering, Beijing University of Posts and Telecommunications, 100876, China

Key words: Multi-core; Redundant heterogeneous; SDON; security gain

Abstract. A redundant heterogeneous multi-core architecture is designed in the Mimic SDON transmission system， which consists of data-receiving module， data-distributing module， multi-core processors module and judgment module. This system realizes the heterogeneity of software and hardware on the multi-core processors module of the optical transmission equipment. The majority voting algorithm is used to determine the data returned by each core. Simulation and experiment results show that the probability of the system being successfully attacked is greatly reduced and the security gain of the system is highly enhanced.

1 Introduction

Software-defined optical network （SDON） is the combination of SDN and traditional optical transmission network which allows the control plane to be decoupled from the data plane. It enables remote controllers to configure the network equipment from different hardware vendors ［1］. SDON combines the advantages of optimizing fragmentation of specific networks in optical layer and great bandwidth providing services， and it provides a unified control platform for different networks ［2］. But SDON system may have some vulnerabilities that viruses and Trojan can use to attack the system， resulting in information leakage and network paralysis.

A large number of studies have proposed methods to enhance the security of SDON systems， such as distributed controller ［3］， elastic control plane ［4］ ， Mimic defense on SDON controller ［5］ and virtual service chain ［6］. Most of these methods focus on the security of control layer. However， the security of physical layer is also important. SDON physical layer is mainly composed of optical transmission equipment， most of which are based on single-core system. But single-core system is vulnerable to attacks when transmitting data， so its security is low.

In this paper， we propose a redundant heterogeneous multi-core architecture in optical transmission equipment. The optical transmission equipment consists of four modules： data-receiving module， data-distributing module， multi-core processors module and judgment module. Heterogeneity of hardware and software is realized on the multi-core processors module of the optical transmission equipment. The majority voting algorithm is used to determine the data returned by each core. And the final data is sent to PL for configuration. The performance of the multi-core system can be evluated by the following aspects： attack probability， security gain， system similarity， heterogeneous degree， and time performance. Simulation and experiment results show that the het-

erogeneous degree and safety performance of multi-core system is significantly improved compared with of single-core system， and the safety performance of five-core system is superior than that of tri-core system. This method greatly increases the security of the physical layer of the SDON system.

2 System architecture

2.1 Mimic SDON transmission system

Fig.1 shows the architecture diagram of the Mimic SDON transmission system. The system can be divided into three parts： application layer， control layer and physical layer. The application layer collects information of SDON controller and displays graphical interface. The control layer mainly realizes centralized control of network resources. The physical layer is mainly composed of Mimic SDON transmission equipment in the optical transmission network.

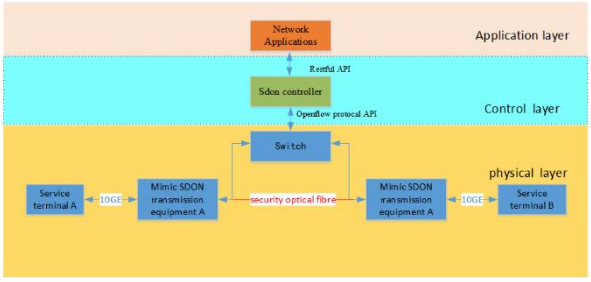


Fig. 1 Architecture diagram of Mimic SDON transmission system

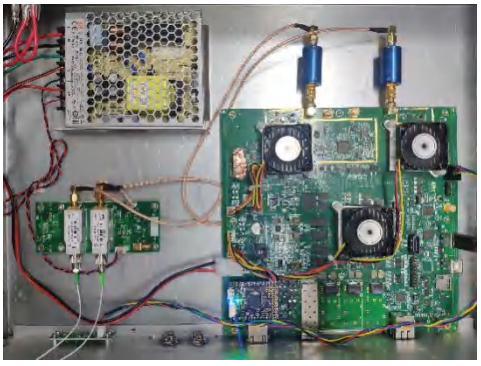


Fig. 2 Internal structure of Mimic SDON transmission equipment

Fig. 3 Front panel of Mimic SDON transmission equipment

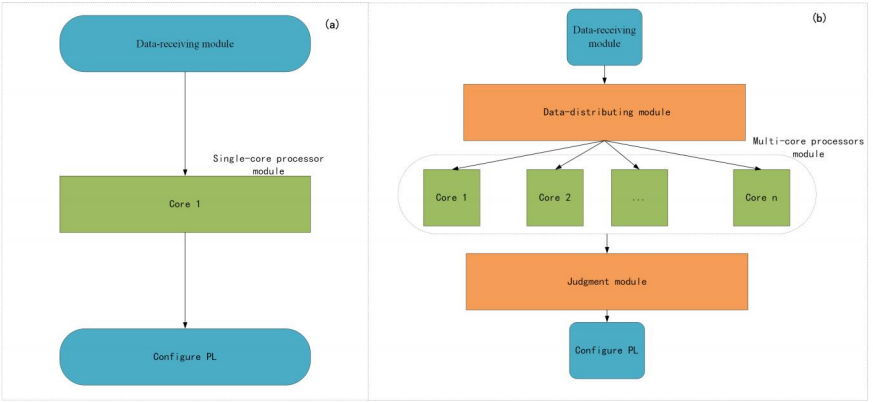
Fig.2 and Fig.3 show internal structure and external interface of the Mimic SDON transmission equipment. Develop SDON management and control interface on the transmission equipment to realize communication with the control layer， and develop SDON line interface to realize connection between optical transmission equipment. In this paper， system security is enhanced by realizing multi-core redundant heterogeneous architecture in the Mimic SDON transmission equipment.

2.2 Multi-core architectural model

Fig. 4 （a） shows that traditional single-core system has only one core as its processor， which processes the data and sends the results to PL for configuration. But the single-core system is easy to be attacked because of its simple structure. Fig. 4 （b） shows that the redundant heterogeneous multi-core architecture consists of four modules： data-receiving module， data-distributing module， multi-core processors module and judgment module. Data-receiving module adopts the SDON management and control interface to receive data sent from the control layer. Data-distributing module copies several pieces of data and sends them to the multi-core processors module. Multi-core processors module is the core of the system. Judgment module determines the processing results of multi-core processors based on the majority voting algorithm. The shared memory is used for communication between multi-core processors， and the interrupt is used to realize the information exchange

Fig. 4 （a） Single-core model of optical transmission equipment. （b） Multi-core redundant heterogeneous model of optical transmission equipment

In a redundant heterogeneous architecture， ex-



ecutors refers to a set of functional equivalent components [7]. Heterogeneity is the implementation of the same function but different structure on executors. The heterogeneous executors of the optical transmission equipment in this system can be divided into two parts: hardware heterogeneity of multi-core and software heterogeneity of algorithms on cores.

Fig. 5 shows that the optical transmission equipment has a quad-core ARM Cortex-A53 with 64-bit and dual-core ARM Cortex-R5 with 32-bit based processing system (PS) and programmable logic (PL) which is integrated into a single device. This system uses four A53 cores and two R5 cores to construct redundant heterogeneous executors at hardware level, and uses different algorithms to construct executors at software level.

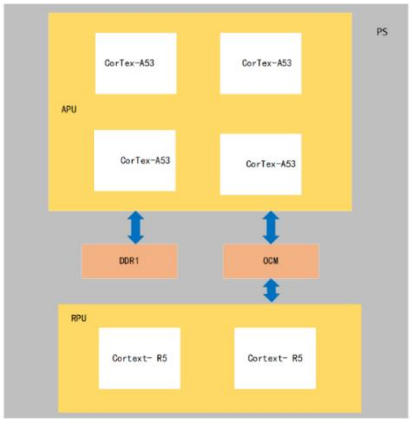


Fig. 5 Multi-core architecture diagram

3 Security analysis

3.1 Similarity of system

In a redundant heterogeneous system, the executor consists of several components. For a system with n executors, the characteristic similarity matrix of the k - th component is [8]:

$$A_k = \begin{bmatrix} 1 & \delta_{12}^k & \delta_{13}^k & \cdots & \delta_{1n}^k \\ \delta_{21}^k & 1 & \delta_{23}^k & \cdots & \delta_{2n}^k \\ \delta_{31}^k & \delta_{32}^k & 1 & \cdots & \delta_{3n}^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \delta_{n1}^k & \delta_{n2}^k & \delta_{n3}^k & & 1 \end{bmatrix}$$

$A_k$is symmetric matrix, δ_{ij}^k represents$(0 < i \neq j <= s, 0 < \delta_{ij} <= 1)$ similarity between the i - th and$j$ - th categories in the$k$ - th component.

The characteristic similarity matrix of redundant heterogeneous system is weighted sum of the similarity of various components [9].

$$A = \sum_{k=1}^{m} \omega_k A_k \tag{1}$$

$$\sum_{k=1}^{m} \omega_k = 1 (0 < \omega_k <= 1) \tag{2}$$

A is characteristic similarity matrix of the system, $\omega_k$is similarity weight, and $m$is number of system components.

In this paper, the executor of the system consists of hardware and software components. Generally speaking, there are more vulnerabilities in hardware construction and less in software. Therefore, the hardware similarity weight ratio is set as 0.7, and the software similarity weight ratio is set as 0.3 in this paper.

The similarity of redundant heterogeneous system is normalization of the sum of the similarities of all the different elements [8].

$$\alpha_n = \frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\delta_{ij}}{C_n^2} \quad (0 < \alpha_n <= 1) \tag{3}$$

$$H_n = \frac{1}{a_n} \tag{4}$$

δ_{ij} represents similarity between the *i - th* and the *j - th* executors in the system. α_n represents similarity of the system with *n*executors, H_n is defined as the heterogeneous degree of the system. The similarity is 1 when construction of all executors in the system are the same. The similarity is closer to 0 when the construction difference of executors is larger. Therefore, the values of A and B are inversely proportional.

3.2 Attack probability model and security gain

In this paper, the attacker without memory is used as the attack source and the attacker's attack ability is characterized by exponential probability density function. Let's suppose the attacker launches several attacks on this system . Then the probability of an executor being successfully attacked within the duration of each attack is [10]:

$$p_i(t) = \int_t \lambda e^{-\lambda t} \tag{5}$$

λ is a constant and is set as 0.3 in this paper.

In order to ensure the validity of majority voting algorithm, the number of executors *n* is set as odd:

$$n = 2k + 1(k = 0,1,2\cdots) \tag{6}$$

The value of $\alpha_n$in this paper is also defined as the random influence factor that affects the probability of the system being successfully attacked. It is assumed that the attacker has same probability of attacking each executor, and the security difference is replaced by random impact factor. For a redundant heterogeneous system with *n* executors, the probability of the system being attacked successfully can be expressed as [11]:

$$P_n(t) = \alpha_n \sum_{k=(n+1)/2}^{n} p_i^k (1 - p_i)^{n-k} \tag{7}$$

For a defender, the probability of an attacker failing to attack reflects its security performance. Therefore, the security gain of system with *n* executors is defined as:

$$\triangle_n(t) = \frac{1 - P_n(t)}{1 - P_1(t)} \tag{8}$$

3.3 Selection of executors

In this paper, the heterogeneous executors of the system can be divided into two parts: hardware heterogeneity of multi-core processors and software heterogeneity of algorithms. Based on the analysis results in Section 3.1, α_n ranges from 0 to 1. At software level, MOSS tool developed by Microsoft is used to calculate the similarity between algorithms working on different cores. Because the similarity is difficult to measure at the hardware level, set the similarity between A53 core and R5 core as 0.65 which represents a medium level of similarity. This paper only considers heterogeneous executor in the case of optical transmission equipment with single, three and five cores.

When there is only one core as the executor, α_1 =1, H_1 =1.

The set of executors for three cores is{ *A*53 - 0, *A*53 - 1, *R*5 - 0 }, the similarity matrix is as follows:

$$A_1 = \begin{bmatrix} 1 & 1 & 0.65 \\ 1 & 1 & 0.65 \\ 0.65 & 0.65 & 1 \end{bmatrix} A_2 = \begin{bmatrix} 1 & 0.233 & 0.300 \\ 0.233 & 1 & 0.240 \\ 0.300 & 0.240 & 1 \end{bmatrix} A = \begin{bmatrix} 1 & 0.770 & 0.521 \\ 0.770 & 1 & 0.512 \\ 0.545 & 0.527 & 1 \end{bmatrix}$$

According to formula (3) and (4), α_3= 0.614, H_3 =1.629.

The set of executors for five cores is{ *A*53 - 0, *A*53 - 1, *A*53 - 2, *R*5 - 0, *R*5 - 1 }, the similarity matrix is as follows:

$$A_1=\begin{bmatrix}1&1&1&0.65&0.65\\1&1&1&0.65&0.65\\1&1&1&0.65&0.65\\0.65&0.65&0.65&1&1\\0.65&0.65&0.65&1&1\end{bmatrix}A=$$

$$\begin{bmatrix}1&0.770&0.773&0.545&0.515\\0.770&1&0.786&0.527&0.512\\0.773&0.786&1&0.571&0.536\\0.545&0.527&0.571&1&0.768\\0.515&0.512&0.536&0.768&1\end{bmatrix}$$

$$A=\begin{bmatrix}1&0.770&0.773&0.545&0.515\\0.770&1&0.786&0.527&0.512\\0.773&0.786&1&0.571&0.536\\0.545&0.527&0.571&1&0.768\\0.515&0.512&0.536&0.768&1\end{bmatrix}$$

According to formula （5）， α_5=0.630， H_5= 1.587.

According to the above analysis， heterogeneous degree of the tri-core and five-core systems is greatly improved compared with that of the single-core system. However， heterogeneous degree of five-core system is slightly lower than that of tri-core system because there are more similar executors in the five-core system.

4 Simulation and data analysis

4.1 Simulation

Based on the analysis results in Section 3， Fig. 6 and Fig. 7 show the simulation diagram of the probability of the system being attacked successfully and the security gain of the multi-core system within the attack duration［10］. Table.1 shows the value of the probability of system being attacked successfully and security gain at certain moments.

The simulation results show that the probability of the multi-core system being successfully attacked is significantly lower than that of the single-core system， and the security gain is also significantly improved compared with the single-core system. Moreover， under the same attack duration， the success probability of the five-core system being attacked is much lower than that of the tri-core system， and the security performance is also significantly improved compared with the tri-core system.

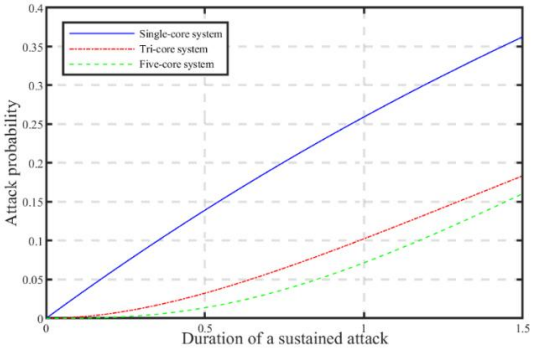


Fig. 6 The probability diagram of multi-core system being attack successfully

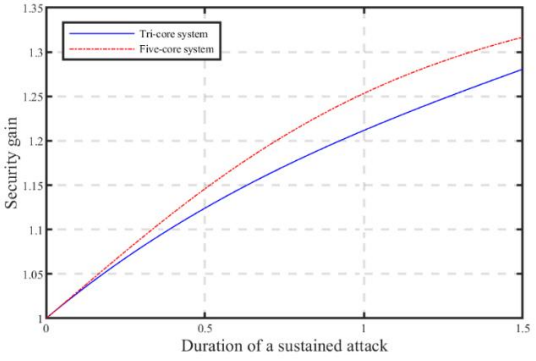


Fig. 7 Multi-core system security gain diagram

Table.1 Attack probability and security gain at some time point

	$P_1(t)$	$P_3(t)$	$\triangle_3(t)$	$P_5(t)$	$\triangle_5(t)$
0.5	13.93%	3.24%	1.125	1.37%	1.147
1	25.92%	10.24%	1.215	7.15%	1.257
1.5	36.24%	18.34%	1.287	16.05%	1.326

4.2 Performance test

The operating time of the optical transmission equipment is tested under the conditions of single， three and five cores.

It can be seen from the Fig. 8 that the data-distributing time of the multi-core system is very low and negligible. Compared with the single-core processor， the total operating time of multi-core system is mainly in the maximum processing of the core， reading register and judging. The maximum data processing time of the core depends on the core with the highest algorithm time complexity. The time of reading

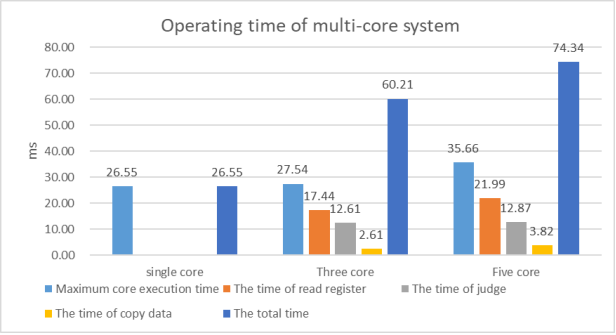


Fig. 8 Operating time diagram of multi-core system

registers for the five cores is slightly longer than that for the three cores. The judging time of the tri-core and the five-core system is roughly the same.

Results show that the operating time of the tri-core system is 2.27 times that of the single-core system, and the operating time of the five-core system is 2.8 times that of the single-core system. When the attack duration is 0.5 (common attack source), the probability of the system being successfully attacked is 13.93% for a single-core system, 3.24% for the tri-core system and 1.37% for the five-core system. The security gains of tri-core system and five-core system are 1.125 and 1.147 respectively.

5 Conclusions

In this paper, an ideal defense mechanism is designed to enhance the security performance of the Mimic SDON transmission system: redundant heterogeneous multi-core architecture in the optical transmission equipment. The optical transmission equipment consists of four modules: data-receiving module, data-distributing module, multi-core processors module and judgment module. Heterogeneity of hardware and software is realized on the multi-core processors module of the optical transmission equipment. The majority voting algorithm is used to determine the data returned by each core. And the final data is sent to PL for configuration.

Firstly, this paper theoretically analyzes the similarity, heterogeneous degree, attack probability, security gain and other parameters affecting security performance of systems. Then we calculate that the value of $\alpha_1$is 1 and the value of H_1 is 1 for single-core system; the value of α_3 is 0.614 and the value of H_3 is 1.629 for tri-core system; the value of α_5 is 0.614 and the value of H_5 is 1.629 for five-core system. Secondly, the attack probability and security gain of single-core and multi-core systems are simulated under different attack duration time. When the attack duration is 0.5, $P_1(0.5)$is 13.93%, $P_3(0.5)$ is 3.24%, and $P_5(0.5)$ is 1.37%; $\triangle_3(0.5)$ is 1.125, $\triangle_5(0.5)$is 1.147. Therefore, the heterogeneous degree and safety performance of multi-core system is much higher than that of single-core system, and the safety performance of five-core system is better than that of tri-core system. Finally, experiment shows that the operating time of single-core system is 26.55ms, that of tri-core system is 60.21ms, and that of five-core system is 74.34ms.

The operation time of redundant heterogeneous multi-core architecture is slightly longer than that of single-core system, but its security performance is greatly improved compared with single-core system. Moreover, this method has low cost with no additional hardware, and is relatively simple to implement, which can significantly improve the physical layer security of SDON system.

References:

[1] JI P. N. Software defined optical network [C]//International Conference on Optical Communications and Networks. IEEE, 2012: 1-4.

[2] CHEN M, SHOU G, HU Y, et al. Enabling software-defined optical networks based on OpenFlow extension [C]//Opto-Electronics and Communications Conference. IEEE,2015:1-3.

[3] Pashkov V, Smeliansky R. On High Availability Distributed Control Plane for Software-Defined Networks [C]//International Scientific and Technical Conference Modern Computer Network Technologies. IEEE,2018:1-10.

[4] CHEN Y, LI Q,YANG Y ,et al. Towards adaptive elastic distributed Software Defined Networking [C]//International Performance Computing and Communications Conference, IEEE ,2015:1-8.

[5] WANG Z P, HU H C, CHENG G Z, et al. Implementation architecture of mimic security defense based on SDN[J]. Journal of Network and Information Security ,2017,3(10):52-61.

[6] LEE W, CHOI Y H, KIM N. Study on virtual service chain for secure software-defined networking [C]// The International Conference on Control and Automation. 2013:177-180.

[7] WANG W, ZENG J J, LI G S, et al. Security analysis of dynamic heterogeneous redundant system[J]. Computer Engineering，2018，44（10）：42-45,50.

[8] LIU Q R, LIN S J, GU Z Y. Heterogeneous redundancies scheduling algorithm for mimic security defense [J]. Journal of Communications，2018. 39（07）：188-198.

[9] ZHANG J X, PANG J M, ZHANG Z, et al. Executors Scheduling Algorithm for Web Server with Mimic Structure [J]. Computer Engineering，2019. 45（08）：14-21.

[10] HU H C, CHEN F C, WANG X P. Performance Evaluations on DHR for Cyberspace Mimic Defense [J]. Journal of Information Security，2016 1（04）：40-51.

[11] FAN Y W, ZHU W J, BAN S H, et al. Dynamic Heterogeneous and Redundancy Data Protection Architecture [J]. Minicomputer system，2019 40（09）：1956-1961.

About the authors

ZHANG Yongzhuang was born in 1996. 11. 16. He received his B. E. degree in electronic engineering from Hebei University of Science and Technology. He is now a MA. Eng candidate of Beijing University of Posts and Telecommunications. His research interests include Embedded development and Optical communication （Email： 1329772564@qq. com）

ZHANG Huibin ［corresponding author］ was born in 1980 . He received his Ph. D. degree in Beijing University of Posts and Telecommunications in 2011. His research interests include Embedded development and Optical communication.（Email： zhanghuibin@bupt. edu. cn）

YANG Chenguang was born in 1998. 8. 14. He received his B. E. degree in Optoelectronic Information Science and Engineering from North University of China. He is now a MA. Eng candidate of Beijing University of Posts and Telecommunications. His research interests include Embedded development and Optical communication （Email： 1361301997@qq. com）

国家基因库生命科学数据可信共享系统研究

丁远彤[1，2]，谈聪[1，2]，陈凤珍[1，2]，王丽娜[1，2]，徐志成[1，2]，潘光明[3]，杨涛[1，2]，杨帆[1，2]，高飞[1，2]，韦振勇[1，2]，游丽金[1，2]，徐翌钦[4，5]，聂永星[4]，魏晓锋[1，2*]

[1]深圳国家基因库；

[2]深圳华大生命科学研究院；

[3]深圳华大智造科技股份有限公司；

[4]深圳国家高技术产业创新中心；

[5]深圳大学中国经济特区研究中心

摘　要：背景　，基因测序技术的迅速发展，使得以基因数据为代表的生命科学数据极具增长，基于基因大数据的解读和临床应用上的突破，极大地提升了人类医疗和健康水平，但海量基因数据也为数据隐私、安全与监管带来严峻挑战。**研究**，依据深圳国家基因库这一国家级综合基因库的一期工程现状和发展需求，我们研究设计了二期工程方案，重构了由数据计算模块和数据汇交模块为核心的面向生命科学数据的可信共享系统方案，并实施前期验证工作。重点讨论了依据层和模块的设计，如何支撑本系统的细节，它们是：1、安全策略；2、安全多方计算（MPC）应用场景、MPC应用框架、MPC功能分层、MPC功能模块组网；3、基于Intel SGX的同态加密（HE）；4、数据标准子模块、数据抽取子模块、数据质控子模块、数据管理同步子模块、归档子模块；5、区块链数据溯源子模块；6、云平台建设。结论　，在系统存储和计算的承载能力为百Pb的规模下，局部功能已经在2020年中国和全球共享流感数据倡议组织（GISAID）合作的新冠病毒基因组分析工作中得到实际验证，证明达到可信共享的设计目标。

关键词：生命科学数据、可信共享、区块链、安全多方计算、同态加密、深圳国家基因库

Study of China National GeneBank Life Science Data Trusted Computing and Sharing System

Ding Yuantong[1,2], Tan Cong[1,2], Chen Fengzhen[1,2], Wang Lina[1,2], Xu Zhicheng[1,2], Tao Yang[3], Yang Fan[1,2], Fei Gao[1,2], Wei Zhenyong[1,2], You Lijin[1,2], Pan Guangming[1,2], Xu Yiqin[4,5], Nie Yongxing[4], Wei Xiaofeng[1,2*]

1.China National GeneBank, BGI-Shenzhen, Shenzhen, 518120, China;

2.BGI-Shenzhen, Shenzhen, 518083, China;

3.MGI, BGI-Shenzhen, Shenzhen, 518083, China;

4.State High-Tech Industrial Innovation Center, Shenzhen, 518083, China;

5.China Center for Special Economic Zone Research, Shenzhen University, Shenzhen, 518083, China

Abstract: Background　With the rapid development of gene sequencing technology, life science data represented by genetic data has grown tremendously. The interpretation of genetic big data and its breakthroughs in clinical applications

have greatly improved the level of human health and wellbeing, but massive genetic data has also brought severe challenges to data privacy, security and supervision. Method Based on the current status and development needs of the first phase of the China National GeneBank, a national comprehensive gene bank, we researched and designed the second phase of the project, reconstructed the life science data oriented trusted sharing system with data computing module and the data collection module as the core, and performed the preliminary verification. We focus on the design of layers and modules that support the system. They are: 1. Security strategy; 2. Secure multi-party computing (MPC) application scenarios, MPC application framework, MPC function layering, MPC function module group Network; 3. Homomorphic encryption (HE) based on Intel SGX; 4. Data standardization sub-module, data extraction sub-module, data quality control sub-module, data management synchronization sub-module, data archiving sub-module; 5. Blockchain data traceability sub-module; 6. Cloud platform construction. Result In conclusion, under the current 100Pb scale of storage and computing capacity, the local functions have been verified in the work of the new coronavirus genome analysis platform between China and the Global Shared Influenza Data Initiative (GISAID) in 2020, and proved to meet the design goals.
Key words: Biological data; trusted sharing; blockchain; secure multi-party computing; Homomorphic encryption; China National GeneBank

1 引言

生命科学大数据是指与人类及其它生物相关的各类数据，包括基因数据、临床病理数据、影像数据、电子病历（EMR）、生活习惯数据、环境数据等诸多不同类型、不同层次和不同维度的数据。在数据获取、存储、分析和分发各环节的需求量均达到或超越如天文学和社交媒体的主要大数据产出领域。研究测算，到2025年，全球仅基因组学数据产出量即可达到2～40Eb/年（10^{18}字节），这充分表明生命科学已经进入大数据时代。

由于基因数据的敏感性，其安全隐私保护要求远高于一般数据。美国已于1996年和2009年分别发布《健康保险隐私及责任法案》（HIPAA）和《经济和临床健康信息技术法案》（HITECH），前者就医疗电子信息的传送、访问和存储做出了详细规定；后者要求数据存储者实施物理、行政与技术的解决方案，用以保护生命科学数据免遭泄漏。美国食品和药品管理局（FDA）在2017年1月就开始与IBM合作开始区块链研究，共同探索电子医疗记录数据、临床试验数据和可穿戴设备中的健康数据如何通过区块链得到更好的共享和审计效果，通过在不可更改的分布式账本上保存所有交易的审计跟踪，实现数据交换过程的可靠性和透明性。MIT、斯坦福等高校也开始投入区块链和密码学技术研究，并与业界展开广泛合作。欧盟也在2018年5月起全面实施“史上最严数据安全法规”的《通用数据保护条例》（GDPR），旨在对个人数据生产与应用的各环节加以安全规范，最大限度地保护个人数据隐私。我国于1998年发布《人类遗传资源管理暂行办法》，并于2019年6月重新修订成为更加严格的《中华人民共和国人类遗传资源管理条例》，并于同年7月1日开始实施，此举加大了对我国人类遗传资源的保护力度。

深圳国家基因库是由国家发展改革委员会、财政部、工业和信息化部、卫生健康委员会四部委于2011年批复同意，以国家发改委和深圳市政府为联合理事长单位的理事会为指导，由深圳华大生命科学研究院承建的国家级综合性基因库。

本文总结我们的前期实践及验证，研究提出国家基因库生命科学数据可信共享系统的建设方案，其局部功能已经在2020年春季中国和欧盟的新冠病毒基因溯源工作中得到实际验证，证明达到了可信共享的设计目标。

2 现状综述

在理论研究方面，文献［1］详细分析和总结了以同态加密（Homomorphic encryption，HE）、安全多方计算（Secure Multi-Party Computation，MPC）为代表的现代密码学技术和以差分隐私为代表的扰动方法技术这三者之间的利弊，指出：HE可在确保在保护数据所有者的利益和隐私的前提下，实现多方数据可信交换和协同计算、联合挖掘；MPC包括混淆电路、不经意传输、零知识证明、秘密分享等技术，可在无可信第三方、多方之间在互不公开数据的前提下实现协同计算；差分隐私技术仅通过随机化和利用随机噪声扰动数据便可以实现，并且在机器学习中部署并不会

带来过多额外的计算开销。文献［2］提出隐私的定义与度量，隐私保护技术及评估，分类及比较数据失真、数据加密、数据限制发布等保护技术。文献［3］提出采用近似值（例如98%）目标、高效的私有计算和优化通信双方协议的方法，比以往方法块2-3个数量级。

在方法和政策方面，文献［4］提出一种协议以隐私保护的方式计算χ^2统计量来防止数据泄露，它只会返回是/否答案，不透露任何其他信息，对于百万级的GWAS（Genome-Wide Association Studie），使用MPC可在2毫秒内完成。文献［5］讨论使用遗忘有限自动机、加法同态加密、混淆电路和秘密分享来保护隐私的基因测试。文献［6］讨论了美国基因组和EMR数据隐私和机密性，在临床、技术和伦理方面，以及生物医学研究中保护隐私的基因型-表型连锁的潜在解决方案。文献［7］提出对于HIPAA隐私和安全规则下的数据和患者同意条件下的数据采用清理结构化或非结构化数据的安全措施方法，而对于伦理学主题采用加密数据的安全措施方法。

在具体实践成果方面，文献［8］提出DNA数据在上传到外包云之前进行隐私化加密，在搜索时采用双线性组上的子组决策隐私保证。文献［9］提出SecureLR（Secure Logistic Regression）模型新的框架，使研究人员可以利用公共云服务器的计算和存储能力，在不损害数据安全性或效率的情况下，对生物医学数据进行学习和预测。文献［10］采用Intel SGX的加速硬件和PREMIX隐私评估软件，针对信息性识别最有区别的单核苷酸多态性（SNP），基于期望最大化（EM）的最大似然估计器来识别单个混合物，根据仿真和1000个基因组数据的实验结果，证明了所提出框架的效率和准确性。文献［11］介绍一种混合框架安全，它可以使用HE在联邦基因组数据集上安全地执行GWAS，并且用Intel SGX确保高效率和私密性，安全性比现有最佳安全计算技术快4.82倍；文献［12］介绍MaskAl方法读取比对方式，结合快速的预处理步骤，建立原始基因组数据与经过清理的读数比对的算法，屏蔽敏感数据，并使用Intel SGX与公众大规模并行化的比对软件云和新兴飞地云，MaskAl几乎与纯文本方法一样准确，比对速度比当前的隐私保护快87%，同时还减少了内存空间和网络带宽。

文献［13］指出在生命科学数据采集、存储、分发和分析方面，均需要新技术来满足未来的挑战。文献［14］采用无信任中心的区块链云共享服务，采用智能合约和访问控制机制，在为患者的EMR隐私提供有效保护的前提下，带来可与通用云服务相媲美的边缘解决方案。文献［15］通过定制化的访问控制协议、对称密码技术和改进的共识机制，实现EMR在跨医院、跨网络共享，实现基于区块链的EMR信息管理系统MedBlock，同时节省电力和网络资源消耗。为了解决严格监管和官僚低效而减缓EMR的创新问题，文献［16］提出一种MedRec原型验证方案，采用分散的记录管理系统为患者解决区块链共享，鼓励医疗利益相关者参加作为区块链“矿工”的网络，为他们提供访问汇总匿名数据作为挖掘奖励，通过证明来维持和保护网络的回报工作。文献［17］尝试将原始EMR存储于云和索引中，并保留在防篡改财团区块链中，区块链的索引确保不能随意修改EMR，智能合约预定访问权限。

文献［18］描述了一种GWAS分析方案，可进行质量控制和群体分层校正分析，同时保持潜在基因型和表型的机密性，证明了该协议可适用于百万人的规模样本。文献［19］通过应用近似值来开发一种保护隐私的半并行GWAS算法——HE方案HEAAN，将Fisher评分和半并行GWAS算法修改为使用多种优化方法对HE数据进行有效计算，替代通过伴随矩阵进行矩阵求逆，避免计算超大规模的多余矩阵，并变换算法转换为近似版本。

文献［20］介绍了一种HE方法，可以掩盖个体的基因型和表型，并且适合定量遗传关联分析，加密密文和未加密明文经过解析可互换，加密使用高维随机线性正交变换密钥，从而避免在具有正态分布误差的线性模型下的数量性状数据保持不变，以防止使用现有密码对密文进行解密的蛮力或降噪攻击。文献［21］提出基于HE的几个隐私保护框架，可在5.6小时内进行100，000个体和500，000个单核苷酸多态性位点（SNPs）的GWAS分析，但该方法仅支持简单分析。文献［22］指出HE的3种局限性，即1、属于公钥密码体制，其算法复杂度高于共享密钥体制和扰乱技

术；2、通常只能实现加法和乘法，不足以支持更多的数据挖掘方法；3、现有的方法通常都只局限于某一种数据挖掘方法的隐私保护。文献［23］指出目前主流的MPC包括2类隐私保护集合交集（Private Set Intersection，PSI）技术协议，即传统的公钥加密、混乱电路、不经意传输的PSI协议和新型的云辅助的PSI协议。

文献［24］提出了一种基于Pailier密码系统和保序加密（OPE）的基因组数据隐私保护方法，采用通过性能和安全分析对该方案进行了评估，该方法可以支持数据查询和提取。文献［25］指出区块链的8种解决方案：1、保护患者和提供者的身份，2、管理医药医疗设备供应链，3、临床研究和数据货币化，4、医疗欺诈检测，5、公共卫生监督，6、启用公开和开放地理标记数据，7、连接物联网自治设备、可穿戴设备、无人机和车辆，提供智能健康城市，8、在危机场景指示和恢复场景中启用区块链的增强现实。

文献［26］提出了一种针对EMR数据的sS-VM（sparse Support Vector Machine）分类器，开发了一种cPDS（cluster Primal Dual Splitting）算法，用于解决分布式的大规模sSVM问题，与其他分布式算法相比，它收敛速度更快，通信开销更小。但该方法仅适用于稀疏数据，另外在梯度交换中是否存在原始数据泄漏风险尚无严格证明。文献［27］针对医疗保健或生物医学区块链列出5个关键点：1、医疗保健或生物医学区块链应用的优势；2、医疗保健应用程序中底层区块链平台的关键特征描述；3、基于PRISMA（系统评价和元分析的首选报告项目）陈述，开发技术的系统评价方法；4、比较10个流行区块链平台的21种医疗保健相关技术功能；5、讨论审查结果和审查局限性。

一些研究还涉及无可信中心的多秘密共享方案：1、参与者联合生成多个共享的随机密码；2、基于零知识证明协议，使任何人可验证参与者分发信息的有效性；3、采用老成员协助新成员获得秘密份额及改变更新多项式次数的方式，分析结果表明该方案具有较高的安全性［28］。基于分裂密钥RSA（split-key RSA）算法，提出一种新的混合架构，在不影响安全性的情况下，对密钥进行了统一的分割，将控制中心端的密钥参数缩短为32bit，并且不随RSA的强度而变化，使得控制中心端的在线计算量显著降低，RSA2048模式下降低为原来的1.6%［29］。设计可有效地防止成员之间的恶意欺诈行为，每个参与者提供相同的秘密份额，在黑盒子中协同产生各自的秘密份额，从而避免可信中心的权威欺骗［30］。共享的多个随机秘密是由参与成员共同产生的，密钥份额的有效性不仅可以被份额持有者自己验证，而且可以被其他任何成员验证，可用于设计电子投票协议、密钥托管协议［31］。利用双线性对的性质，任何人都能验证分发的秘密份额和更新的秘密份额的正确性［32］。基于差分隐私，在不实质上改变系统效用的情况下，有可能阻止对信标服务的攻击，该方法受到共享中心iDASH（隐私与安全研讨会）挑战场景和评估方案设计的限制，应用场景较有限［32］。

另外一些研究强调了区块链应用于医疗保健的最新发展状况、其局限性和未来研究领域［33］。探讨了应用于临床试验的区块链核心功能以及实施过程中的潜在影响［34］。提出了一种结合加密隐私保护技术的新方法，具有区块链在HE和MPC方面的可审核性，通过授权，公民个人决定谁可以查询和访问他们的基因组数据，并确保端到端数据的机密性［35］。讨论区块链在医学领域的潜在应用，包括可互操作的健康数据访问、数据存储和安全、基于价值的支付机制、医疗供应链效率等［36］。探索了一种新颖的方法来公开发布由该研究组收集的超过11，000个数据集，总计超过30 TB［37］。基于围绕区块链分类账构建的对等文件共享网络，为防止数据篡改提供了分布式解决方案［38］。

3　生命科学数据可信共享系统研究

3.1　一期工程现状简介

国家基因库的一期工程的结构如图1的现有技术部分所示。

从结构上看，数据计算包括：各种服务器、云平台的基因安全容器、工具标准流程描述语言、生物信息分析工具、工作流引擎、生物信息分析、生命科学大数据分析挖掘框架等数据库子系统。数据汇交包括：基础的互联网接入、路由及交换，设备包括路由器交换机以及它们的管理系统。

从规模上看，国家基因库已颇具规模。一期工程生物样本库的存储已达到千万样本的存储量级，数字化平台每年产生数个Pb（Petabyte）存储量的海量测序数据。迄今已经积累了近百Pb存储和计算资源，计算能力为691万亿次/秒，裸存储能力是88PB，存储性能为150GB/秒，年可处理20万人类基因组样本标准分析。由此可见，在我国的互联网基础建设具有较高水平的前提下，无论是数据量还是算力，生命科学数据已经形成可观的规模。

从资源管理上看，目前资源分散，尚未统一管理。一期工程基本上只是面对国家基因库本身，尚未形成覆盖、整合和管理散落在企业、高校、医院和科研机构的生命科学数据的能力，造成数据孤岛、管理孤岛现状。

从安全上看，数据外流风险严重。一期工程的数据安全策略只是采用常规的用户名＋密码的数据库方式管理，现阶段一些应用过分依赖NCBI、EBI等国际数据库，使用时需要数据上传，从而导致我国的生命科学数据外流的严重威胁。

3.2 二期工程——可信共享系统需求分析

为了解决上述一期工程中存在的问题，我们提出实时旨在通过建设可信安全共享系统的国家基因库生命科大学数据二期工程的方案，其设计目标是：1、实现数据本身的安全共享、安全计算和统一管理，强化国家基因库在我国生命科学大数据领域的主导地位。2、实现国内各企业、高校、医院和科研机构产生的生命科学大数据的安全共享、安全计算和统一管理。3、兼顾一期工程，平滑升级、柔性过渡。4、为国家统一的生命科学大数据可信共享门户提供系统支撑，以推动生命科学技术和现代健康产业的发展。二期工程如图1中的可信共享系统所示。

依据一期工程的现状和二期工程的目标，作为数据计算核心，需要引入MPC技术，并尝试采用HE技术，建立可信的计算环境，使得国家基因库本身和企业、高校、医院和科研机构等能在不接触对方数据的前提下进行数据分析、共享分析结果。作为数据汇交核心，对于全部用户的接入，需要纳入区块链、数据存储、数据加密等技术，对数据汇交和处理过程进行存证，以保证生命科学数据汇交的各个处理步骤都是可预测、可追溯、可验证和可监管，最大程度地保护数据的安全，实现对于PMC的支撑。

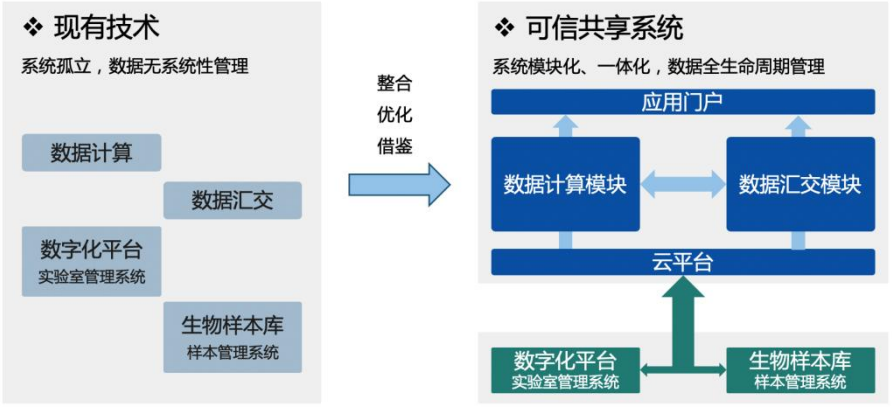


图1 现有技术平台与可信共享系统关系

3.3 可信共享系统创新与设计

依据上述需求分析，我们设计可信共享系统的数据汇交模块和数据计算模块具体如图2所示。其中各部分设计描述如下：

3.3.1 层的新设计

依据可信共享，从业务层面我们划分以下4个层面，以降低系统的复杂度，提升系统的可维护性。

应用层，它用户群通过应用层这一用户门户，提供基于可信共享的数据归档功能、数据管理功能、计算工作空间和工具管理功能，使得能力层响应用户层的业务数据请求，并把处理结果反馈给用户。不同于传统的数据库管理系统，应用层的创新设计是全程采用可信共享支撑，采用区块链管理。

能力层，它是用户直接使用的、与业务相关

图 2　可信共享系统功能框架图

的各系统集合，是系统的关键核心。主要包括数据计算核心模块和数据汇交核心模块。其中，数据计算核心模块包括：计算工作空间管理、工作空间权限管理、生物信息工作流、MPC 和 HE 数据计算算法库；数据汇交核心模块包括：数据标准子模块、数据抽取子模块、数据质控子模块、数据管理子模块、数据归档子模块、数据溯源子模块。它们直接承载着国家基因库开放共享的核心能力，如生物样本库的样本管理系统和数字化平台的实验室管理系统，通过 Restful（Representational State Transfer）标准开发的 Web Service 接口，向数据汇交模块传输数据，并且通过质控等子模块，将数据归档到系统中。能力层的创新设计中，我们基于可信共享，首先在数据的出入口，均采用可信共享设计和管理。其中，在数据汇交模块采用区块链连接管理，对于数据计算模块，逐步修改相关软件系统，支撑 PMC，同时，在服务器中采用内置第六代 Intel SGX 的加速硬件和依据 PREMIX 隐私评估软件修改加速软件，实现 HE 的加速。

数据层，它是应用层和能力层的数据支撑，主要功能是将样本管理系统的样本信息、实验室管理系统的实验信息和测序数据、其他独立汇交的数据做初步的汇聚，为上层应用提供归档和计算数据源。

平台层，它是完成各类数据从抽取、质控、管理、归档到计算等全过程的软硬件设备以及软、硬件设备运行所需要的实体硬件环境的有机组合，是本系统建设的基础平台，包括计算服务器、存储系统、网络设备和平台系统管理软件等。

3.3.2　模块的新设计

限于一期工程中已经划分数据计算和数据汇交 2 大核心模块并且已经运行多年，为了兼顾一期工程，平滑升级、柔性过渡，我们在两大核心模块的基础上，依据可信共享系统的设计目标设计如下：

数据计算核心模块新的工作流程

1、用户通过数据汇交模块的数据接口对接存储的生命科学大数据，采取区块链存证措施和权限管理确保数据安全。

2、通过 WDL 工具流定义工作流程，通过 Docker 镜像进行计算环境封装，通过 Cromwell 调配计算资源。

3、建立 MPC 和 HE 算法库，用户数据存储在不同节点中，发起计算流程后，协同计算方节点收到发起并确认后，由 MPC 节点进行联合计算。

数据汇交核心模块新的工作流程

1、研发设计标准化数据接口对接国家基因库的生命科学大数据现有生物样本库的样本管理系统 BRMS 和数字化平台实验室管理系统 LIMS，从 BRMS 系统中抽取样本相关信息、从 LIMS 抽取样本编号及实验测序相关信息和数据文件等。

2、对获取到的信息和数据文件进行数据质控，通过数据质控的数据文件和元信息会进行

归档。

3、结合区块链技术，对数据流转的全过程进行上链存证，为后期的数据计算模块提供高质量、可溯源、可监管的数据基础。

3.3.3 可信共享系统技术架构新设计

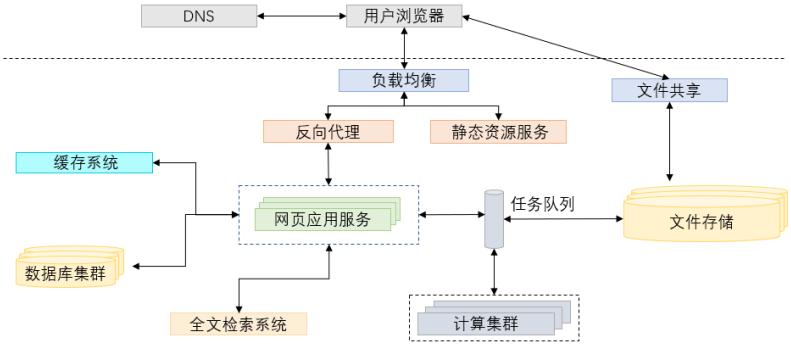


图 3　可信共享系统的技术架构

如图3所示，我们设计可信共享系统的技术架构的新的技术指标是：

性能，支持 1000 QPS（每秒钟请求数），同时500用户在线，服务响应时间满足项目设计目标。

安全，平台应该保证数据的机密存储，保证访问信息的传输安全。防止用户数据的泄露，防止非法访问造成的服务不可用。

可用，所有提供服务需要保证高可用，不允许因为单节点故障而导致服务不可用。

扩展，采用伸缩性支持较好的方案，使得业务增加时可以横向或纵向扩展。

3.3.4 可信共享系统技术新方案

3.3.4.1、应用门户

如图2所示，将数据计算模块、数据汇交模块、云平台的能力集成统一为数据共享门户，建立数据汇交和数据计算的信息通道，具备统一用户认证管理和权限管理。使用高可用的Web应用架构，为用户提供一站式的可视化数据应用服务，主要包括：数据管理、工具管理、工作空间、数据归档。

3.3.4.2、数据计算模块的安全策略

图4　数据计算模块

我们设计数据计算模块整框架如图4所示，由计算工作空间管理模块、工作空间权限管理模块、生物信息工作流模块、安全多方计算（包括MPC和HE）模块、数据计算模块算法库共计5个模块构成，整体计算系统以数据汇交模块存储的海量数据构建的数据集市为支撑，结合灵活工作空间管理模式为核心，同时继承和扩展一期工程的权限管理以及新增创新的二期工程的区块链存证等多种模式确保用户数据安全共享。对用户不同计算场景需求，提供两种计算模式策略：

（1） 单用户模式：通过WDL工具流定义工作流程，通过Docker镜像进行计算环境封装。通过Cromwell调配计算资源。

（2） 多用户模式：MPC模式，HE模式，满足对数据隐私安全有更高要求，不直接共享原始数据的用户联合计算需求。

各个模块具体安全策略设计如下：

计算工作空间管理模块安全策略，可信共享系统采用工作空间构建计算沙箱，设计包含工作空间详情页、数据集文件详情页、工具集详情页、计算监控详情页。计算用户可个性化的组合不同的组件构建工具类、数据集类以及测试或协作计算类的工作空间应用不同共享场景。可信共享系统以工作空间维度管理数据集市中公共数据集和受限数据集，计算用户页可以按工作空间创建个性化数据集或者通过共享空间方式，与不同地域研究者或者团队成员进行协作分析。可信共享系统会结合工作空间权限功能、文件加密以及后续区块链等多种方式确保用户数据集协作分析的安全。

工作空间权限管理模块安全策略，为了解决控制保护隐私数据和易于共享之间的矛盾，传统的方法是数据拥有者管理工作空间的权限分配，但他无法管理授权这对克隆工作空间的派生数据的数据管理。我们设计的权限管理模块将工作空间与授权标签结合。授权标签是工作空间的授权域，包含一系列授权用户组，允许具有相同标签的人访问工空间。当用户使用“授权域”克隆工作区时，授权标签将与新工作空间副本一起保留，并且想要访问该副本的任何人都必须拥有徽章。用户不再需要担心意外共享敏感数据，因为如果用户尝试与没有正确徽章的用户共享克隆的工作空间，那么该用户将无法输入。可信共享系统用户权限默认分为三个级别：READER， WRITER和OWNER。每个访问级别代表一组扩展的权限。并支持更细粒度的开发者文件权限管理。

生物信息工作流模块安全策略，可信共享系统工具流语言通过生物信息分析工具调度引擎Cromwell将WDL描述的workflow转化为批量计算的作业（Job）运行在的计算容器实例中。具体包括：WDL分析流程定义、Dockerfile方式构建镜像、任务投递。

安全多方计算MPC模块安全策略，当涉及多个用户联合计算时采用MPC的计算模式。在安全计算环境的基础上加强数据隐私保护，且该模式有利于进一步拓展可信共享系统的对外数据协作能力，提供更加灵活的联合计算模式，主要实现方式是基于混淆电路和不经意传输协议，将计算逻辑转化为布尔电路，并加密传输电路及标签数据，最后各方解密计算结果。考虑到部分生命科学数据的高度敏感性与不可篡改性，这种框架天然适合于多方敏感数据的协同计算。具体分功能、架构、组网和算法库4个部分，描述如下：

MPC功能设计，MPC技术在本系统中的应用场景如图5所示，用户数据存储在不同网络节点中，发起计算流程后，协同计算方节点收到发起并确认后，由MPC节点进行联合计算。MPC平台允许拥有数据的各方在在不泄漏原始数据的情况下完成协同计算，同时与区块链技术进行结合，制定一个标准化的组学数据共享协议，将每一次计算与交互的日志进行区块链存证，确保计算过程的公开透明，实现多方共赢。 MPC和区块链的整合应用框架如图6所示，用户数据存储在不同网络节点中，发起计算流程后，通过区块链凭证实现数据的授权。各节点在区块链上接收到授权凭证并确认有效后，由MPC节点进行联合计算。

MPC功能主要支持多用户联合计算场景，支持模式包括：平台内部多用户MPC计算，平台内部与外部用户MPC计算，平台外部多用户MPC计算。

MPC架构设计，MPC技术在本系统中的功能分层如图7所示，使用MPC技术进行数据加密计算的主要步骤：计算授权、数据传输、计算执行和计算结果返回。

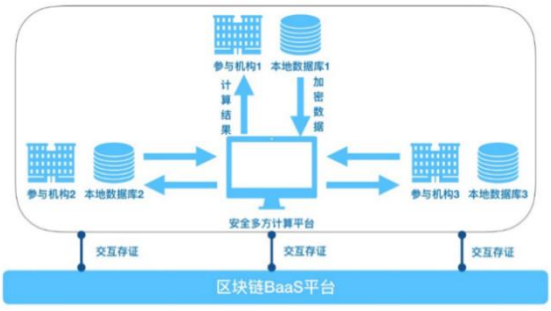


图5　MPC应用场景

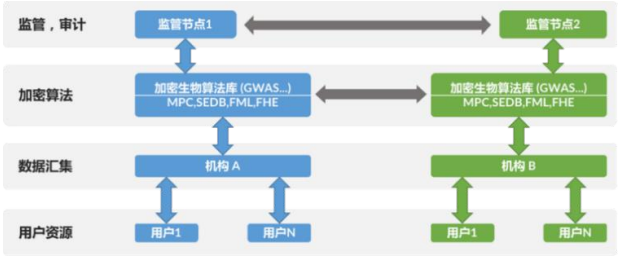


图6　MPC和区块链的应用框架

MPC组网设计，如图8所示，不同功能模块在实现MPC任务时的业务流程如下所述：

1）后台定期和MPC计算节点进行通信，获取可用的MPC计算节点列表，刷新t_mpc_node表里的节点可用状态。支持MPC节点IP白名单设置，t_mpc_node_port列表用于保存mpc节点的端口号状态。

2）启动运行后主动跟WEB后台上报状态（首次启动时需要上传本节点的数据加密公钥，上传格式和加密算法需要支持RSA加密、ECC加密）。后续定期获取WEB后台的查询请求并进行响应，算法执行模块client _mpc。

3）提供交互界面给用户进行算法选择、计算节点选择、可加入计算任务选择；用户选择包括两类：a）加入已发起的计算任务；b）自己创建新的计算任务。用户自行创建任务时可设置任务参数，比如允许他方加入计算的门槛条件、任务等待时长等，WEB前端/Client客户端。

4）后台根据用户创建的新计算任务在t_mpc_task列表中创建新的mpc_task记录，生成全局唯一的task序列号，保存task计算参数；同时根据可用端口号筛选出可参与本次隐私计算的mpc节点，在t_mpc_task_node中插入可参与隐私计算的隐私节点记录，节点管理模块。

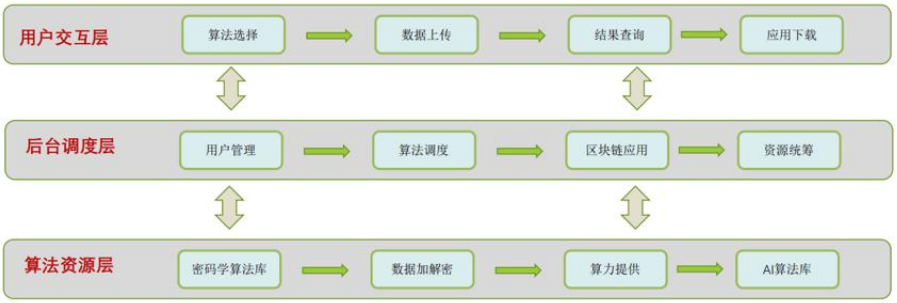


图7　MPC功能分层

MPC算法库设计，数据计算模块算法库分为两部分：通用分析流程算法库和MPC算法库。其中，前者主要由5个子方向共约12个算法工具组成：基于全基因组测序的肿瘤基因组学研究及精准医学相关工具、单细胞测序相关分析工具开发、合成生物学关键酶基因挖掘分析工具、转录组测序的动植物基因表达调控网络分析工具、宏基因组测序相关在线分析工具。后者针对可信共享系统数据场景，我们将设计并开发相关常用生物分析算法的MPC算法库，重点支持全基因组关联分析算法GWAS。

HE整体设计，参考［10-12］给出的方法，尝试采用两种Intel SGX硬件加速，即将Intel SGX与同态加密一起使用的混合方法，和仅使用Intel SGX的安全硬件方法。SGX允许在称为安全区的CPU安全段内执行应用程序的某些部分，允许在不解密数据的情况下对加密数据执行计算。我们设计混合框架SAFETY模型用以解决以下两个问题，即1、HE与SGX融合在一起的问题，2、开发安全且和可扩展的基因组数据计算问题。该模型的主要目标是依靠现有安全计算方案提供的更好的安全保证，以及对任意大小量级的数据而言，

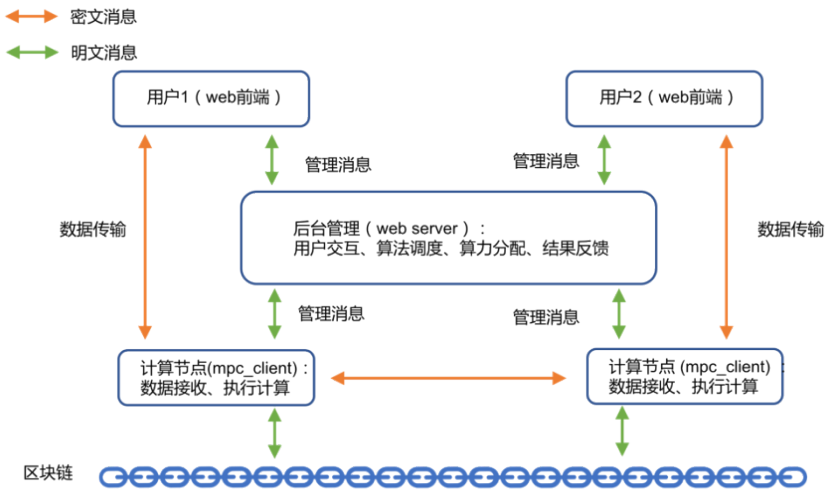


图8　MPC功能模块组网

可以速度更快地执行计算。此外，还克服了HE在处理高阶多项式方面的不足，利用Intel SGX扩展指令集，简化了安全地执行和评估GWAS在联邦架构中的处理流程。我们设计了四项统计测试：连锁不平衡（LD），哈迪-温伯格平衡（HWE），趋势卡方测试（CATT），费舍尔精确测试（FET），以评估各种设置下的安全性。使得每个数据所有者完全不了解参与同一分析的其他数据所有者的贡献，最终结果仅向研究人员透露，而没有透露数据所有者的个人贡献，这使我们能够保留每个数据所有者输出的隐私。

3.3.4.3、数据汇交模块安全策略

数据汇交模块作为可信共享系统的数据源底层，我们的安全策略是设计标准化数据接口对接基因库现有业务系统生物样本库的样本管理系统BRMS和数字化平台实验室管理系统LIMS，从BRMS系统中抽取样本相关信息、从LIMS抽取样本编号及实验测序相关信息和数据文件等。并对获取到的信息和数据文件进行数据质控，通过数据质控的数据文件和元信息会进行归档。在安全方面，采用区块链技术，对数据流转的全过程进行上链存证，为后期的数据挖掘分析和数据库建设提供高质量、可溯源、可监管的数据基础。如图9所示，具体设计是：

图9　数据汇交模块框架

数据标准子模块，建立一套生物组学数据的元数据和数据文件标准，按照数据类型的不同主要包括：项目、样本、实验、测序、组装、变异、蛋白、代谢等。制定各数据类型编号规则，建立

数据字典，设计数据标准展示页面。

数据抽取子模块，根据数据标准规范，建立数据抽取的标准化接口，对接基因库各业务系统，选择性的获取各类型数据标准中覆盖的一部分重要信息。在抽取的过程中按照标准中定义的数据值域范围、字段数据类型、是否必填等进行数据的初步校验。具体包括实验测序信息抽取、样本数据抽取、项目数据抽取、数据整合。

数据质控子模块，主要分为元信息质控和数据文件质控。由于各业务系统本身的标准不一，各项目间的研究内容、填写习惯等差异较大，从各业务模块抽取的数据，需要进行数据清洗和质控。清洗方式包括：按照归档系统的数据质量要求进行数据格式标准化，内容表述规范化，数据文件的完整性和可用性校验等。数据质控子模块实现对数据质量的把控，保障归档数据的完整性、准确性和一致性，有效消除数据在处理、压缩、拷贝、传输和存档过程种出现的破坏。数据质控工作流融入人工审编和机器自动化质控，能有效地组织不同专业背景的审编人员，兼顾数据质量和数据规模两个方面。

数据管理和同步，通过数据质控的项目、样本、实验测序等元数据和数据文件，将从各业务系统抽取到数据汇交模块，为了对汇交的数据进行统一管理，在汇交模块下做账号权限设置和数据分级管理。它们是账号权限设置、数据分级管理、数据同步。

数据归档子模块，致力于组学数据的存储、管理和共享，促进组学数据的再利用。它不仅可接收来自基因库各业务系统抽取的数据，也能归档来自全球的生命组学数据，同时也为可信共享系统的数据汇交提供数据对外共享的能力。数据归档子模块主要包括项目、样本、实验、测序、组装、变异、蛋白、代谢等数据的递交流程建立，存储备份和开放共享。

区块链数据溯源子模块，设计实现数据汇交的全流程可确权、可监控、可溯源，引入区块链技术，建立可信的数据全生命周期管理系统。主要支撑接入数据归档与计算模块，支持数据存证、数据溯源、计算存证、数据监管与审计功能，实现数据全生命周期流程管理。数据溯源能力构建主要基于区块链技术，区块链作为一种集合分布式存储、点对点传输、共识机制、加密算法等技术的组合式创新应用，为数据自治和跨机构共享交换，提供了崭新的解决方案。区块链可以保障数据存证和溯源信息不可被任何中心化平台非法使用、篡改和删除，使得数据交互方可以不依赖第三方机构进行价值传递，并保证交易记录公开透明、不可篡改，极大地降低信任成本，提高交易效率，形成高效的多方利益分配体系，并为数据共享进行安全、透明的追溯审计。

区块链数据溯源子模块功能设计，子模块框架如图 10 所示，支撑接入数据汇交与计算模块，支持数据存证、数据溯源、计算存证、数据监管与审计功能，实现数据全生命周期流程管理。功能包括：数据源存证流程、数据汇交存证流程、数据计算存证流程、数据监管与审计流程。

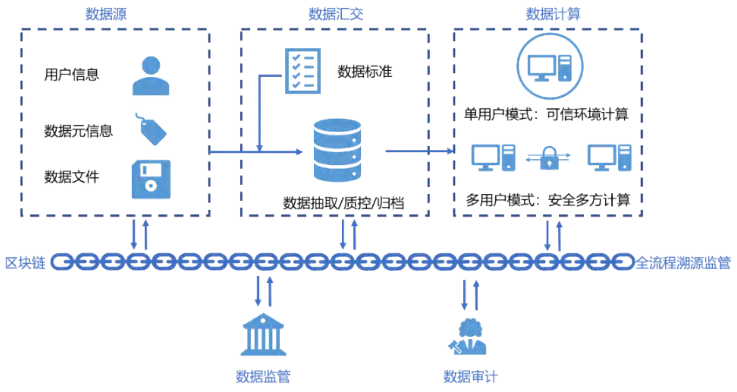


图 10　区块链功能框架

区块链数据溯源子模块架构设计，区块链模块与其它模块组网模式如图 11 所示，区块链模块整体系统架构如图 12 所示，包括：区块链系统、存储模块、计算模块。完整的区块链模块整体系

统架构包括BaaS、PaaS、IaaS分层框架，其中PaaS层的功能模块由区块链底层框架实现，不同的区块链底层框架通过实现PaaS层的功能模块提供不同的区块链应用给上层业务系统调用，例如存证应用，智能合约应用等。BaaS层的功能模块一般由业务系统中通过调用区块链底层框架的应用接口来实现，以支持不同应用场景及业务流程，例如授权审批、监管审计等业务应。

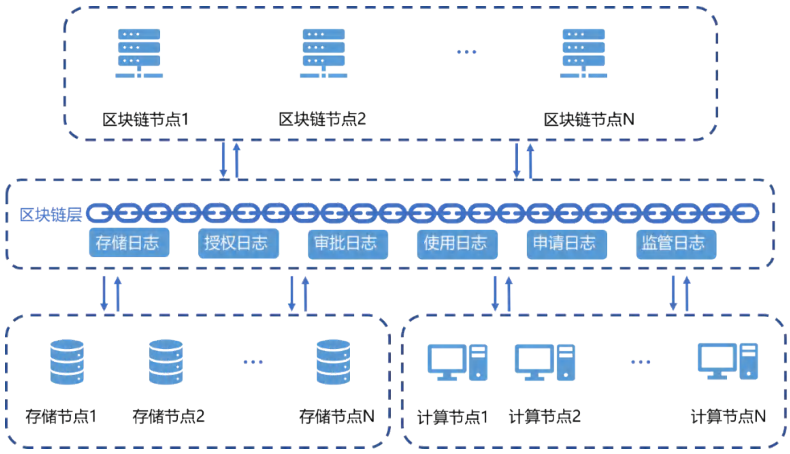


图11　区块链模块与其它模块组网模式

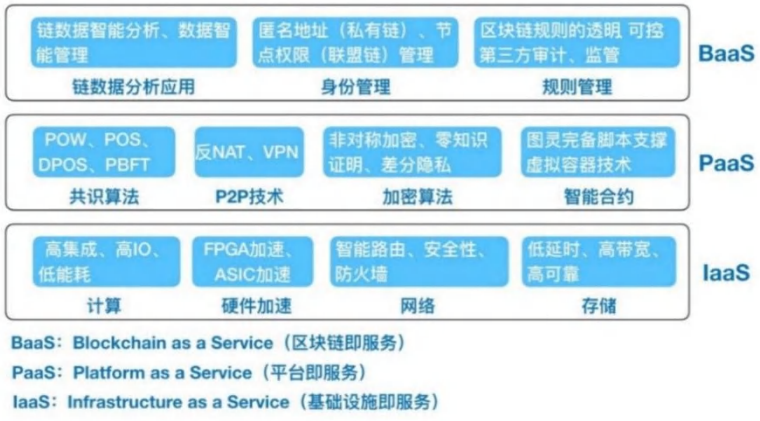


图12　区块链模块整体系统架构

3.3.4.4、云平台

云平台作为可信共享系统的数据汇交模块和数据计算模块提供稳定、可靠、安全、高效的基础能力支撑。面对生命科学数据汇交的复杂性和数据计算的多样性等特征，重点部署基因容器技术（Gene Container Service，GCS）。基因容器作为基因测序端到端完整解决方案，提供数据管理、分析工具平台、流程定义能力、运行流程以及查看流程执行结果的能力，同时以上能力均支持以CLI命令行方式执行。基因容器的使用方式包括可视化界面、Rest API、SDK、命令行，可以满足具备不同信息技术水平的生命科学领域研究人员使用要求。

云服务管理系统，主要包括云服务层（IaaS、PaaS、SaaS）、云服务Console、租户登录Portal门户和集中的运维系统，整体架构如图13所示。

3.4　可信共享系统概念验证

在本次肆虐全球的新冠肺炎疫情中，深圳国家基因库与全球共享流感数据倡议组织（GISAID）为加快新冠病毒基因组数据的共享，加速相关诊断、治疗、防疫技术的研究进展，开发了新冠病毒基因组分析平台（https：//db.cngb.org/virus/secure_evolution/publicdata）。在该平台中，我们初步验证了国家基因库生命科学数据可信共享系统的设计应用，完成了局部范围的验证工作，尤其是局部功能已经在今年中国和全球共享流感数据倡

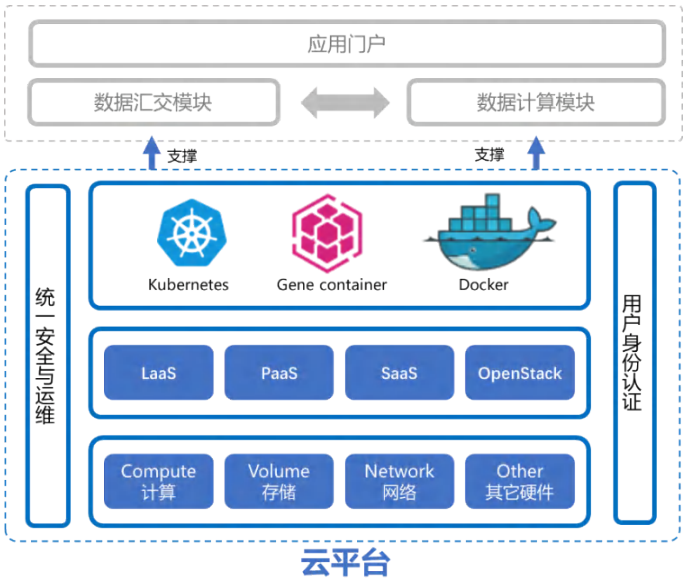


图13　云平台架构及与核心模块关系

议组织（GISAID）合作的新冠病毒基因组分析工作中得到实际验证，证明达到了可信共享的设计目标。

4　结束语

作为国家基因库生命科学数据可信共享系统研究课题，我们在以下几个方面做出了创新贡献：

1、首次完成在生命科学数据领域全系统的可信共享系统顶层设计和详细设计。

2、首次在国家基因库层面针对基因数据常用的生物分析算法，从数据的产生、传输、存储到分析各个环节实现了闭环可信计算。

3、实现了以区块链存证方式进行数据应用监管。

4、实现MPC实现跨机构、跨地域、跨主体的数据安全计算。

5、尝试采用基于Intel SGX硬件加速的HE方法。

6、建立了基于虚拟机的可信计算平台。

限于系统研究的文章篇幅，作为系统研究工作中的算法部分设计的研究成果将在其他文章中进行说明。

参考文献:

[1] 谭作文,张连福. 机器学习隐私保护研究综述[J]. 软件学报,2020,31(07):2127-2156.

[2] 周水庚,李丰,陶宇飞,肖小奎. 面向数据库应用的隐私保护研究综述[J]. 计算机学报,2009,32(05):847-861.

[3] Arellano AM, Dai W, Wang S, Jiang X, Ohno-Machado L. Privacy policy and technology in biomedical data science. Annual review of biomedical data science. 2018 Jul 20;1:115-29.

[4] Bonte C, Makri E, Ardeshirdavani A, Simm J, Moreau Y, Vercauteren F. Towards practical privacy-preserving genome-wide association study. BMC bioinformatics. 2018 Dec;19(1):1-2.

[5] Dugan T, Zou X. A Survey of Secure Multiparty Computation Protocols for Privacy Preserving Genetic Tests. In2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE) 2016 Jun 27 (pp. 173-182). IEEE.

[6] Wang S, Jiang X, Singh S, Marmor R, Bonomi L, Fox D, Dow M, Ohno-Machado L. Genome privacy: challenges, technical approaches to mitigate risk, and ethical considerations in the United States. Annals of the New York Academy of Sciences. 2017 Jan; 1387(1):73.

[7] Asharov G, Halevi S, Lindell Y, Rabin T. Privacy-preserving search of similar patients in genomic data. Proceedings on Privacy Enhancing Technologies. 2018 Oct 1;2018(4):104-24.

[8] Wang B, Song W, Lou W, Hou YT. Privacy-preserving pattern matching over encrypted genetic data in cloud computing. InIEEE INFOCOM 2017-IEEE Conference on Computer Communications 2017 May 1 (pp. 1-9). IEEE.

[9] Jiang Y, Hamer J, Wang C, Jiang X, Kim M, Song Y, Xia Y, Mohammed N, Sadat MN, Wang S. SecureLR: Secure logistic regression model via a hybrid cryptographic protocol. IEEE/ACM transactions on computational biology and bioinformatics. 2018 May 7;16(1):113-23.

[10] Chen F, Dow M, Ding S, Lu Y, Jiang X, Tang H, Wang S. PREMIX: Privacy-preserving EstiMation of individual admixture. InAMIA Annual Symposium Proceedings 2016 (Vol. 2016, p. 1747).

American Medical Informatics Association.

[11] Sadat MN, Aziz MM, Mohammed N, Chen F, Wang S, Jiang X. Safety: Secure gwas in federated environment through a hybrid solution with intel sgx and homomorphic encryption. arXiv preprint arXiv:1703. 02577. 2017 Mar 7.

[12] Lambert C, Fernandes M, Decouchant J, Esteves-Verissimo P. MaskAl: Privacy preserving masked reads alignment using intel SGX. In2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS) 2018 Oct 2 (pp. 113-122). IEEE.

[13] Stephens ZD, Lee SY, Faghri F, Campbell RH, Zhai C, Efron MJ, Iyer R, Schatz MC, Sinha S, Robinson GE. Big data: astronomical or genomical? [J]. PLoS biology. 2015 Jul 7;13(7):e1002195.

[14] Xia Q, Sifah EB, Asamoah KO, Gao J, Du X, Guizani M. MeDShare: Trust-less medical data sharing among cloud service providers via blockchain[J]. IEEE Access. 2017;5:14757-67.

[15] Fan K, Wang S, Ren Y, Li H, Yang Y. Medblock: Efficient and secure medical data sharing via blockchain [J].. Journal of medical systems. 2018;42(8):136.

[16] Azaria A, Ekblaw A, Vieira T, Lippman A, editors. Medrec: Using blockchain for medical data access and permission management[J]. 2016 2nd International Conference on Open and Big Data (OBD); 2016: IEEE.

[17] Liu J, Li X, Ye L, Zhang H, Du X, Guizani M, editors. BPDS: A blockchain based privacy-preserving data sharing for electronic medical records[J]. 2018 IEEE Global Communications Conference (GLOBECOM); 2018: IEEE.

[18] Cho H, Wu DJ, Berger B. Secure genome-wide association analysis using multiparty computation[J]. Nature biotechnology. 2018; 36 (6):547-51.

[19] Kim D, Son Y, Kim D, Kim A, Hong S, Cheon JH. Privacy-preserving Approximate GWAS computation based on Homomorphic Encryption. BMC Medical Genomics. 2020 Jul;13(7):1-2.

[20] Mott R, Fischer C, Prins P, Davies RW. Private Genomes and Public SNPs: Homomorphic encryption of genotypes and phenotypes for shared quantitative genetics. Genetics. 2020 Jun 1;215(2):359-72.

[21] Zhang Y, Blanton M, Almashaqbeh G, editors. Secure distributed genome analysis for GWAS and sequence comparison computation [J]. BMC medical informatics and decision making; 2015: BioMed Central.

[22] 钱萍,吴蒙. 同态加密隐私保护数据挖掘方法综述[J]. 计算机应用研究,2011,28(05):1614-1617+1622.

[23] 申立艳,陈小军,时金桥,胡兰兰. 隐私保护集合交集计算技术研究综述[J]. 计算机研究与发展, 2017,54(10):2153-2169.

[24] Khan A, Manzoor U, Sarwar K, Ahmed M, Tahir M, Anjum A, Alam M, Javaid N, Balubaid MA. Towards preserving privacy of outsourced genomic data over the cloud [J]. Journal of Medical Imaging and Health Informatics. 2017 Oct 1;7(6):1475-82.

[25] Boulos MN, Wilson JT, Clauson KA. Geospatial blockchain: promises, challenges, and scenarios in health and healthcare. International Journal of Health Geographics. 2018. 25.

[26] Brisimi TS, Chen R, Mela T, Olshevsky A, Paschalidis IC, Shi W. Federated learning of predictive models from federated electronic health records[J]. International journal of medical informatics. 2018 Apr 1;112:59-67.

[27] Kuo TT, Zavaleta Rojas H, Ohno-Machado L. Comparison of blockchain platforms: a systematic review and healthcare examples [J]. Journal of the American Medical Informatics Association. 2019 May;26(5):462-78.

[28] 谷婷,简小明,彭晓良,姚鹏,苏忠勇. 无可信中心可公开验证可更新的多秘密共享[J]. 计算机应用研究,2020,37(S1):293-297.

[29] 邓锐,陈左宁. 基于可信计算的群内安全信息共享混合模型[J]. 上海交通大学学报,2014,48(07):914-921.

[30] 张艳硕,李文敬,赵耿,王庆瑞,毕伟,杨涛. 基于特征值的无可信中心的秘密共享方案研究[J]. 电子与信息学报,2018,40(11):2752-2757.

[31] 于佳,陈养奎,郝蓉等. 无可信中心的可公开验证多秘密共享[J]. 计算机学报,2014,37(5):1030-1038.

[32] 王俞力,杜伟章. 向量空间上无可信中心的动态多秘密共享方案[J]. 计算机工程, 2017,43(07):163-169.

[33] Wan Z, Vorobeychik Y, Kantarcioglu M, Malin B. Controlling the signal: Practical privacy protection of genomic data sharing through Beacon services [J]. BMC medical genomics. 2017 Jul; 10 (2): 87-100.

[34] Agbo CC, Mahmoud QH, Eklund JM. Blockchain technology in healthcare: a systematic review[J]. InHealthcare 2019 Jun (Vol. 7, No. 2, p. 56). Multidisciplinary Digital Publishing Institute.

[35] Benchoufi M, Ravaud P. Blockchain technology for improving clinical research quality[J]. Trials. 2017 Dec;18(1):1-5.

[36] Grishin D, Raisaro JL, Troncoso-Pastoriza JR, Obbad K, Quinn K, Misbach M, Gollhardt J, Sa J, Fellay J, Church GM, Hubaux JP. Citizen-Centered, Auditable, and Privacy-Preserving Population Genomics[J]. bioRxiv. 2019 Jan 1:799999.

[37] Yaeger K, Martini M, Rasouli J, Costa A. Emerging blockchain technology solutions for modern healthcare infrastructure[J]. Journal of Scientific Innovation in Medicine. 2019 Jan 24;2(1).

[38] Ortega DR, Oikonomou CM, Ding HJ, Rees-Lee P, Alexandria, Jensen GJ. ETDB-Caltech: A blockchain-based distributed public database for electron tomography[J]. PloS one. 2019 Apr 15;14(4): e0215531.

[作者简介]

丁远彤（1990—），女，博士，深圳华大生命科学研究院和国家基因库副研究员，主要研究方向基因组演化分析、多组学数据挖掘，基因数据安全与隐私保护等。

谈聪，男，博士，深圳华大生命科学研究院和国家基因库副研究员，主要研究方向为生物多组学大数据的分析和挖掘、生物数据库开发与利用、群体基因组学和功能基因挖掘等。

魏晓锋，男，深圳国家基因库生物信息数据库副主任，国家基因库生命大数据平台负责人，主要研究方向魏生命科学领域大数据系统的设计和研发。

次优的通用计算电路不可区分混淆器自动化构造方法

朱率率，韩益亮，李鱼
武警工程大学密码工程学院，陕西西安，710086 网络与信息安全武警部队重点实验室，陕西西安，710086

摘　要：不可区分混淆（$i\mathcal{O}$）的构造问题是近年来一直困扰密码学研究的一个难题，现有的基于多线性映射、函数加密、全同态加密等密码学原语的$i\mathcal{O}$构造均存在不同程度的安全性问题、构造过程不易实现、电路扩展效率不高等缺陷。同时，混淆电路作为另外一个具有相似构造目标的密码分支独立发展，并在电路的混淆上得到了许多重要的结论，可以被借鉴用来设计通用的不可区分混淆的实用化电路设计。本着探索$i\mathcal{O}$实用化构造的初衷，本文从电路的自动化搜索的全新角度审视$i\mathcal{O}$的设计问题，将电路设计映射到图神经网络构造问题中，基于图神经网络的自动演化技术，探索了一种可以实现限定性满足不可区分性和功能保持性的通用$i\mathcal{O}$构造方法：AG$i\mathcal{O}$。该$i\mathcal{O}$的基本架构是基于对偶的对抗性图神经网络架构，针对任意电路功能，通过图枚举得到备用的电路样本集合，然后使用以子电路为粒度的差分演化算法分别独立优化上述对偶的图神经网络，当自动化分类模型从统计上不能有效识别不同的输出电路时，达到该网络的判别器不可区分的状态。测试结果表明，该AG$i\mathcal{O}$架构简单，易于实现，较好的满足了电路的通用性和统计上的不可区分性。

关键词：不可区分混淆、公钥密码、图神经网络、生成式对抗网络

An Automatic Construction Method for Sub-optimal Indistinguishable Obfuscation of Generic Computing Circuits

Zhu Shuaishuai, Han Yiliang, Li Yu
Engineering College of Cryptography, Engineering University of the APF, Xi'an, China, 710086

Abstract: The construction of indistinguishability obfuscation($i\mathcal{O}$) is a long-term concern confusing the researchers. The existing$i\mathcal{O}$constructions are based on primitives of multi-linear map, functional encryption, fully homomorphic encryption. These routines naturally inherent the shortages appeared in security, efficiency and generic abilities. Exploring new approaches satisfying better generic functioning and indistinguishability in $i\mathcal{O}$ construction is still an major problem to solve. In this paper, we presented a new$i\mathcal{O}$framework called AG$i\mathcal{O}$, which can automatically generate sub-optimal $i\mathcal{O}$with functional equivalence and generic circuit obfuscation. The AG$i\mathcal{O}$ achieved indistinguishability by circuit garbling which is a natural tool in constructing obfuscation. Then we designed the graph-based automatic evolvement, which can well achieve sub-optimal circuit generalization. Through our test, the AG$i\mathcal{O}$is simple to deploy and implement, while the efficiency is acceptable.

Key words: Indistinguishability Obfuscation; Public Key; Graph Neural Network; Generative Adversarial Network

1　引言

不可区分混淆器（$i\mathcal{O}$）是近年来密码设计中　　的一个重要概念，最初由实现程序代码的版权保

*本文工作由国家自然科学基金(No.61572521,U1636114)，国家重点研发项目(2017YFB0802000)，武警工程大学创新团队项目(KYTD201805)，武警工程大学基础研究项目(WJY201910)提供资助。

护机制发展而来，达到保留代码的功能而不泄露执行过程的目的。出于相似的目的，*iO*可以对密码算法的功能和执行过程加密，以保护密码算法的执行。从广义上讲，使用安全、高效的*iO*可以便捷的实现非对称加密算法、身份基加密算法和各类密钥封装算法，使数据加密不再受限于密码原语的选择和密钥的处理，如同Deffie-Hellman公钥设计思想，*iO*的有潜力引发下一次密码算法设计的全面革新。*iO*如此强大的构造能力吸引了众多密码研究人员的关注［1］。同时，经过十多年的研究，*iO*设计理论虽然呼之欲出，但在解决安全性、可靠性和实用性等关键问题上依然若隐若现，目前没有一个令人满意的结果。

*iO*在设计上两个基本要求是实现运算功能的保持和电路的不可区分。实际上，有很多其它的密码构造也可以实现这两个基本要求，例如，混淆电路［2~6］，函数加密［7-8］等密码原语。这些构造都可以实现对加密过程的保护，同时不影响程序的输出正确性，因此，在概念上与*iO*非常相似，事实上有许多工作专注于它们之间的相互构造，并取得了重要的成果［9］。另一方面，*iO*理论与其它构造的一个明显的区别是凌驾于电路、密钥和输入数据之上的通用性要求，目前这一点还无法很好的满足。

归纳当前*iO*构造理论中有待克服如下两个方面的问题。首先，*iO*在电路混淆形式化的通用构造问题困扰了研究者很长时间。根据Barak等人［10］对不可区分混淆的形式化定义，构造规模相同、功能相同，并通过严格证明不可区分的密码结构是较难实现的。Garg等人［11］首先利用多线性映射构造通用的*iO*，并将不可区分性规约到多线性映射困难问题上。Bitansky等人［12］利用函数加密构造了新的*iO*方案，Lin等［13~15］通过构造常数阶维度的多线性映射构造新型的困难问题假设，在此基础上构造了更简洁的*iO*，提高了*iO*的效率。其次，通用*iO*实现的复杂性影响了*iO*理论的进一步应用。为了实现Barak等人定义的*iO*形式化语义安全，目前*iO*的所有构造方式均直接或间接的依托现有的密码原语，从而将不可区分的困难性规约到已知的困难问题上。

本文将*iO*的不可区分性构造分成两大类：通过形式化定义并证明不可区分性建立在某个公开的密码原语上，称为理想的不可区分性，与之对应的，将不可区分性建立在外部的统计判定部件在某个代价下无法获取判定优势上，称为次优的不可区分性。本文退而求其次，寻找其它的更简洁的*iO*构造方式满足次优的不可区分性，并寻求在性能上的提升。本文探索通过限定*iO*的构造条件，实现自动化构造满足通用电路功能保持且次优不可区分特性的*iO*。这里的“次优”一方面指的是攻击敌手通过自动化搜索攻击*iO*时，在限定时间内从统计达到足够的不可区分特性，从而取代现有的通过语义安全分析得到理论上的不可区分困难性，另一方面指，在通用电路构造时，使用给定的随机选择的电路作为*iO*的输入，取代语义上任意电路作为输入。本文第2节和第3节分别介绍了*iO*的研究现状和本文用到的重要概念或理论；第4节详细介绍了不可区分混淆AG*iO*的自动化构造过程；第5节和第6节分别从理论和实现角度分析了AG*iO*的可行性、安全性等基本特性。

2 相关工作

混淆（Obfuscation）理论［10］最初是用于保护计算机程序的内部运行结构而提出的，即通常所说的代码保护，达到程序结构不泄露，而功能发挥不受影响的目的。Hada等人［16］提出了函数混淆的概念，指出对程序保护的混淆模型中，要求混淆处理后的程序相当于一个不可拟合的函数，长期以来，这种混淆无法被严格构造出来。Barak等人［10］提出了电路混淆的严格安全定义，即虚拟黑盒安全，并提出了安全上弱化的一种虚拟黑盒，即不可区分混淆器（*iO*）。由于*iO*可以很容易被改造用于构造实现众多的密码原语和密码协议，并可以大大简化构造过程。然而，构造既能够实现混淆的不可区分性，同时又能够实现通用的功能，一直以来是构造通用*iO*难以逾越的障碍，因此，构造满足条件的*iO*成为密码学中一个重要的问题。

2013年， Gentry等人［11，17］提出了多线性映射的构造（multilinear maps），该构造具有非常好的函数映射通用性，接着Garg等人［18-19］利用多线性映射构造了首个通用的$i\mathcal{O}$，并实现了对于相同的输入，其输出电路在计算上满足不可区分性。［7，20-21］等工作利用基于多线性映射构造的$i\mathcal{O}$，构造了语义上安全的单密钥公钥密码体制、多方安全计算、密钥交换协议等多个密码学应用，Sahai等人［20］证明大多数的密码单向计算应用均可以通过$i\mathcal{O}$实现新的构造。然而，随着深入的研究，针对多线性映射的安全问题暴露出来，Cheon［22］中分析了基于整数的多线性映射中采用的全新的安全假设。Chen等人［23］成功攻击了［21，24-25］等利用多线性映射构造的$i\mathcal{O}$。为了弥补多线性映射构造$i\mathcal{O}$的安全缺陷和效率问题，［19］采用弱多线性映射构造了新的$i\mathcal{O}$，避开了多线性映射中已知的缺陷，［13］使用固定编码梯度的多线性映射，［15］沿用相同的思路，证明了梯度为3的三线性映射上的计算困难性，并相应构造了$i\mathcal{O}$，最终，近期的［22，26］成功构造了基于双线性映射的$i\mathcal{O}$，使其安全性建立在成熟的DDH问题上。但$i\mathcal{O}$应该是密码学上具有安全完备性的基础原语，因此，一般认为仅仅在构造工具上的修补对实现$i\mathcal{O}$的完备性没有本质上的帮助。

近年来出现了许多基于新的密码原语或困难问题的$i\mathcal{O}$构造方法，如基于函数加密、全同态加密等。2017年，Gentry等［23］基于有限域上的矩阵分支运算构造了程序的混淆方法。2019年，Agrawal［28］构造了一种基于带噪声的线性函数加密原语的$i\mathcal{O}$，且不使用任何映射结构，并在2020年对这种构造存在的安全问题进行了攻击分析和改进［29］。2020年，Brakerski等人［8］利用全同态加密原语构造了新的$i\mathcal{O}$，使用了一种全同态加密的变体（split FHE）构造了全新的split $i\mathcal{O}$，使$i\mathcal{O}$的不可恢复和不可区分性基于LWE上的经典困难问题。

另一方面，与不可区分混淆有着天然相似性的混淆电路也独立的发展，并在多个方面有着重要的应用。混淆电路最初由Yao［2］提出用于解决多方安全计算问题，主要用电路设计的技巧来隐藏运算电路。之后，Lindell［3］，Bellare［4］等人分别将混淆电路进行了完整的形式化定义，使之成为独立的抽象密码工具，用于密码方案的分析和构造，并逐渐演变成了白盒密码理论。但从已有工作来看，高效的白盒密码对具体的密码方案实现都采用专用的电路，且仅对当前的对称密码算法有效，从而缺少通用白盒密码的可重用构造。2013年，Goldwasser等人［5］首次利用全同态加密原语构造了可重用的混淆电路，但该方案效率不高，使其仅具有理论意义。2019年，Zhang等人［1］指出从构造条件和过程来看，使用混淆电路构造不可区分混淆具有较大的探索意义，同时在解决$i\mathcal{O}$效率问题时，混淆电路也具有一定的优势，这一点也是本文有待探究的一个重要问题。

3 预备知识

在这一节，我们对本文用到的一些定义和结论做简要的介绍。

定义1.

不可区分混淆$i\mathcal{O}$［10］：

给定任意规模的电路C，对任意合法的输入x，构造电路$\tilde{C}$，对任意多项式级别的判别器D，满足$Pr\{D(\tilde{C},C)=1|\tilde{C}(x)\leftarrow i\mathcal{O}\{C(x)\},\tilde{C}(x)=C(x),|\tilde{C}|=|C|\}\leqslant negl(\lambda)$。

根据Lindell、Bellare等人的形式化定义和功能描述［3-4］，我们得出混淆电路如下简洁定义。

定义2.

混淆电路：给定任意电路C，通过电路等效替换、集成、电路加密等方式，重新构造$\tilde{C}$，对任意合法输入x，满足$\tilde{C}(x)=C(x)$。

定义3.

图神经网络GNN［30］：GNN是用图结构表示输入输出或中间连接状态而构造的一类神经网络，用于完成高维度特征的表达、学习和演化。

GNN按照训练方式的不同可以分为带监督功能的GNN和不带监督功能的GNN，前者主要以图结构作为输入，用于特征提取、分类和优化等学习任务；后者以算法规则或计算模式作为输入，用于电路设计、推理和群智演化等高度抽象的学习任务。正因为GNN具有丰富的表达和设计能力，我们尝试将通用电路混淆的过程由GNN自动完成。

定义4.

差分演化[31]：给定一个群体$G=\{x_0,x_1,\cdots,x_n\}$的初始值，以及定义在G上的某约束$F(G)$，通过演化算法$DE(\cdot)$，求解$\tilde{G}=DE\{x_0,x_1,\cdots,x_n\}$满足$F(G)$的最佳组合关系，如最小约束关系$arg\min_\theta\{F(\cdot)\leftarrow DE_\theta(G)\}$，其中$\theta$为$DE(\cdot)$的代价参数。

定义5.

图枚举算法：设μ为关于λ的一个多项式，给定深度为d，输入规模为I，总规模为N的电路C：$DAG(g_0,g_1,\cdots,g_N)$，通过搜索空间$\{0,1\}^N$，输出规模为1的电路$\tilde{C}$，满足对于任意输入x，有$C(x)=\tilde{C}(x)$，则该算法的最大步骤数为$2^I\times\sum_{i=1}^{d-1}2^i\binom{2}{2^i-1}$，即计算复杂度为$\mathcal{O}(\mu(\lambda)(\frac{d}{2}!)2^d)$。

引理1.

非完全覆盖的图枚举[32-33]：对于最大深度为d的电路C及其阶数为q的划分$\{g_0,g_1,\cdots,g_n\}$，存在关于安全参数λ的多项式$\mu(\lambda)$，使得在C上的图枚举算法计算复杂度为$\mathcal{O}(\mu(\lambda)\frac{N}{q}2^{3q-2})$。

从电路的拓扑构造上，阶数越高，等效电路搜索的复杂度越高；另一方面，从电路的输入规模上，规模越大，等效电路存在的可能性越小。因此，在调用图枚举算法时，尽可能选择阶数小的拓扑划分，实际上，在第4节中我们一般令$q\leqslant 5$，而且有效的等效电路多集中在$q\leqslant 4$。所以在非完全覆盖的图枚举场景中，引理1具有很好的适用性。

定理1.

等效电路演化结论：对于输入电路C和任意子电路划分方法，基于图枚举算法的差分演化输出满足门类型及其连接关系的统计不可区分性。具体来说，对于多项式时间电路攻击算法$\mathcal{A}$：

（1）对于任意一个规模为n的子电路$g\in C$及其任意两个不同的演化输出$\tilde{C}_0,\tilde{C}_1$，在多项式时间内，有$Pr\{D(\tilde{C}_0,\tilde{C}_1)=1\}\leqslant negl(n)$；

（2）给定安全参数λ和在多项式时间内获得的有限个演化结果$\{\tilde{C}_0,\tilde{C}_1,\cdots,\tilde{C}_N\}$，对攻击算法$\mathcal{A}$有$Pr\{C\leftarrow\mathcal{A}(1^\lambda,\tilde{C}_0,\tilde{C}_1,\cdots,\tilde{C}_N)\}\leqslant negl(\lambda)$。

证明：见附录。

4 不可区分混淆AG$i\mathcal{O}$的自动化构造

4.1 系统架构

为了构造满足$i\mathcal{O}$必备特性且易于实现的结构，我们借助具有自动化搜索和生成能力的对抗式演化结构。该架构包括三个部分：带抽样功能的图枚举算法、对偶的两个图神经网络演化模型和全局的电路判别器。带抽样功能的图枚举算法中，对于给定的功能F，批量产生参与演化的电路样本，从中再次随机抽样得到C_0和C_1，作为G_0和G_1的输入；在输出编码和判别优势的监督下，对偶的图神经网络演化模型采用差分演化算法更新模型G_0和G_1；判别器D将G_0和G_1的图输出作为待判定的输入，经过基于图神经网络的学习得到当前输入的判别优势Adv，由Adv作为该架构分支或终止的条件。

如图1所示，该架构以任意功能的电路C（x）作为输入，使用对输出电路Adv的监督作为G_0和G_1演化的全局策略，并使用以随机选择x的电路编码输出作为功能监督，以最大置信保持了G_0和G_1中间电路等效的结构演化。G_0和G_1独立决定了各自演化方向，并使D达到对二者输出的可忽略Adv，使局部电路呈现统计上的最大不可区分性。同时，为了优化全局的执行效率，根据输入电路C（x）的规模，架构中设置了判定器的静态淘汰阈值，用于决定演化初始电路C_0和C_1是否具备快速达到预定的不可区分程度。

不可区分混淆AG$i\mathcal{O}$架构可以表示为：

$$AGi\mathcal{O}\leftarrow\{(G_0,G_1)\leftarrow DE[(C_0,C_1)\leftarrow GE(C)]|arg\min_{\lambda,\eta}\{Adv\}\leftarrow D(G_0,G_1)\},$$

其中，$DE(\cdot)$为差分演化部件，$GE(\cdot)$为图枚举部件，$D(\cdot)$为图判定器，G_0和G_1为输入电路C的混淆。若$Adv\leqslant negl(\lambda)$，则上述模型的输出满足不可区分性。

4.2 功能电路枚举

电路的阶数q定义为一个子电路g中包含与门和或门的个数。阶数为q的电路g，其等效电路\tilde{g}定义为，满足$|g|=\tilde{g}$，且对于任意输入x，有$g(x)=\tilde{g}(x)$，即从输入输出编码的角度讲，二者有相同功能表达。对于输入编码长度为n的电路，这个表达的步骤需要搜索功能相同的电路$\{C_0,C_1,\cdots\}$作为备用的潜在输出电路，但搜索规

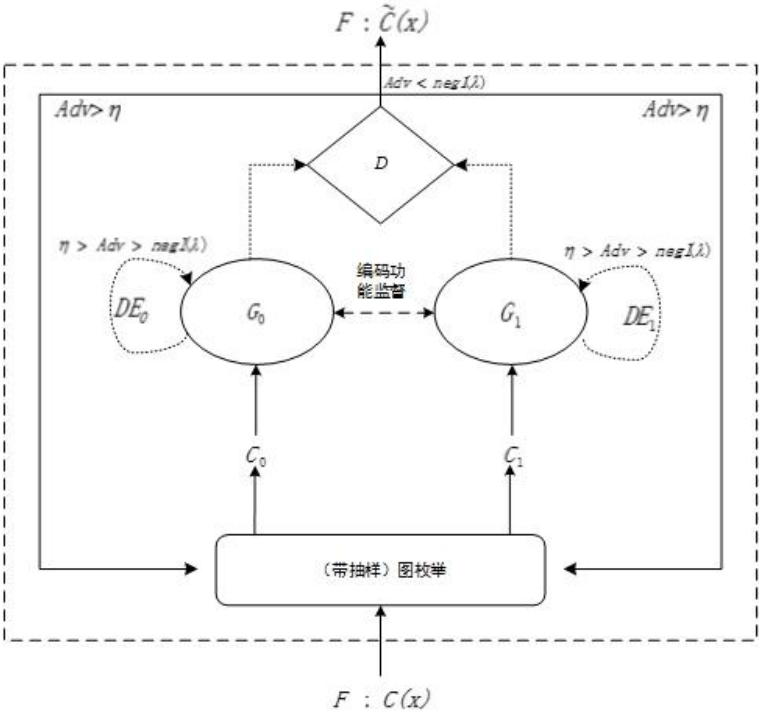


图1.不可区分混淆AG$i\mathcal{O}$架构

模是关于n的指数级规模。因此，设计多项式级的功能电路枚举是设计$i\mathcal{O}$的基本步骤，其本质问题是给定任意电路规模（例如，C中包含N个门，连接结构不限定，电路功能是实现一个函数$F:y=C(x)$，通过抽样枚举可能的子电路构造空间，满足在所在阶数上子电路的功能等效，从而得到集合$S_C=\{C_0,C_1,\cdots\}$，对任意x，满足$C_i(x)=C(x)$，且$|C_i|=|C|$。

算法1描述了从给定的任意电路C出发，通过枚举局部的子电路$\tilde{g}$，产生具有相同电路功能和电路规模的部分混淆电路集合S_C，用于产生统计上不可区分的目标电路$\tilde{C}$。该算法的基本思想是，从带输入节点的子电路g开始，逐一枚举真值表相同的等效子电路$\tilde{g}$。根据q的值，由简单到复杂，产生满足保持电路功能的潜在混淆电路。实际电路C的规模远远超出了多项式时间内可枚举的范围，需要采用局部抽样的方法进行枚举。算法1中采用了均匀的随机对子电路编号进行抽样的方式简化算法复杂度。针对规模较大的输入电路，应按照实际的电路结构分布进行抽样。一种方法是，按照电路的阶数q，统计门电路的类型，并估计门电路跳数的分布函数；然后，带输入节点的子电路g依照该分布的位置选取，采用局部等效替换的方法枚举有效的备选电路。因为抽样方法与AG$i\mathcal{O}$独立，故而子电路抽样的位置和抽样参数可以作为AG$i\mathcal{O}$的密钥信息，但为了简化AG$i\mathcal{O}$架构，本文未讨论带密钥参与的AG$i\mathcal{O}$对输出结果的性能影响。

4.3 基于图神经网络的电路差分演化

G_0和G_1的图演化是产生不可区分混淆的驱动部分，基本方法是按照子电路的阶数逐层对输入的节点类型和连接关系进行局部最大化差分演化，达到对C_0和C_1不可区分重构的目的。演化过程使用判定器D的优势Adv作为图神经网络全局的半监督，并使用$G_0(x)=G_1(x)$作为演化的中断性监督，其中x随机选自子电路真值表的合法输入。中断性监督仅是一个必要的校验步骤，并不能保证结果满足演化的功能等价性。该演化过程分为如下三个基本步骤：

将C_0和C_1按照阶数分别划分为两个子电路的列表$L_0=\{c_{00},c_{01},\cdots,c_{0u}\}$和$L_1=\{c_{10},c_{11},\cdots,c_{1u}\}$，其中$u=\lfloor |C|/q \rfloor$，在每一次$G_0$和$G_1$的演化中，$L_0,L_1$重新随机划分；

（2）对于$i,j\in\{0,1,\cdots,|C|\}$，以子电路为单位输入演化算子$\nabla(c_{0i},c_{1j})=Pr\cdot SW(c_{0i})+(1-Pr)\cdot SW(c_{1j})$，其中，$SW(\cdot)$为图中等效电路所代

表的节点替换，$Pr=\{[c_{0j}(0)-c_{0j}(0)]+[c_{0i}(1)-c_{0j}(1)]\}/(2q)$，即当前等效电路中相同位置门电路结构重复的概率；

（3）在Adv监督下对G_0和G_1进行演化算子的迭代，并将每一代的结果送入判定器D，直到D达到满足要求的判定优势$Adv \leqslant negl(\lambda)$。

附录中的算法2详细实现了上述步骤。

4.4 判别器的设计

判别器D是实现对G_0和G_1演化有效监督的另一个关键部件，它由一个图自动二分类的分类器CL组成，对输入的一个电路对$\{G_b, C\}$，$b\in\{0,1\}$，能够尽最大准确度实现对b的判定。对于电路结构，该自动分类器的首选结构是使用卷积图神经网络分类器GCN，训练的样本为矩阵化的$\{G_0, C\}$和$\{G_1, C\}$，并使用二元交叉熵（BCE）损失函数进行训练监督。在AG$i\mathcal{O}$运行的初期，二分类的分类器，如SVM、逻辑回归、随机森林等经典工具均可满足，但随着子电路阶数的增加，深度GCN结构更具有准确性优势。本文选择用全连接的图卷积网络，每一层的输出特征用如下算子表示：

$$f_i^l=\sigma\left(\sum_{j\in n_i} f_j^{l-1} w_{T_j}^{l-1}\right),$$

其中，w为特征权重，T_j为第j个节点的门类型，根据数据规模，也可在层输入特征值上增加归一化因子和偏置，以提高计算效率和准确性。

判别器D的目的是在统计上近可能的得到高精度的准确率，因此，在实现环节中可以设置m种自动判别分类器，在演化迭代中采用并行判定的方式，取各个算法的最高准确率作为最终判别结果$\tilde{b}$的计算根据，即$Accuracy_{D,\tilde{b}}=\max_{accuracy}(CL_0, CL_1, \cdots, CL_m)$。基于智能算法的二分类工具的构造技术比较成熟，此处不再赘述。

在AG$i\mathcal{O}$启动时，判别器延后于G_0和G_1启动，并以0.5的概率随机读取G_0和G_1的拓扑，并输出判定结果$\tilde{b}$，然后计算该判别器累积的判别准确度为$Accuracy_{D,\tilde{b}}$，则$Adv=Accuracy_{D,\tilde{b}}-0.5$代表了判定$G_0$和$G_1$输出的优势。

5 AG$i\mathcal{O}$的构造分析

本节分别从AG$i\mathcal{O}$功能的通用性和构造的不可区分性两个方面分析其性能。AG$i\mathcal{O}$的通用性体现在，图枚举部件、演化部件和对抗性判定器对于任意输入电路C的功能、拓扑、规模不做任何限定，中间处理过程仅与当前的枚举结果和判定器反馈的优势有关。AG$i\mathcal{O}$的不可区分性通过多层迭代的拓扑差分演化保证。具体分两个方面，一方面从多层阶数子电路替换了原有的子电路，隐藏了输入电路的拓扑结构；另一方面，对每一层的随机抽样决定了等效电路替换操作的不可逆。

5.1 不可区分特性分析

从定性上讲，AG$i\mathcal{O}$的不可区分性，一方面建立在直接在输入C上的子电路等效替换上，使用低阶数替换后的结果$\{C_0, C_1, \cdots, C_s\}$作为执行进一步通用演化的样本；另一方面，建立在对不同样本的演化上，更进一步的自动混淆不同样本间的电路拓扑特征。两个方面的区别在于，前者会尽可能覆盖以固定q划分的所有子电路，效率较高，而后者在输入的C_0和C_1上，按照q对$\{g_{i0}, g_{i1}, \cdots, g_{im}\}$进行逐层抽样，在反馈的Adv监督下迭代演化出新的拓扑，满足对高阶子电路的覆盖。当通过判别器无法对GE_0和GE_1输出的电路特征产生明显的判别优势时，即达到了次优的不可区分性。

从定量上，AG$i\mathcal{O}$的不可区分性可以从搜索$\tilde{C}_b$与$C_{\tilde{b}}$匹配的特征来实现，我们用定理2来计算该过程取得成功的概率。

定理2.

对任意选择的输入规模为$\mu(\lambda)$的电路C和输入变量x，AG$i\mathcal{O}$满足$Pr\{D(\tilde{C}_0 \leftarrow AGi\mathcal{O}(C), \tilde{C}_1 \leftarrow AGi\mathcal{O}(C))=1\} \leqslant negl(\lambda)$。

证明：见附录。

5.2 通用性分析

AG$i\mathcal{O}$的通用性意味着两层含义：首先是，所设计的架构具有可重用性，而不依赖于具体的输入电路和控制密钥，即对于任意不同的输入C_i, C_j，得到的$\tilde{C}_i \leftarrow AGi\mathcal{O}(C_i), \tilde{C}_j \leftarrow AGi\mathcal{O}(C_j)$在拓扑关系上相互独立，且能保持各自的功能特性和相同的不可区分性强度；其次是，密钥的使用与AG$i\mathcal{O}$架构的松耦合，这个方面混淆电路和不可区分混淆追求一致的目标，即算法的白盒执行和密钥的保护同步进行。由于在GNN中，在Adv的半监督演化暂时无法进行准确的定量描述，仅仅通过Adv不能完全体现不同的演化路径在训练结果上的差

异性，因此，为了简化AG$i\mathcal{O}$的架构描述，本文工作没有专门讨论密钥选择对演化的影响，而使用了随机选取的子电路划分和抽样方式。

AG$i\mathcal{O}$的可重用性及不可逆性由定理3来定量描述。

定理3.

对任意输入规模为$\mu(\lambda)$的电路C和输入变量x，AG$i\mathcal{O}$满足$Pr\{\tilde{C}_0(x)=\tilde{C}_1(x)=C(x)|Pr\{D(\tilde{C}_0\leftarrow AGi\mathcal{O}(C),\tilde{C}_1\leftarrow AGi\mathcal{O}(C))=1\}\leqslant negl_0(\lambda)\}\leqslant 1-negl_1(\lambda)$。

证明：见附录。

6 AG$i\mathcal{O}$性能测试

本节根据第4节的基本框架和主要算法实现了一个原型本版的AG$i\mathcal{O}$，初步摸索了AG$i\mathcal{O}$训练中所需要的运行环境和参数设置，同时结合AG$i\mathcal{O}$的设计初衷，对得到的部分结果做了性能、效率等方面的分析。

6.1 环境设置

原型本版的AG$i\mathcal{O}$使用Pytorch在Python3 64bit环境下编写，在E4110@2.1GHz 16core CPU、32GBytes内存和2*1080GTX加速下运行。原型AG$i\mathcal{O}$目前仅支持单比特输出的电路，且由于计算复杂度原因，*GE*的图演化中要求$q\leqslant 5$。在判定器的实现上，我们参考了SDNE网络嵌入的方法[34]对输入的图样本进行特征提取。AGiO实例中所需要的基本参数见表1和表2。

表1 *GE*参数设置

参数名称	值	注释
$\|S\|$	1000	样本容量
N	10~1300	输入电路的规模
q	{2,3,4,5}	子电路的阶数
η	0.25	*GE*重新抽样的优势

表2 D参数设置

参数名称	值	注释
$\|S\|$	1000	图神经网络分类器
N	10~1300	层内特征激活函数
q	{2,3,4,5}	可供训练的样本量
η	0.25	*D*的不可区分精度(终止条件)

6.2 结果分析

现有的$i\mathcal{O}$构造方案绝大多数停留在理论阶段的原因是计算效率低，系统开销大，如[35]中小于100个门电路的$i\mathcal{O}$执行时间在小时级别。本文围绕构造次优$i\mathcal{O}$的目的就是在保证基本特性的前提下，通过折中不可区分特性的实现方法和衡量方法，大幅提高$i\mathcal{O}$执行的效率。图2和图3分别显示了AG$i\mathcal{O}$在输入电路规模小于1300时的存储开销和时间开销，其中，100个门电路的输入，处理时间在600秒左右，运行内存开销约为6Gbytes，在执行效率上有了实质性的提高。由于现有的$i\mathcal{O}$构造方案的执行效率远远低于AG$i\mathcal{O}$，因此我们没有与其它方案做性能的对比，只对*GE*部件是否对子电路进行抽样做了对比测试，从中看出，经过抽样的算法实例随着门电路规模增长呈现远低于指数级的开销增加，而没有经过抽样的算法实例则呈现出指数级开销增长。对不同阶数的子电路抽样之后再进行演化是AG$i\mathcal{O}$实现效率提升的关键。

AG$i\mathcal{O}$的不可区分特性依赖于判定器D的判定优势*Adv*，当判定优势*Adv*依λ可忽略时，我们认为达到了次优的不可区分性。如图4所示，当迭代次数大于20时，图样本的特征迅速减少，*Adv*随着*GE*迭代次数的增加而迅速降低。

图2 AG$i\mathcal{O}$空间开销

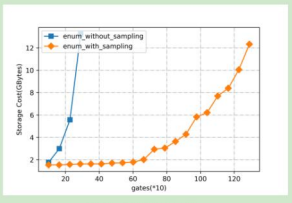


图2 AG$i\mathcal{O}$空间开销

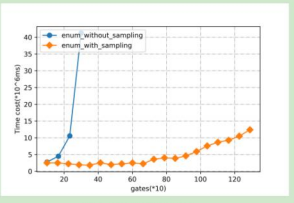


图3 AG$i\mathcal{O}$时间开销

另外，为了直观的体现输入电路*C*的特征，我们分析了*GE*子电路的抽样中，采用不同的q时等效子电路的覆盖情况，如图5所示。显然当q越小时，在计算上越容易进行等效替换，反之，则需要更多轮的迭代尝试；另一方面，当q越小时，进行的等效电路替换仍然能够泄露电路局部的特征，例如，q=2时，自动搜索到的等效电路一般是两个门电路的对偶置换，并不能隐藏对电路局部的统计特征。这种情况下，就需要对*GE*中的电路进行再次抽样和更高阶数的演化。另一方面，局部的等效电路演化并不能完全保证电路规模的恒定，因此，需要设定额外的规模浮动变量$\Delta(G_i)=||G_i|-$

$|C_i||$，当一轮迭代结束时，满足$\Delta(G_i)=0$。我们用n/N作为门电路覆盖情况的衡量参数，其中n为等效替换操作的次数，N为输入规模，当该值逼近1时，意味着几乎所有的计算门都参与了电路演化。针对不同的覆盖情况，在AG$i\mathcal{O}$的训练中适当的对q进行选择，可以进一步提高生成$\tilde{C}$的效率。

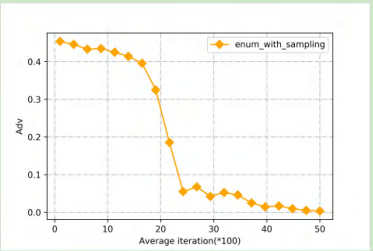


图4 D在一次演化中的判定优势

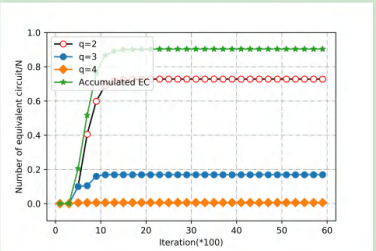


图5 AG$i\mathcal{O}$演化中等效电路的覆盖情况

7 结论

现有的基于多线性映射、函数加密、全同态加密等密码学原语的$i\mathcal{O}$构造均存在不同程度的安全性问题、构造过程不易实现、电路扩展效率不高等缺陷。因此，本文围绕$i\mathcal{O}$的构造问题，尝试了一系列新型的构造思想和技术方法，用于克服现有$i\mathcal{O}$构造中存在的方法和性能的瓶颈。AG$i\mathcal{O}$架构中，采用等效电路自动化搜索的全新方法解决通用$i\mathcal{O}$构造问题，即将通用电路生成映射到图神经网络构造问题中，基于图神经网络的自动演化技术，探索了一种满足统计上次优的不可区分性和功能保持性的通用$i\mathcal{O}$构造方法。该$i\mathcal{O}$的基本架构是基于对偶的对抗性图神经网络架构，对于任意功能的输入电路C，通过图枚举得到备用的替代电路样本集合，然后使用差分演化算法分别独立优化上述图神经网络，并使用一个全局的判别优势监督图神经网络演化的方向，最终达到判别器对该图神经网络不可区分的状态。

AG$i\mathcal{O}$的构造中依然存在诸多不确定的因素，会导致潜在的安全问题。图枚举部件、演化部件和对抗性判定器对于任意输入电路C的功能、拓扑、规模不做任何限定，中间处理结果仅与当前的枚举结果和判定器反馈的优势有关，因此可以达到通用电路处理的目的，但另一方面，中间处理过程与输入电路规模有直接关系，因此，AG$i\mathcal{O}$会泄露出计算执行的资源耗费信息。另外，AG$i\mathcal{O}$中对子电路的抽样方式直接决定了演化路径，从而对整个结构的安全性有重要影响，但本文尚未发现可以准确描述这种制约关系的方法。综上，AG$i\mathcal{O}$的原型架构还存在诸多改进之处以及结构优化的余地。

参考文献:

[1] ZHANG F G ZHANG Z. Garbled circuits and indistinguishability obfuscation. Journal of Cryptologic Research，6(5)：541，2019.

[2] Andrew Chi-Chih Yao. How to generate and exchange secrets. In Proceedings of the 27th Annual Symposium on Foundations of Computer Science，SFCS '86，page 162-167，USA，1986. IEEE Computer Society.

[3] Yehuda Lindell and Benny Pinkas. A proof of security of yao's protocol for two-party computation. J. Cryptol.，22（2）：161-188, April 2009.

[4] BellareMihir, Viet Tung Hoang, and RogawayPhillip. Foundations of garbled circuits. In Proceedings of the 2012 ACM Conference on Computer and Communications Security，CCS '12，page 784-796，New York，NY，USA，2012. Association for Computing Machinery.

[5] GoldwasserShafi, KalaiYael, PopaRaluca, VaikuntanathanVinod, and ZeldovichNickolai. Reusable garbled circuits and succinct functional encryption. pages 555 - 564，06 2013.

[6] Yongge Wang, MalluhiQutaibah, and KhanKhaled. Garbled computation in cloud. Future Generation Computer Systems，62，12 2015.

[7] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. Algorithmica，2016.

[8] BrakerskiZvika, D? ttlingNico, GargSanjam, and MalavoltaGiulio. Candidate iO from Homomorphic Encryption Schemes，pages 79 - 109. 05 2020.

[9] Liu XM Xu HL Zhao QS, Zeng QK. Verifiable computation using re-randomizable garbled circuits. Journal of Software，30（2）：399，2019.

[10] BarakBoaz, GoldreichOded, ImpagliazzoRussell, RudichSteven, SahaiAmit, VadhanSalil, and Ke Yang. On the (im)possibility of obfuscating programs. IACR Cryptology ePrint Archive，2001：69，01 2001.

[11] GargSanjam, GentryCraig, and HaleviShai. Candidate multilinear

maps from ideal lattices. 2013.

[12] Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. Journal of the ACM，65：1－37, 11 2018.

[13] Huijia Lin. Indistinguishability obfuscation from constant-degree graded encoding schemes. pages 28－57，05 2016.

[14] Huijia Lin and Stefano Tessaro. Indistinguishability obfuscation from trilinear maps and block-wise local prgs. pages 630－660，07 2017.

[15] Huijia Lin and Stefano Tessaro. Indistinguishability obfuscation from trilinear maps and block-wise local prgs. In Jonathan Katz and Hovav Shacham，editors, Advances in Cryptology － CRYPTO 2017, pages 630－660，Cham，2017. Springer International Publishing.

[16] HadaSatoshi. Zero-knowledge and code obfuscation. ASIACRYPT 2000. LNCS，1976：443－457, 01 2000.

[17] GentryCraig, GorbunovSergey, and HaleviShai. Graph-induced multilinear maps from lattices. Volume 9015，pages 498－527，03 2015.

[18] GargSanjam, GentryCraig, HaleviShai, RaykovaMariana, SahaiAmit, and WatersBrent. Candidate indistinguishability obfuscation and functional encryption for all circuits. SIAM Journal on Computing，45：882－929, 01 2016.

[19] GargSanjam, MilesEric, MukherjeePratyay, SahaiAmit, SrinivasanAkshayaram, and ZhandryMark. Secure obfuscation in a weak multilinear map model. In Proceedings，Part II，of the 14th International Conference on Theory of Cryptography - Volume 9986，page 241-268，Berlin，2016.

[20] Amit Sahai B. Watersy. How to use indistinguishability obfuscation：Deniable encryption，and more. Proceedings of the Annual ACM Symposium on Theory of Computing，pages 475－484，05 2014.

[21] GargSanjam, GentryCraig, HaleviShai, and RaykovaMariana. Two-round secure mpc from indistinguishability obfuscation. 02 2014.

[22] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehl´e. Cryptanalysis of the multilinear map over the integers. In Elisabeth Oswald and Marc Fischlin，editors, Advances in Cryptology － EUROCRYPT 2015, pages 3 － 12，Berlin, Heidelberg，2015.

[23] Yilei Chen, GentryCraig, and HaleviShai. Cryptanalyses of candidate branching program obfuscators. In Jean-S´ebastien Coron and Jesper Buus Nielsen，editors，Advances in Cryptology － EUROCRYPT 2017，pages 278－307，Cham，2017.

[24] Zvika Brakerski and Guy N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. In Yehuda Lindell，editor, Theory of Cryptography, pages 1 － 25，Berlin, Heidelberg，2014. Springer Berlin Heidelberg.

[25] Boaz Barak, Sanjam Garg, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Protecting obfuscation against algebraic attacks. In Phong Q. Nguyen and Elisabeth Oswald，editors, Advances in Cryptology － EUROCRYPT 2014, pages 221 － 238，Berlin, Heidelberg，2014.

[26] AnanthPrabhanjan, JainAayush, Huijia Lin, MattChristian, and SahaiAmit. Indistinguishability obfuscation without multilinear maps: New paradigms via low degree weak pseudorandomness and security amplification. Advances in Cryptology CRYPTO 2019，pages 284－332，Cham，2019.

[27] DezJorge, Juan del Coz, LuacesOscar, and BahamondeAntonio. Using tensor products to detect unconditional label dependence in multilabel classifications. Information Sciences，329，09 2015.

[28] AgrawalShweta. Indistinguishability obfuscation without multilinear maps: New methods for bootstrapping and instantiation. Advances in Cryptology EUROCRYPT 2019，pages 191－225，Cham，2019.

[29] Shweta Agrawal and Alice Pellet-Mary. Indistinguishability Obfuscation Without Maps: Attacks and Fixes for Noisy Linear FE，pages 110－140. 05 2020.

[30] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems，page 1C21，2020.

[31] Rainer Storn and Kenneth Price. Differential evolution c a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization，11(4)：341－359, 1997.

[32] Wang JS. A general algorithm for enumerating some subgraphs of a simple graph. Journal of Computers,(1):39－45，1986.

[33] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining，KDD '16，pages 1225－1234，New York，NY，USA，2016. ACM.

[34] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Deep neural networks for learning graph representations. In Proceedings of the Thirtieth AAAI Conference on Arti fi cial Intelligence，AAAI'16，page 1145C1152. AAAI Press，2016.

[35] HaleviShai, HaleviTzipora, ShoupVictor, and Stephens-DavidowitzNoah. Implementing bp-obfuscation using graph-induced encoding. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security，CCS '17，page 783C798，New York，NY，USA，2017. Association for Computing Machinery.

基于多级SVM的功耗分析研究

马鹏[1]，刘祥[2]，钟卫东[1]，夏璇[1]
[1]武警部队网络与信息安全保密重点实验室，西安 710086；
[2]武警工程大学研究生大队，西安 710086

摘　要：本文在充分研究和分析当前SVM在功耗分析应用的基础上，设计了一种多级SVM的功耗分析方法。该方法主要是针对当前随着硬件技术的进行以及防护技术的提升，传统的功耗分析要想达到恢复密钥的目的往往需要采集大量的功耗数据，并且需要消耗大量的时间的问题而展开。多级SVM将以汉明重量和密钥为标签进行分类的方式进行结合，首先以汉明重量为标签使用SVM对功耗进行分类，达到缩小密钥范围的目的；然后再以密钥为标签使用SVM对功耗进行二次分类，从而达到恢复其密钥的目的。通过实验验证发现，本文提出的多级SVM在应用功耗攻击上，采集SM4加密算法的功耗进行对比实验发现，与标准SVM相比其密钥恢复性能提高了30%，与CPA相比其密钥恢复性能提高了约68%。

关键词：多级SVM、汉明重量、密钥恢复、标签、分类

Research on Power Analysis based on Multistage SVM

MA Peng[1], LIU Xiang[2], ZHONG Weidong[1], XIAN Xuan[1]
1.Key Laboratory of Network and Information Security,Chinese People's Armed Police,Xi'an shaanxi 710086,China;
2.Graduate Team,Engineering University of Chinese People's Armed Police Force,Xi'an shaanxi 710086,China

Abstract: In this paper, a multi-level SVM power analysis method is designed on the basis of fully studying and analyzing the application of current SVM in power analysis. With the development of hardware technology and the improvement of protection technology, the traditional power analysis needs to collect a lot of power consumption data and consume a lot of time to recover the key. Multi level SVM combines Hamming weight and key as labels. Firstly, it classifies power consumption by SVM with Hamming weight as label, so as to reduce the key range; secondly, it uses SVM to classify power consumption with key as label, so as to recover the key. The experimental results show that the key recovery performance of the proposed multilevel SVM is improved by 30% compared with the standard SVM, and the key recovery performance is improved by 68% compared with CPA.

Key words: multistage SVM; hamming weight; key recovery; label; classification

1　引言

智能卡、手机和RFID标签等嵌入式设备在我们的生活中得到了广泛的应用。这些设备使用加密算法来实现加密和解密操作，以保护数据的安全。然而，在密码算法的执行期间，设备处理的秘密信息将根据与数据和操作相关的功耗[1]、时间[2]和电磁[3]泄露出去。功耗攻击就是利用密码设备在加密过程中泄漏的功耗进行分析与攻击的方法，其中E Brier等人[4]在2004年提出的相关功耗攻击（Correlation Power Analysis， CPA）因较强攻击性和破解密钥的高效性被人们广泛采用。

近年来，侧信道攻击在机器学习技术的基础上探索了一个新的发展方向[5]。机器学习可以从经验中自动学习，通常用于解决分类和回归问题。机器学习技术一般由训练和测试两个环节组成，通常被分为监督学习和无监督学习。对于有监督学习，该算法提供了由输入数据（特征）和相关结果度量（标签）组成的训练示例。许多研究人员已经进行了基于监督学习的侧信道研究，结果

基金项目：大学基础研究基金项目（No. WJY201914）; 军事类研究生资助项目（No. JY2019B167）。

表明，该方法可以显著提高侧信道攻击效率[6][7]。无监督学习可以处理没有结果标签的任务，也可以在侧信道攻击中采用[8][9]。

G.Hospodar等人首次将SVM应用于恢复AES加密模块的密钥中，通过实验发现SVM恢复密钥的性能高于模块攻击的性能[10]。胡晓阳等人结合SVM对RSA算法中平方、乘法操作进行识别，根据密钥与操作的相关性，推断出RSA二进制密钥序列[11]。邓高明等人，提出了一种基于主成分分析（PCA）技术和多分类支持向量机（SVM）的模板分析密码旁路攻击方法，实验表明同等条件下SVM的分类效果好于大多数文献上使用的Bayes判别的分类效果[12]。上述文献中，将SVM用于功耗攻击中的分类中从而达到破解密钥的目的。然而，文献［11］中选择用汉明重量来作为分类标签，但是这样只能缩小正确密钥范围，若想得到正确密钥还需要根据加密算法本身进一步的来推算；文献［12］选择用所有猜测密钥作为分类标签，但是这样一来就大大增加了其计算复杂度。基于此本文提出了一种多级SVM分类方法用于功耗攻击中。该方法将以汉明重量和密钥为标签进行分类的方式进行结合，首先以汉明重量为标签使用SVM对功耗进行分类，达到缩小密钥范围的目的；然后再以密钥为标签使用SVM对功耗进行二次分类，从而达到恢复其密钥的目的。

2 基础知识

2.1 SM4算法

SM4算法[13]是我国的分组密码标准，由于其具有加解密效率高、一定程度的安全性以及硬件实现简单的特点被我国商用密码标准采用，广泛应用于金融、通信加密等邻域。该算法利用32轮的Feistel非线性迭代结构对轮密钥扩展部分进行设计，以此增强密码的安全性。算法的32轮的迭代由一个轮函数来进行控制，轮函数则通过非线性变换τ和线性变换L构造而成。SM4算法解密过程与加密过程类似，只需将加密过程中的轮密钥顺序反转即可得到用于解密的轮密钥。

设输入SM4算法中的明文为$(X_0, X_1, X_2, X_3) \in (Z_2^{32})^4$，输出的密文为$(Y_0, Y_1, Y_2, Y_3) \in (Z_2^{32})^4$，第$i$轮使用的轮密钥$rk_i \in Z_2^{32}$，其中$i$=0，1，2，…，31。其加密运算如下：

$$\begin{aligned} X_{i+4} &= F\left(X_i, X_{i+1}, X_{i+2}, X_{i+3}, rk_i\right) \\ &= X_i \oplus T\left(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus rk_i\right) \end{aligned} \tag{1}$$

密文输出：

$$\left(Y_0, Y_1, Y_2, Y_3\right) = \left(X_{35}, X_{34}, X_{33}, X_{32}\right) \tag{2}$$

2.2 功耗泄漏模型

侧信道攻击技术避开直接分析密码算法复杂的数学结构，通过利用密码芯片在运行密码算法时所泄露的时间、电磁、功耗、声音等物理信息来进行密钥恢复。其中，功耗分析攻击以其攻击成功率高、消耗资源少的特点成为侧信道攻击分析领域研究范围最广、最具代表性的攻击理论。功耗分析攻击通过密码设备、密码芯片在运行过程中产生的功耗信息与密码算法的中间值形成对应关系，并根据这种关系来恢复中间数据的相关信息，进行密钥恢复。

根据功耗产生机理[14]，密码设备产生的动态功耗与数据转换是息息相关的，而功耗泄露模型就是对密码设备的操作数据与功率消耗仿真值之间对应关系的一种刻画。密码设备电路的功耗信息主要包括两个部分：静态功耗和动态功耗。静态功耗是指密码设备在非工作状态下的功率消耗，动态功耗是指在工作状态下输入输出发生改变而产生的功率消耗。其中静态功耗往往浮动较低，对功耗信息影响较小，因此动态功耗成为引起功率变化的主要因素。泄露模型通常根据功耗信息与信号改变之间关系，得出功耗信息与中间数据之间的相关性来进行构建。在功耗泄露模型中，最主要的是汉明重量模型（Hamming Weight Model，HWM）和汉明距离模型（Hamming Distance Model，HDM）。

（1） 汉明重量模型

汉明重量模型是由Messerges等人于2000年提出，他们基于汉明重量模型和绝对差组合函数的二阶DPA思想，对密码芯片中的密码算法进行攻击，成功恢复出所需密钥。汉明重量模型是一种实用性高、所需条件少的泄露模型，不仅可以应用于动态功耗分析攻击也可应用于静态功耗分析攻击。当攻击者对密码设备了解较少或者无法获知连续处理的数据时，攻击者可以使用汉明重量模型对密码设备进行攻击。

在实际电路中，数据的汉明重量与密码设备处理该数据所引起的功率变化存在一定程度上的关联性。攻击者在利用汉明重量模型对密码设备进行攻击时，由于处理汉明重量高的数据所引起的功率变化往往比处理汉明重量低的数据高得多，因此可以假设处理数据中1的个数与密码设备运行时产生的功耗消耗成正比。根据上述假设，建立相应的功耗攻击模型，进行密钥恢复工作。本文中汉明重量为一个二进制字符串中非零字符的个数。表1显示了汉明重量的计算方式。

表1　汉明重量计算

字符	字符串	汉明重量(HW)
0,1	1011011	5
0,1	0000000	0
0,1	1011111111	9
0,1	0011	2

（2） 汉明距离模型

汉明距离模型是在2004年由Brier等人[4]提出，其根据门电路变换的总次数来对功率消耗进行刻画，主要应用于动态功耗分析攻击。汉明距离模型的建立主要基于两个假设：一是相同门电路进行转换时，其引起的功率消耗视为无消耗，即门电路0转换到门电路0与门电路1转换到门电路1所消耗的功率忽略不计；二是非相同门电路进行转换时，其引起的功率消耗具有一致性，即门电路0转换到门电路1与门电路1转换到门电路0所消耗的功率是相同的。根据上述假设，攻击者通过汉明距离模型能够得出门电路变换产生的功耗与汉明距离成线性正相关，并对密码设备运行时产生的功率消耗进行快速计算。

汉明距离模型构建的数学表达式为：

$$HD\left(PT_1,CT_1\right)=HW\left(PT_1\oplus CT_1\right) \tag{3}$$

HD为汉明距离，HW为汉明重量，PT_1为处理前的数据，CT_1为处理后的数据。

门电路变换产生的功耗与汉明距离成线性正相关表达式为：

$$P=\alpha\times HD\left(PT_1,CT_1\right)+L+n \tag{4}$$

P表示为功率消耗，α为相关系数，L为常数，n为噪声。

汉明距离模型尽管其对功率消耗的计算比较粗略，但是凭借其简单易懂的原理，在实际操作过程中得到了广泛的应用。

2.3　支持向量机（Support VectorMachine，SVM）

支持向量机是一种二分类的方法，其基本思想是寻找一个最优分类超平面，使得该超平面到两类点中最近点的距离最大。支持向量机通常可分为三类：线性可分、线性支持及非线性支持向量机，分别对应线性可分、近似线性可分、线性不可分数据集。如图1所示：

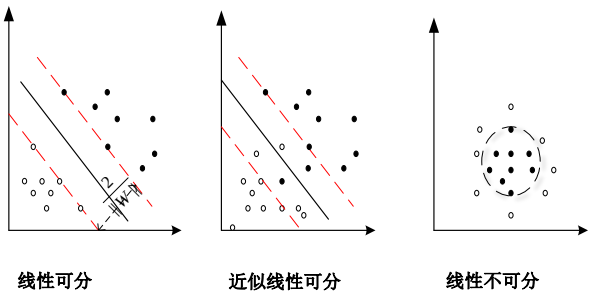


图1　支持向量机的分类

设线性可分数据集为｛（xi，yi）｝，xi∈Rn，yi∈｛-1，1｝，SVM的目标寻求分类超平面W•x+b=0，使得超平面距离两类样本最近点的距离最远，即要求$2/||w||^2$最大。其数学模型：

$$\begin{gathered}\min_{W,b}\left(\frac{\|W\|^2}{2}\right)\\ s.t.y_i(W\cdot x_i+b)\geqslant 1(i=1,\cdots,N)\end{gathered} \tag{5}$$

该数学模型是一个二次凸优化问题，利用拉格朗日乘子法转化为如下对偶形式：

$$\begin{gathered}\max -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}a_ia_jy_iy_j(x_i\cdot x_j)+\sum_{i=1}^{N}a_i\\ s.t.\sum_{i=1}^{N}a_iy_i=0\\ a_i\geqslant 0\end{gathered} \tag{6}$$

通过求解式（6）得到$a^*=(a_1^*,a_2^*,\cdots,a_N^*)$

依据公式$W^*=\sum_{i=1}^{N}a_i^*y_ix_i$，这里将$a_i^*>0$所对应的点叫做支持向量，并用支持向量求解$b^*$，得出决策函数：

$$f(x)=sign(W^*\cdot x+b^*) \tag{7}$$

由于在解决实际问题时都会涉及非线性可分离数据（即数据不能够准确的被线性分离出来），

通常会引入了核函数来将数据映射到高维空间内，从而达到数据再次分离的目的。因此选择或者构造合适的核函数对于SVM的性能至关重要，因为它决定了数据将要进行分类的变换后的特征空间[15]。

3 基于多级SVM的功耗攻击研究

假设明文M，密钥k，假设密钥n位，则汉明重量的可能取值有n + 1种，每一种汉明重量对应一个标签，则标签label1 = $\{HW_1, HW_2, \cdots, HW_{n+1}\}$。密钥空间$c = 2^n$，设每个密钥对应一个标签，则标签label2 = $\{k_1, k_2, \cdots, k_c\}$。针对明文M，分别用每个密钥加密，得到m条功耗曲线，则样本量c*m。

当前多分类SVM的方法主要有一对一和一对多两种方式。一对一的方法：对每两类样本设计一个SVM，则共需要设计$\frac{c(c-1)}{2}$个分类器，每个分类器训练时间复杂度为$O(m^3)$，总的时间复杂度为$O(c^2m^3)$。通常情况下，一对一的分类方法，计算复杂度较低，然而当分类数增加时，计算复杂度会急剧增加。一对多的方法，对每一类数据设计一个SVM，则共需要c个分类器，每个分类器进行训练时，正列样本m个，其余样本都为负列样本，则每一个SVM的训练时间复杂度为$O(c^3m^3)$，则总的时间复杂度为$O(c^4m^3)$。该类方法所需要的模型较少，但是每次训练都需要整个训练样本，且正负样本不平衡，将会严重影响分类准确率。

针对上述一对一和一对多各自存在的缺点，本文提出了一种多级SVM模型，首先根据设计第一级SVM模型，主要将汉明重量当做标签，利用一对多的方法，则需要$n + 1$个分类器，同时，为了避免训练中数据不平横带来的影响，在除正列以外的其他样本中随机抽取与正列相同的样本数作为训练数据集，不仅减少了计算复杂度，同时也解决了非平衡问题。其计算复杂度为$O(c^3m/n^2)$。对于第一级SVM模型，只能得出数据属于哪个汉明重量，并不能得出密钥，因此在第二级SVM模型中，我们试图得到精确的密钥。因此在第二级SVM中以密钥作为标签，同时，针对每个汉明重量对应的一类数据训练一个多分类SVM模型。假设共需要n+1个多分类SVM模型，此时第二级SVM的时间复杂度为$O(c^2m^2\log c)$，综上所述多级SVM模型的总的时间复杂度为$O(c^2m^2\log c + c^3m/n^2)$。对比上述两个模型，多级SVM相对一对一方式，得到了更少的分类器，且决策速度较快；相对于一对多分类器，不仅利用了更少的分类器，同时计算复杂度低，且在训练中，克服了数据非平衡问题对分类器带来的影响。多级SVM模型其流程如图2所示。

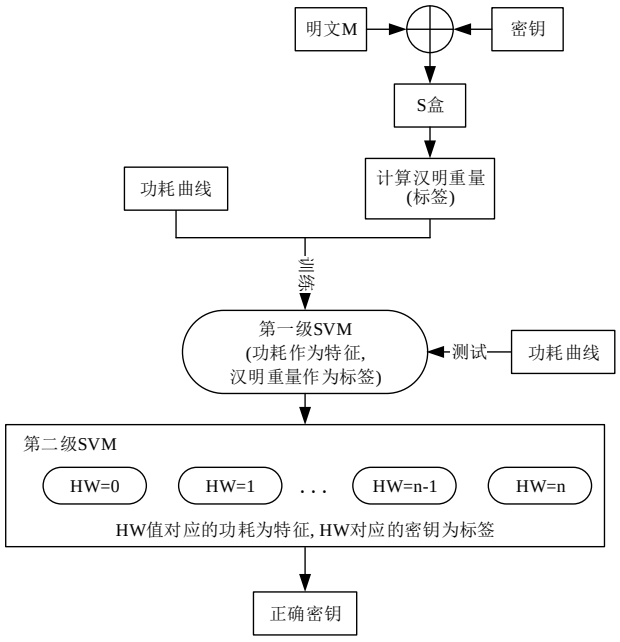


图2 多级SVM模型

结合多级SVM模型，本文提出的多级SVM的流程主要为：

（1）假设子密钥n位，选取一条明文，使用不同的密钥对其进行加密，使用功耗采集平台对S盒输出部位的功耗进行采集，共采集m（m<100）条功耗曲线，则得到SVM训练数据$m*2^n$个。

（2）选取汉明重量模型作为功耗泄漏模型，则其可能存在的汉明重量共有n+1种可能。利用一对多进行训练，把汉明距离相同的密钥作为一类，则有$c_1, \cdots c_{n+1}$类，每次训练时抽取m条正样本，剩余样本中随机抽取m个作为负样本（降低计算复杂度），则可以训练出n+1个模型：$f_1(x), \cdots f_{n+1}(x)$。其决策函数为$HW_value = \max(f_1(x), \cdots f_i(x))$

（3）假设步骤（2）中分类得出的汉明重量为h，则二级SVM对应了C_n^h个子类（n为猜测密钥总

位数）。而后根据汉明重量 *HW_value* 和明文信息获得可能的猜测密钥，进行精细化训练得到二级SVM训练模型 $h_{i,1}(x),\cdots h_{i,l_i}(x)$，其中i汉明重量，$l_i$ 表示汉明重量为i的密钥种类数，其则决策函数 $key=\max(h_{i,1}(x),\cdots h_{i,j}(x))$。

4 实验与分析

4.1 实验方案

为了评估本文方法对功耗攻击性能的提升，选择基于硬件的SM4算法进行选择明文攻击，具体实验步骤如下：

第一步：数据收集。通过在硬件实现的SM4对特定明文进行加密，选择第一个S盒作为泄漏点进行功耗数据的采集，共采集功耗曲线5000条。如图3为采集的一条功耗数据。同时记录其对应的明文与密钥信息，用于模型训练与精度验证。

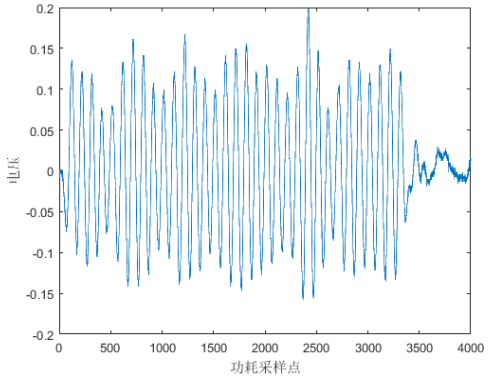


图3 采集的一条功耗数据

第二步：数据预处理。在该步骤中使用机器学习中常用的PCA降维方法来对功耗数据进行预处理，从而达到降低数据维度和去除信号中的冗余信息，并且降低其噪声[16]。

第三步：模型训练。利用训练数据对模型进行训练，并在训练过程中采用5折交叉验证。在训练过程中，选取支持向量机的核函数和参数，并选取最优值进行建模和验证。

第四步：模型评估。利用试验数据对训练后的模型进行了检验，并对模型的预测精度进行了评价。

第五步：实验对比。使用随机明文和固定密钥加密，总共记录了1500条功耗曲线。比较CPA，标准SVM和本文多级SVM进行实验验证，并对实验结果进行比较和分析。

4.2 方案实现与结果分析

在模型构建过程中，对于模型的训练集与测试集二者之间的比例对于最终的密钥预测结果都存在着较大的影响。为了确实二者之间的比例数值，选择不同比例的训练集和测试集进行测试，最终结果如表2所示。由表可知，当比例为8：2时，其最后的密钥预测准确率最高。因此，对比实验中的训练集与测试集比例设定为8：2。

表2 不同训练集与测试集比例的预测准确率

训练集	测试集	比例	SVM预测准确率
4500	500	9:1	0.8640
4000	1000	8:2	0.7830
3500	1500	7:3	0.7543
3000	2000	6:4	0.6760
2500	2500	5:5	0.6304

（一） 分类函数与超参数确定

根据支持向量机的原理，其核函数的选择对于特征转换和后期分类有着重要的影响。因此对支持向量机的四个常用函数（Linear、RBF、Poly、Sigmoid）进行了训练和测试。对应的每个核函数使用等量功耗曲线进行，训练100次后取平均值进行对比，其结果如表3所示。从表3中观察可以发现RBF核函数具有最高的预测精度，因此在对比实验中选择使用RBF核函数。

表3 不同核函数下预测精度值

核函数	均值	最大值	最小值
Linear	0.6875	0.7085	0.6630
RBF	0.7623	0.8080	0.7167
Poly	0.5911	0.6088	0.5733
Sigmoid	0.5545	0.6061	0.4850

在RBF核函数确定后，对于核函数的性能主要受参数γ和惩罚因子C的影响。如果C的值越大，则越容易导致出现过拟合的现象。其中C是惩罚因子，表示对误差的宽容度。C的值越大，表示对于测试和训练误差的容忍度越低，但是C太大会导致模型过拟合；相反如果惩罚因子C的值太小，模型就会欠拟合。另外一个重要参数γ只是对于高斯径向基核函数（RBF），它指的是RBF的幅宽，它的值是对每个支持向量对应的高斯的作用范围产生影响，从而影响模型的泛化能力。如果γ的值

设的太大，高斯核函数的概率分布就会变得又高又窄，这样它只会对支持向量样本数据附件产生影响，而对于未知样本数据分类结果就会变得很差。而如果γ设的过小，则会造成平滑效应太大，无法在训练集上得到特别高的准确率，也会影响测试集的准确率。其实在理论上，RBF 核函数中的γ接近无穷小时，SVM 模型可以拟合任意的非线性数据，但是这样会导致模型的训练准确率很高，而测试准确率就很低。在进行对比实验之前，采用交叉验证法对不同参数的预测精度进行了计算和比较，而后选择预测精度最高时的参数γ和惩罚因子C进行实验[17]。

（二） 特征提取

相关系数可以检验两个变量之间的相关性，在侧信道攻击中，可以用它来分析实际加密数据与功耗信息之间的相关性。根据Pearson相关系数公式，计算功耗曲线的每个采样时刻点与S盒输出中间值之间的相关系数（绝对值）。之所以取绝对值，这是因为在对功耗曲线的功耗信息进行统计时，计算出的相关系数有正有负，它们代表着加密中间值与功耗信息泄漏点的正、负相关性，而绝对值的大小表示着相关程度的高低。如图4所示，中间值与功耗曲线采样点的相关系数在某些位置点出现尖峰，表明10000个采样点中尖峰对应的功耗泄漏点与中间值高度相关。从表4统计可知，相关系数绝对值≥0.2的特征点就有70个，相关系数（绝对值）≥0.01的点达到3540个。

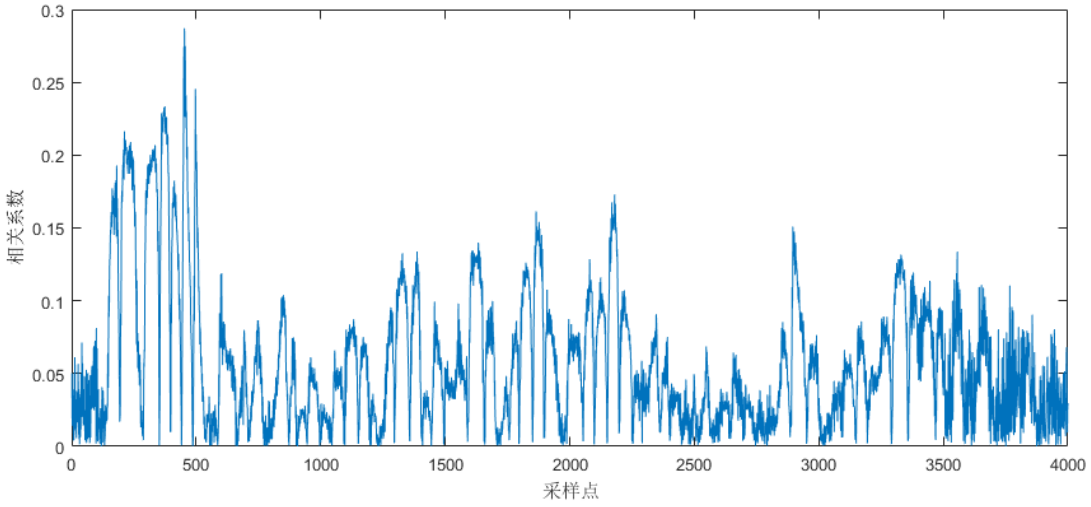


图4 中间值与功耗曲线采样点的相关系数图

表4 分类准确率与相关系数（绝对值）关系表

相关系数（绝对值）	特征点数	SVM分类准确率
R≥0.01	3540	95.43%
R≥0.03	2632	92.67%
R≥0.05	1852	87.54%
R≥0.1	631	82.68%
R≥0.2	70	68.32%

（三） 结果分析

根据上述实验的结果，明确支持向量机的核函数为RBF函数，参数C=56，γ=0.00195，训练集与测试集比例为8：2。结合3.1节实验方案中的实验步骤，依据相关系数大小来选取不同数量的特征点进行功耗分析，采用功耗模型为汉明重量模型，分别进行CPA、SVM分类和多级SVM分类攻击。其中CPA为从采集的功耗数据中随机采取1000条功耗，SVM和多级SVM分类同样随机选取1000条功耗，分别执行100次实验后取平均值，从而验证本文改进方案的性能。如图4为SM4算法执行选择明文攻击情况下某个S盒对应的攻击结果。

从表4和图5分析可知，CPA攻击准确率与特征点选取有关，当选取相关系数比较大的特征点，CPA攻击的准确率就越高，反之准确率就越低。这是因为相关系数较大的采样时刻点，其泄露的功耗信息越多，功耗信息与S盒中间值的相关性就越大。但是，SVM 多分类的准确率与中间值的相关系数呈负相关，这与CPA攻击的情况恰恰相反。

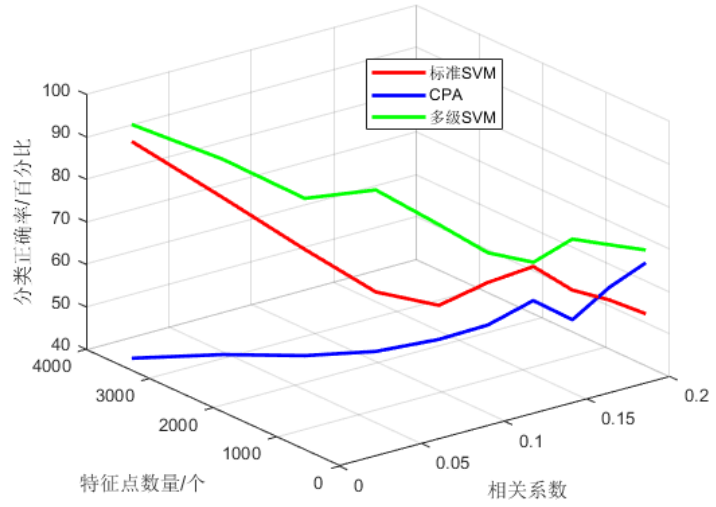


图5　三种方法分类正确率与相关系数、特征点数量三维关系图

这是因为特征点的个数越多，功耗曲线综合泄露信息也就越多，不同中间值之间的汉明重量之间的特征差异就越大，SVM的多分类准确率就越高。但是前提是这些点确实与操作数中间值相关，无关的特征点再多对攻击效果也无任何帮助。而多级SVM分类曲线趋势与SVM类似，区别在于多级SVM的最优分类准确率要好一些。SVM直接对密钥进行分类，类别多，计算复杂度高，准确率低，而多级SVM先对汉明重量进行筛选，而后对密钥进行分类，二次分类逐步缩小密钥类别，从而能够进一步的提高准确率。

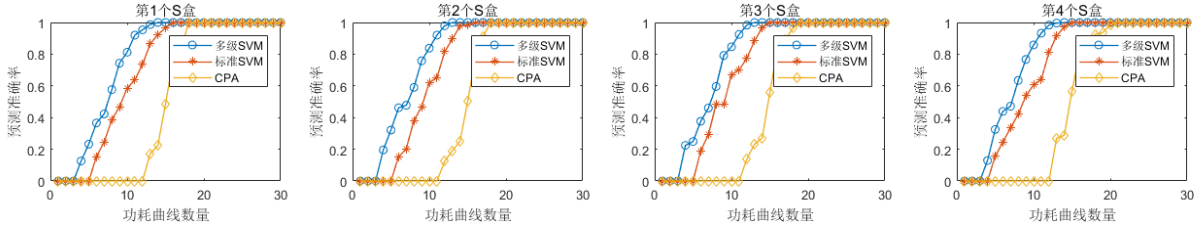


图6　4个S盒猜测密钥结果

如图6所示为SM4在执行选择明文攻击的情况下对应4个S盒的猜测密钥结果，其中横坐标为功耗曲线数量，纵坐标为猜测密钥的准确率。由图6可知，多级SVM方法下猜测4个S盒密钥所需功耗曲线数量分别为：4，4，4，4，即共计16条功耗曲线就可以猜测出全部的正确密钥；而标准SVM所需数量分别为：6，6，6，5，即共计23条功耗曲线就可以猜测出全部的正确密钥；而CPA所需数量分别为：13，12，13，13，即共计51条功耗曲线就可以猜测出全部的正确密钥。因此，与标准SVM相比，多级SVM的方法将SM4算法的密钥恢复能力提高了约30%，与CPA相比则提升了约68%。

5　结束语

本文在充分研究和分析当前SVM在功耗攻击应用的基础上，设计了一种多级SVM的功耗攻击方法。该方法将以汉明重量和密钥为标签进行分类的方式进行结合，首先以汉明重量为标签使用SVM对功耗进行分类，达到缩小密钥范围的目的；然后再以密钥为标签使用SVM对功耗进行二次分类，从而达到恢复其密钥的目的。将本文提出的多级SVM应用在功耗攻击上，采集SM4加密算法的功耗进行对比实验发现，与标准SVM相比其密钥恢复性能提高了30%，与CPA相比其密钥恢复性能提高了约68%。

虽然本文的改进方法对于功耗攻击的性能上有了提升，但是其实质是对于SVM分类方法的重复使用，在二次分类上以密钥为标签进行分类的方法上，其计算复杂度还有进一步降低的可能。下步将在二次分类上对其进行展开深入的研究，从而能够更好地提升相关功耗攻击性能，降低其计算复杂度。

参考文献:

[1] Kocher, Paul & Jaffe, Joshua & Jun, Benjamin.（1999）. Differential Power Analysis. Advances in Cryptology, CRYPTO'99. 1666.

[2] Kocher P C . Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS，and Other Systems［M］// Advances in Cryptology — CRYPTO '96. Springer Berlin Heidelberg，1996.

[3] Agrawal, Dakshiet al. "The EM Side – Channel (s) : Attacks and Assessment Methodologies. ”(2003).

[4] Brier E, Clavier C, Olivier F. Correlation power analysis with a leakage model［C］//International Workshop on Cryptographic Hardware and Embedded Systems. Springer，Berlin，Heidelberg，2004：16-29.

[5] MitchellTom M （1999). Machine learning and data mining. Communications of the ACM，42（11），30-36.

[6] Hetwer B, Gehrer S and Güneysu Tim（2019). Applications of machine learning techniques in side-channel atacks: a survey. Journal of Cryptographic Engineering.

[7] Prou ff E, Strullu R, Benadjila R, Cagli E and Dumas C（2018). Study of deep learning techniques for side-channel analysis and introduction to ASCAD database. Cryptology ePrint Archive, Report 2018/053. htps://eprint. iacr. org/2018/053.

[8] Heyszl J, Ibing A, Mangard S, De Santis F and Sigl G（2014). Clustering algorithms for non-profled single-execution atacks on exponentiations. In：Francillon A and Rohatgi P（eds.) Smart Card Research and Advanced Applications：12th International Conference, CARDIS 2013，Berlin，Germany，November 27-29，2013. Revised Selected Papers，pp. 79－93. Springer, Cham.

[9] Picek S, Heuser A and Guilley S（2017). Template atack versus Bayes classifer. J. Cryptogr. Eng. ，7（4），343-351.

[10] HospodarG. , GierlichsB. , D. MulderE. , VerbauwhedeI. , and J. Vandewalle，“Machine learning in side-channel analysis：a first study, "Journal of Cryptographic Engineering，Vol. 1，No. 4，pp. 293－302，2011.

[11] 胡晓阳,陈开颜,张阳,等. 基于支持向量机的RSA电磁旁路分析方法[J]. 计算机工程与应用，2019(15)：141-146.

[12] 邓高明, 张鹏, 赵强,等. 基于PCA和SVM的电磁模板分析攻击[J]. 计算机测量与控制，2009，17(009)：1837-1839.

[13] 杨家昌. 基于国密算法安全芯片的设计,实现与验证[D]. 2019.

[14] 李伟键. 功耗泄漏模型与能量分析攻击[C]// 2011年亚太青年通信学术会议(APYCC2011). 2011.

[15] Kotsiantis S B, Zaharakis I, Pintelas P. Supervised machine learning: A review of classification techniques［J］. Emerging artificial intelligence applications in computer engineering，2007，160：3-24.

[16] 于赛. 基于分组密码算法的侧信道分析与实现[D]. 2019.

[17] Kai Wang, Yingjian Yan, Chunsheng Zhu. Exploiting wavelet transform and support vector machine algorithm to perform side channel attacks on advanced encryption standard (AES)［P］. Artificial Intelligence，Information Processing and Cloud Computing，2019.

[作者简介]

马鹏（1993—），男，研究生在读，主要研究方向：机器学习，侧信道攻击与防御，流形学习。

刘祥（1994—），男，研究生在读，主要研究方向：机器学习，流形学习，图像处理。

量子计算物理体系综述

荆丽娜[1，2]，刘晓楠[2]，尹美娟[3]，穆清[2，3]，王美玲[2，3]，江舵[3)]

[1]郑州大学中原网络安全研究院 郑州 450000；

[2]数学工程与先进计算国家重点实验室 郑州 450000；

[3]信息工程大学网络空间安全学院 郑州 450000

摘　要：量子计算在理论上已远超经典计算信息存储和计算的能力，成为当代研究热门。新量子算法的不断提出证明了量子计算无与伦比的优越性，对传统密码和信息安全构成极大威胁，目前量子计算物理实现成为现阶段量子计算技术的研究热点和瓶颈问题。超导、离子阱、核磁共振、半导体量子点、光学、拓扑等多种方案层出不穷，其中有的已经让其他方案望尘莫及，有的关键技术亟待突破，有的正处于萌芽之中。本文通过对主流的几种量子计算物理体系进行综合性介绍，希望能够为从事量子计算研究的学者提供参考。

关键词：量子计算物理体系、超导、核磁共振、离子阱、半导体量子点

Overview of Quantum Computing Physics System

Jing Lina[1,2], Liu Xiaonan[2], Yin Meijuan[3], Mu Qing[2,3], Wang Meiling[2,3], Jiang Duo[3)]

1.Department of Zhong Yuan Network Security Research Institute,Zhengzhou University,Zhengzhou,450000;

2.State Key Laboratory of Mathematical Engineering and Advanced Computing,Zhengzhou45000;

3.School of Cyberspace and Security,Information Engineering University,Zhengzhou450000

Abstract: In theory, quantum computing has far surpassed the information storage and computing capabilities of classical computing, and has become a hot topic in contemporary research. The continuous development of new quantum algorithms proves the unparalleled superiority of quantum computing and poses a great threat to traditional cryptography and information security. At present, the realization of quantum computing physics has become a research hotspot and bottleneck problem in quantum computing technology at this stage. Superconductivity, nuclear magnetic resonance, ion trap, optics, semiconductor quantum dots, topology and other solutions are emerging in an endless stream. Some of which have left other solutions behind, some of which are in urgent need of breakthroughs, and some are in the bud. This article comprehensively introduces several mainstream quantum computing physics systems, hoping to provide references for scholars engaged in quantum computing research.

Key words: quantum computing physical system; superconductivity; nuclear magnetic resonance; ion trap; semiconductor quantum dot

1　引言

当前广泛使用的密码安全是基于高度复杂的数学难题设计的，如利用RSA加密体制，其原理为在有限的时间内无法在经典计算机上被破解。随着超级计算机计算能力的提升，尤其是近年来是量子计算机的迅猛发展，解决这些问题变得容易。换言之，一旦大规模的量子计算机诞生有可能威胁通信、贸易与金融业务的安全保密问题。与此同时，量子通信以量子态作为信息载体，利用量子叠加和量子纠缠等属性进行信息传递，基于量子力学中的不确定性、测量坍缩和不可克隆三大原理提供了无法被窃听和计算破解的绝对安全性保证。由此可见，量子技术是目前乃至未来的研究热门。

那如何在物理上实现量子计算机呢？科学家

基金项目：国家自然科学基金资助项目（No.61972413, No.61701539）。

们最初的设想是从传统电路的逻辑出发，只要能找到一个易于测量的双态体系，如光子的偏振方向，电子、原子核的自旋方向或原子、离子本身任意两个离散的能级构成的二能级系统，将它们组成大规模的阵列，再通过光、电、磁场等干涉方法组成一系列“量子逻辑门”对其进行操控，就可以构建出初步的量子计算机［1］。遗憾的是，原子尺度的粒子很容易受到外界环境的噪声干扰，从而由于退相干的原因失去量子效应，进而导致量子比特发生错误。同时，由于测量引起的塌缩使得每一个逻辑比特的纠错和读取都需要5000-10000个物理比特来进行辅助。因此，量子计算物理实现面临着巨大困难。

随着近些年微观尺度控制技术的飞速发展，人们积极的探索了各种可能实现量子比特的物理系统［2］。在衡量一个物理体系是否具有实现量子计算的潜力时，我们通常会用到DiVincenzo判断［3］：

1，能够形成易于表征的量子比特，且具有扩展能力；

2，能够将量子比特初始化到一个简单的量子态；

3，量子比特拥有远长于操作时间的消相干时间；

4，具有一套通用的完备的量子门操作；

5，比特的状态能够被精确地测量。

这五个判据中，尤其以第三条最为重要。我们必须得到足够长的消相干时间，来完成一个量子算法所需的全部操作［4］。因此，对于各种物理体系来说，目前最重要的指标就是尽可能延长消相干时间，缩短门操作时间，从而可以在有限的消相干时间内进行更多的门操作。因此，在量子比特的操控实验中，测量其消相干时间是十分重要的任务［5］。

目前已有一些物理体系满足这五个条件，备受人们期待能实现量子计算的物理系统有基于约瑟夫森结的超导量子系统［6］［7］［8］、离子阱系统［9］［10］、液体核磁共振系统［11］［12］［13］、半导体量子点系统［14］、光学量子系统［15］以及量子拓扑系统［16］等。

2　量子物理系统

2.1　超导量子系统

超导量子系统的核心是约瑟夫森结（Josephson junction）和一系列电容电感形成的非线性谐振电路。非线性谐振电路放置在极低温的环境中，处于高能级的量子比特会释放能量到基态，以此实现量子比特的初始化［2］。通过能量与二能级间能极差相近的微波光子与二能级的相互作用可以实现快速高效的门操作和非破坏性测量［2］。两个超导量子比特之间可以通过电容或电感等形式进行耦合［17］，实现高效可靠的扩展。器件的制备基于比较成熟的微纳米加工技术，非常有利于超导量子比特的规模化。

作为固态系统的超导量子比特［18］［19］，具有备受学界和业界的关注。常见的三种基本的超导量子比特，根据所操控的宏观自由度不同可分为超导位相量子比特、超导磁通量子比特、超导电荷量子比特［20］［21］。电荷涨落、磁通涨落以及准粒子噪音等因素会引起量子态退相干问题，目前最为流行的Transmon/Xmon量子比特设计就是基于此改进得到的。经过多年的发展，超导量子比特的退相干时间已经可以满足几百次门操作的需要，超导量子计算方案是现阶段认为最可能实现量子计算的方案。

超导量子比特和量子计算在最近的几年内取得了令人瞩目的进展，主要有两个显著的优势。其一，超导量子比特中的量子态是一个宏观量子态，因此很容易实现比较强的耦合。这也使得超导量子比特容易操控、易于读出、易于耦合，但同时也易于受到环境干扰。早期超导量子计算研究的重点之一就是如何降低环境影响，提高比特的量子相干性。经过十余年的发展，超导量子比特的退相干时间从1纳秒量级提升到了10-100微秒量级，极大地克服了退相干短板。其二，具有良好的可扩展性，有利于向更大规模集成化发展。我们可以像制造半导体芯片那样来制造“量子芯片”，一旦底层的技术产生突破，半导体工业顶尖的微纳米技术可以很快移植过来，从工业化角度讲，真可谓是一项巨大的优势，要知道半导体经过半世纪的发展，已然成为一艘工业巨舰［22］。

正因如此，国际上众多科技巨头对超导量子

计算偏爱有加，斥巨资研发的如著名的Google、IBM、Intel等，国内IT巨头包括阿里、腾讯、华为等，投资量子计算的重点都在超导方案上[22]。2018年3月3日，谷歌量子人工智能实验室发布狐尾松（Bristlecone）量子处理器，该处理器可实现72个量子比特长度上的单比特门操纵，单量子比特门最佳保真度99.9%，双量子比特门的最佳保真度99.4%[23]。

2.2 离子阱系统

离子阱系统是人们最早尝试实现量子化二能级的系统。早在1995年，奥地利Innsbruck大学的Cirac和Zoller首次提出利用失谐激光束照射和激光冷却实现量子比特的受控幺正变换和初态制备的线性离子阱量子计算系统，并展示了如何利用被俘获离子做一个控制非门，从此离子阱方案引起广泛关注[24]。离子阱量子计算的基本原理是：首先是利用电荷与电磁场的交互作用力牵制带电粒子体运动，将其稳定的囚禁在系统中[25]。然后通过激光冷却技术将被囚禁的离子冷却到其运动基态，再利用激光独立寻址单个离子实现单个量子位的操控，两个量子位的纠缠通过两个离子的内态与外态的耦合来实现[26]。

离子阱系统主要挑战在于外加激光强度、频率及相位的不稳定性，微型化和集成化存在巨大的困难，以及高度集成时如何保持有较高的高保真度[24]。2018年12月11日，IonQ公司发布了基于离子阱的量子计算机，他们利用模块化设计绕开了很难同时操控超过20个以上离子的难题，制备了有160个存储量子比特和79个量子比特的系统。尽管离子阱系统具有相对较好的退相干时间和很高的门操作保真度，但运行效率相对较低和对激光器等设备的海量需求限制了离子阱系统的发展[25]。

总之，离子阱技术目前仍面临四大难点[23]：一是离子阱暂时难以存储多条离子链；二是由于外加激光强度、频率及相位的不稳定，且离子对电场噪声敏感导致的消相干问题；三是可扩展性差；四是体积庞大，小型化尚需时日。

2.3 液体核磁共振系统

核磁共振（Nuclear Magnetic Resonance，NMR）现象是电磁波与原子中原子核相互作用的现象[27]。核磁共振最早是由Purcell和Bloch在1946年发现的[28][29]，他们发现暴露在强磁场中的磁性原子核，会与频率范围在射频区（大概在10MHz到1000MHz）的电磁波发生相互作用，磁性原子核吸收电磁波能量后在不同能级间发生共振跃迁，进而产生共振吸收信号。原子核的磁性来源于原子核的一个内禀物理属性：原子核的自旋。经过人们的研究发现，当原子核中的质子数为奇数，或者原子核中的中子数为奇数时，原子核有自旋。1997年，Chuang和Cory提出用液体NMR研究梁子计算后，NMR又被时代赋予新的意义--用来实现量子计算[30]。

核磁共振可以作为量子计算基于两点[31]：一是液体样品中的每个分子的核自旋大都与其他分子自旋隔绝，所以每个分子可以做独立的量子计算机，射频脉冲可以对样品中的每个分子的某一核自旋进行一致性操作。二是大量统计自旋态即有效纯态的变换特性与纯态一样，并且自旋间的相互作用恰好是量子算法理论所要求的。

在液态核磁共振系统中，用自旋为1/2的核来承载量子信息[32]。核自旋与外界强磁场相互作用，核自旋的能级产生劈裂（塞曼劈裂），产生的二能级系统用作一个量子比特，自旋方向与外磁场一致（即自旋向上）以及自旋方向与外磁场相反（即自旋向下）的两个本征态记为$|0\rangle$和$|1\rangle$，是量子计算基矢。单比特量子门操作由射频电磁波实现。射频电磁波的频率和核自旋在静磁场中的拉莫频率相同，用以控制核自旋在$|0\rangle$态和$|1\rangle$态之间的变换。两比特量子门操作是利用不同核自旋之间的耦合结合射频电磁波来实现的。

核磁共振方案由于液相或固相中核自旋体系与环境耦合较弱，环境对其影响很小，因此核磁共振系统实现的量子比特具有较长的退相干时间，一般可以达到秒的量级[33]。另外，由于核磁共振成像技术有很高的应用价值，经过多年的发展，核自旋体系的动力学行为研究已经很成熟，这为量子比特的门操作的实现打下了坚实的基础。早在2001年，核磁共振系统就展示了量子计算的实验实现，Vandersypen等人在核磁共振系统中展示了Shor算法将15分解为3*5的过程[34]。在核磁共振量子计算迅速发展的同时，也备受争议，主要集中在以下两个方面。其一，核磁共振量子计算的初始态--有效纯态的信噪比随着量子位数的增

加呈指数衰减［35］。其二，在高温近似下，对于较少的量子位情况，核磁共振系统中没有纠缠。

2.4 半导体量子点系统

由于经典计算机主要基于半导体技术，基于半导体开发量子计算也是物理学家研究的重点领域。现在的主要方法是在硅或者砷化镓等半导体材料的表面制作电极，在电极包围的点上利用电场产生势阱捕获少量甚至单个电子，取出电子在势阱中的能级作为量子比特［23］。基于量子点中电子的不同自由度编码量子比特［36］，科研人员开发出了单电子自旋编码量子比特［37］、空穴编码量子比特［38］［39］、电荷量子比特［40］［41］、以及采用多个电子操控编码的杂化量子比特［42］［43］［44］、自旋单态-三重态量子比特［45］［46］、交换量子比特［47］［48］等。根据电子的泡利不相容原理，通过自旋和电荷之间的关联，可以通过普通的电子开关门对电子自旋进行控制，完成包括单量子比特操作、两量子比特操作及结果的读出等在内的对量子自旋编码的量子比特的各种操作［23］。

2007年，代尔夫特大学的Vanderspyen团队在同一块半导体量子点器件上实现了量子比特制备、量子逻辑门操作、量子相干与测量等自旋量子计算的全部基本要素。2014年，新南威尔士大学测得自旋量子比特的退相干时间高达120微秒、保真度高达99.6%［49］。2017年，日本理化研究所在硅锗系统上获得了退相干时间20微秒、保真度99.9%的量子比特［23］。2018年中国科技大学半导体量子芯片研究组郭国平教授与其同事肖明、李海欧和曹刚等人创新性地设计并制备了半导体六量子点芯片，并在实验上实现了三量子比特的Toffoli门操控，成为国际上首个在半导体量子点体系中实现的三量子比特逻辑门，为未来集成化半导体量子芯片的研制奠定了坚实基础。到目前为止，单量子比特和双量子比特的逻辑门操控的保真度已经分别达到99.9%［50］［51］和98%［52］。半导体量子点的制备可与现有的半导体芯片工艺完全兼容、易于大规模集成，并且能够继承和发展现代半导体先进技术，为后续部署带来极大便利［36］。

2.5 光学量子系统

光学量子体系将光子的两种不同的振幅、路径或者轨道角动量等自由度作为量子比特，采用常用的量子光学手段进行读取和操控。光子极化是最常用的量子比特编码自由度，将光子的水平极化状态$|H\rangle$和竖直极化状态$|V\rangle$编码为$|0\rangle$和$|1\rangle$，来表示一个量子比特［53］。对于光子极化量子比特的操控和测量有成熟的半波片技术，光子极化的任意幺正旋转都可以由两块四分之一波片（QWP）和一块半波片（HWP）实现，而其测量则可以使用一块四分之一波片、一块半波片、一个极化分束器和探测器的组合实现［54］。光子路径自由度的操控需要借助分束器类的器件完成，测量通过调整路径相对相位以及借助非均匀分光的分束器实现［54］。对于轨道角动量自由度可以用螺旋台阶的相位片实现不同角动量模式的变换与混合，用Dove棱镜可以实现轨道角动量依赖的相对相位移动［55］。光子的频率、时间等自由度信息可以采用EOM、AOM等器件实现相应的操作来进行量子比特编码。

由于光子与环境相互作用很小，光学量子计算具有相干时间长、操作手段简单以及易扩展性等优点。但也正是由于光子之间相互作用微乎其微，导致两量子比特之间的逻辑门操作难以实现［23］。现阶段光量子计算需要的光路体积较大，集成化还有一定的难度。

2.6 拓扑量子计算系统

遵循量子力学规律而生的量子计算机的物理实现主要困难在于量子态很容易受到环境干扰产生退相干现象，难以实现大规模，拓扑量子计算系统因此而被提出。其利用非阿贝尔任意子，在受拓扑保护的任意子量子系统中，量子信息能够以非局域的形式被存储起来，这使得量子信息能够抵抗局部环境噪音的影响，从而在硬件上解决量子退相干问题［56］。为了实现量子计算，首先要在某种系统中创造出一系列任意子-反任意子，然后将这些任意子的两种熔接结果作为量子比特的两个能级，再利用编织进行量子比特的操控，然后通过测量任意子的熔接结果得到比特的末态［57］。

拓扑量子计算建立在全新的计算思路之上，应用任意子的交换相位，交换过程的“编辫”程序实现量子计算的信息处理。拓扑学研究几何形象在几何元素的连续形变下保持变的性质。如果

构成量子比特的元素是拓扑不变的，基于这些量子比特的运算结果也具有拓扑不变性。由此构造的量子计算对环境干扰、噪音、杂质有很大的抵抗能力。但拓扑量子计算尚停留在理论层面，实际上还未把这些理论付诸成器件化的现实［23］。

3 总结与展望

不可否认的是，目前量子比特的实现方式仍然存在不少缺陷和亟待改进的地方，类比于经典计算机，今天的量子计算机还处于经典计算机的电子管时代，连最底层的物理载体还未完成选定。不同的公司和高校关注重心亦不相同，详细如下：

基于超导、核磁共振、离子阱、光学、半导体量子结构、拓扑等不同物理载体实现的量子计算机亦各有优劣。比如，超导可扩展性强并可依托现有的成熟集成电路工艺，但对环境要求极为苛刻，需要极低温环境；核磁共振相干时间较长且可在室温下工作，但纠缠难度大，难以扩展；离子阱的保真度和相干时间较高，但可扩展性差；光子虽然相干时间较长，可扩展性好，但两量子比特门操作难；半导体量子点易于集成且可扩展，但退相干及保真度不足；还有拓扑量子对环境干扰有很大的抵抗能力，但目前还停留在理论层面。种种诸如此类的困难，使得多年过去人们也只能在实验室内做到小规模量子比特的量子计算原型。目前就物理实现方案方面，谷歌和IBM均基于超导线路，英特尔同时布局硅量子点和超导两种路线，微软则看好全新的拓扑路线，Honeywell则侧重离子阱路线［58］。

从目前发展脉络上看，各个体系有先有后，优点缺点也不尽相同。群雄逐鹿，鹿死谁手，尚未可知。有观点认为，未来量子计算机的实现可能是多种途径混合，比如利用离子阱量子比特的长相干时间做量子存储，超导量子比特的较高保真做操控等。也有观点认为，根据不同的量子计算用途，可能使用不同的量子计算方法，就像CPU更适合任务多而数据少的日常处理，而GPU更适合图像处理这种单一任务而数据量大的处理。随着各个国家对科研的进一步投入，相信在不久的未来量子计算物理实现方面会有很大的突破和进展。

参考文献：（参考文献格式参照2015年新标准GBT 7714-2015）

［1］ 李亦超.借用“平行宇宙”的算力——量子计算现状与展望［J］.自然杂志，2019，41（05）：364-369.

［2］ 李贺康.超导量子计算相关器件的制备工艺研究［D］.中国科学院大学（中国科学院物理研究所），2019.

［3］ David P. DiVincenzo. The Physical Implementation of Quantum Computation. 2000， 48（9-11）：771-783.

［4］ Alexandre Blais， Steven M. Girvin， Wil-

	超导	离子阱	核磁共振	半导体量子点	光学	拓扑
比特操作方式	全电	全光	全电	全电	全光	NA
量子比特数目	50+	70+	7	4	48	从0到1的过程中
相干时间	~50us	>1000s	2s	~100us	长	理论上无限长
两比特门保真度	99.4%	99.9%	99%	92%	97%	理论上可以到100%
两比特门操作时间	~50ns	~10us	~3ms	~100ns	NA	NA
可实现门数	~10^3	~10^8	~10^2	~10^3	NA	NA
业界支持(典型列举、非全部)	国外：谷歌、IBM、英特尔 国内：本源量子、浙大、南大、北京量子院	国外：IonQ、NIST、Honeywell 国内：清华、中科大	国外：IBM 国内：北大、量旋	国外：英特尔、普林斯顿、代尔夫特 国内：本源、中科大	国外：Xanadu、MIT 国内：中科大	国外：微软、代尔夫特 国内：清华、北大、物理所
技术成熟度	物理性质清楚，控制方法成熟		研究重点还在制备、控制方法和基本物理性质			基本单元尚未发现，处于最初构想阶段

liam D. Oliver. Quantum information processing and quantum optics with circuit quantum electrodynamics. 2020，16（9-11）：247-256.

［5］王保传.半导体量子点中量子比特编码研究［D］.中国科学技术大学，2017.

［6］Sébastien Léger，Javier Puertas-Martínez，Karthik Bharadwaj，et al. Observation of quantum many-body effects due to zero point fluctuations in superconducting circuits. 2019，10（1）：856-861.

［7］Shao-Ming Fei. Entanglement in IBMQ superconducting quantum computer with 53 qubits. 2020，2（3）：n/a-n/a.

［8］Ming - Jie Tao，Ming Hua，Na - Na Zhang，et al. Quantum simulation of clustered photosynthetic light harvesting in a superconducting quantum circuit. 2020，2（3）：n/a-n/a.

［9］Gan H C J，Maslennikov Gleb，Tseng Ko-Wei，et al. Hybrid Quantum Computing with Conditional Beam Splitter Gate in Trapped Ion System. 2020，124（17）：170502.

［10］胡长康.基于囚禁离子系统的绝热量子调控实验研究［D］.中国科学技术大学，2019.

［11］余琦.基于核磁共振体系的量子态制备与分团问题研究［D］.中国科学技术大学，2017.

［12］张泽.基于核磁共振系统的拓扑物态量子模拟［D］.哈尔滨工业大学，2019.

［13］Lino Jéssica Boreli Dos Reis，Sauer Stephan P A，Ramalho Teodorico Castro. Enhancing NMR Quantum Computation by Exploring Heavy Metal Complexes as Multiqubit Systems：A Theoretical Investigation.. 2020，124（24）：4946-4955.

［14］陈明博，徐永强，曹刚，郭国平.半导体量子点与谐振腔杂化系统的强耦合［J］.科学通报，2020，65（23）：2427-2438.

［15］陈明城.实验光学量子计算［D］.中国科学技术大学，2017.

［16］季文韬.基于金刚石氮—空位色心的动力学拓扑量子模拟［D］.中国科学技术大学，2020.

［17］徐达.基于超导量子比特的量子模拟［D］.浙江大学，2018.

［18］Ming - Jie Tao，Ming Hua，Na - Na Zhang，et al. Quantum simulation of clustered photosynthetic light harvesting in a superconducting quantum circuit. 2020，2（3）：n/a-n/a.

［19］Fritz Henneberger，Oliver Benson. Semiconductor Quantum Bits. 2016.

［20］戴坤哲.量子模拟及量子算法在超导量子系统中的应用［D］.南京大学，2018.

［21］赵凡.超导Al/Al_2O_3/Al约瑟夫森结和超导Nb悬空桥的制备工艺研究［D］.南京大学，2018.

［22］金贻荣.超导与量子计算［J］.自然杂志，2020，42（04）：301-310.

［23］冯晓辉，李雅琪，周斌，王翠林.2019年量子计算发展白皮书（上）［N］.中国计算机报，2019-10-21（008）.

［24］陈瑞亭.量子计算物理实现体系［J］.电脑知识与技术，2015，11（36）：139-140.

［25］董振铭.量子势阱对量子态的影响的新应用——量子纠缠态的制备和激光的制造［J］.科技视界，2019（26）：101-103.

［26］武文博.芯片阱上离子的稳定囚禁［D］.国防科技大学，2017.

［27］冯冠儒.量子模拟的核磁共振实验研究［D］.清华大学，2014.

［28］Purcell E M，Torrey H C，Pound R V . Resonance Absorption by Nuclear Magnetic Moments in a Solid［J］. Physical Review，1946，69：37-38.

［29］Bloch F，Hansen W W，Packard M . Nuclear Induction［J］. Physical Review，1946，69（3-4）：127-127.

［30］王碧雪.噪声下量子系统的核磁共振实验研究［D］.清华大学，2018.

［31］杨晓冬.用液体核磁共振实现量子计算［D］.中国科学院研究生院（武汉物理与数学研究所），2003.

［32］张泽.基于核磁共振系统的拓扑物态量子模拟［D］.哈尔滨工业大学，2019.

［33］李行.和乐量子门等量子算法的核磁共振实验研究［D］.清华大学，2017.

［34］Vandersypen L M，Steffen M，Breyta

G，Yannoni C S，Sherwood M H，Chuang I L. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance.［J］. Nature，2001，414（6866）.

［35］ 李可仁. 核磁共振中量子控制和量子模拟［D］.清华大学，2018.

［36］ 刘峘，李舒啸，李海欧，郭国平. 硅半导体量子计算进展［J］. 信息通信技术与政策，2020（07）：27-37.

［37］ Maurand R， Jehl X， Kotekar-Patil D，et al. A CMOS silicon spin qubit［J］.Nature Communications，2016，7（1）：133-137

［38］ Daniel Brunner， Brian D. Gerardot，Paul A. Dalgarno， et al. A Coherent Single-Hole Spin in a Semiconductor［J］. 2009， 325（5936）：70-72.

［39］ Michael H. Kolodrubetz， Jason R. Petta. Coherent Holes in a Semiconductor Quantum Dot［J］. 2009， 325（5936）：42-43.

［40］ Cao G， Li H O， Tu T，et al.Ultrafast universal quantum control of a quantum-dot charge qubit using Landau－Zener－Stückelberg interference［J］. Nature Communications， 2013， 4 （1）：104401-26.

［41］ Li H O， Cao G， Yu G D，et al.Conditional rotation of two strongly coupled semiconductor charge qubits ［J］. Nature Communications，2015， 6 （1）：1217-1265.

［42］ Li H O， Cao G， Yu G D， et al. Controlled Quantum Operations of a Semiconductor Three-Qubit System ［J］. Physical Review Applied， 2016， 9（2）.

［43］ Kim D， Shi Z， Simmons C B， et al. Quantum control and process tomography of a semiconductor quantum dot hybrid qubit［J］. Nature， 2014.

［44］ Cao G，Li H O，Yu G D，et al. Tunable Hybrid Qubit in a GaAs Double Quantum Dot.［J］. Physical review letters，2016，116（8）：086801.

［45］ Maune B M， Borselli M G， Huang B， et al. Coherent singlet-triplet oscillations in a silicon-based double quantum dot ［J］. Nature，2012， 481（7381）：344-7.

［46］ Shulman M D， Dial O E， Harvey S P， et al. Demonstration of Entanglement of Electrostatically Coupled Singlet-Triplet Qubits ［J］. Applied Physics Letters， 2015， 104 （14）：103108-205.

［47］ Medford J， Beil J， Taylor J M， et al. Self-consistent measurement and state tomography of an exchange-only spin qubit［J］. Nature Nanotechnology，2013，8（9）：654-659.

［48］ Eng K， Ladd T D， Smith A， et al. Isotopically enhanced triple-quantum-dot qubit［J］. Science Advances， 2015， 1 （4）： e1500214-e1500214.

［49］ Veldhorst M， Hwang J C C， Yang C H， et al. An addressable quantum dot qubit with fault-tolerant control fidelity［J］. Nature Nanotechnology，2014，9（12）：981-985.

［50］ Yoneda J， Takeda K， Otsuka T， et al. A quantum-dot spin qubit with coherence limited by charge noise and fidelity higher than 99.9%［J］. Nature Nanotechnology， 2018，13（2）：102-106.

［51］ Chan K W， Huang W， Yang C H，et al. Assessment of a silicon quantum dot spin qubit environment via noise spectroscopy ［J］. Physical Review Applied， 2018， 10（4）：044017.

［52］ Yang C H，Chan K W，Harper R，et al. Silicon qubit fidelities approaching incoherent noise limits via pulse engineering［J］. Nature Electronics， 2019，2（4）：151-158.

［53］ 黄合良. 线性光学量子计算研究［D］. 战略支援部队信息工程大学，2018.

［54］ 蔡昕东. 光量子计算及其算法实现［D］.中国科学技术大学，2015.

［55］ 张辉，杨夏. 量子计算机的0到N［J］. 自然杂志，2020，42（04）：311-320.

［56］ 何映萍，洪健松，刘雄军. 马约拉纳零能模的非阿贝尔统计及其在拓扑量子计算的应用［J］.物理学报，2020，69（11）：67-85.

［57］ Ady Stern， Netanel H. Lindner. Topological Quantum Computation—From Basic Concepts to First Experiments. 2013， 339（6124）：1179-1184.

［58］ 张海懿，崔潇，吴冰冰.量子计算技术产业发展现状与应用分析［J］.信息通信技术与政策，2020（07）：20-26.

[作者简介]

荆丽娜（1996-），女，硕士，学生，主要研究方向为量子计算。

刘晓楠（1977-），男，博士，副教授，硕导，主要研究方向为量子计算、高性能计算等。

尹美娟（1977-），女，博士，副教授，主要研究方向为信息安全、数据挖掘、高性能计算等。

穆清（1985-），男，硕士，讲师，主要研究方向为量子计算。

王美玲（1995-），女，硕士，学生，主要研究方向为先进计算。

江舵（1985-），男，硕士，学生，主要研究方向为量子计算。

基于分解的多目标进化算法的执行体生成方法

王俊超[1]，卫今[2, 3]，张帆[1]，庞建民[1]

[1]国家数字交换系统工程技术研究中心 河南 郑州 450002；

[2]复旦大学计算机科学技术学院 上海 200433；

[3]复旦大学大数据试验场研究院 上海 200433

摘　要：传统软件安全防御技术存在着防御措施滞后、攻防双方不均衡等问题，导致了网络安全问题日益频发。近年来，网络空间拟态防御技术（Cyberspace mimic defense，CMD）已经成为了一种有效应对未知系统漏洞的手段和方法。软件多样化编译技术可以作为CMD技术中执行体的生成方法，即基于给定的源代码，在编译层次将执行体的源代码编译为不同大小和外部特征的可执行程序。基于已有的多样化编译手段，本文主要解决了执行体的生成问题，即如何有效地选择和组织这些编译手段从而实现最优的执行体多样化，并具备较好的抵抗已知攻击的能力。本文综合考虑了单个执行体的攻击抵抗能力以及执行体种群的多样化属性，提出了一种基于分解的多目标进化算法（Multi-objective Evolutionary Algorithm Based on Decomposition，MOEA/D）的执行体生成方法。实验结果证明本文所提出的方法一方面能够保证执行体种群的多样性，一方面能够提升单个执行体的抗攻击能力。

关键词：软件安全、网络空间拟态防御、软件多样化编译、多目标优化、基于分解的多目标进化算法

1　引言

近年来，软件漏洞已经成为网络安全中重要的威胁之一。由于攻击者通常可以多次使用动态的攻击手段对保持静态的软件进行编译或者调试，例如：较为典型的攻击行为包括恶意终端攻击[1]（Man-At-The-End，MATE），该种攻击会对软件的机密性和完整性造成严重的破坏。

软件多样化防御手段的提出旨在降低MATE攻击者的适用性，通过增加攻击难度和成本来打破攻守不平等的现状，并且可以在一定程度上保护系统免受软件安全漏洞攻击的影响。但是当前的软件多样化的防御手段往往只针对于某种特定类型的攻击方法，每种防护措施都会因自身的不安全性导致被黑客成功绕过。例如，目前软件系统已广泛应用的地址空间布局随机化（Address Space Layout Randomization，ASLR）[2]可以被改进的return-into-lib（c）（RILC）等攻击攻破[5] [6]。Canary[3]保护机制通过在返回地址前插入Canary word来检测攻击行为，但是该防御方法是一种事后弥补的手段，并且相继被相关攻击手段绕过[7] [8]。控制流完整性（control-flow integrity，CFI）[4]防御方法也存在自身的局限性，在不访问程序源的情况下，往往无法获得程序的精确控制流程图，并且执行严格的CFI检查往往也会带来较高的性能成本[9]。编译器级别的软件多样化技术被称作多样化编译技术[10]。常见的具体实现方法包括：指令替换，控制流混淆和垃圾代码注入等，该种方法可以增加攻击者反编译的难度，但是单一的多样化编译手段也易被攻破。常用的一些多样化编译手段已经被集中到相关工具中，例如OLLVM[11]、Tigress[12]、Hikari[13]、Armariris[14]和multcompiler[15]等。

基于上述存在的问题，网络空间拟态防御技术（Cyberspace mimic defense，CMD）[16]为实现安全的软件防御技术提供了新的思路，其基本思想是使用功能等价的多元化或多样化软硬构件搭建动态异构冗余（Dynamic Heterogeneous Redundancy，DHR）架构，通过策略调度和重构重组等多维动态的不确定性机制，使得攻击难以实施。在本文的安全模型中，利用软件多样化容错的思想降低执行体相关程序代码同时出现相同错误的可能性。由于DHR架构安全性的前提是保证功能异构冗余的执行体之间的代码具有最少的相关性，

通讯作者：卫今，Email: jwei17@fudan.edu.cn

因此采用何种方式满足软件模块相异性成为本文模型中执行体生成方式的研究重点。

本文在DHR架构的基础上应用多样化编译技术，通过将源代码编译为不同的可执行程序。由于多样化编译手段的种类和排列决定了拟态架构的安全性，因此研究执行体生成的策略显得尤为重要。本文在该想法的基础上提出了一种基于分解的多目标进化算法[17]（Multi-objective Evolutionary Algorithm Based on Decomposition，MOEA/D）的执行体生成策略，并以执行体多样性、抵抗性为算法的主要考虑因素。由于执行体的编译手段可以作为该执行体的重要性质，针对采用不同编译手段生成的执行体环境，本文假设使用相同编译手段生成的执行体具有相同的属性和抵抗攻击的能力。基于该假设，参考Shannon-Wiener信息理论[32]定义出执行体种群的多样化属性，并且在不同攻击成功情况下定义出执行体的抵抗性属性。针对上述两种属性，确定出本算法的优化问题，即：应该如何选择执行体以同时最大化执行体种群多样性和抵抗攻击能力两种参数。解决该多目标优化问题时，在使用MOEA/D优化方法的基础上结合本文问题进行了改进，增加了遗传算法中突变和交叉的两个遗传算子。在实验部分验证了本文算法的有效性，通过实验可以验证算法中定义的相关变量，例如：攻击数目、所需执行体数目和编译方式数目等参数对执行体多样性和抵抗性两个指标的影响。

本文中将多样化编译手段应用于执行体生成、并进行执行体生成的相关研究的工作是具有创新性的。综上，本文的贡献主要包括：

创造性的提出了一种基于MOEA/D算法的执行体生成策略，本文的优化算法综合考虑了执行体种群的多样性和抵抗攻击能力两个指标。

对单个执行体的多样性和执行体环境的抵抗性采取了合理的手段进行度量，使得两种抽象参数可以定性和定量的进行分析。

在使用MOEA/D算法时，结合本文的具体问题进行了改进，引入突变和交叉的遗传算子。

在实验部分验证了了攻击数目、所需执行体数目和编译方法数目参数对执行体种群的多样性和抗攻击能力两个参数的影响。实验结果证实达到了预期的效果。

2 . 相关工作

2.1 网络空间拟态防御架构（cyberspace mimic defense，CMD）

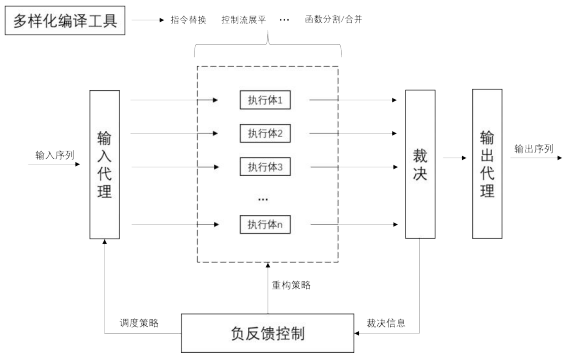


图1 多样化编译在DHR架构中的应用

网络空间拟态防御技术（CMD）[16]是为了解决网络空间中相关应用层次上基于未知漏洞、后门或病毒木马等不确定性的威胁，而提供的具有普适性的防御理论和方法。图1为该理论所提出的动态异构冗余（Dynamic Heterogeneous Redundancy， DHR）架构，其中包括异构的执行体环境、输入输出代理、功能等价的异构执行体、多模裁决和负反馈控制部分。异构的执行体理论上可以是任何一种软硬件结合的实现方法，可以是网络、平台、系统、部件或者模块、构件等不同层面、不同粒度的设备或者设施。在文本中利用不同的多样化编译手段具体实现执行体的异构冗余特性。此外，DHR架构中其他模块的功能简要介绍如下：输入代理将输入激励导入相关执行体，输出代理将多模裁决模块输出的裁决信息进行输出；裁决模块收集并比较多个不同执行体的输出信息，从而有效规避攻击者的攻击扰动，其表决策略使用丰富的裁决信息和策略来降低相对错误的概率；负反馈控制环节通过比较裁决信息，基于策略对可能被攻击成功的执行体进行清洗恢复和状态同步，并将执行体的调度策略传递给输入代理。经过实验证实，拟态系统可以容忍基于未知的漏洞和后门的外界扰动以及基于未知木马和病毒的渗透扰动，实现内外防护一体化[18]。

2.2 传统软件攻击以及软件防御技术

缓冲区溢出攻击[19]通过在内存边界写数据来攻击栈中缓存区的内容。针对此攻击典型的防御

手段包括ASLR[2]和Canary[3]。ASLR通过对堆、栈、共享库映射等线性区布局的随机化，增加攻击者预测目的地址的难度，防止攻击者直接定位攻击代码位置。 Canary方法通过检查插入返回地址前的canary word是否被修改，来判断是否发生了缓冲区溢出攻击，但是该方法只可以检测，不能够预防缓冲区溢出攻击。代码注入攻击[20]通过注入攻击代码来改变原代码的功能，例如更改指向代码指针的新代码。现在已有的一些防御手段，包括Data Execution Prevention—DEP（或者W□X）[21]和指令集随机化（Instruction Set Randomization， ISR）[22]等对代码注入攻击取得了很好的防御效果。DEP要求包含代码的页面不可执行、包含数据的页面不可写来达到保护的效果。ISR通过对指令集进行加密、解密操作，使得攻击者无法解读当前指令的相关含义，从而达到抵抗攻击的效果。代码重用攻击[23-27]作为一种较为新颖、强大的攻击手段，该攻击手段的原理是通过重组现有代码中的片段来实现自己的功能代码。该攻击主要包括三类：1）return-into-libc攻击通过栈溢出来覆盖正常函数栈中的返回地址，使得函数调用返回时执行libc中攻击者指定的函数。2）Return-Oriented Programming （ROP）攻击利用以ret为结尾的程序片段，通过巧妙设计的gadgets来实现攻击的流程。3） 与ROP攻击类似，Jump-oriented programming（JOP）攻击利用程序的间接跳转和调用指令来改变程序的控制流程。针对代码重用攻击的手段提出的防御技术主要包括： ASLR[2]、CFI[4]和软件故障隔离（Software Fault Isolation，SFI）[28]，其中CFI技术通过对每个间接指令进行检查来分析程序的控制流程，可以很好的预防代码重用攻击。SFI利用沙箱技术是将不被信任的代码限制在特定的代码块中，从而防止代码的重用攻击。不难看出，现有的软件防御方法采用了一种相对单一的软件防护方法，在一定程度上能够增加攻击者的攻击难度。攻击技术的改进将会使防御技术失效。

2.3 多样化编译技术以及相关工具

多样化编译技术可以看作一种程序编译算法，将某段程序作为输入，输出是一个功能上等效、但更难以理解和分析的程序。文献[10]列举了当前使用的多样化编译技术，包括：不透明谓词插入、变量分割/合并、数据流展平、指令替换、垃圾代码注入、虚拟化混淆等。实际上，单一的多样化编译手段并不能有效地预防当前所有的攻击情况，例如文献[29]通过代码解读、模式识别等方法成功破解了不透明谓词、编码算术、无用代码注入、控制流扁平化等技术；文献[30]通过代码解读与数据恢复攻击成功攻破了指令替换、虚拟化混淆等算法；文献[31]通过静态代码理解攻击检测恶意的重新包装的Android应用程序来成功攻破了指令集重排、增添或者移除系统调用等手段。

当前的多样化编译手段已经被集成到一些自动化工具中，例如由瑞士西北应用科技大学安全实验室研发的obfuscator-llvm（OLLVM）[11]和Hikari[13]。OLLVM是2010年6月份发起的项目，该工具支持控制流扁平化、指令替换、虚假控制流程等手段，兼容所有语言，包括：C，C++，Objective-C， Ada和Fortran；和目标平台，包括：x86， x86-64， PowerPC， PowerPC-64， ARM，Thumb。相比较而言，Hikari更为轻便、易于移植，并且控制流扁平化、基本块拆分、指令集替换、字符串加密、函数调用混淆等多样化编译手段。上海交通大学的GoSSIP小组设计的一种可以迭代更新的LLVM混淆框架：Armariris[14]，支持包括字符串加密、控制流扁平化、指令替换等手段，此工具支持x86、ARM等框架，并且可以在此基础上进行上开发出更多插件，满足开发者的更多需求。Tigresss[12]是C语言的一种多样化的虚拟器，支持多种防御静态和动态逆向工程以及反虚拟化攻击的防御措施，同样也是一个源到源的转换器：它以C源程序作为输入，并返回一个新的C程序作为输出。支持控制流扁平化、虚拟化混淆、拆分合并函数等手段。multicompiler[15]将随机性引入编译过程来优化软件的安全性，并基于LLVM构建的多个功能等效但内部不同的软件执行体，已经验证的支持的测试平台包括：Firefox，Apache，Python和LLVM。主要支持的手段包括：1）代码随机化，包括：函数排序、CPU寄存器变量分配、指令调度、插入nop指令和指令替换；2）堆栈布局随机化，包括：对堆栈元素进行重新排序、对堆栈元素和框架进行填充；3）全局变量随机化，包括：重新排列全局变量的顺序，随机添加填充值来破坏对全局变量的攻击。现有的多样

化编译方法种类繁多。在拟态防御技术中，如果能够将现有的多样化编译方法进行组合和应用，将会极大地提升系统的安全性。

3 基于MOEA/D的执行体生成算法

本章节的内容安排如下：3.1节介绍本文算法的应用场景。3.2节介绍本文算法的输入输出参数和相关符号定义。3.3节分析本文的优化问题模型。在第4节中，3.4.1节首先对MOEA/D算法的原理和概念进行介绍，3.4.2节说明本文算法在现有MOEA/D算法基础上做出的改进。

3.1 算法的应用场景

如图2所示，攻击者对执行体环境进行攻击，在本文的安全模型中，执行体的编译方法包括：指令替换、控制流展平、函数分割/合并等多样化编译方法。假设每种执行体的编译方式只有一种，本文算法的目的是决定执行体中编译算法的数量和组合，从而使执行体环境的安全性达到最高。

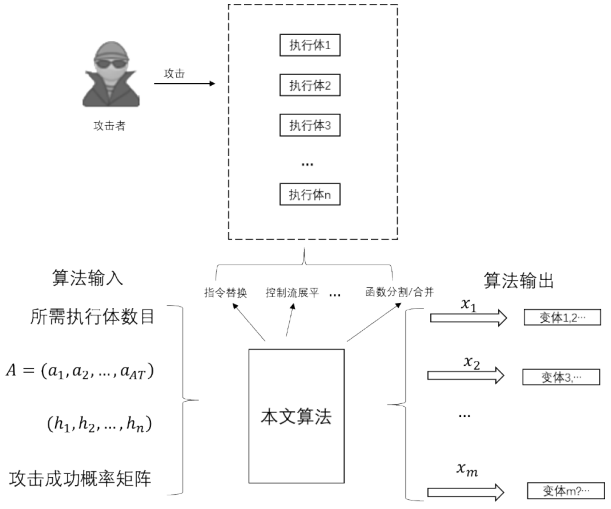


图2 本文算法模型

3.2 参数定义

根据图2，本文算法的输入包括所需执行体数目、攻击序列、攻击使用概率向量、攻击成功概率矩阵；算法的输出包括每种编译方法所对应执行体的数目，相关参数定义如下：

1) 问题输入参数

定义1：攻击数量（AttackNum，AT）：表示一个攻击序列中攻击种类的数目。

定义2．攻击序列A：表示一个攻击序列，$A=(a_1,a_2,..,a_{AT})$。其中，攻击序列中每种攻击方式都不相同，攻击种类的数量等于设置好的参数AT。

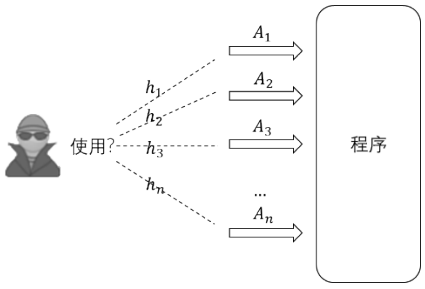


图3 攻击者采用一定概率使用攻击手段

定义3：攻击方式使用概率向量H：如图2所示，攻击者对可利用攻击的使用偏好不同，因此，我们使用$H=(h_1,h_2,...,h_{AT})$来表示攻击者选择可利用攻击向量的概率，其中$\sum_{i=1}^{AT}h_i=1$。h_i越高表明该攻击手段被利用的可能性较高，反之，被利用的可能性较低。

定义4：needVariantNum，nVN：用户指定执行环境中所需执行体的总数。

定义5：多样化编译方法数目（MethodNum，MN）：表示生成执行体的多样化编译方法的个数。每种编译方式代表一种执行体生成方法，我们认为同种编译方式生成的执行体属性和抵抗攻击的能力是相同的。

定义6：攻击成功可能性av_{ji}：本文认为在实例化的攻击中，攻击并不总是成功，而是会以一定概率成功。av_{ji}表示第j种攻击a_j在第i种执行体v_i上攻击成功的可能性。

定义7：攻击成功概率矩阵SucceedMat：该矩阵的第j行第i列的值为第j种攻击在第i种执行体v_i上攻击成功的可能性av_{ji}。

2) 问题输出参数

每种编译方式对应输出的执行体个数，$X=(x_1,x_2,...,x_{MN})$，其中x_i表示利用第i种编译方式生成执行体的数目，并且满足：$\sum_{i=1}^{MN}x_i=nVN$。

本文中出现的相关符号及其含义如下表所示：

3.3 本文的优化模型

在生成执行体时，本文主要考虑执行体的多样性和抵抗性指标，来实现安全性最高的执行体生成策略。

表1　相关符号及其含义

符号	含义
A	攻击序列
v_i	第i种编译方式的执行体
H	攻击使用概率向量H
$av_{\{ji\}}$	攻击a_j在第i种编译方式的执行体v_i上攻击成功的可能性
p_i	某个编译方式的执行体在整个执行体环境的占比
MethodNum，MN	多样化编译方法数目
AttackNum，AT	攻击数量
needVariantNum，nVN	用户指定所需执行体数量
AttackPF	攻击向量组
SucceedMat	攻击成功概率矩阵
likelihood	攻击成功概率
diversity	程序多样性程度

1) 执行体多样性指标

由于采用不同种编译方式来对执行体进行生成，借鉴Shannon-Wiener的信息理论[32]，本文认为同种编译方式生成的执行体属于同一类别且属性相同，因此使用下面式子表示执行体种群的多样性：

$$R = -\sum_{i=1}^{MN} p_i * \ln(p_i) \tag{1}$$

其中MN表示执行体编译方式的数目，p_i表示某个编译方式实现的执行体在整个执行体种群的占比，即$p_i = \frac{\text{第i种编译方式的执行体数目}}{nVN}$。使用指数的幂运算，可以将原式转化为式（2）：

$$Diversity = e^R = \prod_{i=1}^{MN} \frac{1}{p_i^{p_i}} \tag{2}$$

值得说明的是，某些编译方式的执行体数目可能为0，即$p_i = 0$。

2) 执行体抵抗性指标

本文假定当攻击者攻击成功大多数执行体（50%）时，对执行体环境的攻击最终是成功的。

因此执行体环境被攻击成功有以下几种情形：

情景1：当攻击者攻击成功的执行体数为$\left\lceil \frac{nVN}{2} \right\rceil$时，即为攻击成功；

情景2：当攻击者攻击成功的执行体数为$\left\lceil \frac{nVN}{2} \right\rceil$+1时，即为攻击成功；

…

情景nVN-$\left\lceil \frac{nVN}{2} \right\rceil$+1：当攻击者攻击成功的执行体数为N时，即为攻击成功。

可以定义在某次攻击中，被第j种攻击手段攻击成功的执行体编译方式集合可以表示为$S_{success}$，与之类似，没有被攻击成功的执行体编译方式集合为$S_{nosuccess}$。因此，攻击者对于多执行体执行环境攻击成功的概率可以表示为：

$$Likelihood = \sum_{j=1}^{AT} h_j \prod_{i \in S_{success}} av_{ji}^{x_i} \prod_{k \in S_{nosuccess}} (1 - av_{jk})^{x_k} \tag{3}$$

其中$\sum_{i \in S_{success}} x_i \geqslant \left\lceil \frac{nVN}{2} \right\rceil$

特殊情景：下面考虑两种特殊情形的攻击成功概率。同时为了便于分析，可以用$av_{j\min}$表示使用第j种攻击方式时，对某种执行体攻击成功概率的最小值；$av_{j\max}$表示使用第j种攻击方式时，对某种执行体攻击成功概率的最大值。即假设：

$$av_{j\min} = \min\{av_{j1}, av_{j2}, ..., av_{jMN}\}, av_{j\max} = \max\{av_{j1}, av_{j2}, ..., av_{jMN}\}。$$

Case1：当没有执行体被攻击成功时，此时的攻击成功概率可以表示为：

$$Likelihood_{case1} = \sum_{j=1}^{AT} h_j \prod_{i=1}^{MN} (1 - av_{ji})^{x_i} \tag{4}$$

其中x_i表示每种编译手段生成执行体的个数，并且满足$\sum_{i=1}^{m} x_i = N$

在case1情况下，攻击成功概率最小值为：

$$Likelihood_{case1\_\min} = \sum_{j=1}^{AT} h_j (1 - av_{j\max})^{MN} \tag{5}$$

式（5）表示此时，在执行体环境中，均采用同一种编译方式生成的执行体，并且对该种攻击手段抵抗能力最强。

在case1情况下，攻击成功概率最大值为：

$$Likelihood_{case1\_\max} = \sum_{j=1}^{AT} h_j (1 - av_{j\min})^{MN} \quad (6)$$

式（6）表示此时，在执行体环境中，均采用同一种编译方式生成的执行体，虽然所有的执行体并没有被攻击成功，但是对该种攻击手段抵抗能力最弱。

综上：

$$Likelihood_{case1\_\min} \leqslant Likelihood_{case1} \leqslant Likelihood_{case1\_\max}$$

Case2：当只有一种执行体被攻击成功时：

在该情境下多执行体执行环境被攻击成功的概率表示为：

$$Likelihood_{case2} = \sum_{j=1}^{AT} h_j \sum_{i=1}^{MN} (1 - av_{j1})^{x_1} (1 - av_{j2})^{x_2} ... av_{ji}^{x_i} (1 - av_{ji+1})^{x_{i+1}} ... (1 - av_{jMN})^{x_{MN}} \quad (7)$$

在case2情况下，攻击成功概率最小值为：

$$Likelihood_{case2\_\min} = \sum_{j=1}^{AT} h_j av_{j\min}^{x_{\min}} (1 - av_{j\max})^{nVN - x_{\min}} \quad (8)$$

其中$x_{\min}$表示攻击成功概率最小的执行体数目。

式（8）表示，在该执行体环境中，有一个执行体被攻击成功，并且攻击成功概率为最小值av_{jmin}；其余执行体均采用同一种抵抗能力最强的编译方式。

在case2情况下，攻击成功概率最大值为：

$$Likelihood_{case2\_\max} = \sum_{j=1}^{AT} h_j av_{j\max}^{x_{\max}} (1 - av_{j\min})^{nVN - x_{\max}} \quad (9)$$

其中$x_{\max}$表示攻击成功概率最大的执行体数目。

式（9）表示，在该执行体环境中，有一个执行体被攻击成功，并且攻击成功概率为最大值av_{jmax}；其余执行体均采用同一种抵抗能力最弱的编译方式。

综上：

$$Likelihood_{case2\_\min} \leqslant Likelihood_{case2} \leqslant Likelihood_{case2\_\max}$$

3) 本文多优化问题目标

本文的优化问题可以表示为：

$$\begin{aligned} &\min imize\ F = (-Diversity, Likelihood) \\ &subject\ to: \sum_{i=1}^{MN} x_i = nVN \\ &\sum_{i=1}^{AT} h_i = 1 \\ &h_i \geqslant 0 \\ &p_i \geqslant 0 \\ &av_{ji} \geqslant 0 \end{aligned} \quad (10)$$

其中式子（10）中表示本文的多目标优化问题的优化目标为Diversity和Likelihood两个指标。限制条件包括：输出结果中每种编译方式生成的执行体数目之和为所需执行体数目；使用每种攻击手段的概率之和为1；模型中出现的概率值均大于0。

4) 指标的归一化

值得说明的是，本文在使用MOEA / D对上述两个指标求解最优值时，由于两个目标函数的值之间的差距较大，不能简单地直接将它们进行聚合。因此采取以下指标的归一化方法：

a） 多样性指标保持原值。

b） 原执行体抵抗性指标除以最大值，并将结果取对数作为抵抗性指标：

$$Likelihood = e^{\frac{Likelihood}{Likelihood_{\max}}} \quad (11)$$

3.4 基于MOEA/D的执行体生成算法

本节介绍利用MOEA/D算法对式（10）所表示的优化问题的求解过程。在本章节中，3.4.1节首先对MOEA/D算法的原理和概念进行介绍，3.4.2节具体说明本文的算法流程，包括突变和交叉两个遗传因子的工作流程。

3.4.1 MOEA/D算法

MOEA/D算法[17]是一种基于分解的多目标算法。与其他多目标进化算法相比，MOEA/D具有目标数量的可拓展性和计算高效性，并具有以下特点：

1） MOEA/D将分解引入到多目标进化计算中，通过使用MOEA/D框架来解决多目标优化问题.

2） 因为MOEA/D算法是同时优化N标量子问题而不是直接将多目标优化问题作为一个整体来解决，因此会降低传统多目标遗传算法（Multi-objective Evolutionary Algorithm，MOEA）的多样性保持和适应度分配的难度。

3） MOEA/D利用相邻子问题解的信息同时优化N标量子问题。相对来说，MOEA/D不会重复的优化标量子问题，因为它利用了子问题之间的协同进化机制，所以算法的计算复杂度比较低。

根据文章[23]，MOEA/D算法表示为：

算法1. 基于分解的多目标进化算法（MOEA/D）

输入：MOP，，子问题的数量N，N个均匀分布的权重向量$\lambda^1,\lambda^2,\cdots,\lambda^N$，每个权向量领域被权向量的个数T，一个停止条件；

输出：EP。

步骤1：初始化：

1.1：设置EP为空集；

1.2：计算任意两个权向量之间的欧式距离，然后计算每个权向量最接近的T个权向量。对于$i = 1,2,\cdots,N$，设置$B(i) = (i_1,\cdots,i_T)$。

1.3：随机或者通过问题特定的方法产生一个初始种群。设置$FV^i = F(x^i)$。

1.4：随机用问题特定的方法$z = (z_1,z_2,\cdots,z_m)^T$。

步骤2：更新：

对于$i = 1,2,\cdots N$,循环以下步骤

2.1：复制：随即从$B(i)$中选择两个索引k,l，以x^k和x^l为父代使用遗传算法生成新个体y。

2.2：更新y：使用一个启发式算法改进y来生成y'。

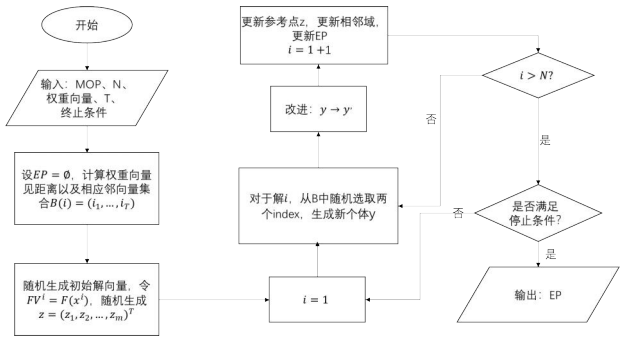
2.3：更新参考点z：对于所有$j = 1,2,\cdots,m$，如果$z_j \leqslant f_f(y')$，那么设置$z_j = f_f(y')$。

2.4：更新邻域解：对于$j \in B(i)$，如果$g^{te}(y'|\lambda^j,z) \leqslant g^{te}(x^j|\lambda^j,z)$，那么设置$x^j = y'$和$FV^i = F(y')$。

2.5更新EP：从EP中移除被$F(y')$支配的所有向量；如果EP中没有向量支配$F(y')$，就将$F(y')$加入到EP中。

步骤3：如果满足停止准则，停止并输出EP，否则，转向步骤2。

算法流程示意图见图3：



3.4.2 本文算法

本算法在实施中，在MOEA/D算法中设置2个遗传算子，分别是突变、交叉。突变表示一个执行体中某个或几个位置产生一个新执行体。交叉表示切换两个执行体中的一部分，并产生两个新后代。值得说明的是，需要确保经过突变和交叉算子之后生成的后代有效，即生成的后代中所有执行体数量的总和依然为nVN。下面介绍两种遗传算子的工作过程：

1) 突变

传统的个体突变算法是在个体中随机选择某些位置进行突变，然而，这种方法并不能够确保生成的个体是有效的，从而无法进一步进行适应度的计算等操作。因此，本文首先随机选择突变点，对于该突变点，在0和原来的值之间随机选择一个数值作为新值。具体流程为：首先随机选择一个突变点，检查执行体总数是否超过设定的数值，如果不超过，则随机设置一个新值，再次从头开始随机选择一个点；如果超过则减小该突变点的原值。下面为程序的某个实际突变过程：

mutSet initial individual：

[1， 0， 2， 0， 5， 0， 0， 4， 1， 0]

Latest individual：

[1， 0， 7， 0， 5， 0， 0， 4， 1， 0]

从中可以看出原执行体中的一部分进行了突变，该过程可以用图4表示：

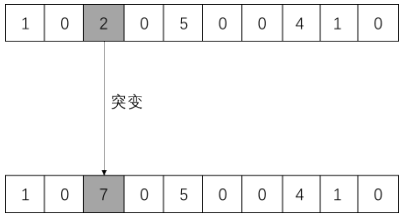


图4 突变过程示意图

2) 交叉

两个执行体ind1，ind2在交叉过程中，各随机选择两个交叉点cxpoint1，cxpoint2。当cxpoint2>=cxpoint1时，将cxpoint2的值加一，从而避免了交叉中cxpoint1与cxpoint2相等的情况。最后将ind1和ind2中cxpoint1到cxpoint2之间的值进行交换。

下面是一个交叉的实例：

cxSet initial individual：

[2， 1， 2， 0， 5， 0， 0， 4， 1， 0]

[3， 1， 0， 2， 1， 7， 0， 3， 1， 0]

After switch individual：

[2， 1， 2， 0， 5， 0， 0， 3， 1， 0]

[3， 1， 0， 2， 1， 7， 0， 4， 1， 0

从该实例中可以看到两个执行体的两个部分进行了交叉，过程可用图5进行表示：

结合算法1，本文执行体生成算法如下：

值得说明的是，由于MOEA/D算法的本质思想是将一个多目标优化问题分解为若干个子问题，

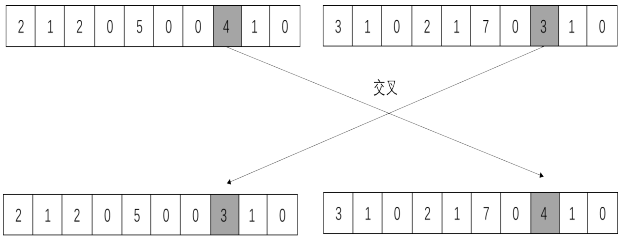


图5　交叉过程示意图

算法2. 基于MOEA/D算法的执行体生成算法

输入：攻击成功可能性Likelihood和执行体环境多样性Diversity表达式，迭代次数。

输出：攻击成功可能性Likelihood和执行体环境多样性Diversity优化值。

步骤1：初始化：参照算法1，计算两个子问题权向量之间的欧式距离，并采用随机或者特定方法产生一个初始种群和最优解。

步骤2：更新：使用突变或者交叉算子生成新的个体，并计算新个体中Likelihood和Diversity的值，如果新产生的值满足更新条件，则更新算法1中参考点、邻域解、Likelihood和Diversity的相关值。

步骤3：当满足迭代次数后，停止并输出Likelihood和Diversity的值，否则，转向步骤2。

每个子问题只利用相邻的几个子问题的信息进行优化，使得MOEA/D算法的复杂度较低。

4　实验

本文实验部分考虑3.3节中case1和case2两种情况下算法的结果。其中4.1节表明所使用的数据集的取值情况，4.2节分析在不同参数取值下，当优化问题取得最优解时，Likelihood和Diversity的计算结果的变化情况。本文使用仿真实验对本算法进行验证，具体环境参数如下： windows10系统，CPU：X64， 16GB内存。

4.1　数据集

分析下面几种参数的不同取值情况对本文优化算法结果的影响：

1） 攻击方式数目（AttackNum，AT）：设置AT的取值范围为［10，19］。

2） 所需执行体数目（needVariantNum，nVN）：设置nVN的取值范围为［20，29］。

3） 多样化编译手段数目（MethodNum，MN）：设置取值范围为［10，19］。

4.2　实验结果

在进行仿真实验时，本文设迭代次数为20次，当达到迭代次数后，停止迭代并输出Likelihood和Diversity的值，并认为此时两个参数的值为优化问题的解。

4.2.1　case1：当没有执行体被攻击成功时

图6表示AT对Likelihood和Diversity两个指标影响规律。图6.a表示当nVN参数固定时（nVN=20），4种不同MN取值情形下（MN=10-14）AT对Likelihood和Diversity的影响情况。图上侧的5条曲线表示为Likelihood的变化情况，可以看出Likelihood随着AT的变化会上下浮动，但是基本上变化趋势保持不变。图下侧的5条曲线表示为Diversity的变化情况，可以看出Diversity指标不会随着AT的变化而改变，这是因为Diversity只与MN和nVN有关。图6.b表示当MN参数固定时（MN=10），4种不同nVN取值情形下（nVN=20-24）的参数的变化规律，与图6.a类似，Likelihood随着AT的变化会上下浮动，但是基本上变化趋势保持不变；而Diversity指标不会随着AT的变化而改变。

图7表示nVN对Likelihood和Diversity两个指标的影响规律。7.a表示当MN参数固定（MN=10）时，4种不同AT取值情形下（AT=10-14）指标变化情况；7.b表示当AT参数固定（AT=10）时，不同MN取值情形下（MN=10-14）指标变化情况。7.a和7.b两图的变化趋势相同，可以看出Likelihood随着nVN指标单调递减。其中图7.a中AT取值不同的所有情形中，Diversity的值没有变化，可以进一步证明Diversity和AT取值无关的结论。图7.b中Diversity随着nVN参数的变化上下浮动，但是没有一个明显的变化趋势。

图8表示MN对Likelihood和Diversity两个指标影响规律。8.a表示当nVN参数固定（nVN=20）时，4种不同MN取值情形下（MN=10-14）指标变化情况；8.b表示当MN参数固定（MN=10）时，不同nVN取值情形下（nVN=20-24）指标变化情况。8.a和8.b两图的变化趋势相同，可以看出Likelihood随着MN指标单调递增。其中图8.a中AT取值不同的所有情形中，Diversity的值没有变化，可以进一步证明Diversity和AT取值无关的结论。图8.b中Diversity随着MN参数的变化上下浮动，但是没有一个明显的变化趋势。

。

4.2.2　case2：当一种执行体被攻击成功时

在case2情形下，即一种执行体被攻击成功时，图9~11分别表示AT、nVN和MN三种参数分

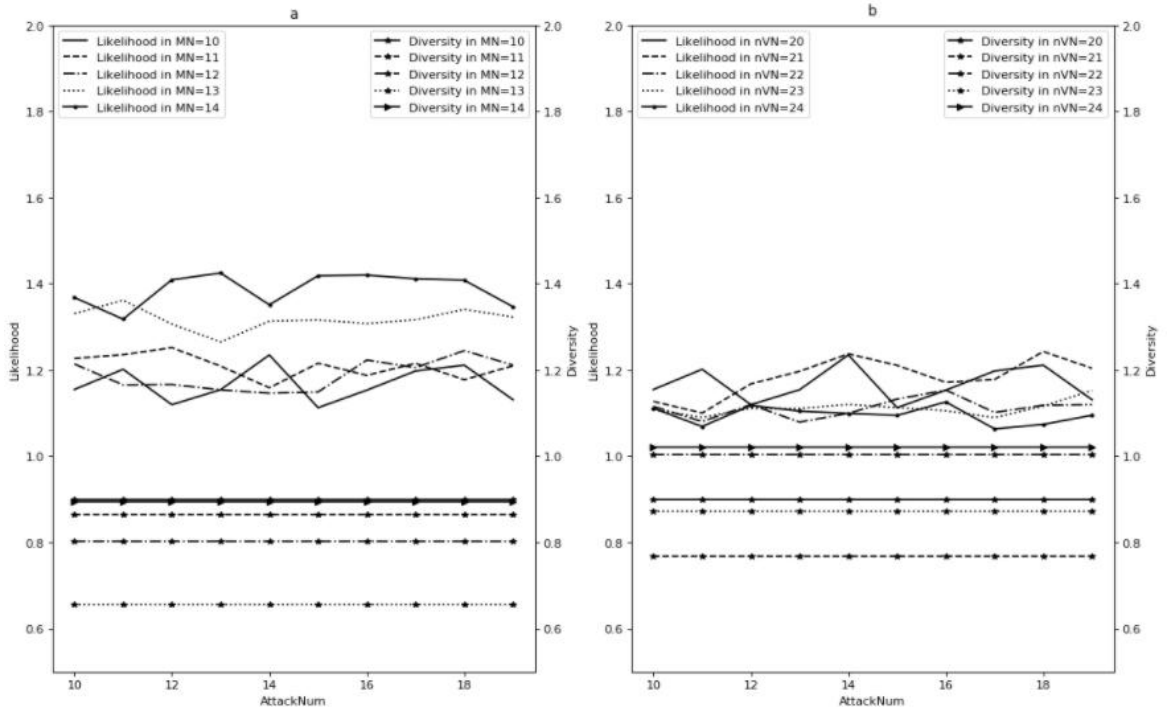


图6　case1情形下AT对Likelihood和Diversity两个指标影响规律

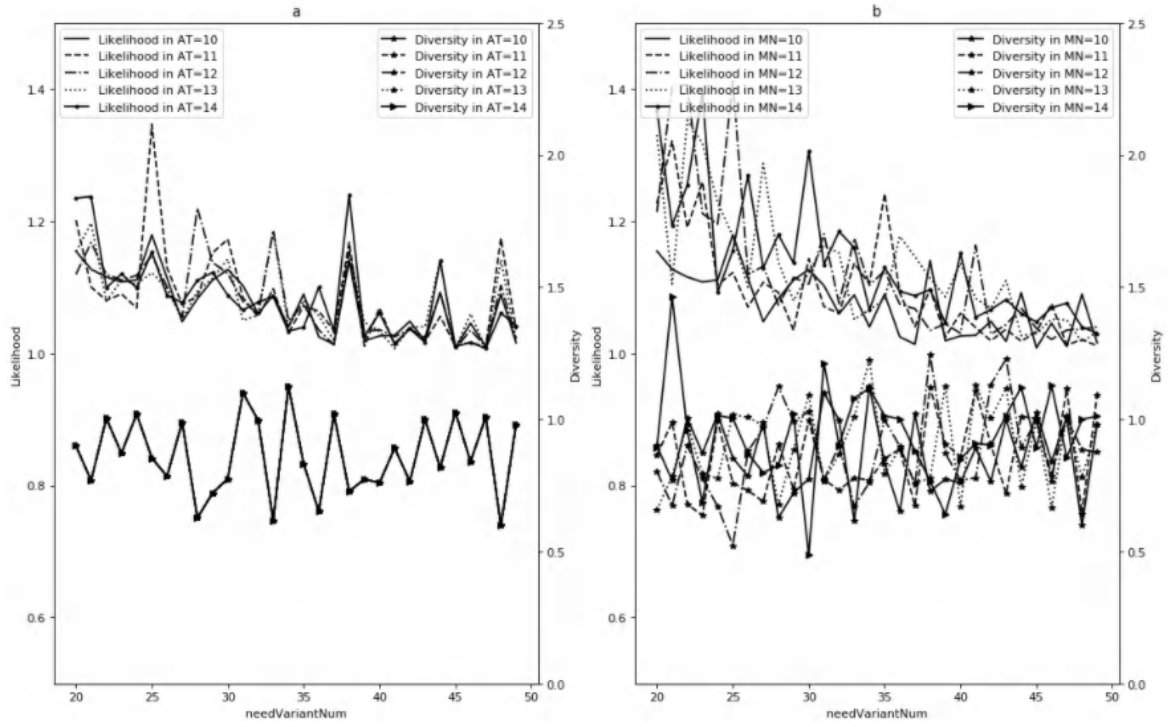


图7　case1情形下nVN对Likelihood和Diversity两个指标影响规律

别对Likelihood和Diversity两个指标的影响。从图中可以看出数据变化趋势与case1基本相同，也可以证明case1中的实验结果的正确性。

5　总结

针对拟态防御技术中多样化执行体生成问题，本文提出了一种基于MOEA/D算法的多执行体生

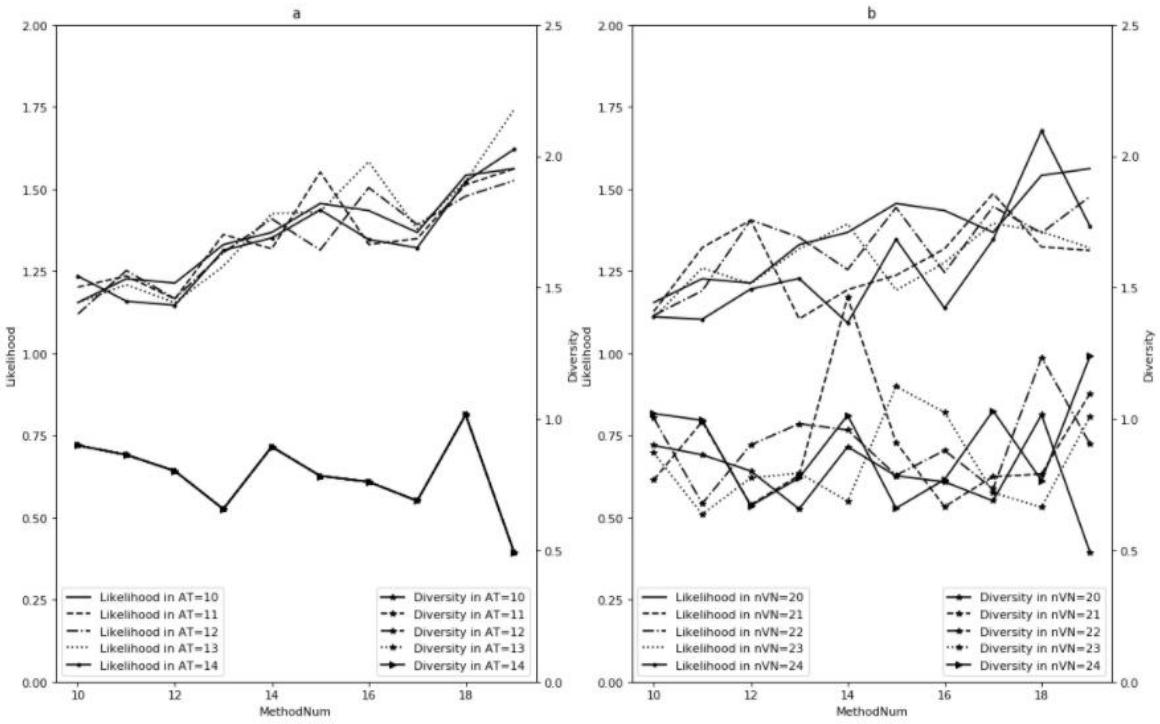


图8　case1情形下MN对Likelihood和Diversity两个指标影响规律

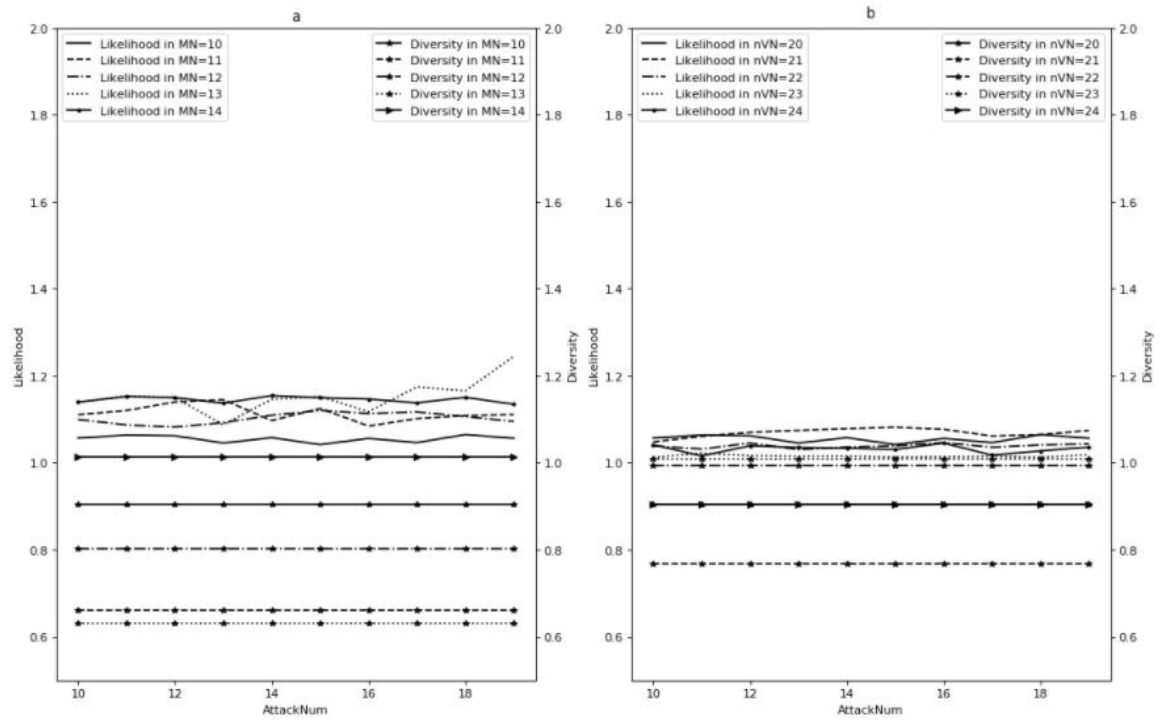


图9　case2情形下AT对Likelihood和Diversity两个指标影响规律

成算法。该算法综合考虑了执行体的多样性和攻击抵抗性两种因素，利用基于分解的多目标进化算法（MOEA/D）同时优化上述两个目标。该算法复杂性在可接受的范围内，并且在实验环节验证了本文方法的可行性。但是本文的执行体生成策略未考虑执行体的异构性和性能等因素，需要在后续研究中进一步完善相关工作。

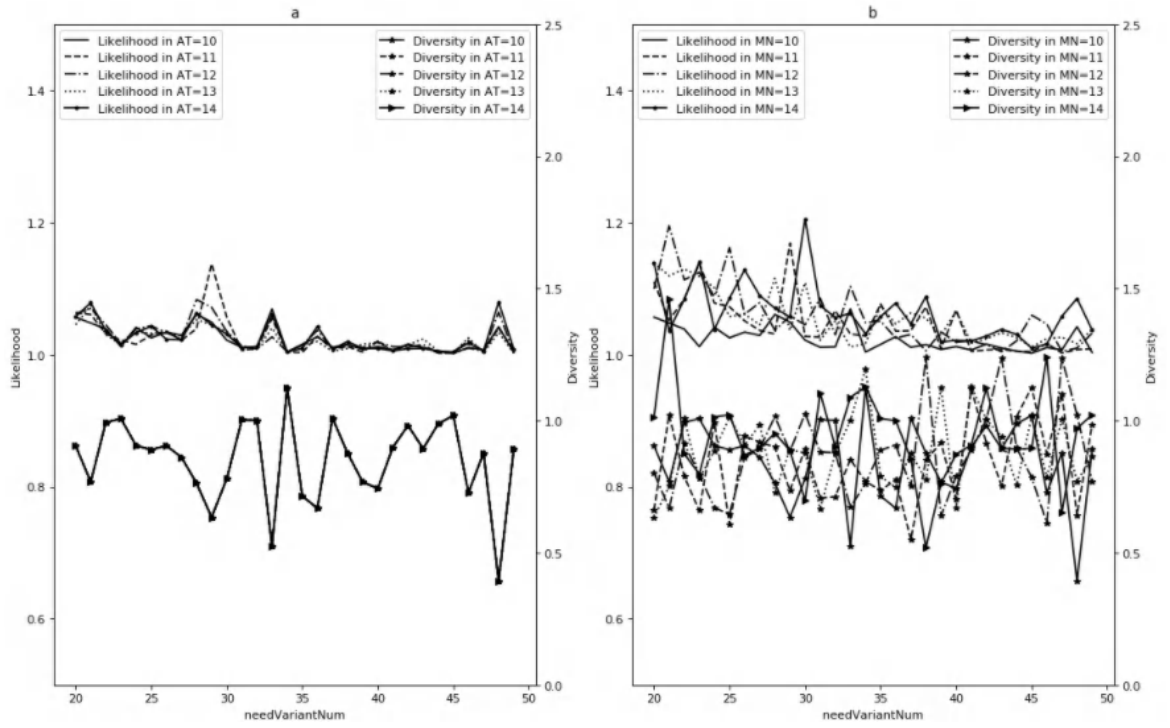


图10　case2情形下nVN对Likelihood和Diversity两个指标影响规律

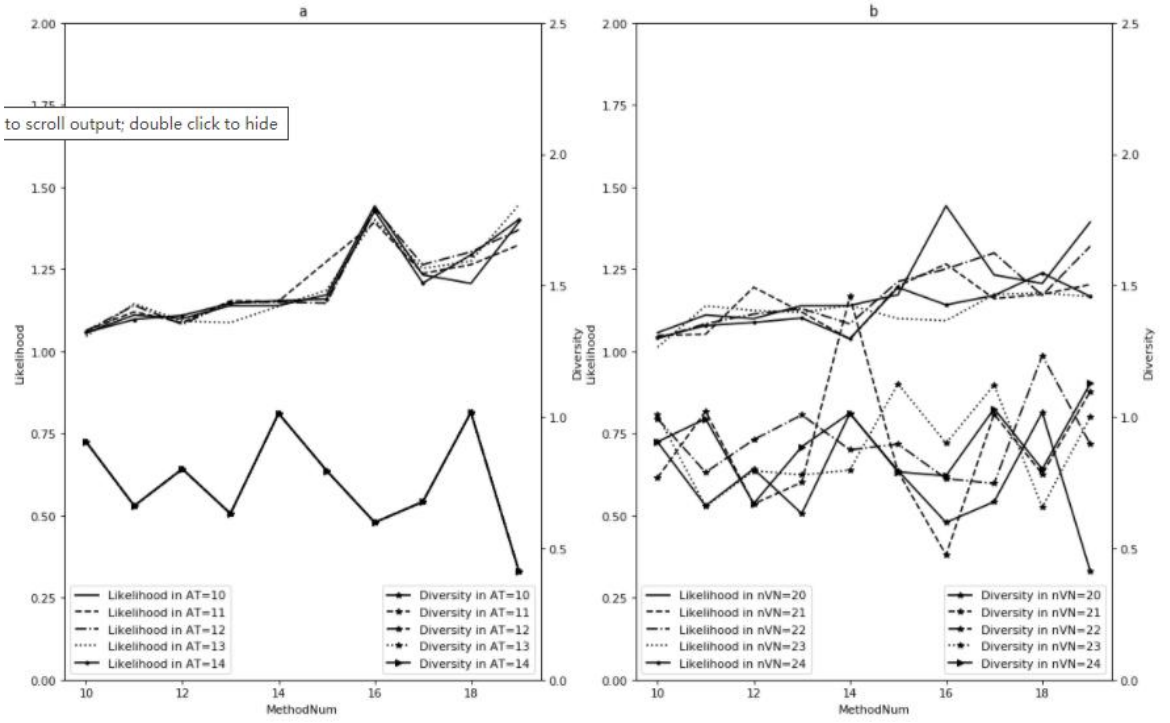


图11　case2情形下MN对Likelihood和Diversity两个指标影响规律

参考文献:

[1] Dolev D, Yao A. On the security of public key protocols[J]. IEEE Transactions on information theory，1983，29(2)：198-208.

[2] TeamPAX“PAX Address Space Layout Randomization，”https://pax. grsecurity. net/docs/aslr. txtFoundations of Computer Science，SFCS '81，pages 350 - 357，Washington，DC，USA，1981. IEEE Computer Society.

[3] CowanC.，PuC.，MaierD.，HintonyH.，WalpoleJ.，BakkeP.，BeattleS.，GrierA.，WagleP.，and ZhangQ.，“StackGuard：automatic

adaptive detection and prevention of buffer-overflow attacks, " in USENIX SEC，1998.

[4] AbadiMart´ın, BudiuMihai, ErlingssonUlfar, and LigattiJay. Control-flow integrity. In Proceedings of the 12th ACM conference on Computer and communications security (CCS'05), pages 340 – 353. ACM，2005.

[5] Nergal. The advanced return-into-lib(c) exploits: PaX case study. http：//phrack. org/issues/58/4. html.

[6] ShachamHovav, PageMatthew, PfaffBen, GohEu-Jin, ModaduguNagendra, and BonehDan. On the effectiveness of addressspace randomization. In Proc. 11th ACM Conf. Computer and Communications Security (CCS)，pages 298 – 307，2004.

[7] Gerardo Richarteet al. Four different tricks to bypass stackshield and stackguard protection. World Wide Web，1，2002.

[8] BittauAndrea, BelayAdam, MashtizadehAli, MazieresDavid, and BonehDan. Hacking blind. In Proc. IEEE Symp. on Security and Privacy (S&P)，pages 227 – 242，2014.

[9] GoktasEnes, AthanasopoulosElias, BosHerbert, and Georgios ¨ Portokalidis. Out of control: Overcoming control-flow integrity. In Proc. IEEE Symp. on Security and Privacy (S&P)，pages 575 – 589，2014.

[10] Banescu S, Pretschner A. A tutorial on software obfuscation［M］// Advances in Computers. Elsevier，2018，108：283-353.

[11] https：//github. com/obfuscator-llvm/obfuscator/wiki/Installation

[12] https：//tigress. wtf/

[13] https：//github. com/HikariObfuscator/Hikari/wiki/Usage

[14] https：//github. com/GoSSIP-SJTU/Armariris

[15] https：//github. com/securesystemslab/multicompiler

[16] Wu J X. Cyberspace mimic defense［M］. Springer International Publishing，2020

[17] Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition［J］. IEEE Transactions on evolutionary computation，2007，11(6)：712-731.

[18] 任权, 贺磊, 邬江兴. 基于离散马尔可夫链的不同抗干扰系统模型分析%Analysis of different anti-interference system models based on discrete time Markov chain[J]. 网络与信息安全学报，2018，004(004)：30-37.

[19] ChewM. and SongD. . Mitigating buffer overflows by operating system randomization. Technical Report CMU-CS-02-197，epartment of Computer Science，Carnegie Mellon University，2002.

[20] Banescu S, Collberg C, Pretschner A. Predicting the resilience of obfuscated code against symbolic execution attacks via machine learning［C］//26th {USENIX} Security Symposium ({USENIX} Security 17). 2017：661-678.

[21] PaX. Homepage of The PaX Team，2001. http：//pax. grsecurity. net.

[22] HiserJ. , Nguyen-TuongA. , CoM. , HallM. , and DavidsonJ. W. . ILR: Where'd my gadgets go? In Proceedings of the 33rd IEEE Symposium on Security and Privacy, S&P '12，pages 571 – 585，2012.

[23] Nergal. The advanced return-into-lib(c) exploits: PaX case study. Phrack Magazine，11（58），2001. http：//www. phrack. org/issues. html？ issue=58&id=4.

[24] KrahmerS. . x86-64 buffer overflow exploits and the borrowed code chunks exploitation techniques，2005.
http：//www. suse. de/
no-nx~krahmer/. pdf.

[25] ShachamH. . The geometry of innocent flesh on the bone: Return-intolibc without function calls (on the x86). In Proceedings of the 14th ACM Conference on Computer and Communications Security，CCS '07，pages 552 – 561，2007.

[26] TranM. , EtheridgeM. , BletschT. , JiangX. , FreehV. W. , and NingP. . On the expressiveness of return-into-libc attacks. In Proceedings of the 14th Interntional Symposium on Recent Advances in Intrusion Detection，RAID '11，pages 121 – 141，2011.

[27] ZhangM. and SekarR. . Control flow integrity for COTS binaries. In Proceedings of the 22nd USENIX Security Symposium，SEC '13，pages 337 – 352，2013.

[28] McCamantS. and MorrisettG. . Evaluating SFI for a CISC architecture. In Proceedings of the 15th USENIX Security Symposium，SEC '06，pages 209 – 224，2006.

[29] SalemA. and BanescuS. . Metadata recovery from obfuscated programs using machine learning. In Proceedings of the 6th Software Security，Protection and Reverse Engineering Workshop，page 8. ACM，2016.

[30] BanescuS. , CollbergC. , GaneshV. , NewshamZ. , and PretschnerA. . Code obfuscation against symbolic execution attacks. In Proc. of 2016 Annual Computer Security Applications Conference. ACM，2016.

[31] ZhangF. , HuangH. , ZhuS. , WuD. , and LiuP. . Viewdroid: Towards obfuscationresilient mobile application repackaging detection. In Proceedings of the 2014 ACM conference on Security and privacy in wireless & mobile networks，pages 25 – 36. ACM，2014.

[32] Hernandez-Castro J, Rossman J. Measuring software diversity, with applications to security[J]. arXiv preprint arXiv:1310. 3307，2013.

基于软件多样化的软件系统安全性度量

卫今[1, 2]，王俊超[3]，张帆[3]

[1]复旦大学计算机科学技术学院 上海 200433；

[2]复旦大学大数据试验场研究院 上海 200433；

[3]国家数字交换系统工程技术研究中心 河南 郑州 450002

摘　要：软件多样化技术是一种能够有效提升软件安全性的技术手段，网络空间拟态防御（Cyberspace mimic defense，CMD）技术所提出的动态异构冗余（Dynamic Heterogeneous Redundancy， DHR）架构为实现高可用、高可靠的软件多样化技术提供了理论基础。然而现有的软件多样化技术在网络空间拟态防御技术中的应用缺乏一种有效的安全性评估手段。本文使用不同的软件多样化编译技术对DHR架构中执行体的源代码进行编译，并在此基础上提出了该系统安全性的度量方法，该度量方法综合考虑了单个执行体的攻击抵抗性和历史抵抗性两个指标。在实验部分分析了攻击时间、执行体数目等因素对安全性相关指标的影响情况，验证了本文所提出的度量算法的有效性。

关键词：软件安全、网络空间拟态防御、动态异构冗余架构、软件多样化编译、抵抗性度量

1　引言

软件多样化技术是一种针对软件安全威胁常用的防御手段，在不改变程序语义的基础上，通过增加程序的复杂性提升攻击的成本和难度。多样化编译[17]是一种编译层面的软件多样化技术，通过改变源代码编译后的形态增加攻击者理解代码的难度，该技术的具体实现方式包括：指令替换，控制流混淆和垃圾代码注入等。多样化编译方法的安全目标是保证编译后代码的语义不能够被攻击者识破，但是攻击者往往会采用逆向工程及相关手段对保护后的代码进行恶意分析[1-6]。例如：动态符号执行[23]就是一种用常用的反编译手段。

现如今，软件多样化技术仍然缺少可用的方法来精确描述其安全性指标。Collberg[7]首次从Potency和resilience两个方面来表示多样化编译后代码的复杂程度，之后相继有研究对这两个指标进行了补充和创新[8][9]，但是这些研究只能描述源代码复杂度的增加情况，并不能精确表述编译后代码的安全性。因此，本文认为现如今软件多样化技术面临两个问题：1）如何实现一种较为安全的软件多样化防御手段？2）如何去评价该防御手段的安全性和对于攻击的抵抗能力?

针对上述两个问题，网络空间拟态防御[10]（Cyberspace mimic defense，CMD）技术给出了解决问题的新思路。CMD的主要思想是运用内生安全机制来应对系统内部因为未知漏洞后门或者病毒木马等引发的不确定扰动，同时也提供了一种具有普适性的防御理论。该理论所提出的动态异构冗余（Dynamic Heterogeneous Redundancy，DHR）架构以成熟的异构冗余可靠性理论为基础，结合广义鲁棒控制架构，构造出动态化、多样化的防御场景。本文通过对DHR架构中的执行体实施多样化编译手段，使其对外表现出不同的形态特征，从而实现一种安全性高的软件多样化防御手段。该防御手段有以下两个优点：

1） 利用DHR架构的特性，可以同时实现多种多样化编译手段，提高了单一多样化编译方法的安全性，从而实现一种高可用、高可信的软件多样化防御技术。

2） DHR架构中有较为全面、系统的算法来度量其安全性[11][12]，结合多样化编译的特性，本文提出了一种度量软件多样化技术安全性的方法。

通讯作者：王俊超，Email: wangjunchao11@126.com

综上所述，本文通过在DHR架构的基础上，利用多样化编译方法实现冗余异构的执行体，从而提出了一种新型的软件多样化防御技术。该技术在理论上可以抵御大部分攻击者对软件的恶意反编译或者调试，以及攻击者试图造成的系统输出错误、系统异常或者恶意的更改行为，并且可以及时发现并抵抗执行体中的未知病毒、系统漏洞、后门等引发的功能型攻击。本文的软件多样化安全技术，可以实现动态、冗余的多样化编译功能，可以弥补单一多样化编译手段的缺陷，可以实现一种高可靠、高可信的软件防御手段。

本文提出了一种针对该软件多样化技术安全性的度量方法。在定义单个执行体的安全性时，结合了攻击抵抗性和历史抵抗性两个指标。利用反编译工具执行攻击的时间来表示单个执行体的攻击抵抗性，并使用执行体的历史裁决结果表示单个执行体的历史抵抗性。基于以上定义，分析了执行体集合的几种安全情形，并给出了相关概率表达式。通过仿真实验验证了本文算法的有效性，在假设抵抗性满足指数分布的前提下，模拟出攻击抵抗性和历史抵抗性两个指标随时间变化的规律，并进一步分析出变体数目和攻击时间等因素对执行体集合安全性的影响。

综上，本文的贡献有：

提出了一种安全性和可靠性较高的软件多样化防御技术。该技术结合了DHR架构和多样化编译手段，可以有效缓解当前软件防御技术面临的安全危机。

提出了一种有效度量软件多样化技术安全性的方法。通过考虑执行体自身的攻击抵抗性，并结合其历史裁决信息，对执行体集合的几种安全情形进行了分析。

在实验环节对算法模型中的攻击抵抗性和历史抵抗性指标进行了度量，并分析出相关因素对执行体集合安全性的影响规律。

2 背景知识

软件多样化技术通过增加攻击者的难度和成本打破攻守不平衡的现状。例如：地址空间布局随机化[13]（Address Space Layout Randomization，ASLR）通过随机化程序堆、栈和代码分段的基地址，迫使攻击者相关函数的位置进行猜测，进而减少攻击成功的概率。Canary保护机制[14]能够在返回地址前插入Canary word，从而当缓冲区溢出时，系统能够检测出来返回地址被覆盖的异常情况。指令集随机化[15, 36]（Instruction Set Randomization，ISR）用来抵御代码注入攻击，通过使用某种加密算法对代码实施加密，使得攻击者没有密钥进行解密，进而无法执行相应的攻击代码。指令地址随机化[16]（Instruction Location Randomization，ILR）通过指令地址随机化阻止攻击者重用代码的行为。

多样化编译技术[17]作为当前常用的软件多样化技术，按照转换目的包括：1）常数转换手段，包括：不透明谓词、数据转换、白盒加密等；2）变量转换手段，包括：变量分割/合并、变量重新排列、数据流展平、堆栈随机化等；3）代码逻辑转换手段，包括：指令重新排序、指令替换、垃圾代码注入、虚拟化混淆、控制流展平等；4）代码抽象转换手段，包括：合并/拆分函数、删除注释、扰乱标识符名称、函数参数随机化等。

当前的多样化编译方法已经被集成到一些自动化工具中，obfuscator-llvm（OLLVM）[18]是由瑞士西北应用科技大学安全实验室于2010年6月份发起的项目，是一套针对LLVM的代码多样化编译工具。该工具支持控制流扁平化、指令替换、虚假控制流程等编译手段，兼容的语言和平台较广。Armariris[19]是由上海交通大学的GoSSIP小组设计的一种可以迭代更新的LLVM多样化编译框架，所实现的功能包括字符串加密、控制流扁平化、指令替换等。此工具支持x86、ARM等框架，并支持相关插件的开发，满足开发者的更多需求。Hikari[20]是由瑞士西北应用科技大学设计的一款轻便、易于移植的多样化编译工具，主要工作在编译过程的中间表示层面（Intermediate Representation level，IR level），并支持控制流扁平化、基本块拆分、指令集替换、字符串加密、函数调用等手段。Tigresss[21]是C语言的一种多样化的编译器，支持多种静态防御和动态逆向工程以及反虚拟化攻击的防御措施。支持控制流扁平化、虚拟化混淆、拆分合并函数等多样化编译手段。Multicompiler[22]工具的基本思想是基于LLVM构建的多个功能等效但内部不同的软件变体，已经验证的支持的测试平台包括：Firefox，Apache，Python

和LLVM。主要支持的技术包括：代码随机化、栈布局随机化和全局变量随机化。

3　基于DHR架构的软件多样化技术

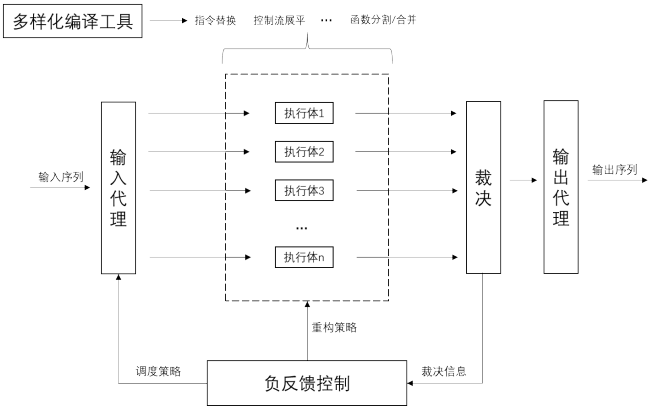


图1　本文结合DHR架构提出的软件多样化技术

本文所提出的安全模型在DHR架构基础上，使用多种多样化编译手段对执行体进行编译，从而满足了执行体的异构冗余特性，并且构建出一种高可用、高可信的软件多样化防御技术。图1为本文提出的安全技术的示意图，该架构包括输入输出代理、异构执行体、裁决和负反馈控制机制，相关工作流程如下：

1） 输入代理：输入代理将输入序列导入相应的执行体。

2） 输出代理：输出代理负责输出裁决机制的裁决信息，输出的信息也被称作输出序列。

3） 异构执行体：功能等价但实现方式不同的执行体。在本文模型中，可以使用第2节介绍的现有的多样化编译工具完成不同执行体的编译。每个执行体的编译手段可以是任何一种或者几种方法的组合，但是应该保证没有两个执行体的实现方法是完全一样的。

4） 裁决机制：拟态裁决机制不同于常规的多模冗余裁决机制。不只是使用相对正确公理做出多数或者少数、一样或者不一样的判别，而是使用丰富的裁决信息和策略降低相对错误概率，并且在攻防博弈环境中尽可能地隐匿目标对象的防御行为。

5） 负反馈控制：负反馈控制机制接受裁决信息，并根据控制通道给定的算法和参数或通过自身学习产生的控制策略，形成输入代理和可重构的执行体操作指令。给输入代理的指令用于将外部输入信号导向到指定的异构执行体，以便能够动态选择异构执行体的元素组成；给重构执行体的操作指令用于确定重构对象以及相关重构策略，从而形成闭环的负反馈控制系统。

DHR架构的安全性主要表现在三个方面：1）执行体集合自身抵抗攻击的能力：在成熟的可靠性理论中，异构冗余系统具有感知攻击造成的随机性故障的内生特性。2）裁决机制：从执行体的多模输出矢量中选择满足裁决算法的输出矢量，具有检测和容错的功能。3）负反馈控制对出错的执行体根据相关算法启动清洗恢复、替换迁移或重构重组。出错执行体在很大概率上可能是出现了随机性故障或者遭到蓄意攻击。本文的安全度量算法忽略了负反馈控制机制的安全因素，主要从执行体自身抵抗性及其历史裁决结果来评价执行体集合的安全性。

总之，本文提出的软件多样化防御手段以动态异构冗余形态的广义鲁棒控制架构为基础，构造出动态化、多样化的软件安全防御场景。

4　本文提出的安全性度量模型

本文提出的安全度量模型从执行体自身抵抗性及其历史裁决结果来评价执行体集合的安全性。

4.1和4.2节给出本文中使用到的相关参数定义和符号，4.3节从攻击抵抗性和历史抵抗性两个方面分析单个执行体的安全性度量，在4.4节分析了执行体集合的几种安全情形。

4.1　参数定义

定义1：执行体集合A：$A=(a_1,a_2,...,a_n)$，其

中n为执行体集合的数目。

定义2：不同抵抗性的影响力因子γ（简称为影响力因子）：表示攻击抵抗性和历史抵抗性两个指标在单个执行体安全性定义中所占比重。

定义3：裁决周期T：两次裁决过程的时间间隔。

定义4：单个执行体的攻击抵抗性$R_a^i(t)$：表示执行体a_i没有被攻击成功的概率。

定义5：反编译攻击时间$T_a{}^i(t)$：表示反编译工具对执行体a_i攻击成功所需的时间。

定义6：单个执行体的历史抵抗性$R_h^i(t)$：表示执行体a_i裁决结果的可信程度。

下表列举出了本文出现的符号及其含义

表1：本文出现的符号及其含义。

符号	含义
A	执行体集合
a_i	执行体集合中第i个执行体
n	执行体集合中执行体数目
γ	不同抵抗性的影响力因子(简称为影响力因子)
λ_i	执行体a_i的失效率
p_j^i	执行体a_i的第j次裁决结果
T	两次裁决过程的时间间隔
AT	执行体被攻击成功的时刻
R_a^i	执行体a_i没有被攻击成功的概率
T^i	反编译工具对执行体a_i攻击成功所需的时间
R_h^i	执行体a_i裁决结果的可信程度
R^i	执行体a_i的安全性
R_{si}	第i种安全场景下系统的安全性

4.3 单个执行体的安全性度量

多样化编译后，执行体的代码功能保持不变，但是代码的复杂性会增加，从而会增加攻击者使用反编译工具实施逆向工程的难度。执行反编译相关攻击手段的时间可以表示执行体的攻击抵抗性[23]。

在DHR架构的裁决机制中，如果某个执行体的输出结果与最终裁决结果一致，可以认为该执行体的输出结果有可信度。经过多次裁决，某个执行体的历史可信度可以作为衡量其抵抗性的一类指标。

因此，本文认为某个执行体a_i的安全性表示为攻击抵抗性和历史抵抗性的加权求和，根据定义4和定义6中攻击抵抗性和历史抵抗性的相关概念，执行体a_i的安全性可以表示为：

$$R^i(t) = \gamma R_a^i(t) + (1 - \gamma) R_h^i(t), 0 \leqslant \gamma \leqslant 1。 \quad (1)$$

其中γ为不同抵抗性的影响力因子（简称为影响力因子）。

下面对攻击抵抗性和历史抵抗性指标进行具体说明：

4.3.1 单个执行体的攻击抵抗性度量

对于编译后的代码来说，复杂性强的代码会使攻击者在执行逆向工程时付出大量的时间和精力，会直接决定此次攻击是否成功。执行反编译的攻击需要付出的时间往往可以表示多样化方法的效果和可靠性。例如在文献[24]中，使用执行逆向工程的时间来表示执行体抵抗攻击的能力。据此，对于执行体a_i，攻击抵抗性的表示如下：

$$R_a^i(t) = T^i(t) \quad (2)$$

式（2）表明单个执行体的攻击抵抗性等于攻击者利用反编译工具执行逆向工程的时间，如果执行攻击所需时间越长，则表明该执行体较难被攻击，其攻击抵抗性越强；反之，该执行体的攻击抵抗性越弱。

本文认为攻击经验是可以累积的，攻击者在执行攻击中会根据情况调整攻击策略，包括所使用的反编译工具的种类等。本文假设还未攻击时，对执行体攻击成功的概率为0；当攻击次数无穷大时，对同一个执行体的攻击是肯定可以成功的。因此可以认为执行体的攻击抵抗性随攻击时间t的增加而减小，反编译攻击执行的时间也随着t增加而变小。在本文中有：

$$R_a^i(t) = T_a{}^i(t) = e^{-\lambda_i t}, 0 < \lambda_i < 1 \quad (3)$$

其中λ_i为执行体a_i的失效率。式（3）表示执行体a_i的抵抗攻击性和攻击成功所需要的时间都服从参数为λ_i的指数分布。

下面对失效率λ_i的概念做以下说明：

在连续时间内，攻击可靠性$R_a^i(t)$表示在时刻t之前执行体未被攻击的概率，即：

$$R_a^i(t) = P(AT > t) \quad (4)$$

其中AT表示该执行体被攻击成功的时刻。

失效率λ_i表示t时刻未被攻击成功的执行体，在单位时间Δt内被攻击成功的概率。根据概率论中分布律的有关概念，在$(t, t + \Delta t)$内被攻击成功的概率可以表示为：

$$p(t < AT \leqslant t + \Delta t | AT > t) \quad (5)$$

因此Δt时间内平均失效率可以表示为：

$$\frac{p(t < AT \leqslant t + \Delta t | AT > t)}{\Delta t} \tag{6}$$

当$\Delta t \to 0$，可以得到t时刻的失效率λ_i和抵抗性$R_a^i(t)$之间的关系为：

$$\begin{aligned}\lambda_i(t) &= \lim_{\Delta t \to 0} \frac{p(t < AT \leqslant t + \Delta t | AT > t)}{\Delta t} = \lim_{\Delta t \to 0} \frac{p(t < AT \leqslant t + \Delta t, AT > t)}{P(AT > t)\Delta t} \\ &= \lim_{\Delta t \to 0} \frac{p(t < AT \leqslant t + \Delta t)}{P(AT > t)\Delta t} = \lim_{\Delta t \to 0} \frac{(1 - R_i(t + \Delta t)) - (1 - R_i(t))}{R_i(t)\Delta t} \\ &= \frac{-R_i'(t)}{R_i(t)}\end{aligned} \tag{7}$$

4.3.2 单个执行体的历史抵抗性度量

执行体的历史裁决信息可以表示该执行体输出结果的可信度，对拟态架构的裁决策略也有重要的应用意义。根据不同执行体的历史裁决信息可以识别出“较易”输出错误的执行体，从而更好的帮助DHR架构更快的执行安全降级和失效停止、提高系统的安全性和可靠性。

本文通过记录执行体裁决结果的历史信息来表示执行体输出结果正确或者错误的概率，并使用历史抵抗性指标来表示执行体的这种性质，历史记录较差的执行体被认为是在系统运行中较容易出错的执行体，相应执行体的历史抵抗性也较差；反之，历史抵抗性越强。

本文使用下面的两个步骤构建执行体裁决结果的历史信息：

1） 对于执行体集合$A = (a_1, a_2, ..., a_n)$，第j次裁决对应的历史裁决信息集合为$P = (p_j^1, p_j^2, ...p_j^n)$，如果在该次裁决中执行体$a_i$输出结果与最终裁决模块输出结果相同，则表明此执行体通过了裁决结果， 则这个输出对应的执行体历史信息值$p_j^i = 1$，否则$p_j^i = 0$。

2） 对于执行体a_i，累计历史信息值并作归一化处理，假设裁决周期为T，经过时间t后，执行体a_i的历史抵抗性可以表示为：

$$R_h^i(t) = \frac{1}{t/T} \sum_{j=1}^{t/T} p_j^i \tag{8}$$

在本文中，如果执行体的输出结果与裁决机制输出结果相同，可以认为该次裁决结果具有可信性，但是该结论在共模逃逸[10]的特殊情况下可能是错误的。在多模裁决情况下，裁决单元会将执行体输出的多数值作为最终结果输出，如果此时恰好多数执行体被攻击成功并且输出结果一致，则会导致裁决机制无法屏蔽错误的输出结果，造成攻击逃逸的现象。按照式（8）的定义方法，共模逃逸情况下可能会增加被攻击成功的执行体被信任程度，这种情况是应该被避免的。但是本文在建立模型时忽略了该种情况的发生，因为在实际工程中，共模逃逸情况发生的概率非常小，大概为5.5×10^{-14}[10]。

4.4 执行体集合的安全性度量

本文考虑m/n裁决策略的CMD系统，对于有n个执行体的DHR架构，至少有m个执行体不被扰动并输出正确结果时，可以认为CMD系统可以抵抗此次攻击情况。

下面是几种m/n裁决策略的系统的安全场景：

安全场景1：m个执行体输出正确

场景1的安全性概率可以表示为：

$$R_{s1}(t) = \sum_{i=1}^{C_n^m} \prod_{a_j \in correct_i} R^j(t) \prod_{a_k \in A - correct_i} (1 - R^k(t)) \tag{9}$$

其中$correct_i$集合表示在第i中情况下，输出正确的执行体集合，A-$correct_i$表示输出错误的执行体集合，$correct$集合总共有C_n^m种情况。式（9）表示C_n^m种情况下的概率之和，其中每种情况下的概率都等于执行体输出正确或者错误的概率乘积。

安全场景2：m+1个执行体输出正确

如式（9），此种安全情况下的执行体集合安全性可以表示为：

$$R_{s2}(t) = \sum_{i=1}^{C_n^{m+1}} \prod_{a_j \in correct_i} R^j(t) \prod_{a_k \in A - correct_i} (1 - R^k(t)) \tag{10}$$

与式（9）相似，式（10）可以看出在该种场景下，$correct$集合总共有C_n^{m+1}种情况。

安全场景n-m+1：n个执行体输出正确

$$R_{sn}(t) = \prod_{i=1}^{n} R^i(t) \tag{11}$$

式（11）表示在该种场景下没有执行体被攻击成功，此时系统的安全性等于所有执行体的安全性的乘积。

所以m/n裁决策略执行体集合的安全性可以表

示为几种场景抵抗性概率的总和：

$$R(t) = R_{s1}(t) + R_{s2}(t) + ... + R_{sn}(t) \quad (12)$$

在本文实验中研究下面考虑两个特殊安全场景：

Case1： n个执行体输出正确，即没有执行体被攻击成功，如式（11），即：

$$R_{sn}(t) = \prod_{i=1}^{n} R^i(t)$$

Case2： n-1个执行体输出正确，即只有一个执行体被攻击成功。

$$R_{sn-1}(t) = (1 - R^1(t)) \prod_{i=2}^{n} R^i(t) + R^1(t)(1 - R^2(t)) \prod_{i=3}^{n} R^i(t) + ... + \prod_{i=1}^{n-1} R^i(t)(1 - R^n(t)) \quad (13)$$

5 实验

实验目的：通过仿真实验模拟出单个执行体的攻击抵抗性和历史抵抗性两个指标根据时间变化的规律，并分析攻击时间和变体数目等参数对执行体集合安全性的影响。5.1节对本文用到的数据集进行说明，5.2节-5.4节分析实验结果。

5.1 数据集

1） 执行体数目n：假设执行体集合中的执行体数目取值范围为3-5个：$n = 3 \sim 5$。

2） 攻击时间t：考虑10个单位时间内执行体相关安全性指标的变化规律。

3） 失效率λ_i：随机生成每个执行体的失效率λ_i，并保证没有两个执行体的失效率是相同的。

4） 裁决次数t/T：假设每个单位时间内执行一次裁决过程，即$T = 1$，并考虑在攻击时间t内每个执行体的历史抵抗性。

5.2 单个执行体的攻击抵抗性度量

首先随机生成的5个执行体的失效率，根据式（3）可以表示出执行体的攻击抵抗性随时间的变化规律。本次实验中，5个执行体的失效率分别为：$\lambda = (0.56, 0.62, 0.52, 0.48, 0.45)$，图2表示在攻击时间t内5个执行体的攻击抵抗性的变化规律。

从图中可以看到， 5个执行体的攻击抵抗性随攻击时间的增加而减小。但是因为失效率的不同，攻击抵抗性减小的程度不同。

5.3 单个执行体的历史抵抗性度量

在生成执行体裁决结果时，本文假设在单个时间内有一次裁决过程，即$T = 1$。参考5.2节设

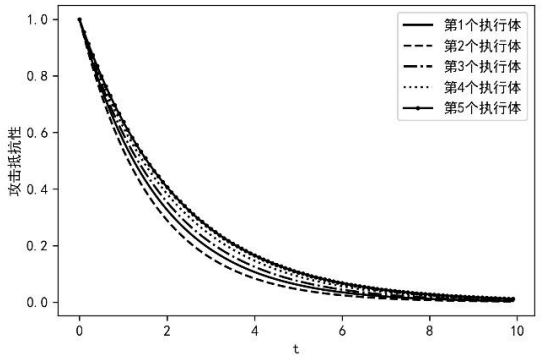


图2 5个执行体的攻击抵抗性变化规律

置的执行体失效率，以每个执行体的失效率相应生成其裁决结果，即第i个执行体以λ_i的概率裁决错误。表2为第1个执行体10次裁决结果示意图。将5个执行体的历史裁决结果进行记录，并根据式（8）计算其历史抵抗性。图3为5个执行体在10次裁决过程中历史抵抗性的变化情况。

表2 执行体a_1的10次裁决结果记录

裁决次数	1	2	3	4	5	6	7	8	9	10
裁决结果	√	√	×	×	×	×	×	√	×	×

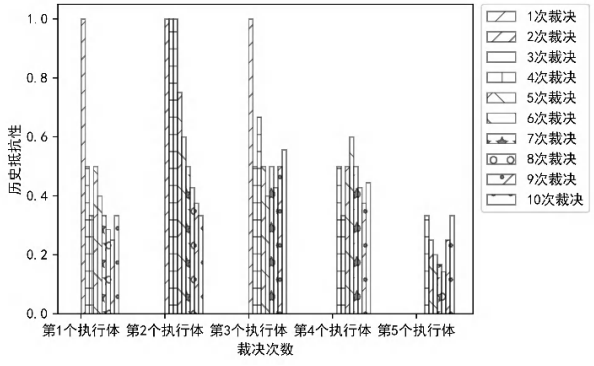


图3 5个执行体的历史抵抗性变化规律

5.4 影响力因子γ对执行体集合安全性的影响。

根据上文分析可得，攻击抵抗性和历史抵抗性是与攻击时间相关的特性。受随机因素的影响，在初始时刻，根据“相对正确”公理，由于没有或者有极少的攻击行为，有理由相信执行体的抵抗攻击能力较强，因此在式（4）中，R_a^i所占权重应该较大；反之，历史抵抗性R_h^i是由0次或者少量裁决结果决定的，相比较而言缺少可信度，因此其所占权重应该较小。并且随着攻击时间的增加，

攻击抵抗性所占权重应逐渐减小；历史抵抗性所占权重应相应逐渐增加。

本文假设在初始时刻，执行体集合的安全性等于攻击抵抗值R_a^i，即$\gamma = 1$。本次实验中假设γ的值满足sigmoid函数的变化趋势。Sigmoid函数的表达式为：

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$

图4为该函数图像。sigmoid函数特性是在固定的区间段之间单调递增输出，并且越接近于区间上下极限时变化越缓慢，越远离区间极限时变化越显著。

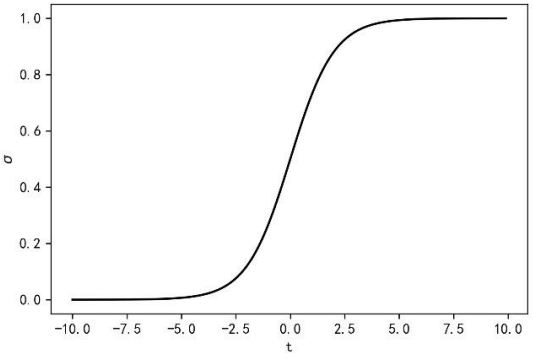


图4　sigmoid函数

由于γ函数满足单调递减的变化趋势，因此γ随时间的变化可用下面的公式进行表示：

$$\gamma(t) = \frac{1}{1 + e^{t}}$$

γ和1-γ的值随时间变化的 情况如图5所示：

根据式（11）和（12），本次实验分析了case1和case2情形下执行体集合安全性的变化趋势。图

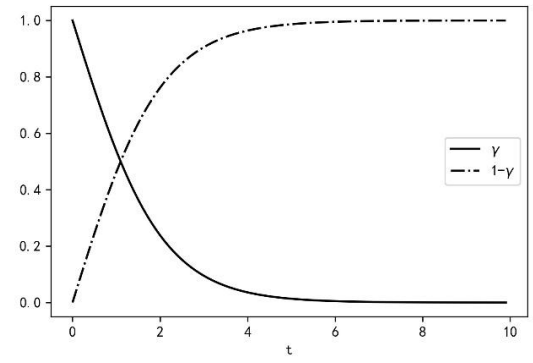


图5　γ和1-γ随时间变化的图像

6表示case1（没有执行体被攻击成功）情形下执行体集合安全性变化情况，可以看出系统的安全性随着攻击时间的增加而降低；并且随着执行体数目的减少，安全性也会随之降低。从图7可以看出case2（单个执行体被攻击成功）情形下执行体集合安全性的变化情况与case1类似，安全性会随着攻击时间增加和执行体数目的减少而降低。

从图6和图7可以得出结论，为了保证执行体集合的安全性，应该减少执行体的上线时间并增加执行体数目。在工程实践中，在成本允许情况下，可以适当增加执行体数目，并且对执行体根据特定算法实施清洗、上线等操作

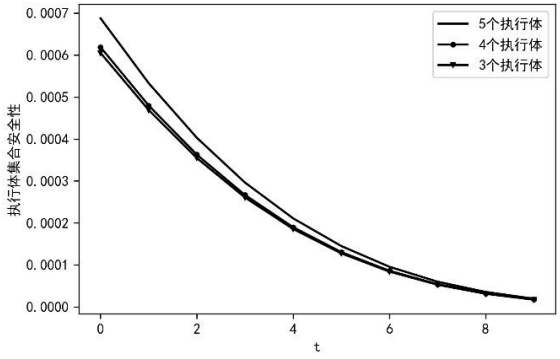


图6　case1情形下执行体集合安全性变化规律

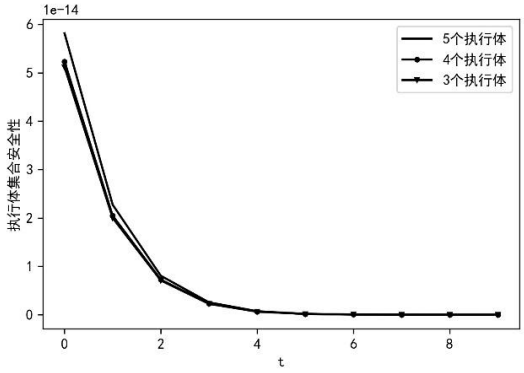


图7　case2情形下执行体集合安全性变化规律

6　相关工作

6.1　软件多样化技术安全性度量手段

Collberg[7]首次提出了用于测评多样化编译技术的相关指标， Potency用于评价编译后的代码相比原始代码的复杂程度，resilience用于评价编译后代码使反编译器逆向分析的难度。Ceccato[8]使

用平均值重新定义potency的计算方法，首先在不同程序中得到该指标的一组值，最终将该组值的平均数作为potency的最终表示值。Udupa [9] 提出使用原始代码和反编译代码之间控制流程图的距离来描述Resilience指标，并且讨论了在静态分析的情形下，多样化编译技术对于代码复杂性的提升情况。Heffner 和 Collberg [25] 提出了一种面向编译过程的优化器：Obfuscation Executive （OE）。通过使用potency指标对多样化编译手段进行测评，选择使用结果最优的编译方式。Karnick 等人 [26] 从嵌套复杂度、控制流复杂度、变量复杂度， 和程序长度这四个方面去评价多样化编译后代码的potency指标。Anckaert等人 [27] 从指令、控制流，数据流和数据四个方面去评价resilience指标。Wu等人 [28] 使用线性回归的模型来分析和预测混淆转换的potency指标。文章 [23] 利用反编译工具执行攻击时需要的时间来表示被多样化编译后的程序抵抗符号化攻击的难易程度。

6.2 CMD技术安全性度量手段

现阶段关于CMD技术安全性的度量主要分为两种研究方向，第一种方向是计算执行体异构性或者抗攻击能力 [11] [12] [29-31]。文章 [11] [12] 通过不同组件之间的相似性或者执行体相似的漏洞集来定义执行体间的异构性；文章 [11] [29-31] 通过定义攻击序列或者攻击过程计算不同攻击情况下执行体抵抗攻击的概率。安全性研究的第二种方向是裁决机制输出结果的准确性，裁决单元将执行体的历史裁决信息作为参考依据，从而较好的规避出错执行体对裁决模块输出结果的影响，相关的算法包括：自适应一致裁决算法 [32]，基于模糊聚类的表决容错算法 [33] 和基于自检测的多数一致表决算法 [34] 等。

现有理论通过随机Petri网（General Stochastic Petri Net，GSPN）模型建立拟态防御架构的抗攻击性模型，并利用连续时间马尔可夫链（Continuous-Time Markov Chain，CTMC）来量化分析拟态系统的稳态概率，进而得出拟态系统抵抗性的一种表示方法 [10] [35]。但是马尔科夫链模型并没有考虑到前序攻击状态对现阶段的攻击情况的影响结果。该模型可以用于系统级别的宏观分析，难以刻画拟态防御架构中真实的攻击场景和抵抗场景。

7 总结

针对软件多样化技术中安全性度量问题，本文提出了一种合理的解决方法。该方法综合考虑执行体自身的攻击抵抗性及其历史裁决信息，并对执行体集合的几种安全情形进行了分析。在实验部分验证了该算法的可行性和合理性。但是本文未考虑拟态防御架构中负反馈环节及调度策略带给系统的安全影响，需要在后续研究中进一步完善该工作。

参考文献:

[1] UdupaS. K., DebrayS. K., and MadouM.，“Deobfuscation：reverse engineering obfuscated code," in Proc. of the 12th IEEE Working Conference on Reverse Engineering，2005.

[2] ChandrasekharanS. and DebrayS.，“Deobfuscation：improving reverse engineering of obfuscated code," 2005.

[3] GuillotY. and GazetA.，“Automatic binary deobfuscation，”Journal in Computer Virology，2010.

[4] CooganK., LuG., and DebrayS.， “Deobfuscation of virtualizationobfuscated software：a semantics-based approach," in Proc. of the 18th ACM Conference on Computer and Communications Security，2011.

[5] YadegariB., JohannesmeyerB., WhitelyB., and DebrayS.，“A generic approach to automatic deobfuscation of executable code，”in Proc. Of the 2015 IEEE Symposium on Security and Privacy，2015.

[6] BichselB., RaychevV., TsankovP., and VechevM.，“Statistical deobfuscation of android applications，”in Proc. of the ACM SIGSAC Conference on Computer and Communications Security，2016.

[7] COLLBERG C., THOMBORSON C., AND LOW, D. A taxonomy of obfuscating transformations. Tech. rep., Department of Computer Science, The University of Auckland, New Zealand, 1997.

[8] CECCATO M., CAPILUPPI A., FALCARIN P., BOLDYREFF, C. AND A large study on the effect of code obfuscation on the quality of java code. Empirical Software Engineering 20，6（2015），1486－1524.

[9] UDUPA S., DEBRAY S., MADOU, M. AND Deobfuscation: reverse engineering obfuscated code. In 12th Working Conference on Reverse Engineering (2005).

[10] Wu J X. Cyberspace mimic defense［M］. Springer International Publishing，2020.

[11] 张杰鑫, 庞建民, 张铮．拟态构造的Web服务器异构性量化方法［J］. 软件学报，2020(2).

[12] 刘勤让, 林森杰, 顾泽宇．面向拟态安全防御的异构功能等价体调度算法[J]. 通信学报，2018，v. 39;No. 373(07)：192-202.

[13] PAX Team, "PAX Address Space Layout Randomization，”https：//pax. grsecurity. net/docs/aslr. txt.

[14] CowanC., PuC., MaierD., HintonyH., WalpoleJ., BakkeP.,

BeattleS. ,GrierA. , WagleP. , and ZhangQ. ，“StackGuard：automatic adaptive detection and prevention of buffer-overflow attacks, " in USENIX SEC，1998.

[15] KcG. S. , KeromytisA. D. , and PrevelakisV. . Countering code-injection attacks with instruction-set randomization. In Proceedings of the 10th ACM Conference on Computer and Communications Security，CCS '03，pages 272 – 280，2003.

[16] HiserJ. , Nguyen-TuongA. , CoM. , HallM. , and DavidsonJ. W. . ILR: Where'd my gadgets go? In Proceedings of the 33rd IEEE Symposium on Security and Privacy，S&P '1 pages 571 – 585，2012.

[17] Banescu S, Pretschner A. A tutorial on software obfuscation［M］// Advances in Computers. Elsevier，2018，108：283-353.

[18] https://github. com/obfuscator-llvm/obfuscator/wiki/Installation.

[19] https://github. com/GoSSIP-SJTU/Armariris.

[20] https://github. com/HikariObfuscator/Hikari/wiki/Usage.

[21] https://tigress. wtf/.

[22] https://github. com/securesystemslab/multicompiler.

[23] Banescu S, Collberg C, Pretschner A. Predicting the resilience of obfuscated code against symbolic execution attacks via machine learning［C］//26th {USENIX} Security Symposium ({USENIX} Security 17). 2017：661-678.

[24] BANESCU S. , COLLBERG C. , GANESH V. , NEWSHAM Z. , PRETSCHNER, A. AND Code obfuscation against symbolic execution attacks. In Proc. of 2016 Annual Computer Security Applications Conference (2016), ACM.

[25] HEFFNER K. , COLLBERGAND, The obfuscation executiveC. . In International Conference on Information Security (2004)，Springer，pp. 428 – 440.

[26] KARNICK M. , MACBRIDE J. , MCGINNIS S. , TANG Y. , RAMACHANDRAN, R. AND A qualitative analysis of java obfuscation. In proceedings of 10th IASTED international conference on software engineering and applications, Dallas TX, USA（2006）.

[27] ANCKAERT B. , MADOU M. , DE SUTTER B. , DE BUSB. , DE BOSSCHERE K. , PRENEEL, B. AND Program obfuscation: a quantitative approach. In Proc. of the ACM workshop on Quality of protection (2007)，ACM，pp. 15 – 20.

[28] WU Y. , FANG H. , WANG S. , QI, Z. AND A framework for measuring the security of obfuscated software. In Proc. of 2010 International Conference on Test and Measurement (2010).

[29] 王 伟,曾俊杰,李光松,等. 动态异构冗余系统的安全性分析[J]. 计算机工程,2018,44(10)：42-45,50.

[30] 李卫超, 张铮, 王立群,等. 基于拟态防御架构的多余度裁决建模与风险分析[J]. 信息安全学报，2018，3(05)：64-74.

[31] 顾泽宇, 张兴明, 林森杰. 基于安全策略的负载感知动态调度机制[J]. 计算机应用，2017，37(011)：3304-3310.

[32] 欧阳城添, 王曦, 郑剑. 自适应一致裁决算法[J]. 计算机科学，2011(07)：130-133.

[33] 王平. 基于动态模糊聚类的输出数据一致表决容错方法[C]// 2007'仪表，自动化及先进集成技术大会论文集(二). 2007.

[34] 周海涛, 朱纪洪. 基于自检测的多数一致表决算法[J]. 清华大学学报: 自然科学版，2005，45(4)：488-491.

[35] 任权, 贺磊, 邬江兴. 基于离散马尔可夫链的不同抗干扰系统模型分析%Analysis of different anti-interference system models based on discrete time Markov chain[J]. 网络与信息安全学报，2018，004(004)：30-37.

[36] 马博林,张铮,等. 基于指令集随机化的抗代码注入攻击方法[J]. 信息安全学报，2020，5(04)：30-43.

基于硬件加速的万兆UDP/IP协议栈设计与实现

董永吉
解放军战略支援部队信息工程大学，河南郑州 450002

摘　要：针对传统的基于软件协议栈无法满足高速数据传输处理的需求，本文提出了一种基于FPGA实现的UDP/IP协议栈设计方案，解决了高速数据传输实时性差的问题，通过实际测试表明，该设计最高可以达到9.32Gbps传输速率，满足10Gbps带宽下线速处理的需求，与传统软件实现相比，处理能力更接近理论极限。
关键词：FPGA、万兆以太网、硬件加速、UDP协议

Design and Implementation of 10G Ethernet UDP/IP Protocol Stack Based on Hardware

DONG Yong-ji
PLA Strategic support force information engineering university,450002, China

Abstract: Aiming at the problem that the traditional software protocol stack can not meet the performance requirement of high-speed data transmission, this paper presents a design scheme of UDP protocol stack based on hardware acceleration. The design uses FPGA to realize UDP/IP protocol stack, and solves the problem of poor real-time performance of high-speed data transmission. The actual test shows that the design can reach the highest level. To 9. 32 Gbps transmission rate, it meets the need of 10 Gbps bandwidth downlink speed processing. Compared with traditional software implementation, the processing capacity is closer to the theoretical limit.
Key words: FPGA; 10G Ethernet; Hardware Accelerated; UDP Protocol

0　引言

随着网络技术的飞速发展、网络带宽的迅速提升，数以万计的视频、图像等业务数据逐渐成为高速链路传输的主要组成，面对海量数据的传输压力，传统操作系统内置协议栈或以软件为核心的协议栈加速技术已经不能满足高速、低延迟传输的需求。

根据网络处理经验，1Gbps的以太网传输需要1GHz的处理器频率，面对万兆高速网络的传输需求，大量CPU计算资源都被将消耗网络协议的处理上。基于硬件实现的UDP/IP协议栈，因通过硬件固化协议栈的处理逻辑，进而能提供高吞吐率、低延迟、高带宽的传输性能[1]。

鉴于FPGA兼备灵活可配置和高速并行处理的特性，尤其适合在高速传输领域固化协议栈来提升系统的处理能力。候义合[2]等人提出了一种基于FPGA实现MAC及UDP/IP协议栈的方法，旨在解决千兆以太网高速传输的问题；杨阳[3]等设计了一种基于FPGA的万兆光纤以太网的高速传输方法；崔鹤[4]等利用硬件实现了UDP/IP协议数据的封装和拆封；Francesc Moll Echeto[5]面向DAQ应用，基于INTEL FPGA实现了一个通用的千兆UDP/IP协议核；Nianyun Liu[6]等人提出了一个适用于大型传感器网络的UDP/IP协议栈，比软件处理提升了8倍的性能。面对万兆应用场景，王禹衡[7]设计了一个10G以太网UDP/IP处理器视频传输接口；夏杨[8]设计了一种万兆以太网分发平台，满足万兆以太网UDP传输的需求，而国外厂商PLDA[9]、Fraunhofer HHI[10]、Intilop[11]以及Dini Group[12]等公司也分别提出价格不菲的商用IP核。

综上可知，当前硬件协议栈研究主要集中在

基金项目：国家重点研发计划项目(2019YFB1802502）

千兆以太网环境，而万兆以太网的硬件协议栈是当前的应用热点。本文基于FPGA实现的UDP/IP协议栈为面向10G网络环境设计，具有以下优势：1.集协议帧的解析、处理、发送和接收于一体，大幅提升系统UDP应用的传输效能。2.基于FPGA可重构可配置的特点，可以部署于不同场景的网络环境中；3.面向服务器应用设计，可支持多端口多点业务的并发连接。

2 硬件结构设计思路

为了提高UDP协议的传输效能，本文提出一种基于FPGA的UDP协议栈设计方案，该设计基于斯坦福大学的NetFPGA-10G [13] 板卡实现，NetFPGA-10G是一个有4个10Gb/s的PCI Express适配卡，拥有一个Virtex-5XC5VTX240T-2 FPGA主处理芯片，能够尽可能多地支持各种应用。本方案利用该板卡PCI Express接口的便携扩展能力强的特点，方便在服务器应用场景中通过扩展物理接口的方式，部署该硬件协议栈到服务器系统中，提升服务器中各种应用UDP协议的加速传输处理。

FPGA作为硬件子系统设计中的重点和难点，根据FPGA硬件结构特点实现了协议处理功能，并将硬件部分划分为10GMAC接口模块、协议解析模块、ARP响应模块、ICMP响应模块、协议封装模块、UDP协议处理模块、PCIE-DMA模块，模块划分如图1所示：

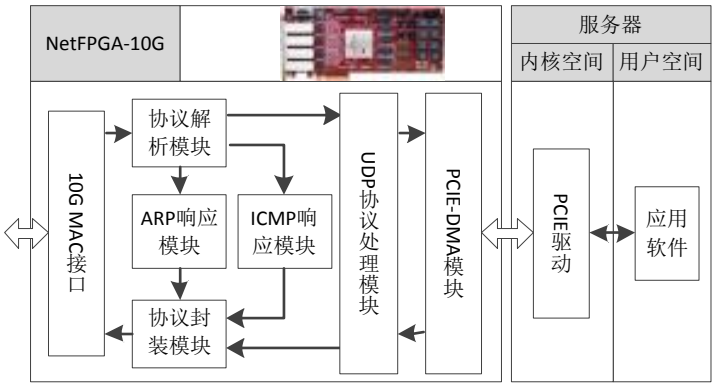


图1 系统结构组成框图

（1）协议解析模块

该模块主要实现数据报文的协议解析处理，并为后续响应报文的生成和封装提取用户的相关信息。首先该模块对到达系统的数据报文协议逐层进行解析，最终实现对本地支持的UDP协议请求协议报文的解析、本地协议报文的提取以及无关报文的过滤三部分功能，鉴于FPGA硬件结构的特殊性，无法完整实现所有协议的解析和处理，针对于设计的应用场景，实现的具体处理算法流程如下图所示：

具体处理算法步骤如下表所示：

（2）ARP响应模块

该模块实现ARP请求、响应报文的构建以及ARP缓存表的维护功能。基于FPGA构建该模块时，采用FPGA片内BLOCK RAM存储ARP表项，鉴于硬件资源的有限性，故在标准的ARP协议处理的基础上，该模块增加了硬件资源这一限制条件，即当BLOCK RAM中存储的ARP表即将溢出时，会选择性删除老化时间最长的表项，进而维持整个ARP表的正常操作。

首先根据ARP请求报文信息，构建ARP响应报文；其次响应协议封装模块请求，提供IP-MAC映射服务，并定时刷新ARP缓存表，根据老化时间删除过期表项；再次根据需求，对于缓存表中未存在的IP-MAC映射关系，主动组织ARP请求报文内容。ARP缓存表维护的算法流程如下图所示：

（3） ICMP协议响应模块

该模块实现了ICMP协议的一个子集。每当收到ICMP请求报文之后，根据当前系统的状态，按照ICMP协议规定，对应生成相应的响应报文。本文设计的UDP协议栈目标部署在服务器端，故根据服务终端的协议特性，实现了包括“回显应答”、“超时”和“目标不可达”三类响应功能。其中“回显应答”用于支持客户端的PING信息，“超时”用于支持数据段TTL过期，“目标不可达”

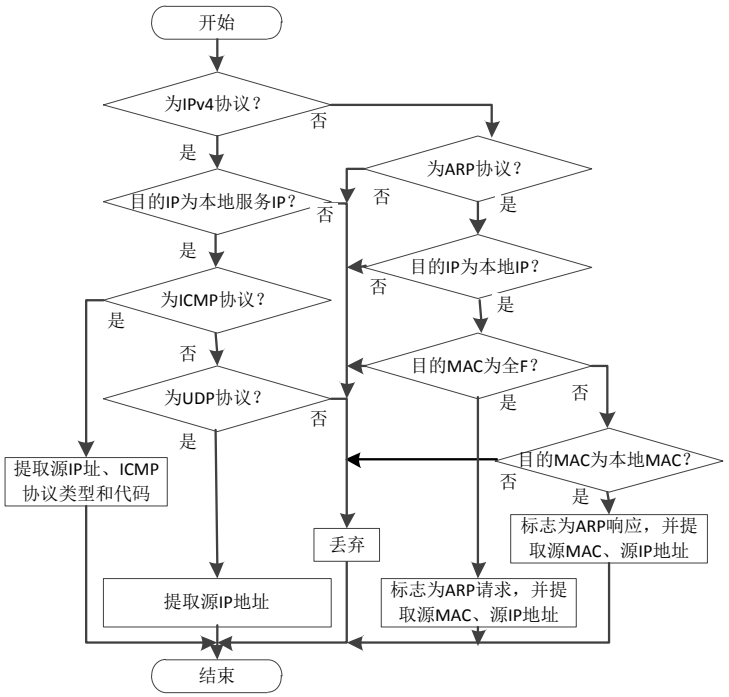


图 2　协议处理流程图

表 1　协议处理步骤

协议处理步骤如下：
step 1)：判断到达以太网帧的以太类型是否为0x0800，如果为0x0800（IPv4协议），则执行step 2，否则执行step7；
step 2)：判断到达的IPv4协议报文中目的IP字段是否匹配本地配置的多个服务IP，如果命中本地多个服务IP中的一个，则执行step 3，否则丢弃该报文；
step 3)：判断到达的IPv4协议报文头中协议字段是否为ICMP，如果为ICMP协议，则执行step 4，否则执行step 6；
step 4)：提取到达的ICMP协议报文中源IP地址和ICMP协议号，用于后续ICMP响应报文的生成，并跳转结束协议解析处理；
step 5)：判断到达的IPv4协议报文头中协议字段是否为UDP，如果为UDP协议，则执行step 6，否则丢弃该报文；
step 6)：提取到达的UDP协议报文中源IP地址，用于后续UDP响应报文的封装，并跳转结束协议解析处理；
step 7)：判断到达以太网帧的以太类型是否为0x806，如果为0x0806（ARP协议），则执行step 8，否则丢弃该报文；
step 8)：判断到达的ARP协议报文中目的IP字段是否匹配本地配置的多个服务IP，如果命中本地多个服务IP中的一个，则执行step 9，否则丢弃该报文；
step 9)：判断到达数据帧的目的MAC地址是否为全F（广播地址），如果为全F，则执行step 10，否则执行step 12；
step 10)：判断到达数据帧的目的MAC地址是否为本地MAC，如果是本地MAC地址，则执行step 11，否则丢弃该报文；
step 11)：提取到达的ARP协议报文中源MAC地址和源IP地址，用于构建ARP缓存表以及后续APR响应报文的重组，并跳转结束协议解析处理；
step 12)：提取到达的ARP协议报文中源MAC地址和源IP地址，用于构建ARP缓存表以及后续ARP响应报文的生成，并跳转结束协议解析处理。

用于指示服务器端未开放的端口。

（4） 协议封装模块

该模块主要实现两个功能，其一是对UDP数据报进行协议封装；其二是对封装后的UDP协议数据帧、ICMP协议数据帧和ARP协议数据帧进行数据合流，统一发送到10G MAC接口。其中针对UDP报进行IP协议的封装时，需要根据用户的IP地址（目的IP地址）信息在ARP响应模块的ARP缓存表中匹配查找表项，并获取该IP地址对应MAC地址信息，进而将该IP包封装成数据帧，协议封装模块的结构图如下所示：

软件应用运行在服务端，软件功能在提供服务之前，需要通过配置端口向硬件协议栈申请/释放服务端口，硬件模块根据服务端口的状态，决定提供或是拒绝服务。

当该模块接收到来自线路的数据时，若通信

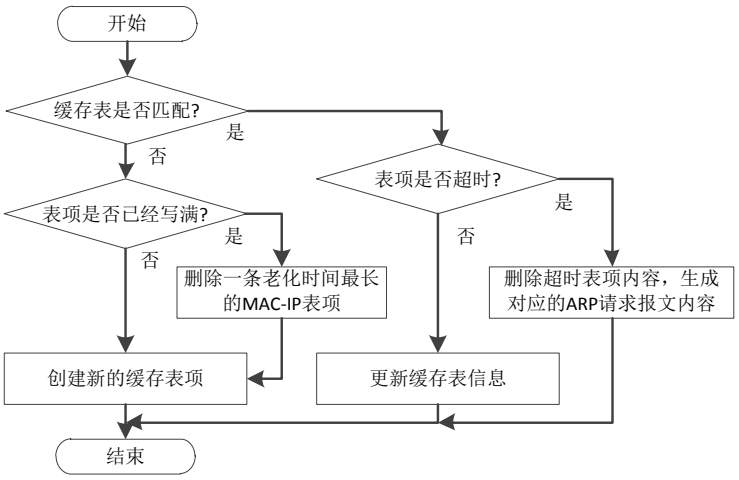
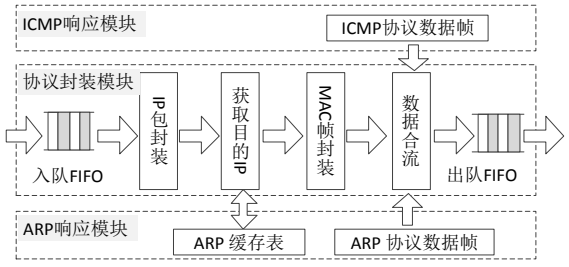


图 3　ARP 缓存表维护算法流程图

表 2　ARP 缓存表维护步骤

step 1)：判断使用待发送数据包的目的 IP 地址与缓存表空间中的表项进行匹配，如果匹配，执行 step 2，否则执行 step 3；
step 2)：检查表项空间中的表项地址上是否已经全部建立了缓存表项（表项空间是否已经被使用完毕），如果是，执行 step 6，否则执行 step 7；
step 3)：依据缓存表中存储的时间戳与当前时刻进行对比，判断待匹配的缓存表项是否已经超时，如果是，执行 step 4，否则执行 step 5；
step 4)：删除超时的缓存表项信息，并生成对应的 ARP 请求报文内容；
step 5)：更新缓存表项的时间戳信息；
step 6)：删除缓存表空间中的一条老化时间最长的表项；
step 7)：在表项空间中未被占用的地址上创建新的缓存表项；



（5）　UDP 协议处理模块

该模块主要实现 UDP 协议的封装和拆装，以及 UDP 校验和的计算，以及与应用软件的套接字接口和数据通道。UDP 协议处理模块与应用软件接口如下图所示：

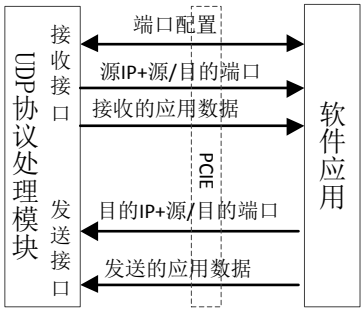


图 5　UDP 协议处理模块应用接口

的目的端口为开放状态，硬件则进行 UDP 协议的拆装，提取通信套接字和接收数据；若通信的目的端口为关闭状态，硬件则丢弃该 UDP 数据段。

当该模块接收到来自软件的数据时，若通信的源端口为开放状态，硬件则根据软件提供的套接字和应用数据，封装成 UDP 数据段；若通信的源端口为关闭状态，硬件则丢弃该应用数据。

为了实现服务器场景下对众多 UDP 端口状态的管理，该模块内部采用 BRAM 建立了一个地址为 16bit 位宽的端口状态表，软件系统通过申请/释放操作将该查找表中对应端口号设置为开放/关闭，状态表中共有 2^{16} 个地址，可以将 UDP 所有的端口号一一映射到该表中，且每个表项内有 1 bit 指示位用于标志该端口号是否开放，如下图所示，53 端口（DNS 协议）状态开放。

（6）PCIe-DMA 模块

PCIe-DMA 模块主要包括 Xilinx PCIe IP 核和 DMA 模块。PCIe IP 核直接在 Vivado 中调用，用来实现 PCIe 协议的物理层和数据链路层。如图 5 所示，PCIe-DMA 模块主要实现了 UDP 协议处理模块与上位机软件的透明通信。

	地址位	数据位
2^16	0	0
	1	0
	2	0
	……	0
	53	1
	……	0
	2^16 - 1	0

图 6　服务端口开放标志

验证。验证的硬件平台依托于斯坦福大学的NetF-PGA-10G板卡，整个系统采用verilog语言开发，基于Xilinx公司的Vivado 2016.4工具进行开发，并使用Xilinx ISIM工具进行了仿真验证，仿真结果表明该协议栈设计能够正确完成ARP、ICMP、IPv4、UDP等协议报文的发送和接收等功能。如下图7所示，协议栈实现了10G速率下数据线速的接收和发送功能。

3　设计验证

本设计的验证分为功能仿真验证和物理性能

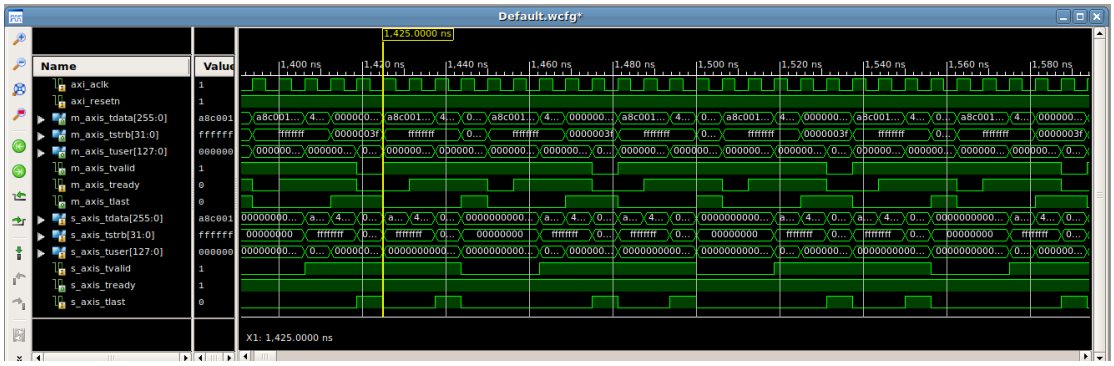


图 7　功能测试结果

为了对本文所提出的设计方案进行性能验证，采用斯博伦TestCenter3.61的两个10G接口进行实际发包测试，采用逐步增大测试数据包MTU长度的方法，测试本系统在处理不同数据包长情况下的性能，测试结果如下图所示。

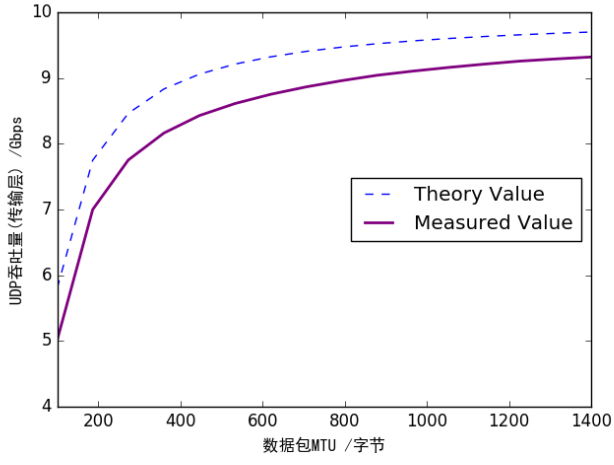


图 8 性能测试结果

从图8中看以看出，随着数据报文MTU值的增加，硬件协议栈对于UDP协议负载的传输能力越发贴近理论值，当MTU达到1400字节时，协议栈对于UDP数据可以达到9.32Gbps的传输速率，接近理论值9.7Gbps，达到了设计的要求，满足10G连路上线速传输数据的需求。

4 结束语

传统基于FPGA的UDP/IP协议栈研究都是集中在一对一的点到点传输，无法适用于10G网络环境下提供多点并发连接的服务器场景，本文研究了一种面向服务器应用场景的UDP硬件协议栈，本设计充分利用硬件在并行处理上的优势，流水化设计UDP协议处理流程，将UDP协议处理固化在FPGA中，降低于传输层协议对处理器的处理需求，提升系统的传输效能，通过实验结果表明，该协议栈可以适用于10G网络UDP协议的线速处理。

参考文献:

[1] 10G以太网MAC控制器的设计与验证[D]. 杨莹. 中国科学技术大学 2016

[2] 基于FPGA+MAC+PHY的千兆以太网数传系统设计[J]. 侯义合，张冬冬，丁雷. 科学技术与工程. 2014(19)

[3] 杨阳,刘剑,蒋廼偶,李赛辉. 一种基于FPGA的万兆光纤以太网高速传输方法[J]. 雷达与对抗,2015,03:23-27.

[4] 崔鹤,刘云清,盛家进. 基于FPGA的UDP/IP协议栈的研究与实现[J]. 长春理工大学学报(自然科学版),2014(2):133-137.

[5] Francesc Moll Echeto. "Design and implementation of an UDP/IP Ethernet hardware protocol stack for FPGA based Systems [D]." Universitat de Barcelona，2019. 01

[6] Nianyun Liu, Zhiqiang Xu，"The Design of High-Speed Hardware UDP/IP Stack Based on FPGA for Large-Scale Sensing Systems"，Journal of Internet Technology，vol. 18, no. 3 ，pp. 579-587，May. 2017.

[7] 王禹衡。基于FPGA的10G以太网UDP/IP处理器视频传输接口设计[D]. 沈阳工业大学 2018. 06

[8] 基于FPGA的万兆以太网数据分发平台设计[D]. 夏杨. 北京理工大学 2016

[9] https://www. plda. com/products/fpga-ip/xilinx/fpga-ip-tcpip/quicktcp-xilinx/.

[10] LangenbachU. , BertheA. , TraskovB. , WeideS. , HofmannK. , and GregoriusP. , "A 10 gbe tcp/ip hardware stack as part of a protocol acceleration platform, " in Consumer Electronics Berlin (ICCE-Berlin), 2013. ICCEBerlin 2013. IEEE Third International Conference on. IEEE，2013，pp. 381－384.

[11] http://www. intilop. com/tcpipengines. php/.

[12] http://www. dinigroup. com/new/TOE. php/.

[13] 基于NetFPGA10G的网络数据包加密实现及其低功耗研究[D]. 刘宇寒. 杭州电子科技大学 2018

[作者简介]

董永吉（1983-），男，副研究员，博士，研究方向为高速网络技术。

一种基于多决策器强化学习的关系抽取算法

张建朋，张晓斌，李辉，陈福才
信息工程大学，河南 郑州 450002

摘　要：远程监督方法因其能够将海量文本自动对齐知识库进行关系抽取，从而减少大量人工标注的成本，是当前关系数据获取的重要来源之一。然而由于数据集存在大量的噪声，现有基于远程监督的关系抽取方法难以实现高质量的关系抽取。为了解决此问题，本文使用强化学习的方法，结合现有的深度神经网络，提取原始数据集中可以用于关系分类的高质量句子，剔除噪声数据，从而提高原始数据集的质量。此外，由于对不同关系应采取不同的提取策略，本文提出了一种基于多决策器的强化学习方法，即对特定的关系使用独立的决策策略，将关系类别考虑到影响决策的因素之中。实验结果表明，该模型在真实的数据集上比现有方法能够取得更好的关系抽取效果。

关键词：强化学习、深度学习、关系抽取、远程监督

A Relation Extraction Algorithm Based on Multi-Decision Reinforcement Learning

Zhang Jianpeng, Zhang Xiaobin, Li Hui, Chen Fucai
Information Engineering University, Zhengzhou 450002, China

Abstract: Distant supervision method has become an important source of data acquisition because it can automatically align massive text to extract relationships from knowledge base, thus it reduces the cost of manual annotation. Due to the large amount of noise in the data set, the existing remote extraction-based relationship extraction method is difficult to achieve high-quality relationship extraction. To solve this problem, this paper uses the reinforcement learning, in conjunct with the existing deep neural network, to extract high-quality sentences that can be used for relation classification in the original data set and remove the noisy data set, and thus improves the quality of the original data set. Meanwhile, since different extraction strategies should be adopted for different relations in principle, this paper proposes a kind of reinforcement learning method based on multiple-decision makers, that is, using independent strategies for specific relationships, and takes the relation into the factors that influence the decision. Experiments showed that the model has achieved good results on the real-world datasets.

Key words: Reinforcement learning; deep learning; relation extraction; distant supervision

1　引言

随着Web2.0的迅速发展和移动社交媒体的广泛应用，网络上的文本数据呈现爆炸性增长的趋势，由于数据的繁杂冗余，数据难以准确高效地获取。针对这一问题，信息抽取技术应运而生。信息抽取技术是将网络中的非结构化或半结构化文本甚至是多媒体信息中进行抽取，从而形成可被机器直接理解的规则化，结构化的事实信息，能够辅助人们实现对海量文本数据的获取。目前的信息抽取技术，主要分为实体抽取，关系抽取和事件抽取，本文主要研究关系抽取。关系抽取主要解决文本中实体之间关系的分类归属问题，这也是构造复杂知识库系统和知识图谱的关键步

基金项目：自然科学基金群体项目（No. 61521003），国家自然基金青年基金项目(62002384)，国家重点研发计划项目(2016QY03D0502)，郑州市协同创新重大专项（162/32410218）
通讯作者：张晓斌 zhangxiaobin@pku.edu.cn

骤。其作为一个底层要素，关系抽取已被广泛应用于诸如智能检索，问答系统，机器翻译，文本摘要等智能应用。

所谓的关系抽取任务，是给定一段文本 ，一般是一个句子，以及该文本中包含的实体对 e_1 和 e_2，目标是确定文本中两个实体之间的关系。关系抽取的结果可以由一个三元组 $<e_1,r,e_2>$ 描述，其中 即为两个实体的之间的关系。例如，在“Bill Gates is the founder of Microsoft.”这个句子中，“Bill Gates”和“Microsoft”是句中的两个实体，“founder”是表征这两个实体之间关系的词语，关系抽取的结果就可以用三元组 $<\text{Bill Gates, founder, Microsoft}>$ 表示。

现有的关系抽取算法，主要分为有监督、无监督、半监督、联合抽取方法，开放式抽取方法，远程监督方法等［1］。本文的研究属于远程监督方法的范畴。远程监督由Mints等人［2］提出，是为了解决有监督方法中人工标注难的问题，可以多介绍一下远程监督的定义。然而，现有的基于强化学习的方法都只是训练了一个决策器，并未有效考虑实体之间的关系类别。在强化学习里，关系也可以作为决策网络的一个重要特征，因为决策的目标是当前的句子是否被选取，而不是将句子归类，因此关系类别也可以作为输入加入到决策网络中。

基于此，本文提出了一种基于多决策器强化学习的远程监督关系抽取算法，该方法根据训练标签关系类别对训练集进行分类，并对每个分组分别采用独立的强化学习算法，去除各自类别的噪声数据。由于该方法是将数据按关系类别进行分类，相当于将关系类别考虑进了策略网络中的特征，在一定程度上提高了决策的准确性，从而改善了关系抽取的效果。此外，本文采用的强化学习算法是策略梯度（Policy Gradient）算法［3］，策略梯度算法是一个回合制算法，即决策所获得的奖励只在回合结束时更新。该算法将策略网络化，使难以用价值函数表达的策略可以用神经网络去表达。对于那些拥有高维度或连续状态空间的状态而言，策略梯度算法效率更高，同时也具有更好的收敛性。

2 相关工作

针对大规模的无标注的数据，远程监督方法可以大大降低对人工选取的依赖，能够自动地抽取文本中的实体对，从而扩充知识库的规模。远程监督是利用已有的三元关系组，通过Freebase［4］等关系型知识库自动标注大规模文本。标注的规则如下：对知识库中的一个三元体 $<e_1,r,e_2>$，将文本数据中所有包含实体 e_1 和 e_2 的句子加入到可训练数据中，并将该句子的关系标注为 ，所有三元体都根据这个规则对文本数据进行标注，最终便可构建一份可用于训练的标注语料。从语料的构建过程可以看出，远程监督假设文本中所有包含实体对的句子均满足相应三元体对应关系，但事实并非所有的句子都满足这一关系，因此语料库中含有大量的噪声。为了解决此问题，许多学者进行了多种尝试，Hoffmann等人［5］提出一种基于知识的弱监督关系抽取方法，来解决同一个实体对可能出现多个关系的问题。Surdeanu等人［6］提出一种多示例多标签学习框架来进一步解决改善这个问题。Zeng等人［7］将卷积神经网络和多示例学习方法［8］结合起来，并假设所有包含某一个实体对的句子（称为一个句袋）中至少有一个句子真实表达了对应标签表示的关系，但这样却丢失了许多正确实例。Lin等人［9］利用句子层级的注意力机制对句袋中的句子赋予权重，用以描述对应标签关系的重要程度，但仍难以完全摒弃错误实例的影响，无法将错误实例在训练数据中移除。同时由于语料库没有完全对齐，数据中也会出现一个实体对的所有句子都是错误正例的情况， Zeng和Lin等人使用的方法则难以适应上述情况。因此，Feng等人［10］使用强化学习的方法，将环境反馈部分的预测准确率作为奖励，抽取训练数据中的正确实例，再利用神经网络方法对训练数据进行关系分类，得到了不错的效果。Qin等人［11］同样也是利用强化学习方法对关系进行抽取，不同之处在于Qin等人利用验证集前后回合F1值的变化作为奖励去指导策略网络，从而去除训练数据中的噪声数据。文献［1］提出将注意力机制加入到句子级别的实例抽取中，并对Wikipedia等知识库进行实体知识背景的扩充。文献［19］提出了无标

签的远程监督算法，有效避免了知识库的标签数据对当前的关系类型的判断造成误判，有效的降低了噪声数据的影响，提高了关系抽取的准确性。文献［20］提出将半监督学习与句子级的远程监督进行融合的方法，该方法有效地使用了负类数据，屏蔽了噪声数据的影响。文献［21］采用了多尺度注意力机制的远程监督抽取算法，该算法有效地使用了多尺度特征，将注意力机制与句子相关性进行比较，从而提高远程关系抽取任务。文献［22］提出远程监督和卷积神经网络融合的方法对关系进行抽取，并将其扩展到远程监督数据上，有效地降低了错误标签的传播。

3　本文模型概述

本文模型由两个关键模块组成，如图1所示，分别为强化学习模块和关系分类模块，强化学习模块又可分为句子选择模块和环境反馈模块。模型首先对原始数据按照关系进行分组，通过强化学习模块去除数据中的噪声，再通过关系分类模块将强化学习训练获得的高质量句子运用现有的句子或句袋级别的关系分类方法进行分类。

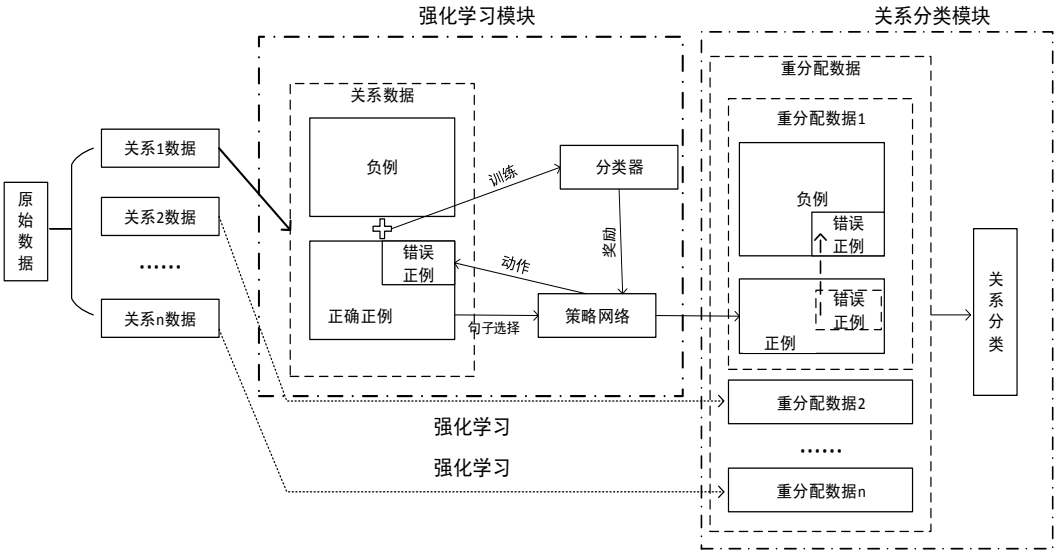


图1　模型结构

3.1　强化学习模块

强化学习模块可分为句子选择模块和环境反馈模块。如图2所示。句子选择模块会根据奖励的大小，更新策略网络，有选择地从噪声数据中筛选正确的句子，而环境反馈模块则将这些筛选的句子作为训练数据，更新深度神经网络，并反馈评估该句子的奖励给句子选择模块。这两个模块在训练过程中相互作用，共同影响，构成强化学习模块。

3.1.1　句子选择模块

句子选择模块定义为：给定一组<句子，关系类别>对，记为 $X=\{(x_1,r_1),(x_2,r_2),\quad,(x_n,r_n)\}$，其中，$x_i$ 是一个与两个实体 (h_i,t_i) 相关联的句子，r_i 是一个由远程监督产生的包含噪音的关系标签。该模块是为了确定哪句话真正描述了该关系，从而确定是否选择这句话作为训练实例。在句子选择模块中，输入是根据 r_i 已经进行分组的句子集，每个句子 x_i 都有一个相应的动作 a_i 来指示 x_i 是否被选择为关系分类的实例。

如图2所示，在强化学习中，决策器（Agent）不断地采取行动（Action），之后转移到下一个状态（State），并且获得一个奖励（Reward），从而更新决策器的策略网络，与环境（Environment）完成交互。该模型的句子选择模块和环境反馈模块就是一个强化学习机制，句子选择模块是一个决策器，它与关系分类器构成的

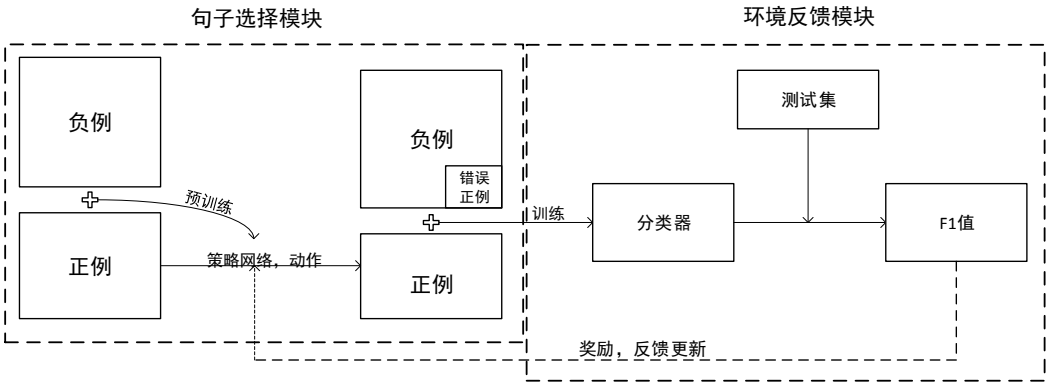


图2　强化学习模块

环境进行交互。句子选择模块遵循策略函数来决定在每个状态中选择哪个动作（是否选择当前句子），然后选择完成时转移到下一个状态并从环境反馈模块中得到奖励。我们接下来将从状态，行动和奖励三个方面介绍句子选择模块。

3.1.1.1　状态

状态包含了当前的句子以及当前回合已经选择了的句子，状态是由一个实值向量 $F(s_i)$ 表示的，该向量编码了如下的信息：

（1）当前句子（未经过词向量映射）记为 $F_1(s_i)$；

（2）所选句子集的表示，即所有被选取的句子经词向量映射后所得到的向量表示的平均值，记为 $F_2(s_i)$；

由状态的定义可知，在当前状态和当前状态下所执行的动作确定之后，系统的下一个状态也确定了，即 $P(s_{i+1}|s_i, a_i) = 1$。

对状态的处理可以达到状态满足马尔科夫性的目的，即系统的当前状态只与上一个状态有关。需要注意的是，当前句子未经词向量映射处理是因为后续决策网络中要将词向量映射层中的词向量作为网络参数进行微调。

3.1.1.2　动作

动作是模型根据当前环境的状态而做出的一个决策，可由策略函数 $\pi_\Theta(s_i, a_i)$ 表示，其中　表示模型中需要学习的参数，a_i 表示模型做出的决策动作，当 $a_i = 0$ 时表示句子选择器没有选择当前回合　中的第　个句子，当 $a_i = 1$ 时则表示选择了回合　中的第　个句子。

策略网络的输入是状态 $F(s_i)$，当前句子 $F_1(s_i)$ 经词向量　映射得到 $F_1'(s_i)$，再与所选句子集 $F_2(s_i)$ 连接得到向量 $\mathrm{I}(s_i)$，该向量作为卷积神经网络的输入，输出得到 $\mathrm{O}(s_i)$，卷积核参数为 W_c 和 b_c。

随后，经过全连接层，可得当前状态的策略函数如下：

$$\pi_\Theta(s_i, a_i) = P_\Theta(a_i \mid s_i)$$
$$= \frac{a_i \exp(W_{\theta_1} * O(s_i) + b_1) + (1 - a_i)\exp(W_{\theta_0} * O(s_i) + b_0)}{\exp(W_{\theta_0} * O(s_i) + b_0) + \exp(W_{\theta_1} * O(s_i) + b_1)} \quad (1)$$

所需训练参数为 $\Theta = \{\mathrm{V}, \mathrm{W}_c, W_{\theta_0}, W_{\theta_1}, b_c, b_0, b_1\}$，根据所需参数设置，进一步优化参数。

3.1.1.3　参数优化

对于每一个回合 r，如果评测集上的F1值相对于上一个回合有所增加，说明该回合下句子选择的动作相对合理，于是在更新训练参数时赋予

一个正奖励，反之同理。奖励R的定义下一节会具体给出。本文定义损失函数为：

$$loss = \sum_{i=1}^{|r|} R \log \pi_{\Theta}(s_i, a_i) \tag{2}$$

其中R是环境反馈模块的奖励，动作a_i由策略函数$\pi_{\Theta}(s_i, a_i)$给出，状态s_{i+1}由状态转移概率$P(s_{i+1} | s_i, a_i)$给出，在当前状态和动作已知后，下一个状态就可以完全确定，即$P(s_{i+1} | s_i, a_i) = 1$。

3.1.2 环境反馈模块

环境反模块定义为：给定一个句子x_i和该句子包含的实体对(h_i, t_i)，目标是预测x_i的关系r_i。该模块以句子选择模块中筛选出的句子作为训练实例，采用卷积神经网络（CNN）[12]进行训练，同时给句子选择模块一个奖励反馈，以优化其策略函数。

环境反馈模块是在句子层级上进行关系预测，首先通过句子选择模块直接过滤掉有噪声的句子，然后通过关系分类模块进行句子层级的分类。该方法与之前的句袋模型方法有着显著的不同，例如Zeng等人[7]是通过选择句袋中最能表示当前关系的一个句子来降低噪声的影响，而Lin等人[9]则是通过减少噪声句子的权重来降低噪声的影响。本文的模型是对经过处理后的已过滤句子进行训练和测试，相比较而言，本模型对噪声数据的处理更为合理，

众所周知，CNN已被广泛应用于诸如图像分类，视频检测，自然语言处理等诸多领域，并以其有效性和高效性获得了广泛关注。在环境反馈模块中，为了提高效率，同时保证较好效果，本文采用了CNN模型来预测关系并反馈奖励，并在其非线性层获得该句子的一个表示向量，作为句子选择模块中句子的状态表示之一。CNN模型基本沿用了Zeng等人[13]的做法，下面简要介绍该模块。

3.1.2.1 输入层

（1）句子清洗。需要将句子进行大写转换为小写，标点符号的分离等等操作。

（2）使用Word2Vec[14]训练单词。在这里，首先使用Word2Vec工具在大量Wiki语料上进行训练，完成后，保留模型的权重参数，然后训练SemEval-2010 task 8数据集，这可以在一定程度上提高词向量的表示性能。

（3）位置向量标记。将实体位置标记为0，实体左边的依次标记为-1，-2，-3……，右边的标记为1，2，3……，之后再将其映射到随机初始化的向量，由于有两个实体，所以要对应到两个向量。

（4）在进行数据预处理和向量映射之后，原始数据的句子就可以用向量$x = (\text{Word}, \text{P1}, \text{P2}), x \in {}^{d}_{s}$表示，World代表词向量，P1代表实体1的位置向量，P2代表实体2的位置向量。

3.1.2.2 卷积层

通过指定一定大小的窗口值来提取某一类特征。该窗口值称为一个卷积核，卷积层可以有多个卷积核，卷积层的输入是一个矩阵$M = \{x_i | i = 1, 2, \quad, d_m\}, x_i \in {}^{d}_{s}$。假设卷积核滑动窗口大小为$k \times d_s$，让该窗口在矩阵　上滑动，共有$d_c$个卷积核。经过卷积操作，有$C = CNN(M), C \in {}^{d_s \times (d_m - k + 1)}$，这里卷积核参数$W_c \in {}^{d_c \times k \times d_s}, b_c \in {}^{d}_{s}$。

3.1.2.3 池化层

池化层作用是将模型提取的特征进行过滤，不仅能去除一些冗余信息，还能减少网络的节点数进而减少训练参数数量。池化层的输入是C，令$C = \{c_i | i = 1, 2, \quad, d_s\}, c_i \in {}^{d_m - k + 1}\}$，则$P = \{\max(p_i), i = 1, 2, \quad, d_s\}$。

3.1.2.4 全连接层

在全连接层，将池化层后的节点个数降到关系类别数，再通过softmax非线性层，从而得到所有关系类别下的分到各关系类别的概率$p(r | x)$。

$$p(r | x) = \text{softmax}(W_f * \tanh(P) + b_f) \tag{3}$$

其中全连接层的参数$W_f \in {}^{d_r \times d_s}, b_c \in {}^{d}_{r}$，$d_r$是关系类别数。

3.1.2.5 优化

对于给定的句子集合 $\hat{S}$，目标是最大化概率 $p(r|x)$，因此损失函数定义如下：

$$loss(\Phi) = -\frac{1}{|\hat{S}|}\sum_{i=1}^{|\hat{S}|}\log p(r_i|x_i;\Phi) \tag{4}$$

这里训练参数 $\Phi = \{W_c, b_c, W_f, b_f\}$

3.1.2.6 奖励

一般而言，评估一个句子选择器的好坏，就是评估在该句子选择器下筛选出来的句子集合的质量，句子集合质量越好，则用它们训练出来的关系分类器的效果就越好。为了评估句子选择器的一系列行为，我们采用了一个简单的CNN模型对句子选择器选择出来的句子进行评估，主要是考虑到简单的网络对训练集的质量更为敏感。可以用该句子集训练出来的二分类器在评测集上的F1值作为评价句子选择器性能的标准。很自然地，可以用F1值的前后变化作为该回合的奖励。

$$R = \alpha(F_1^i - F_1^{i-1}) \tag{5}$$

这里得到的奖励，将作为句子选择器的输入，用于更新策略网略，指导策略网络的决策按照我们希望的方向靠近。

3.2 关系分类模块

在3.1中的强化学习模块中，我们采用了基于策略的强化学习方法生成新的关系数据集，并通过选取原始数据中的高质量句子来重新分配训练数据集。对于重新分配后的数据集，就可以利用Zeng等人［7］和Lin等人［9］提出的模型进行训练，进而评估模型的性能。与之前的工作类似，采用hold-out方法来评估关系分类器的性能。对于测试集的数据，我们也将其通过决策网络，得到新的测试集，同重新分配的训练数据集一起，构成关系分类模块的数据集。

4 实验

4.1 实验数据

关系抽取模型采用的是由Riedel等人［15］收集的数据集。这个数据集是将来自Freebase的实体对与纽约时报语料库（NYT）对齐而生成的。NYT语料库的实体由斯坦福大学实体识别工具［16］进行识别。以纽约时报2005-2006年的句子为训练语料，2007年的句子为测试语料。数据包含52个实际关系和一个特殊关系，即Other类，表明首尾实体之间没有关系，这也是实验里负例的来源。

4.2 实验参数

句子选择器的动作空间只包括两个操作，即选取和不选取。因此，可以将句子选择器建模为二分类器。实验采用单窗口CNN作为策略网络。详细的超参数设置如表1所示。至于词向量，我们直接使用Lin等人［9］实验发布的词向量文件。位置向量则采用随机初始化的策略。

表1 超参数设置

句子选择模块参数		环境反馈模块参数	
参数	值	参数	值
词向量维度 d_w	50	词向量维度 d_w	50
位置向量维度 d_p	5	位置向量维度 d_p	5
预训练时批处理大小	64	CNN卷积窗口个数	3
CNN卷积核窗口个数	1	CNN卷积核个数	256
CNN卷积核个数	128	奖励系数 α	100

4.3 决策网络的训练

4.3.1 数据分组

在NYT数据集中，包含了52个实际关系和一个Other类（表2罗列了数据集中数量较多的8类关系）。然而，有的关系句子数量太少，不足以生成训练该关系的训练数据，因此，本实验只训练了前8类关系的决策器，剩余数据保持原样。

表2 关系数量分布

关系	数量	关系	数量
/location/location/contains	75969	/people/person/nationality	11446
/location/country/capital	11216	/people/person/place_lived	9829
/location/neighborhood/neighborhood_of	9472	/location/country/administrative_divisions	8860
/location/administrative_division/country	8860	/business/person/company	7987
NA（Other类）	414162	其他关系	12287

4.3.2 预训练

预训练是强化学习相关工作中的一种常见策略，用于加速策略网络的收敛。例如，DeepMind著名的AlphaoGo［17］通过学习很多棋谱，用来对策略网略进行预训练。然而，在远程监控关系提取任务中，数据存在大量的噪声。要想获得较精确的数据集，只能让语言专家为数据集做一些人工标注，但这要花费很大的时间精力，而且也不符合远程监督的初衷。在语料库中，由于正确的正例在数量上比错误正例要多，，我们可以假设原始数据的标注是正确的，进而在此基础上进行预训练。因此，对于一个特定的关系类型，我们直接将远程监督的正集合作为正集合，随机抽取远程监督负集合的一部分作为负集合，作为预训练的数据集。

4.4 模型性能比较

Zeng等人［7］和Lin［9］等人的模型都是解决远程监管关系抽取问题的强大模型。Zeng等人［7］将多实例学习与深度神经网络相结合，只提取句袋中的一个最能表现当前关系的句子，预测实体对之间的关系，并首次提出了PCNN（Piecewise Convolutional Neural Networks），即在池化层通过两个实体位置将特征图分为三段进行池化，其目的是为了更好的捕获两个实体间的结构化信息；Lin等人［9］将一个实体对的所有句子组合起来，并赋予每一个句子一个权重，这样就可以为该实体对生成一个综合的关系表示。本文后续将通过这两个模型来验证数据集的性能，间接验证强化学习模型的性能。为了验证本文所提出模型的有效性，我们利用基于CNN模型和PCNN模型上的两种经典算法在原始数据集和经过本文模型处理后的数据集上进行对比试验。我们采用句子的决策网络将错误正例移动到负例样本集中，进而提取正确实例来重新分配NYT数据集。然后利用上述两种模型上的算法来预测此重分配后的数据集的关系，并将其性能与原始NYT数据集的性能进行比较。基于CNN和PCNN模型上的两种算法为多实例模型（ONE）和句子级选择注意力模型（ATT）。

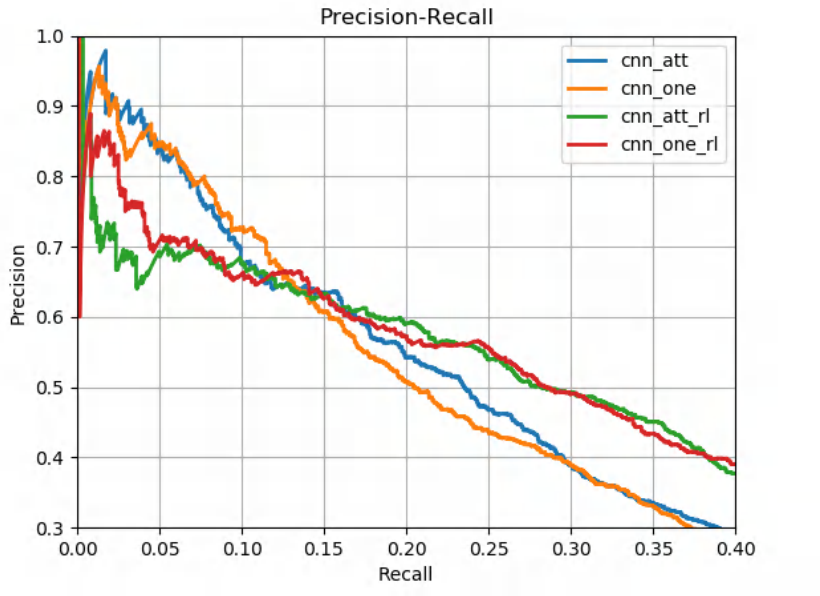


图3 CNN相关PR曲线

图3是基于CNN模型上的两种算法在原始数据集和经过处理后的数据集上的召回率-准确率对比图，图4是基于PCNN模型的召回率-准确率对比图，由图中我们看到两种算法在经过处理后的数据集上表现出了更好的效果。在图3中，当召回率（recall）大于0.2之后，原始数据集和经过本文模型处理后的数据集在算法上的性能提升变得非常明显，而且提升越来越大。在图4中，经过本文模型处理后的数据集在算法上的精度一直比原始数据集要高。这表明在强化学习的协助下，同一

模型可以通过更合理的训练数据集来实现明显性能提升。同时，对比图3和图4，我们可知基于PCNN模型的表现要比基于CNN模型更加的稳定。

为了进行更直观的比较，我们计算了每一条PR曲线的AUC（Area Under Curve）值，如表3所示，它反映了这些曲线下的面积大小，面积越大，代表着算法性能越好。这些比较结果更加直观清楚的表明了我们基于强化学习方法的有效性。

表3　各曲线的AUC值

曲线	AUC值	曲线	AUC值
cnn+att	0.3036	pcnn+att	0.3270
cnn+one	0.3018	pcnn+one	0.3233
cnn+att+rl	0.3339	pcnn+att+rl	0.3427
cnn+one+rl	0.3292	pcnn+one+rl	0.3412

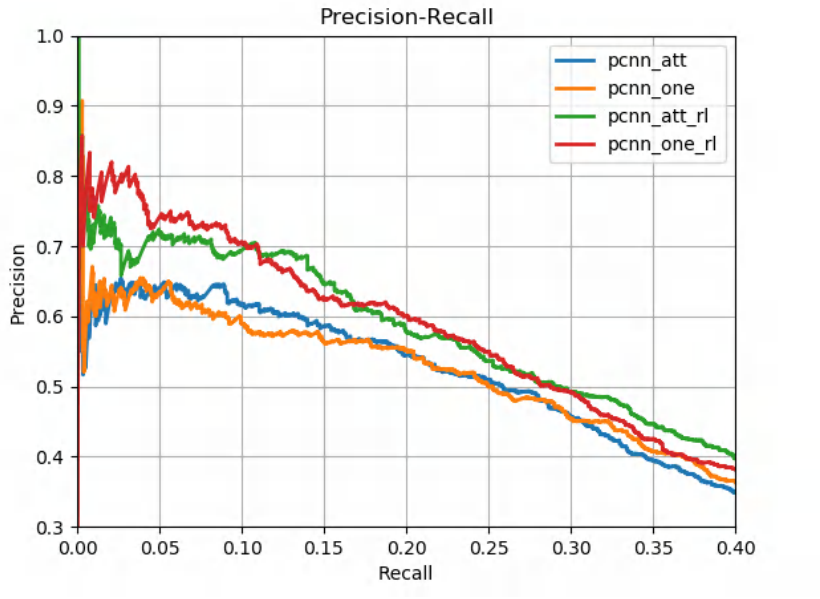


图4　PCNN相关PR曲线

5　结束语

本文提出了一种基于多决策器强化学习的远程监督关系抽取算法。模型的目标是学习一个判定标记数据准确性的策略网络，用来移除数据当中的错误正例，进而优化训练数据，最大限度地提高关系分类的性能。与先前的研究相比，本文的工作是在数据层面进行了优化，先对数据集按关系类别进行分组，再通过策略网络对数据集的标签重分配，即移除错误正例标签，最后组合这些分组数据得到质量相对较高的新数据集。通过在多种现实数据集进行实验验证，实验结果表明本文模型可以筛选出更合理的数据集进行训练从而使得关系抽取任务的性能得到较大提升，证明了本文基于多决策器强化学习模型的有效性。

参考文献:

[1] PAWAR Sachin, PALSHIKAR Girish K. and BHATTACHARYYA Pushpak. Relation extraction : A survey[J]. 2017.

[2] MINTZ Mike, BILLS Steven, SNOW Rion, et al. Distant supervision for relation extraction without labeled data[C]. Joint Conference of the Meeting of the Acl & the International Joint Conference on Natural Language Processing of the Afnlp：Volume，2009：

[3] SUTTON Richard S, MCALLESTER David A, SINGH Satinder P, et al. Policy gradient methods for reinforcement learning with function approximation [C]. Advances in neural information processing systems，2000：1057-1063.

[4] BOLLACKER Kurt, EVANS Colin, PARITOSH Praveen, et al. Freebase: A collaboratively created graph database for structuring human knowledge [C]. Proceedings of the 2008 ACM SIGMOD international conference on Management of data，2008：1247-1250.

[5] HOFFMANN Raphael, ZHANG Congle, LING Xiao, et al. Knowledge-based weak supervision for information extraction of overlapping relations[C]. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics：Human Language

Technologies-Volume 1，2011：541-550.

[6] SURDEANU Mihai, TIBSHIRANI Julie, NALLAPATI Ramesh, et al. Multi-instance multi-label learning for relation extraction［C］. Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning，2012：455-465.

[7] ZENG Daojian, LIU Kang, CHEN Yubo, et al. Distant supervision for relation extraction via piecewise convolutional neural networks［C］. the 2015 Conference on Empirical Methods in Natural Language Processing，2015：1753-1762.

[8] DIETTERICH Thomas G, LATHROP Richard H and LOZANO-PéREZ Tomás. Solving the multiple instance problem with axis-parallel rectangles［J］. Artificial intelligence，1997，89(1-2)：31-71.

[9] LIN Yankai, SHEN Shiqi, LIU Zhiyuan, et al. Neural relation extraction with selective attention over instances［C］. ACL，2016：

[10] FENG Jun, HUANG Minlie, ZHAO Li, et al. Reinforcement learning for relation classification from noisy data［J］. 2018.

[11] QIN Pengda, XU Weiran and WANG William Yang. Robust distant supervision relation extraction via deep reinforcement learning［J］. arXiv preprint arXiv:1805. 09927，2018.

[12] YANN Lecun, LEON Bottou, YOSHUA Bengio, et al. Gradient-based learning applied to document recognition［C］. Proceedings of the IEEE，1998：2278-2324.

[13] ZENG Daojian, LIU Kang, LAI Siwei, et al. Relation classification via convolutional deep neural network［C］. COLING，2014：2335-2344.

[14] MIKOLOV Tomas, CHEN Kai, CORRADO Greg, et al. Efficient estimation of word representations in vector space［J］. arXiv preprint arXiv:1301. 3781，2013.

[15] RIEDEL Sebastian, YAO Limin and MCCALLUM Andrew. Modeling relations and their mentions without labeled text［C］. European Conference on Machine Learning & Knowledge Discovery in Databases，2010：

[16] FINKEL Jenny Rose, GRENAGER Trond and MANNING Christopher. Incorporating non-local information into information extraction systems by gibbs sampling［C］. Meeting on Association for Computational Linguistics，2005：

[17] SILVER David, HUANG Aja, MADDISON Chris J.，et al. Mastering the game of go with deep neural networks and tree search［J］. Nature，2016，529(7587)：484-489.

[18] Ji, Guoliang, et al. "Distant Supervision for Relation Extraction with Sentence-Level Attention and Entity Descriptions.”Thirty-First AAAI Conference on Artificial Intelligence，2017，pp. 3060－3066.

[19] Wang, Guanying, et al. "Label-Free Distant Supervision for Relation Extraction via Knowledge Graph Embedding. " EMNLP 2018：2018 Conference on Empirical Methods in Natural Language Processing，2018，pp. 2246－2255.

[20] 余小康,陈岭,郭敬,蔡雅雅,吴勇,王敬昌. 结合从句级远程监督与半监督集成学习的关系抽取方法[J]. 模式识别与人工智能，2017，30(01)：54-63.

[21] 蔡强,郝佳云,曹健,李海生. 采用多尺度注意力机制的远程监督关系抽取[J]. 中文信息学报，2018，32(01)：96-101.

[22] Zeng, Daojian, et al. "Adversarial Learning for Distant Supervised Relation Extraction.”Cmc-Computers Materials & Continua，vol. 55，no. 1，2018，pp. 121－121.

[作者简介]

张建朋，男，1988年生，信息工程大学助理研究员，主要研究方向为大数据分析。

张晓斌，男，1993年生，信息工程大学硕士研究生，主要研究方向为文本挖掘、实体关系抽取（通信作者 zhangxiaobin@pku. edu. cn）。

李辉，男，1996年生，信息工程大学硕士研究生，主要研究方向为知识图谱。

陈福才，男，1974年生，信息工程大学研究员、硕士生导师，主要研究方向为大数据分析与处理。

A Sensitivity Analysis of Attention-Gated Convolutional Neural Networks for Sentence Classification

Liu Yang, Zhang Jianpeng*, Gao Chao, Qu Jinghua, Ji Lixin

National Digital Switching System Engineering and Technological R&D Center, Zhengzhou, 450002, ChinaInformation Engineering University, Zhengzhou, 450002, China

Abstract: In this paper, we investigate the effect of different hyperparameters as well as different combinations of hyperparameters settings on the performance of the Attention-Gated Convolutional Neural Networks (AGCNNs), e. g. , the kernel window size, the number of feature maps, the keep rate of the dropout layer, and the activation function. We draw practical advice from a wide range of empirical results. Through the sensitivity analysis, we further improve the hyperparameters settings of AGCNNs. Experiments show that our proposals could achieve an average of 0. 81% and 0. 67% improvements on AGCNN-NLReLU-rand and AGCNN-SELU-rand, respectively; and an average of 0. 47% and 0. 45% improvements on AGCNN-NLReLU-static and AGCNN-SELU-static, respectively.

Key words: ***Sentence Classification***; ***Sensitivity Analysis***; ***Attention-Gated Convolutional Neural Network***

I Introduction

Natural language is inherently the unstructured data that is difficult to process and comprehension for computers. Sentence classification is one of the most essential and challenging tasks of natural language processing. Recently， Convolutional Neural Networks （CNNs） have achieved remarkable results on a number of practically important sentence classification tasks ［1］-［4］. Among them， Attention-Gated Convolutional Neural Network （AGCNN） ［4］ improves the capability of the pooling layer in the standard CNN ［1］ to find the most significant features by introducing an attention gating mechanism. AGCNN is mainly constructed by two convolutional layers for different uses （see Fig. 1）. AGCNN can generate the hierarchical abstract representation of the input text through the two convolutional layers. AGCNN not only allows to precisely control the length of dependencies but also enables nearby input text elements to interact at lower layers while distant text elements interact at higher layers. These characteristics make AGCNN suitable for processing the unstructured text data after the text data is mapped into word embeddings ［5］， ［6］.

The robust empirical results achieved by AGCNN demonstrate that AGCNN can be used as a substitute for traditional baseline models， e.g.， statistical features-based methods ［7］-［9］， and standard CNNs ［1］. However， AGCNN requires practitioners to set a number of hyperparameters compared to these traditional methods and the performance of AGCNN is quite sensitive to these parameters. Moreover， it is extremely costly to explore the appropriate parameter setting combination of the model in practice. Because for an AGCNN model， we need to set a large number of parameters （e.g.， kernel window size of the first convolutional layer and at-

tention-gated layer， the number of feature maps of the first convolutional layer and attention-gated layer， the dropout rate， and the activation function） and each parameter has a large value space. Also， the GPU memory usage of the model is large， and the training speed of the model is relatively slow. For example， the usage of GPU memory can reach 4377 MB， when the number of feature maps for the attention-gated layer is 10， dataset is Subj ［10］， and other parameters follow the original settings ［4］； and running 10-fold cross-validation costs a lot of time， from about 40 minutes （CR ［11］ dataset） to about 2 hours （Subj dataset）. These problems make the application and fine-tuning of AGCNN inconvenient.

Various emerging methods have been proposed to explore hyperparameter optimization， including random search ［12］， ［13］， Bayesian optimization ［14］， ［15］， and the combination of Bayesian optimization and Hyperband ［16］. However， the vast number of possible hyperparameter configurations requires expertise to restrict the search space， and most of these sophisticated methods are very costly.

We are inspired by the previous empirical analyses of Coates et al. ［17］， Breuel ［18］， and Zhang et al. ［19］ on the neural networks， which explored the factors in unsupervised feature learning， the hyperparameter settings in stochastic gradient descent， and the hyperparameter settings in CNNs， respectively. Our aim in this work is to investigate and analyze the sensitivity of AGCNN to each hyperparameter setting of the architecture and to provide reasonable scope and appropriate advice for the fine-tuning of each hyper-parameter， from a large number of empirical results.

The main contributions of our work are summarized as follows：

We investigate the sensitivity of AGCNN to each hyperparameter through a series sensitivity analysis across six different essential datasets， and we draw practical advice and reasonable scope for the tuning of each hyperparameter from a wide range of empirical results.

We explore the effect of different combinations of hyperparameter settings on the performance of AGCNN and analyze to what extent different hyperparameter settings contribute to the performance of AGCNN.

We improve the hyperparameter settings of AGCNN， and experiments demonstrate that our proposals achieve an average of 0.81% and 0.67% improvements on AGCNN-NLReLU-rand and AGCNN-SELU-rand， respectively； and an average of 0.47% and 0.45% improvements on AGCNN-NLReLU-static and AGCNN-SELU-static， respectively.

The rest of the paper is organized as follows. In Section II， we review the attention mechanism and architecture of AGCNN. In Section III， we introduce the datasets and configuration of the experiment. Experimental results are summarized in Section IV. Finally， conclusions are given in Section V.

II Related Work

A Attention mechanism

Attention mechanism in neural networks has attracted much attention and has been applied to a variety of neural network architectures， e.g.， encoder-decoder ［20］. Recently， the application of the attention mechanism in CNNs has become a new research hotspot ［21］. As shown in Fig. 1， the AGCNN model constructs an attention-gated layer before the pooling layer to generate attention weight from feature’s context windows by using specialized convolution kernels. These specialized kernels are all one-dimensional， and their window sizes are different to obtain the different grained context attention weights of the same feature. The attention-gating mechanism on the feature maps before the pooling operation can help the pooling layer down-sample the genuinely significant abstract features.

B AGCNN Architecture

Fig. 1 demonstrates a simplified illustration of the AGCNNs’ architecture. As depicted in Fig. 1，

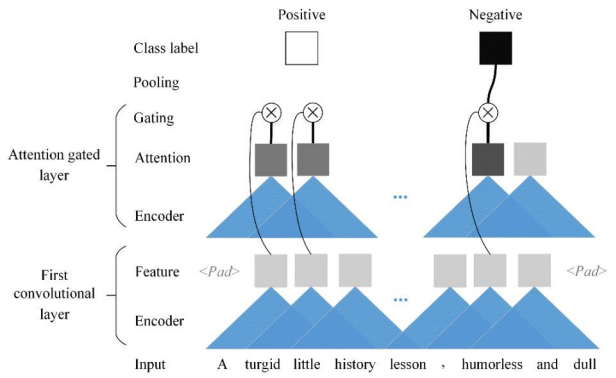


Figure 1 Illustration of the attention-gated convolutional neural network.

AGCNN consists of a convolutional layer, an attention-gated layer, a pooling layer, and a fully-connected layer with dropout operation [22] and Softmax output. The attention-gated layer contains a gating layer and a convolutional layer, where the convolutional layer (padded on the feature map when it is necessary) is used to generate attention weights.

The entire workflow of the AGCNN is as follows. First of all, each word in the input sentence is converted into a word embedding by looking up in the (pre-trained) word embedding matrix. Secondly, the abstract features are generated by the first convolutional layer and activated by activation functions. And then, through the convolution of the attention-gated convolutional layer, we can get the attention gating weights (or called the attention weights). Next, we activate these gating weights and multiply with the abstract feature maps of the first convolutional layer. Finally, we feed these attention-gated feature maps into the pooling layer and the fully connected layer to obtain the prediction output.

III Experiment Configuration and Datasets

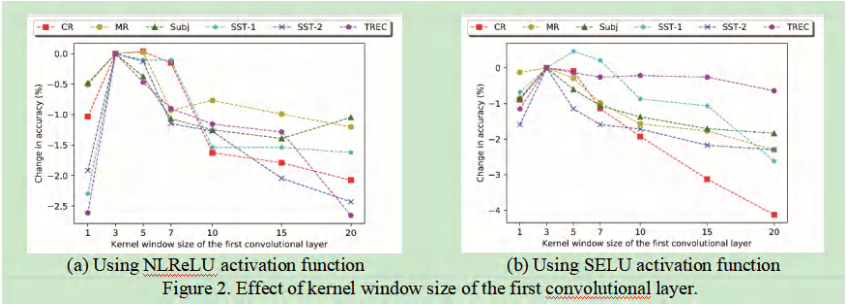
The baseline we choose is the AGCNN-static model (with a single channel and 'static' word embeddings) [4]. For the hyperparameter settings, we set all the kernel window size to 3, and the number of feature maps is 100 and 1 for each convolutional layer, respectively. The keep rate is 0.5. Other experiment settings follow the original settings [4].For consistency, we use the same data preprocessing steps for all the datasets as described in previous work [1], [4]. All the reported results in this paper are from 10-fold cross validation over all datasets. All experiments run with TensorFlow [23] on two NVIDIA Tesla M40 GPUs.

For investigating each parameter's effect, we hold all other settings fixed as baseline model settings and vary only the component of interest. For each configuration, we replicate the experiment 10 times and take the average as the final result. For the experiment results which are plotted as line plots, we only show the percentage change in accuracy from an arbitrary baseline model point. Since the original model has different versions based on two activation functions, i.e., Scaled Exponential Linear Unit (SELU) [24], and Natural Logarithm rescaled Rectified Linear Unit (NLReLU) [4], the model is analyzed separately based on the two activation functions. We emphasize here that in this paper, our aim is not to improve the state-of-the-art results, although the experiments show that our proposals can improve the performance of AGCNN. We aim at analyzing the sensitivity of the model on the hyperparameters and analyzing how much each setting contributes to the performance of the model.

We use six essential datasets, including one topic classification dataset (TREC [25]), one subjective/objective classification (Subj [10]), and four positive/negative classification datasets (CR [11], MR [26], SST-1 and SST-2 [27]). The data preprocessing steps we use are consistent with previous work [1], [4]. These datasets are briefly summarized as follows: (1) CR: Customer reviews of various products. (2) MR: Movie reviews dataset. (3) Subj: The snippets of movie reviews and plot summaries for movies from the internet. (4) SST-1: Extension of MR but with train/dev/test splits provided and fine-grained labels. (5) SST-2: This dataset is derived from SST-1 but removes neutral reviews and converts to two labels. (6) TREC: This dataset requires classifying questions into six question types (wheth-

er the question is about the person， location， numeric information）.

Table 1 summarizes the classification accuracy of AGCNN-static model （with the baseline setting in this paper） on the six essential datasets.



(a) Using NLReLU activation function (b) Using SELU activation function

Figure 2. Effect of kernel window size of the first convolutional layer.

Table 1 Classification accuracy (%) of baseline on different datasets.

Model	CR	MR	Subj	SST−1	SST−2	TREC
AGCNN−NLReLU	84.34±0.89	80.72±0.85	92.78±0.37	47.47±0.97	85.60±0.43	93.48±1.21
AGCNN−SELU	85.18±0.73	80.83±0.69	93.15±0.43	47.02±0.72	86.15±0.44	93.68±1.02

IV Sensitivity Analysis

A Effect of kernel window size of the first convolutional layer

Fig. 2 （a） and 2 （b） show the influences （on the change in the percentage of accuracy） of the kernel window size for the first convolutional layer （the kernel window size of the baseline point is 3）. As shown in Fig. 2 （a） and 2 （b）， as the size of the kernel window increases， the accuracy of the model will decrease significantly， which is especially apparent when the model’s activation function is SELU. For the reasons that the model performance is degraded when the kernel window size is too large or too small， we believe that too large a context window cannot extract important fine-grained semantic information； similarly， too small a context window cannot extract coarse-grained semantic information. Therefore， it is necessary to combine convolution kernels of multiple different context window sizes to extract the semantic information of different granularities from the input sentences at the same time.

We then explore the case where the first convolutional layer has multiple kernel window sizes. Based on previous discussions， we permutate and combine several different window sizes from small to large. The results are reported in Table 2 and 3. It can be seen that as the combined window sizes become larger， the performance of the model decreases； and when the combination adds small window sizes， the performance rises again， and the model performs best at （1，2，3，4，5）. Besides， it can be seen from the error bars of the classification accuracy in Table 2 and 3 that as the granularity of the window size combination becomes more abundant， the fluctuation of the classification accuracy of the model becomes smaller.

Therefore， the richer the granularity of the window size combination helps to improve the performance of the model. The contextual window size of the first convolutional layer’s kernels directly decides the n-gram word embedding information that AGCNNs is capable of capturing from the sentence by the convolution kernels. The combination of multiple consecutive window sizes can make the model capture different granularities information from sentences and improve the performance of the model.

B Effect of kernel window size of the attention-gated layer

As shown in Fig. 3 （a） and 3 （b）， the effect of the kernel window size of the attention-gated

layer varies between different datasets. It is noted that the performance change of the model on most datasets is more significantly as the kernel window size increases when the activation function is SELU. Comparing Fig. 2 and Fig. 3， the fluctuation of the performance caused by the change of window size in the attention-gated layer is more obvious. Therefore， AGCNN is more sensitive to window size changes in the first convolutional layer.

The performance of the model can also be improved by optimizing the window size combination of the attention-gated layer. We then explore the case where the attention-gated layer has multiple kernel window sizes （set the combination of the first convolutional layer’s kernel window size to （1，2，3，4，5））. As reported in Table 4， through the grid-search we find the combination （1，3，5，7） performs the best.

When the kernel window size is odd， the attention weight for each target feature is obtained from its symmetric context window； while for the even window sizes， the attention weight is obtained from the target feature’s asymmetric context window ［4］. Experiments show that odd window sizes can make the model perform better， since the odd-numbered attention-extracted windows are symmetric， which is beneficial to the generation of attention weights.

C Effect of number of feature maps of the first convolutional layer

As shown in Fig. 4 （a） and 4 （b）， the accuracy of the models rises rapidly as the number of feature maps in the range of 10-100， and rises slowly after 100， and tends to be stable in the range of 200-600. The increase in the number of feature maps increases the number of corresponding convolutional kernels with the same window size. Meanwhile， the increase in the number of feature maps also leads to a significant increase in the amount of model parameters and reducing the efficiency of model operation.

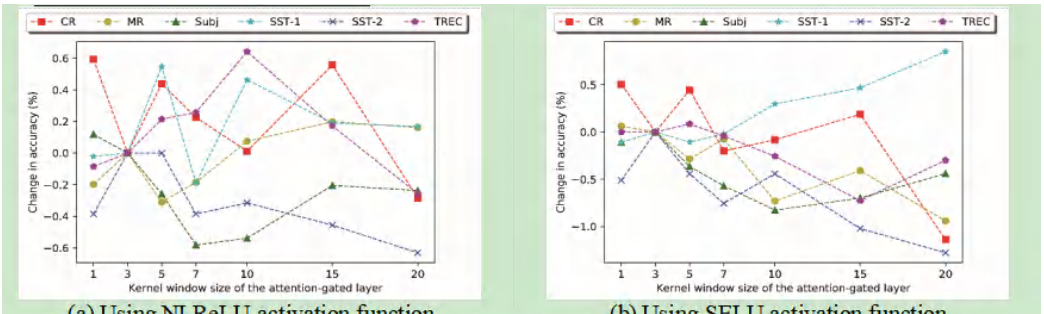
The choice of the number of feature maps should

Table 2 Effect of multiple kernel window sizes of the first convolutional layer (using NLReLU, on the CR dataset).

Multiple Kernel Window Sizes	Accuracy (%)
(1,2,3)	85.70±0.54
(2,3,4)	85.83±0.28
(3,4,5)	85.37±0.15
(4,5,6)	85.21±0.14
(1,2,3,4)	85.63±0.27
(2,3,4,5)	85.53±0.39
(3,4,5,6)	85.29±0.27
(1,2,3,4,5)	**85.92 0.38**
(2,3,4,5,6)	85.37±0.36
(1,2,3,4,5,6)	85.53±0.44

Table 3 Effect of multiple kernel window sizes of the first convolutional layer (using SELU, on the SST-1 dataset).

Multiple Kernel Window Sizes	Accuracy (%)
(1,2,3)	47.50±0.48
(2,3,4)	47.66±0.34
(3,4,5)	47.48±0.30
(4,5,6)	47.23±0.40
(1,2,3,4)	47.60±0.38
(2,3,4,5)	47.81±0.11
(3,4,5,6)	47.34±0.21
(1,2,3,4,5)	**47.90 0.29**
(2,3,4,5,6)	47.45±0.27
(1,2,3,4,5,6)	47.75±0.34



(a) Using NLReLU activation function (b) Using SELU activation function

Figure 3. Effect of kernel window size of the attention-gated layer.

consider the memory allocation and model's performance comprehensively. Experimental results show that increasing the number of convolution kernels within a certain range can help the convolutional layer abstract more efficient and rich features, thus improving the performance of the model. Beyond this range, the increase of accuracy brought about by the addition of convolution kernels is limited. As illustrated in Fig. 4, 20 or more can be a good choice for the number of feature maps of the first convolutional layer. For the tuning of number of feature maps, 100 to 400 is an appropriate scope for the first convolutional layer.

D Effect of number of feature maps of the attention-gated layer

As shown in Fig. 5 (a) and 5 (b), as the number of feature maps of the attention-gated layer becomes larger, the performance of the model on most datasets increases first and then decreases. The performance of the model on most datasets do not fall below the baseline. The percentage change in performance of the model when using SELU is more notable than that of the model when using NLReLU. Comparing Fig. 4 and Fig. 5, AGCNN is more sensitive to the number of feature maps of the first convolutional layer. Fine-tuning the number of feature maps of the attention- gated layer within a certain range can improve the performance of the model.

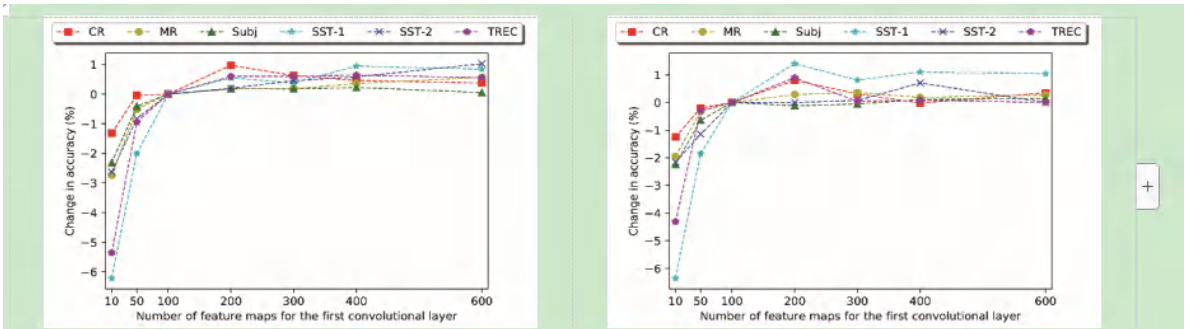
The convolution kernels of the attention-gated layer operate directly on the abstract features which are the n-gram word embedding-based abstract features extracted from the input text. Increasing the number of feature maps within a certain range, which increases the number of kernels corresponding to the same context window size, can help the model extract more efficient and varied attention weights, thereby helping to improve the performance of the model. However, an increase of the number of feature maps of the attention-gated layer also results in a significant increase in the amount of the model's parameters and the memory usage of GPU. Therefore, a comprehensive consideration should be given to setting the appropriate number of feature maps. The number of feature maps of the attention-gated layer should be fine-tuned in the range of 10 to 50 when the GPU memory allocation and computing power are sufficient.

Table 4 Effect of multiple kernel window sizes of the attention-gated layer (using NLReLU, on the CR dataset).

Multiple Kernel Window Sizes	Accuracy (%)
(1,2,3)	85.60±0.14
(1,2,3,4)	85.67±0.28
(1,2,3,4,5)	85.87±0.11
(2,3,4)	85.61±0.35
(3,4,5)	85.84±0.28
(2,3,4,5)	85.51±0.37
(1,3,5)	85.89±0.20
(1,3,5,7)	**86.15 0.36**
(2,4,6)	85.76±0.32
(2,4,6,8)	85.54±0.33

E Effect of keep rate of the dropout layer

Dropout [22] is a very important regularization method for AGCNN. It is used to prevent the model from overfitting. In this section, we explore



(a) Using NLReLU activation function (b) Using SELU activation function

Figure 4. Effect of number of feature maps of the first convolutional layer.

the effect of the keep rate （opposite of dropout rate） on the performance.

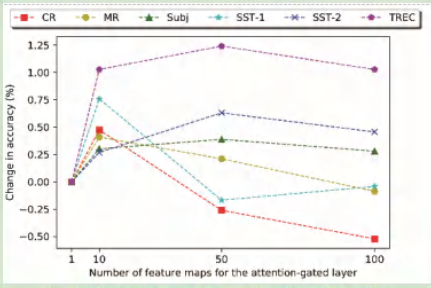
As shown in Fig. 6 （a） and 6 （b）， a too large or too small keep rate will result in a significant drop in the model’s performance. If the keep rate is too large， the model is easy to overfitting； and if the keep rate is too small， the learning of the model is insufficient， which leads to the decline in the model performance.

From Fig. 6 （a） and 6 （b）， one can see that the non-zero keep rate can help with the model at some points from 0.2 to 0.8， depending on datasets. Meanwhile， it can be seen that the model is susceptible to the change of keep rate when it experiments particularly on SST-1. This indicating that the training of the model on this dataset is hard， and also the model is easy to overfit this dataset. The keep rate of the model can be fine-tuned on the particular datasets in the range of 0.2-0.8 to find the most appropriate value.

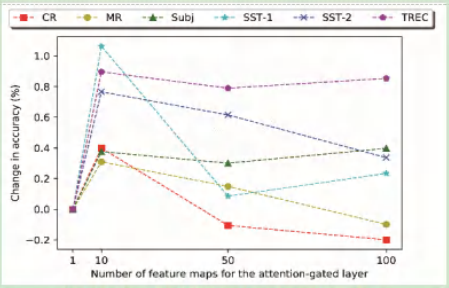
F Effect of activation functions

Activation functions play a crucial role in achieving remarkable performance in deep neural networks. Sigmoid， ReLU [28]， Softplus [28]， Leaky ReLU （LReLU） [29]， Parametric ReLU （PReLU） [30]， Exponential Linear Unit （ELU） [31]， and SELU [24] are all fairly-known and widely-used activation units.

In this section， we set ReLU as the baseline to illustrate the change percentage in the accuracy of other models using

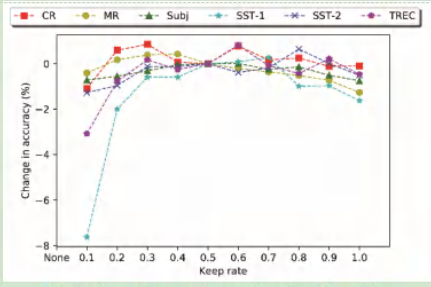


(a) Using NLReLU activation function

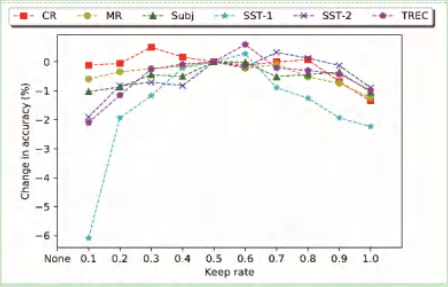


(b) Using SELU activation function

Figure 5. Effect of number of feature maps of the attention-gated layer.



(a) Using NLReLU activation function



(b) Using SELU activation function

Figure 6. Effect of keep rate of the dropout layer.

Table 5 Effect of different hyperparameter setting combinations.

Model	CR	MR	Subj	SST−1	SST−2	TREC
CNN−static−A	84.74±0.89	81.03±0.54	93.04±0.34	45.45±0.45	86.65±0.21	92.06±0.78
CNN−static−B	85.33±0.68	81.14±0.96	93.01±0.55	46.60±0.30	86.32±0.74	93.21±0.59
AGCNN−ReLU	85.80±0.43	81.44±0.61	93.70±0.94	47.71±0.31	86.53±0.45	**94.52 0.86**
AGCNN−NLReLU	85.82±0.31	81.60±0.23	93.82±0.35	48.01±0.31	**87.21 0.25**	94.28±0.68
AGCNN−SELU	**86.43 0.24**	**81.74 0.16**	**93.90 0.28**	**48.33 0.17**	87.14±0.58	94.28±0.64

different activation functions compared to the baseline across different datasets.

As shown in Fig. 7， SELU is the best performing activation function， followed by NLReLU and ELU. However， ReLU performs the best on the TREC dataset. Moreover， one can see that the better performing activation functions are the ones tend to transform each layer’s skewed neurons to approximately “normal” or adequately suppress the distribution of each layer’s neuron activations.

The choice of activation function is essential. Although using SELU could achieve better results in most cases， it seems that AGCNNs are too sensitive to hyperparameters’ changes when using SELU（compared with NLReLU）. The activation function used should be determined in conjunction with the application scenario and requirements.

G Effect of different parameter setting combinations

As shown in Table 5， in this section， we will add different parameter settings step by step on the standard CNN-static model to form different combinations of parameter settings. We aim at investigating how much these hyperparameter settings contributes to the performance of AGCNN. Models used for comparison are as follows：

CNN-static-A： Standard CNN-static model.

CNN-static-B： All the initializations are same as AGCNN-static but the attention-gated layer is not used， and the activation function is ReLU.

AGCNN-ReLU（-static）， AGCNN-SELU（-static） and AGCNN-NLReLU（-static）： The activation functions used by each model are ReLU， SELU and NLReLU， respectively.

As reported in Table 5， from the comparison of CNN-static-A and CNN-static-B， the change of the parameter initialization method brings about an average improvement

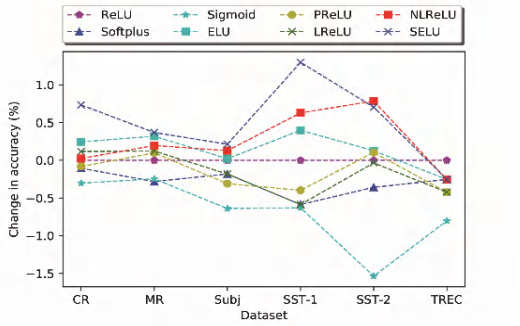


Figure 7 Effect of activation functions.

of 0.44%. The comparison between CNN-static-B and AGCNN-ReLU（-static） shows that the introducing of the attention-gated layer achieves an average performance improvement of about 0.68%. From the comparison of AGCNN-ReLU （ -static），AGCNN-NLReLU（-static） and AGCNN-SELU（-static）， the use of activation functions NLReLU and SELU bring about the average improvements of 0.17% and 0.35%， respectively. Therefore， the attention-gated layer contributes the most to the performance of AGCNN， followed by the initialization method and the choice of activation function.

Based on the above practical results and conclusions， we improve the hyperparameter settings of AGCNN. We set the number of feature maps for the first convolutional layer and the attention-gated layer to 200 and 10， respectively. The multiple kernel win-

Table 6 Classification accuracy (%) of different hyperparameter settings.

Model	CR	MR	Subj	SST−1	SST−2	TREC
AGCNN−NLReLU−rand	82.09±0.39	78.33±0.35	91.56±0.32	44.41±0.38	83.54±0.24	92.52±0.39
Ours	82.95±0.48	79.01±0.15	92.07±0.44	45.08±0.26	84.42±0.25	93.78±0.11
AGCNN−NLReLU−static	85.82±0.31	81.60±0.23	93.82±0.35	48.01±0.31	87.21±0.25	94.28±0.68
Ours	86.34±0.23	81.83±0.10	93.79±0.16	48.48±0.18	87.25±0.20	95.35±0.35
AGCNN−SELU−rand	82.33±0.49	78.29±0.31	91.84±0.25	44.72±0.18	83.69±0.26	92.93±0.41
Ours	83.02±0.43	78.83±0.24	92.27±0.15	45.30±0.44	84.35±0.12	94.04±0.36
AGCNN−SELU−static	86.43±0.24	81.74±0.16	93.90±0.28	48.33±0.17	87.14±0.58	94.28±0.64
Ours	86.67±0.19	81.95±0.08	93.72±0.12	48.74±0.08	86.90±0.33	95.23±0.41

dow sizes for the first convolutional layer and the attention-gated layer are set to （1， 2， 3， 4， 5） and （13， 5， 7）. Keep rate is 0.5. The results are summarized in Table 6. Compared with the baseline models， our proposals can achieve an average of 0.81% and 0.67% improvements on AGCNN-NLReLU-rand and AGCNN-SELU-rand， respectively； and an average of 0.47% and 0.45% improvements on AGCNN-NLReLU-static and AGCNN-SELU-static， respectively.

V Conclusions

In this paper， we investigate how sensitive the model's performance is with respect to the changes in the configurations of the parameter settings and conduct an extensive sensitivity analysis of AGCNNs for sentence classification. We then explore and analyze how much different parameter setting combinations contribute to model’ s performance. Meanwhile， for those interested in using AGCNNs for sentence classification in the real-world sentence classification scenarios， we draw practical advice by summarizing from these wide ranges of empirical study. Also， in this work， we improve the performance of AGCNN by improving the hyperparameter settings of AGCNN.

References:

[1] Y. Kim，“Convolutional neural networks for sentence classification"，arXiv preprint arXiv:1408. 5882，2014.

[2] WangP. , XuJ. , XuB. , LiuC. , ZhangH. , WangF. , HaoH. "Semantic clustering and convolutional neural network for short text categorization", Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing，pp. 352 - 357，2015.

[3] ErM. J. , ZhangY. , WangN. , PratamaM. ，"Attention pooling-based convolutional neural network for sentence modelling"，Information Sciences，vol. 373，pp. 388-403，2016.

[4] LiuY. , JiL. , HuangR. , MingT. , GaoC. , ZhangJ. ，“An Attention-Gated Convolutional Neural Network for Sentence Classification”，arXiv preprint arXiv: 1808. 07325v3， 2018. (Intelligent Data Analysis，in press).

[5] MikolovT. , ChenK. , CorradoG. , DeanJ. “Efficient estimation of word representations in vector space", arXiv preprint arXiv: 1301. 3781，2013.

[6] MikolovT. , SutskeverI. , ChenK. , CorradoG. S. , DeanJ. "Distributed representations of words and phrases and their compositionality", Advances in Neural Information Processing Systems，pp. 3111-3119，2013.

[7] McCallum, K. NigamA. “A comparison of event models for naive bayes text classification”，AAAI-98 Workshop on Learning for Text Categorization，pp. 41-48，1998.

[8] IkonomakisM. , KotsiantisS. , TampakasV. ，"Text classification using machine learning techniques"，WSEAS transactions on computers，vol. 4，pp. 966-974，2005.

[9] WangZ. , QianX. "Text categorization based on LDA and SVM", International Conference on Computer Science and Software Engineering，pp. 674-677，2008.

[10] B. Pang，L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts"，Proceedings of the 42nd annual meeting on Association for Computational Linguistics，Barcelona，pp. 271，2004.

[11] HuM. , LiuB. "Mining and summarizing customer reviews", Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle，pp. 168-177，2004.

[12] BengioY. “Practical recommendations for gradient-based training of deep architectures”，Neural Networks: Tricks of the Trade，pp. 437 - 478，Springer，Berlin，Heidelberg，2012.

[13] MendozaH. , KleinA. , FeurerM. , SpringenbergJ. T. , HutterF. , “Towards automatically-tuned neural networks” ， Workshop on Automatic Machine Learning，pp. 58-65，2016.

[14] Yogatama, N. A. SmithD. “Bayesian optimization of text representations”，arXiv preprint arXiv:1503. 00693，2015.

[15] BergstraJ. , YaminsD. , CoxD. D. “Making a science of model search：Hyperparameter optimization in hundreds of dimensions for vision architectures"， Proceedings of the 30th International Conference on Machine Learning，pp. 115-123，2013.

[16] ZelaA. , KleinA. , FalknerS. , HutterF. ，“Towards automated deep learning：Efficient joint neural architecture and hyperparameter search", arXiv preprint arXiv：1807. 06906，2018.

[17] CoatesA. , NgA. Y. , LeeH. "An analysis of single-layer networks in unsupervised feature learning", International conference on artificial intelligence and statistics，pp. 215 - 223，2011.

[18] BreuelT. M. “The effects of hyperparameters on SGD training of neural networks”，arXiv preprint arXiv:1508. 02788，2015.

[19] ZhangY. , WallaceB. ，"A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification"， Proceedings of the Eighth International Joint Conference on Natural Language Processing，pp. 253-263，2017.

[20] VaswaniA. , ShazeerN. , ParmarN. , UszkoreitJ. , JonesL. , GomezA. N. , KaiserL. , PolosukhinI. ，“Attention is all you need”，Advances in Neural Information Processing Systems，pp. 5998-6008，2017.

[21] YinW. , SchützeH. , XiangB. , ZhouB. “Abcnn: Attention-based convolutional neural network for modeling sentence pairs", arXiv preprint arXiv:1512. 05193，2015.

[22] SrivastavaN. , HintonG. , KrizhevskyA. , SutskeverI. , SalakhutdinovR. ，"Dropout: a simple way to prevent neural networks

from overfitting"，The Journal of Machine Learning Research，vol. 15，pp. 1929-1958，2014.

[23] AbadiM.，BarhamP.，ChenJ.，ChenZ.，DavisA.，DeanJ.，et al.，"Tensorflow: A system for large-scale machine learning"，USENIX Symposium on Operating Systems Design and Implementation，vol. 16，pp. 265－283，2016.

[24] KlambauerG.，UnterthinerT.，MayrA.，HochreiterS.，“Self-normalizing neural networks”，Advances in Neural Information Processing Systems，pp. 971-980，2017.

[25] LiX.，RothD. "Learning question classifiers", Proceedings of the 19th International Conference on Computational Linguistics，pp. 1-7，2002.

[26] B. Pang，L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales"，Proceedings of the 43rd annual meeting on Association for Computational Linguistics，pp. 115-124，2005.

[27] SocherR.，PerelyginA.，WuJ.，ChuangJ.，ManningC. D.，NgA.，PottsC.，“Recursive deep models for semantic compositionality over a sentiment treebank”，Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing，pp. 1631-1642，2013.

[28] Nair, G. E. HintonV. “Rectified linear units improve restricted boltzmann machines”，Proceedings of the 27th International Conference on Machine Learning，pp. 807-814，2010.

[29] MaasA. L.，HannunA. Y.，NgA. Y. "Rectifier nonlinearities improve neural network acoustic models", Proceedings of the 30th International Conference on Machine Learning，pp. 3，2013.

[30] HeK.，ZhangX.，RenS.，SunJ. “Delving deep into rectifiers：Surpassing human-level performance on imagenet classification"，Proceedings of the IEEE International Conference on Computer，pp. 1026-1034，2015.

[31] ClevertD. A.，UnterthinerT.，HochreiterS.，"Fast and accurate deep network learning by exponential linear units (elus)"，arXiv preprint arXiv：1511. 07289v5，2016.

基于多队列的随机参数归一化方案设计与实现

张坤，李合元，张兴明，吴少勇，李顺斌*

之江实验室，杭州 310012

摘　要：基于动态异构冗余的拟态防御体系作为一种革命性、颠覆性的理论与技术已经在网络空间安全领域发挥重要作用。异构执行体作为拟态主动防御理论的基础，既要保证多执行体与外部通信会话的随机性，又要保证异构执行体间执行结果输出的归一化。针对此问题，本文提出了一种基于多队列的异构执行体间随机参数归一化的方案。仿真和实验结果表明，所设计的多队列随机参数归一化装置，能为拟态防御系统提供统一的随机参数，能有效同步异构执行体间随机参数的乱序请求。

关键词：拟态防御、随机参数、多队列、归一化

Design and Implementation of Random Parameter Normalization Scheme based on multi-queue Architecture

Zhang Kun, Li Heyuan, Zhang Xingming, Li Shunbin

Zhejiang Lab, Hangzhou 310012, China

Abstract: As a revolutionary and subversive theory and technology, the mimic defense system based on dynamic heterogeneous redundancy has been playing an important role in the field of cyberspace security. Heterogeneous executors are the basis of the mimic active defense theory. It is necessary to guarantee the randomness of external communication sessions, and meanwhile, ensure the normalization of the output between different heterogeneous executors. In response to this problem, this paper proposes a normalized scheme to deal with the random parameter requests from heterogeneous executors based on multi-queue Architecture. Simulation and experimental results show that the proposed random parameter normalization device based on multi-queue architecture can provide uniform random parameters for the heterogeneous executors of the mimic defense system, effectively synchronizing the of random parameter requests in disorder.

Key words: Mimic Defense; Random Parameter; Multi-queue; Normalization

1　引言

目前，基于传统网络防护理念和技术的网络系统对于攻击者来说，由于目标对象构造和运行机制的静态性、相似性及确定性，只要发现一个漏洞或预设一个后门，就可以采取“里应外合、隐匿配合”的攻击方式，形成“单向透明”的行动优势，让防御者处处被动，陷于不断的针对攻击者的具体攻击而修补漏洞、甚至“无法设防”的窘境［1］。

以动态、异构、冗余架构起的拟态防御理论，使网络信息系统的安全可信不再以软硬构件的“无毒无菌”为前提，因而能充分适应经济技术全球化的产业生态环境［2］。目前，拟态防御技术已在路由器［3］

和Web服务器［4］等系统上进行了原理性验证和实践，相关的产品化工作和其他原理验证研究也在同步开展中［5］。拟态防御系统的关键操作在于，对多个等价异构执行体的输出结果进行多数表决，及时检测出受攻击对象并执行清洗操

基金项目：本论文受国家重点研发计划项目（2020YFB1804800），之江实验室开放课题资助（编号：2018FD0ZX01），浙江省市场监督管理局“数字经济标准化试点项目”（项目编号：ZJCT5-2019063）。

通讯作者：李顺斌，lishunbin@zhejianglab.com

作［6］。上述表决能生效的前提是，等价异构执行体在正常运行情况时对相同输入具有相同输出相应；而在经受攻击者攻击时，执行体输出表现出与运行环境相关的差异。上述模型也被称之为IPO模型［7］。

然而在许多关键的通信会话场景，等价异构执行体在没有受到攻击时输出也会表现出随机性，具体表现有：1、为保证通信会话新鲜度与安全加密强度由执行体自生成的随机数；2、标记消息时间由执行体自生成的时间戳；3、为区分不同会话由执行体自生成的随机数；4、同一时刻，执行体所须的随机参数可能大于1个，并且随机参数请求的顺序亦可能为乱序；等等。然而，为了避免增加攻击路径，拟态防御系统实行严格的单向联系机制，不同执行体之间不能相互沟通联系。为此，不同执行体间乱序的随机参数请求归一化协同处理问题成为实现拟态防御系统的共性关键问题。

针对上述问题，本文提出基于多队列的随机参数归一化硬件解决方案，为拟态防御系统中的异构执行体的乱序随机参数请求与同步提供并行高效的解决方案。

2　系统结构

随机参数归一化装置在拟态防御系统中的位置与作用如图1所示。主要负责接收来自于不同执行体发起的随机参数请求，并将其缓存和标记，然后将归一化的随机参数结果反馈给相应的执行体。保证不同执行体相同功能会话在不同时间发起的随机参数请求，均能得到一个相同的归一化随机数值。

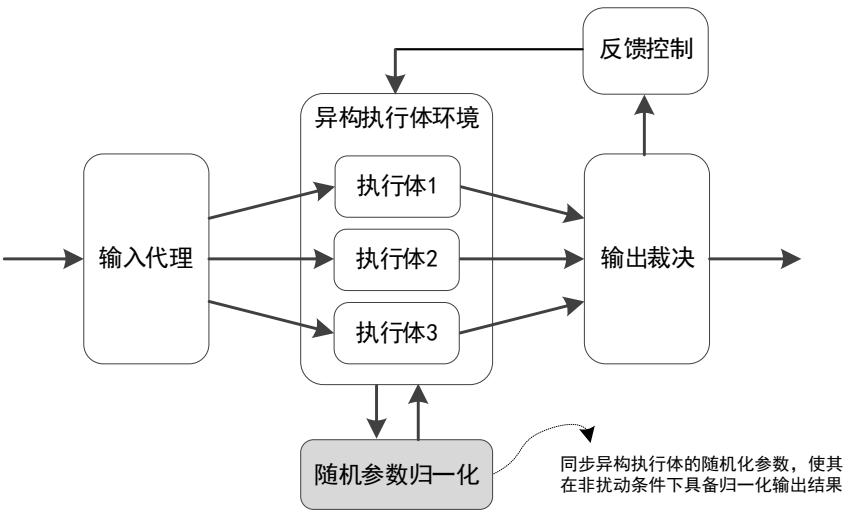


图1　随机参数归一化装置在拟态防御系统中的位置与作用

本文提出的随机参数归一化装置整体架构如图2所示。包含随机参数服务控制接口、随机参数生成器、MD5计算器、随机参数缓存队列、以及队列搜索匹配引擎5大模块。各模块主要功能如下：

随机参数服务控制接口：负责接收执行体的随机参数请求，识别发出随机参数请求的执行体、提取请求的随机参数类型和随机参数请求的具体信息，并将获取的随机参数反馈至对应执行体。

随机参数生成器：根据所请求的随机参数类型，在队列搜索匹配引擎的控制下生成对应的随机数，写入到随机参数缓存队列。

MD5计算器：将所请求的随机参数具体信息按照MD5规范计算512比特MD5散列值，将散列值截短后输入到队列搜索匹配引擎。

随机参数缓存队列：记录着不同执行体的存取标识、MD5散列数据截短值、随机数/定时器时间戳。

队列搜索匹配引擎：根据发起随机参数请求的执行体编号与MD5散列值，在随机参数缓存队列中搜索匹配随机参数结果，完成对随机参数的生成、标记与反馈控制。

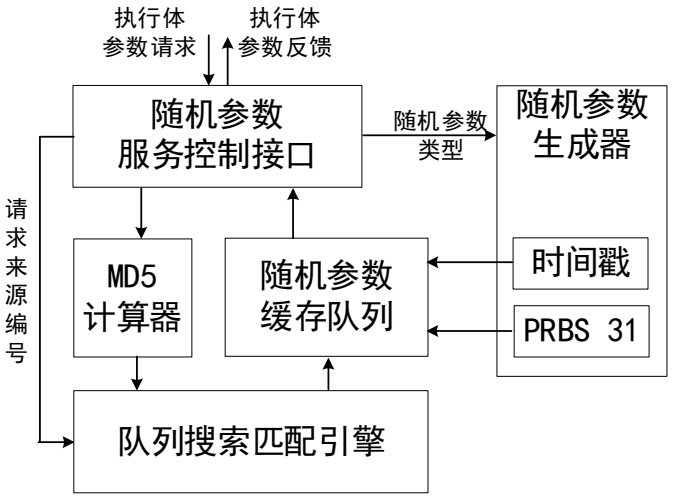


图2　随机参数归一化装置整体结构图

3　工作流程

1、异构执行体中的任意一个或多个发送随机参数请求后，随机参数服务控制接口将不同执行体的请求进行整形和编码标记，然后将整形标记过的执行体请求同时输出到队列搜索匹配引擎、MD5计算器和随机参数发生器中进行处理；

2、队列搜索匹配引擎对请求的执行体编号与16个随机参数缓存队列中的执行体编号有效位进行逐次比对，如果队列中已经存在其他执行体标记过的随机参数，则取出此请求的MD5计算结果与队列中的MD5标识ID进行匹配。

3、若MD5匹配命中，则证明此队列的随机参数对应的其他执行体的请求与此请求为同类，反馈此随机参数回执行体；

4、若所有队列均为空或所有队列匹配均未命中，证明此类请求之前未被其他执行体发送至随机参数归一化装置处理，则从16个队列中选出序号最小且为空的队列进行标记、缓存随机参数、反馈随机参数至执行体。

其中，队列搜索匹配引擎算法的三个执行步骤：队列识别、MD5匹配、结果反馈，具体内容如下图3伪代码所示。

此外，此随机参数归一化装置还为16个随机参数缓存队列分别配置了一个定时器，用以计算每个队列的归一化时间，当某个队列被请求占用了过多时间导致定时器计数达到限定值时，清空队列并将队列中标志位的状态反馈给拟态裁决器，作为执行体安全状态的判决依据之一。

4　硬件实现

随机参数归一化装置全部使用硬件描述语言Verilog进行设计，按照主要功能划分为五个模块，下面对每个模块的实现进行单独介绍：

随机参数服务控制接口：使用一个任务存储队列接收来自于不同CPU的乱序随机参数请求。使用一个状态机对相应的标志位进行轮询，按照轮询结果将请求数据分发到MD5计算器、队列搜索引擎和随机数生成器等模块进行后续处理。

随机参数生成器：包含一个高速并行PRBS31随机数发生器［8］和一个时间戳生成器，根据请求中的标志位判断具体输出的随机数种类。

MD5计算器：使用高层次综合技术对MD5算法进行优化和处理，综合生成硬件实现的MD5计算器［9］。因128位的标准MD5散列值会占用较大的硬件寄存器资源，故将其截短为前29位用以表征对应的执行体请求。

随机参数缓存队列：为了高效率的存取队列中的数据，使用了16个缓存寄存器实现此队列。如图4所示，每个寄存器按照位数划分为执行体编号有效位、MD5标识ID、随机参数和定时器。

队列搜索匹配引擎：如图5所示，使用一个有限状态机完成队列的搜索和匹配。系统以检测各个队列定时器时间是否溢出开始，逐个搜索各个队列是否匹配命中，最后以更新队列并输出反馈给执行体相应的随机数结束，反复循环。

```
输入：各个执行体请求数据：request_data
输出：反馈到各个执行体的数据：random_number
从序号最小的队列开始比较 i = 0:
STEP1:
    while(i < 16){
        if(request_data[31:29]&queue_buffer[i][63:61] = request_data[31:29])      //队列被前一任务占用
            Jump to STEP3                                                        //跳转到步骤3
        else{
            if(queue_buffer[i] = 0){                                             //队列为空
                queue_empty_flag = i                                             //标记最小序号空队列
                Jump to STEP3                                                    //跳转到步骤3
            }
            else
                Jump to STEP2                                                    //跳转到步骤2
        }
    }
STEP2:
    wait(md5_data_valid = 1){                                                    //等待请求数据request_data的MD5运算结果有效
        if(md5_match = 1){                                                       //MD5匹配命中
            executive_system = queue_buffer[i]                                   //取出队列中的随机参数random_number，反馈至对应的执行体
            queue_buffer[i][63:61] = 001 or 010 or 100                           //相应执行体标志位置1
        }
        else
            Jump to STEP3                                                        //跳转到步骤3
    }
STEP3:
    if(!queue_query_finish){                                                     //16个队列未轮询完毕
        i = i + 1                                                                //队列标识加1
        Jump to STEP1
    }
    else{                                                                        //16个队列轮询完毕，所有队列中无匹配项
        i = 0                                                                    //队列计数清零
        if(queue_empty_flag = 1){                                                //最小序号空队列被标记
            Jump to empty queue j                                                //跳转到queue_empty_flag标记的空队列j
            queue_buffer[j] = random_number                                      //向空队列填充随机数
            executive_system = random_number                                     //反馈至对应的执行体
        }
        else{                                                                    //16个队列全被占用，无可用队列存放随机数
            executive_system = queue_overflow                                    //反馈队列溢出标志至执行体
            queue_buffer[0~15] = 0                                               //清空16个队列
            Jump to STEP1
        }
    }
```

图3　队列搜索匹配引擎算法伪代码

图4　随机参数缓存队列结构示意图

5　功能验证及性能分析

本文提出的随机参数归一化方案需针对不同执行体的乱序输入请求进行同步，使其在各种复杂的请求组合下依然能够稳定高效的完成归一化随机数的输出。故将整个验证过程分为仿真验证和系统验证两步，以充分验证其在各种复杂输入激励下的反馈状况。

5.1　仿真验证

编写 testbench 进行行为仿真，通过不同的激励验证随机参数归一化装置功能和时序的正确性。

仿真分为正常 case 验证和异常 case 验证，其中正常 case 主要通过模拟三个执行体输出大量复杂的请求验证反馈结果的一致性。而异常 case 包括定时器计时超出、随机参数缓存队列溢出等，验证其装置在异常或错误状态下的处理和反馈。

图 6 为随机参数归一化装置的部分仿真结果，包括输入输出数据、16 个缓存队列中的参数存取切换和相应的定时器计数状态。由图中可以看出，16 个参数缓存队列针对三个执行体的输入不断更

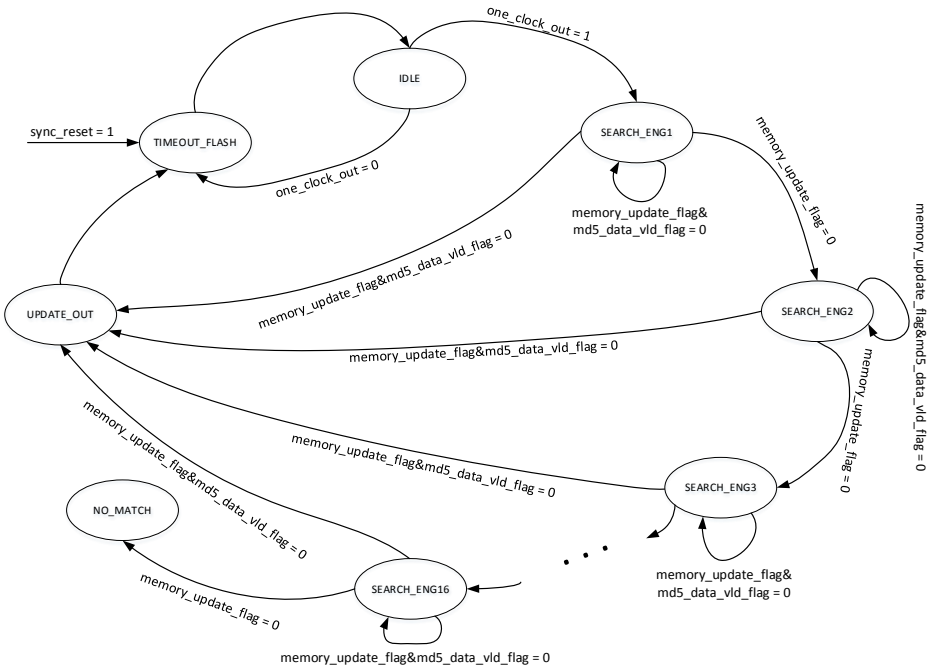


图5　搜索匹配引擎状态机

新相应的标志、MD5计算值、随机数和定时器计数值。此外还有其他复杂case也同样如下图保持着有序的更新和反馈，因篇幅有限，在此不再赘述。

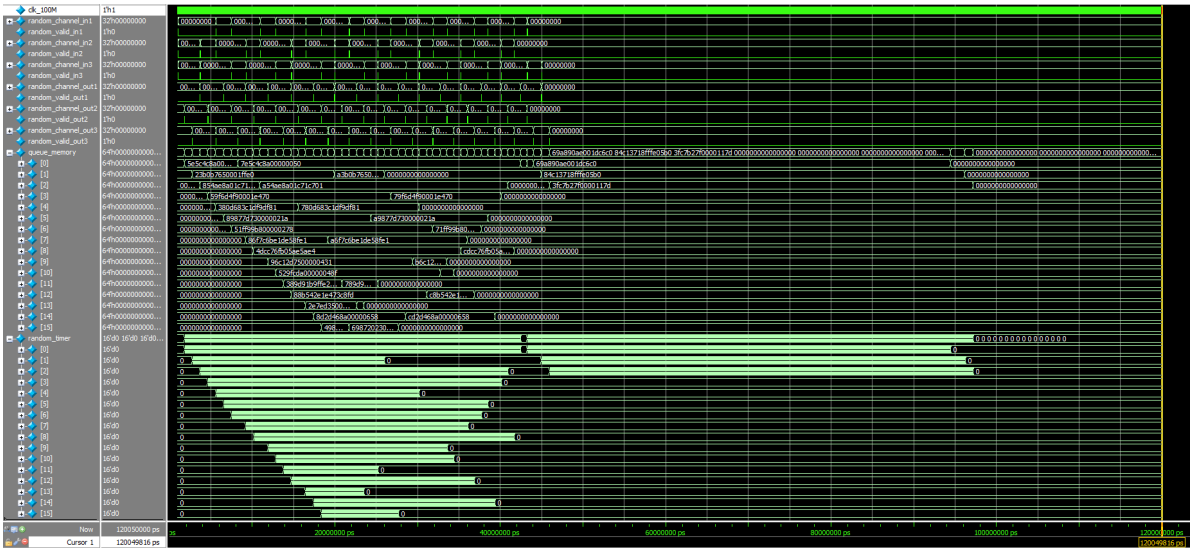


图6　随机参数缓存仿真结果

5.2　系统验证及性能分析

按照行为仿真testbench的验证思路，编写可综合的Verilog硬件电路，生成激励并验证装置的输出。使用搭载了Xilinx的xc7z100ffg900-2芯片的评估板进行板级验证，将经综合、布局布线后的bit流下载到评估板的FPGA中，使用Xilinx提供的FPGA内部虚拟逻辑分析仪捕获相应信号。

如下图7所示。因受FPGA内部RAM资源的制约，虚拟逻辑分析仪无法捕获足够数量和深度的信号，但通过设置不同的捕获条件，可观察到各个信号的时序细节与行为仿真时的输出一致，进而验证了此随机参数归一化装置实现的正确性。

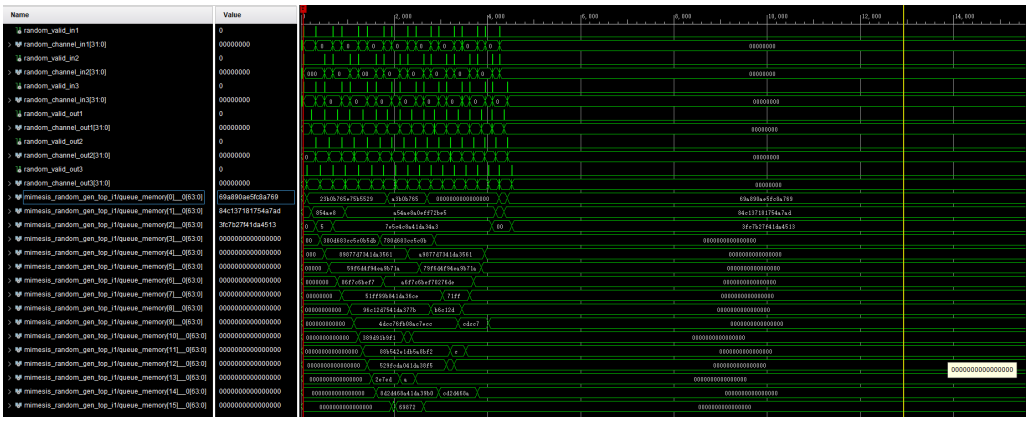


图7　随机参数缓存实验结果

下面对其性能指标进行统计分析：

1、因MD5计算器对时序的要求较高，在时钟频率上升到100MHz以上的之后，MD5计算器的时序难以收敛，故100MHz为此在此FPGA器件型号下的最高时钟频率；

2、在三个执行体同时发出请求时，装置的MD5计算器和队列搜索匹配引擎需要排队处理请求，此时归一化随机数反馈延时达到最大的1870ns，合计187个时钟周期；

3、在三个执行体分时发出请求时，装置可即时进行相应的处理，此时归一化随机数反馈延时为最小的780ns，合计78个时钟周期；

4、功耗方面，Xilinx的Vivado工具估算总功耗约为0.389W，其中静态功耗0.218W，动态功耗0.171W。因为在算法实现过程中使用了较少的逻辑资源，故动态功耗较低。

整体性能统计如表1所示。

表1　整体性能统计

最高时钟频率	最大反馈延时	最小反馈延时	静态功耗	动态功耗
100MHz	1870ns(187 cycle)	780ns(78 cycle)	0.218W	0.171W

表1　整体性能统计

资源消耗方面如下表2所示，表中分别统计了各个子模块针对Xilinx的FPGA器件xc7z100ffg900-2的Slice、LUT、Flip-Flop和Block RAM的资源使用量，并对总体的资源消耗和占用器件资源总量的百分比做出了计算。

器件资源(总量)	Slice(69350)	LUT(277400)	Flip-Flop(554800)	Block RAM(755)
总体消耗(百分比)	2273(3.28%)	6645(2.40%)	3697(0.67%)	7.5(1.00%)
队列搜索匹配引擎	1097	2806	1710	0
MD5运算器	1152	3817	1917	7.5
PRBS发生器	24	22	70	0

表2 FPGA资源消耗统计

以上统计结果表明，本文提出的基于多队列随机参数归一化算法的实现占用了较少的FPGA资源，并达到了较高的性能。

5.3　扩展分析及优化方案

实验中设定的是使用16个随机参数缓存队列存放3个异构执行体发出的乱序请求，在实际的拟态防御系统中可满足绝大多数的使用场景需求。但也会在少数特殊场景中存在更多的异构执行体或需要更多的缓存队列的应用，用户可根据表1和表2中罗列的相关数据进行预评估系统性能及FPGA资源使用量。

尽管此方案可高效稳定的完成多异构执行体的随机参数归一化任务，并在实际环境验证中达到了良好的效果，但还有一些优化空间可进一步提升装置的性能，其具体思路如下：

1、如参考文献［10］中所描述的使用全流水架构算法实现MD5计算器，可提升MD5算法的时钟频率，在多执行体同时提出请求时并行计算MD5的输出值，从而缩短随机参数的归一化反馈

延时；

2、 在某些需要多队列缓存随机数（如64、128、256或更多）的特殊场景，可将队列搜索匹配引擎改为并行匹配搜索算法实现，虽然消耗了更多的FPGA硬件资源，但却可以进一步缩短随机参数归一化反馈延时；

3、 使用更高纳米制程的FPGA器件实现此算法，可提升系统最大时钟频率，缩短随机参数反馈延时。

以上几种方法虽然可达到缩短随机数归一化反馈延时、提升系统时钟频率的目的，但需要说明的是，这两个参数并不是衡量此算法优劣的唯一标准，在实际的使用中，还需综合考虑诸如占用资源、能效比、器件价格等诸多指标，结合实际需求选择合适的实现方案，从而达到一个最佳的平衡点。

6 结束语

本文提出了一种适用于拟态防御系统的多队列随机参数归一化方案。该装置基于多队列的硬件结构，采用MD5散列算法将来自于不同执行体的不同随机参数需求归一化为统一的32位地址表达，能够高效稳定的处理拟态防御系统的随机参数归一化请求。通过大量的仿真与上板验证实验表明，此随机参数归一化装置可保证参数的匹配精度，满足拟态防御系统的随机参数归一化需求，提升随机参数的生成与同步效率。

文末还针对系统时钟频率的提升和归一化参数反馈延时的缩减提出了几个优化方案，用户可在项目中根据实际的需求进行选择。

参考文献:

[1] 邬江兴．鲁棒控制与内生安全[J]. 网信军民融合，2018，010（03）:23-27.

[2] 邬江兴．拟态防御技术构建国家信息网络空间内生安全[J]. 信息通信技术，2019(6).

[3] Ma Hailong, Yi Peng, Jiang Yiming, et al. Dynamic Heterogeneous Redundancy based Router Architecture with Mimic Defenses [J]. Journal of information security，2017，2(1):29-42.

[4] Tong Qing, Zhang Zheng, Zhang Weihua, Wu Jiangxing. Design and Implementation of Mimic Defense Web Server [J]. Journal of software，2017，28(4):883-897. http://www. jos. org. cn/1000-9825/5192. htm.

[5] 武兆琪, 张帆, 郭威,等．一种基于执行体异构度的拟态裁决优化方法[J]. 计算机工程，2020，046(005):12-18.

[6] 高明, 罗锦, 周慧颖,等．一种基于拟态防御的差异化反馈调度判决算法[J]. 电信科学，2020，036(005):73-82.

[7] Jiangxing Wu. Research on Cyber Mimic Defense [J]. Journal of Cyber Security，2016.

[8] 刘宇．基于FPGA的并行PRBS序列的实现[J]. 国外电子测量技术，2008(05):6-8.

[9] 王波涛, 韩国栋, 张效军．基于FPGA的MD5算法设计与实现[J]. 通信技术，2010(01):69-71.

[10] 谭健, 周清雷, 斯雪明,等．全流水架构MD5算法在拟态计算机上的实现及改进[J]. 小型微型计算机系统，2017，038(006):1216-1220.

[作者简介]

张坤 1989年生，男，硕士学历，之江实验室工程师。主要研究方向为工业互联网、网络信息安全等。

李合元 1985年生，男，硕士学历，之江实验室工程师，中级工程师。主要研究方向为工业互联网、网络安全、网络通信、图像传输与处理、数字电路设计等。

张兴明 1963年生，男，教授，现任之江实验室首席科学家助理，主要研究领域为信息与通信系统、拟态计算及拟态安全。

吴少勇 1981年生，男，硕士学历，之江实验室高级工程师，主要研究领域为时间敏感网络、网络安全、未来网络等。

李顺斌 1990年生，男，2018年获浙江大学信息与通信工程专业博士学位，现任之江实验室工业互联网研究中心主任助理，主要研究方向包括异构计算、可重构计算、拟态安全等。

基于区块链的FICS工控系统安全研究

薛镭，吴少勇，林会肖，杨汶佼，王延松，张汝云
之江实验室，310023

摘　要：针对工业控制系统中组态服务器和数据库服务器中心化面临的单点故障、数据恶意篡改等安全风险，提出一种基于Fabric联盟链和Ceph分布式文件存储的工控系统安全解决方案FICS（Fabric based Industry Control System）。该方案采用Ceph对工业控制系统生产运营关键数据文件进行分布式存储，并将数据文件操作记录以账本形式记录于Fabric联盟链，从而充分利用Ceph的分布式、多副本、高性能，以及Fabric联盟链的加密认证、去中心化、数据不可篡改、智能合约等特点，有效解决传统工业控制系统中服务器中心化、组态管理软件缺乏行为监管、数据可篡改等安全风险。

关键词：联盟区块链、工业控制安全、分布式文件存储、智能合约

Research on FICS Industrial Control System Security Based on Block Chain

XUE Lei, WU Shaoyong, LIN Huixiao, YANG Wenjiao, WANG Yansong, ZHANG Ruyun
Zhejiang Lab, Hangzhou, 310023, China

Abstract: In order to solve the problems of single point of failure and malicious data tampering in centralized configuration server and database server of industrial control system, a security solution FICS (Fabric based Industry Control System) is proposed, which applied fabric alliance block chain and Ceph distributed file storage technology. In this scheme, Ceph is used to store the key data files of production and operation of the industrial control system, and the operation records of the data files are recorded in the fabric alliance chain in the form of account books, so as to make full use of Ceph's distributed, multi copy, high-performance, as well as the features of fabric alliance chain, such as encryption authentication, decentralization, data unforgeability, intelligent contract, etc. , to effectively solve the traditional problems In the industry control system, server centralization, lack of behavior supervision and data tampering are the security risks.

Key words: Alliance Block Chain; Industrial control safety; Distributed File System; Smart contract

1　引言

工业控制系统作为包括电力、军工、水电、石油、天然气、化工、交通运输、药品制造、加工厂（食品、饮料、纸质）和高端制造业（汽车、航空和耐用品）等行业广泛使用的信息化控制系统，是涉及国民经济及公共安全的关键信息基础设施，同时也是非法入侵者恶意攻击的重点目标。因此，工业控制系统的信息安全防护尤为重要，增强工业控制系统的安全性要求也日益迫切[1]。

近年来，随着比特币的出现，区块链技术得到了快速发展，并因其所具有的数据加密、分布式记账、去中心化、共识机制等安全特性，引起学术界和产业界的高度重视；目前已经在金融、供应链和物联网等行业中逐步开始应用，对于提升数据安全性已取得良好成效。

本文在分析研究了工控系统安全风险，以及区块链技术安全性的基础上，提出了一种基于Fabric联盟链和Ceph分布式存储的工业控制系统安全实现方案，以解决传统工业控制系统中行为缺乏监管、数据可篡改、服务器中心化等问题，从而使工业控制系统升级成为网络更安全、行为

基金项目：浙江省市场监督管理局“数字经济标准化试点项目”（项目编号：ZJCT5-2019063）。

可追溯、数据更可信的新型工业控制系统，更好地支撑制造业数字化、网络化、智能化发展。

2　工控系统现状与区块链分析

2.1　工控系统现状

工业控制系统是完成实时数据采集、工业生产流程监测控制的管控系统，常被应用于现代大型企业规模生产与运营控制。工业控制系统主要包括：过程控制、数据采集（SCADA）、分布式控制（DCS）、程序逻辑控制（PLC）等。经过多年积累和优化后，系统层级划分已经比较成熟[2][3]。典型的工业控制系统网络结构如图1所示：

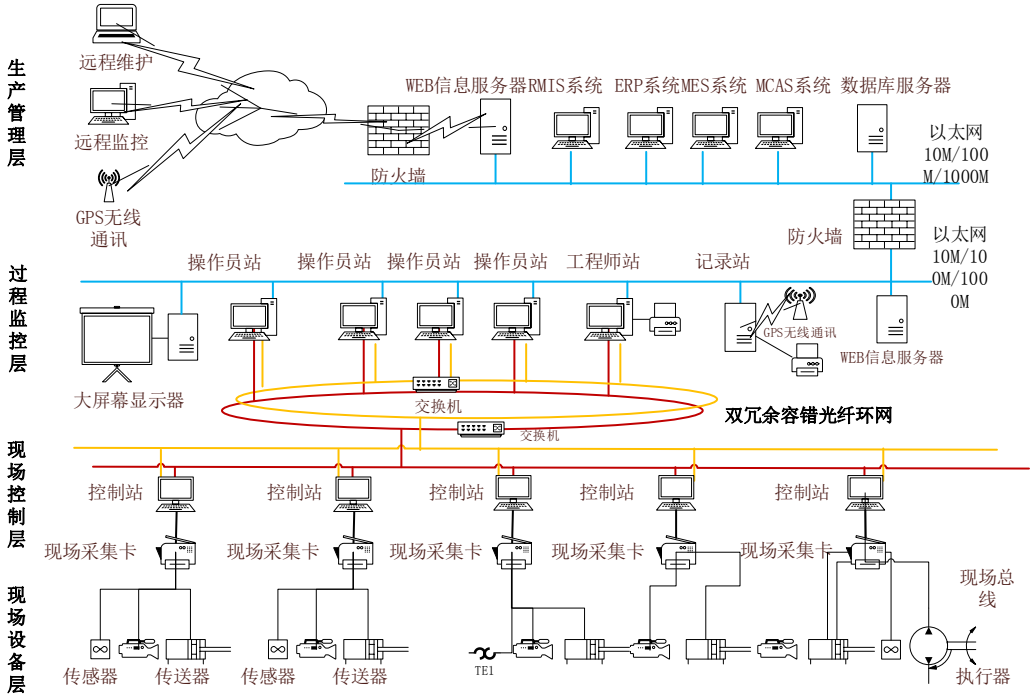


图1　工业控制系统网络架构图

1）生产管理层（MES系统）：提供网络的集中式管理功能；

2）过程监控层（DCS系统、PLC系统、SCADA系统）：提高网络的集中式监控功能；

3）现场控制层（RTU、DCS控制器、PLC控制器）：用于集中操作，分级管理，分布式现场控制功能；

4）现场设备层（变送器、执行器、传感器）：通过现场总线，工业协议实现现场控制功能。

组态管理软件是工业控制系统中的重要组成部分，既是过程监控层中的数据收集处理中心，也是生产管理层远程监视中心和数据转发中心，为用户提供了高可靠性实时运行环境和功能强大的开发工具。操作人员可以利用组态软件将各种功能软件进行适当的“组装连接”（即组态），极为方便地生成满足控制系统要求的应用系统，包括查看生产现场实时数据及流程画面、自动打印各种实时/历史生产报表、自由浏览各个实时/历史趋势画面、及时得到并处理各种过程报警和系统报警、修改生产过程参数和状态、管理部门提供生产实时数据等。

组态管理软件信息流图如图2所示。

2.2　工控系统安全性分析

工业控制系统安全风险主要包括工业协议隐患、操作系统漏洞、服务器中心化风险、组态管理软件风险、安全策略风险等五个方面：1）目前Modbus、DNP、ProfiBus等主流的工业控制协议在设计时还都是主要考虑通讯的可用性和实时性，对安全性普遍考虑不足，缺乏身份认证、数据加密保护等机制；2）数量庞大的操作员站、工程师站安装Windows操作系统和Linux系统，网络管理员出于对保障企业日常生产的稳定运营考虑，担心影响效率，基本不对操作系统进行版本和补丁升级，导致系统中各类站点存在大量的安全漏洞；

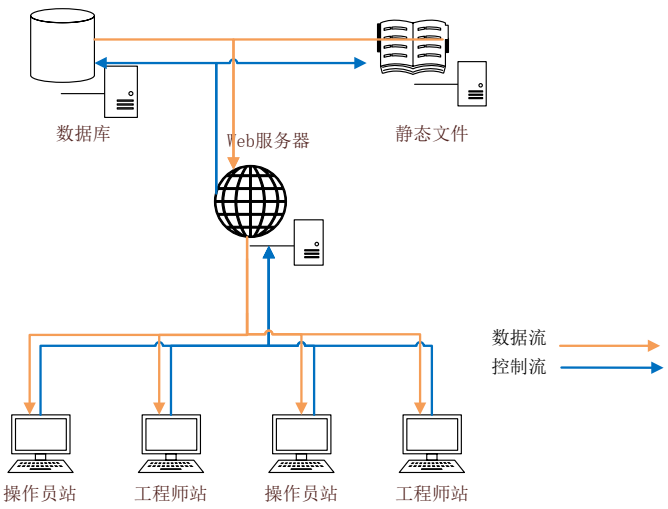


图2　组态管理软件信息流图

3）大量的运营数据和现场控制文件集中保存在服务器上，一旦服务器遭到网络攻击，数据被篡改，将导致整个工业控制系统瘫痪，给企业带来无法估量的损失。另外，地震、强风、暴雨等自然灾害也是影响工业控制系统物理安全的重大威胁，易造成设备损毁、网络瘫痪、数据丢失等工业事故；4）组态管理软件中，很多登录界面用户密码安全强度低，服务器端缺乏身份认证措施。SQL注入式、DoS攻击、伪装身份等攻击方式极大地威胁着组态管理软件及后台数据库服务器数据的安全，数据被篡改或丢失后的恢复能力不足，数据可信度不高。另一方面，工业控制系统中由于组态管理软件很少有安全测试和审计，系统出现故障除了更新很难进行取证；5）安全策略及技术标准不够完善，信息管理员及系统操作人员安全意识不足也是很大风险[4][5]。

当前，工业协议隐患、操作系统漏洞等工控系统风险可以采用异构冗余、执行裁决等技术实现工控系统内生安全拟态防御思想，增强系统安全性；安全策略也可以通过企业相关安全管理规范和规章制度的制订与员工安全培训得以完善和提升。而工控系统服务器中心化及数据可篡改、组态管理软件缺乏行为监管等安全风险则主要是通过在系统内计算机系统安装防护杀毒软件和网络防火墙解决[6][7][8]，实时性低，安全效率不高，需要引入新的技术手段或设计新的系统架构才能有效解决。

2.3　区块链分析

区块链技术源自比特币，是由网络中不同节点共同维护，使用密码学保证传输和访问安全，从而实现数据一致存储、难以篡改的分布式记账系统。按照系统是否具有节点准入机制，区块链可分类为联盟链、私有链、公有链。在联盟链和私有链中，节点加入退出需要系统的许可；而在公有链中，节点可以随时自由加入和退出。从技术角度看，区块链技术以P2P网络技术、分布式账本技术、非对称加密、共识机制技术和智能合约技术等五大技术作为技术支撑[9]-[15]：

1）P2P网络技术，根据其路由查询结构可以分为四种类型，分别是集中式、纯分布式、混合式和结构化模型。这四种类型也代表着P2P网络技术的四个发展阶段。比特币采用的是混合式模型，而现今公链大多采用的是结构化模型。在结构化网络的具体实现上，大都采用DHT（Distributed Hash Table，分布式哈希表）算法的思想。基于DHT算法思想的具体实现方案有Chord、Pastry、CAN和Kademlia等算法。

2）分布式账本技术，本质上是一种可以在多个网络节点、多个物理地址或多个组织构成的网络中进行数据分享、同步和复制的去中心化数据存储技术。

3）非对称加密，采用椭圆曲线加密算法（Elliptic Curve Cryptography，简称ECC），应用场景主要包括信息加密、数字签名和登录认证等。

4）共识机制技术，目前较为主流的共识算法

有 POW （Proof Of Work）、POS（Proof of Stake）、DPOS

（Delegated Proof of Stake）、PBFT（Practical Byzantine Fault Tolerance）等。每种算法在实际应用中时有各自的优缺点。不同应用场景中的区块链往往会采用不同的共识算法。

5） 智能合约技术，智能合约程序不只是一个可以自动执行的计算机程序，它本身就是一个系统参与者，对接收到的信息进行回应，可以接收和储存价值，也可以向外发送信息和价值。这个程序就像一个可以被信任的人，可以临时保管资产，总是按照事先的规则执行操作。

从安全优势上看，区块链技术使用全新的加密认证技术和去中心化共识机制去维护一个完整的、分布式的、不可篡改的账本，让参与者在无需相互认知和建立信任关系的前提下，通过一个统一的账本系统确保信息安全。区块链在点对点网络上由许多分布式节点和计算机服务器来支撑，任何一部分出现问题都不会影响整体运作，而且每个节点都保存了区块链数据副本。所以区块链可提供极高的业务连续性、可靠性、容错性，能够有效预防故障与网络攻击。此外，由于所有文件都能够以代码或分类账的形式体现，通过对区块链上的数据处理程序进行设置，智能合约及自动交易就可能在区块链上实现，从而提升业务自动化水平[16]-[21]。

3 FICS 工控系统安全解决方案

针对现有工控系统中的服务器中心化及数据可篡改、组态管理软件缺乏行为监管等问题，综合考虑联盟链弱中心化、强可控性、强拓展性、交易速度较快等特点，Fabric 可插拔实现各种功能的模块化架构，具有强大的容器技术来承载各种主流语言的技术优势[22][23]，我们设计了一个基于 Fabric 的工控系统安全解决方案——FICS（Fabric based Industry Control System），其网络架构如图 3 所示。

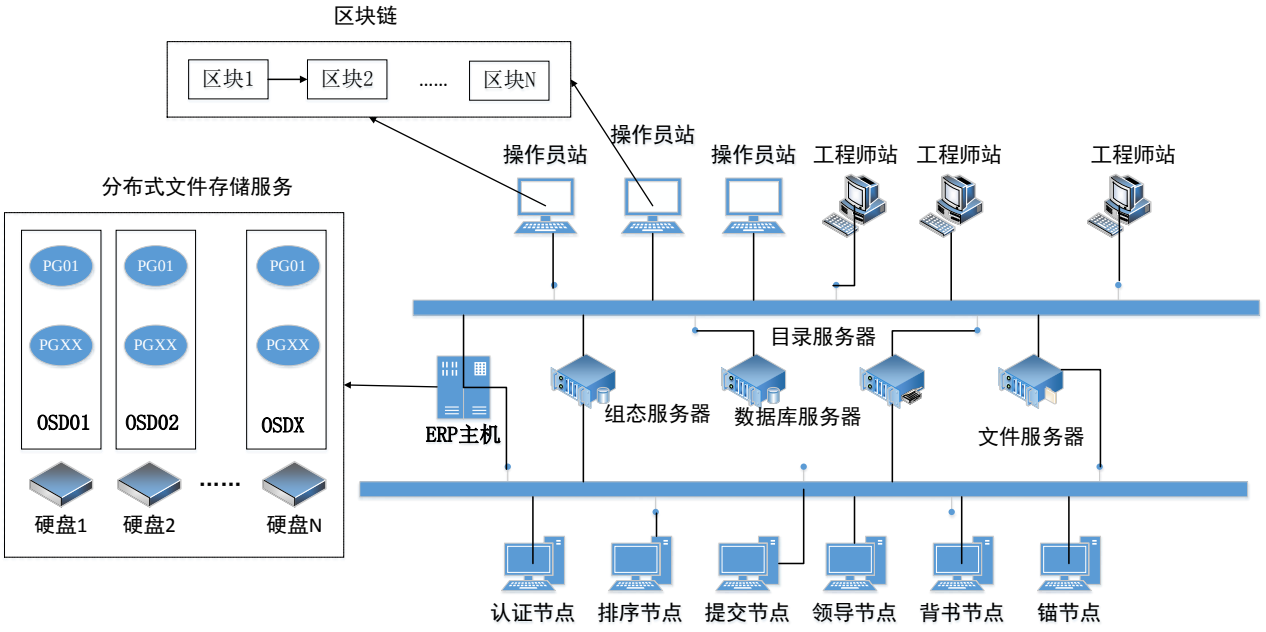


图 3 FICS 系统网络结构图

FICS 系统中包括了传统工业控制系统中的普通节点、承担区块链系统监管角色的特殊节点、分布式文件系统及区块链。其中，普通节点包括：组态服务器、数据库服务器、文件服务器、工程师站、操作员站、目录服务器等；承担区块链系统监管角色的特殊节点包括以下类型：

背书节点（Endorser）：主要提供 Process Proposal 方法供客户端调用，完成对交易提案的背书处理；每一个安装了智能合约的节点，都可以成为背书节点。

提交节点（Committer）：负责维护区块链和账本结构。

排序节点（Orderer）：负责区块链排序。

认证节点（CA）：负责网络中所有证书的管理，实现标准的PKI（Public Key Infrastructure）架构。

领导节点（Leader）：一个机构可能在某个通道中有多个节点，其中只需要一个领导节点来接收交易，然后由它负责将交易分发给其他节点。

锚节点（Anchor）：用于机构之间的通信，它使得不同机构间的对等节点了解彼此的存在。

上述特殊节点是FICS系统实现网络监管功能的必要构成部分。

3.1 FICS安全访问机制

FICS普通节点成员身份认证基于标准的X.509证书，采用PKI体系为每个成员生成数字证书以标识用户身份。FICS利用PKI体系发布数据证书，结合MSP（Membership Service Provider）组件进行身份认证和权限控制。认证节点提供对工程师站、操作站、组态服务器、数据库服务器等节点用户登录和注册的数字证书管理功能。系统通过MSP标识检查身份证书有效性、证书路径检查是否存在用户证书到认证节点的有效路径及CRL检查证书是否被吊销完成身份认证。

工程师站、操作站、组态服务器、数据库服务器等运营生产节点成员必须被许可才能加入网络，通过实体注册来获得长时间的根据实体类型生成的身份凭证。在用户使用过程中，这样的证书允许操作证书颁发机构TCA（Transaction Certificate Authority）颁发匿名证书。操作证书被用来对提交交易授权。操作证书存储在区块链中，并对审计集群授权；否则操作是不可链接的。FICS系统中各节点必须注册到系统中，获得证书后才能进行区块链操作，如图4所示。

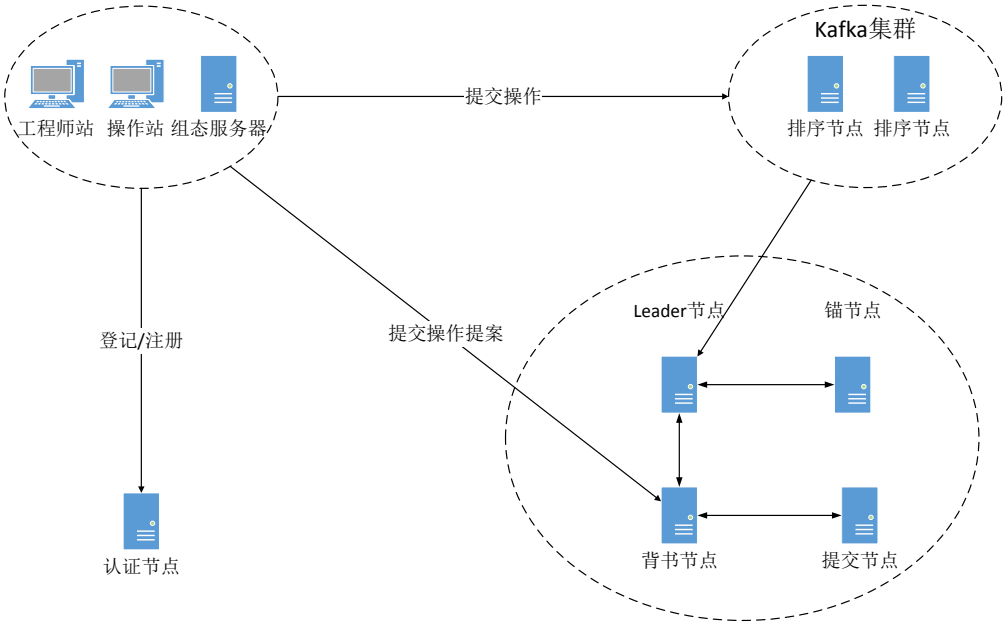


图4 FICS节点安全访问

安全访问机制提供了工程师站、操作站、组态服务器、数据库服务器等运营生产节点接入网络的身份合法性，提升了工业控制系统的网络安全性。

3.2 FICS监管可信机制

为了提高FICS系统的整体安全可信度，需要其中承担系统监管功能的特殊节点在接入网络时具备不可篡改的身份可信。为此，FICS在认证节点、背书节点、排序节点、领导节点、锚节点等监管节点设备上加入了可信根，通过集成专用微控制器在软件栈初装或重启时对其进行静态度量和验证。先启动的软件对后一级启动的软件进行度量，实现基于硬件的物理可信传递，如图5所示。

特殊节点启动时检测BIOS和操作系统的完整性和正确性，保障其硬件配置和操作系统没有被篡改过，所有系统的安全措施和设置都不会被绕过；在节点设备启动后，对所有的软件应用可进

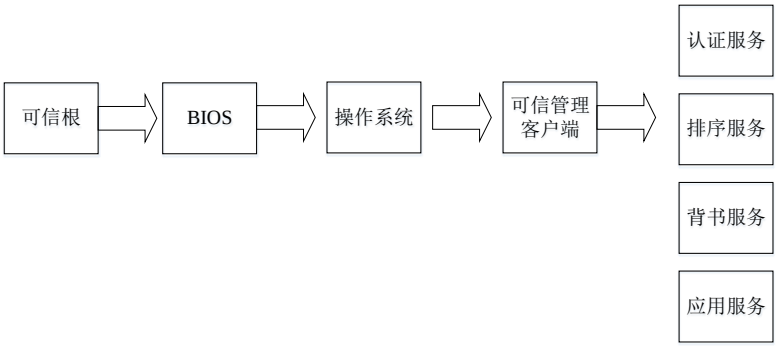


图5　物理可信链

行实时监控，若发现应用被篡改立即采取保护措施。

监管可信机制保证了特殊节点系统和应用的完整性，从根本上解决了特殊节点的信息安全风险，使其具备可信的监管功能。

3.3　FICS文件账本管理

FICS系统中工控生产运营节点上的文件以账本方式进行管理。文件账本由区块链和状态数据库两部分组成。每个文件进行指纹采集即哈希计算获得摘要信息，并结合操作时间、操作名称、操作人信息，形成一个区块。每个区块中包含若干个文件操作的数据，不同区块所包含的文件操作数量可以不同。区块之间用哈希链关联：每个区块头包含该区块所有操作的哈希值以及上一个区块头的哈希值。链式结构确保FICS中每个区块的数据不可更改以及每个区块之间的顺序关系不可更改，如图6所示。

FICS系统中每个对文件记账的节点都是互相复制的状态机，节点之间需要保持相同的账本状态。为了实现系统中节点状态的一致性，每个节点需要通过共识过程对账本状态的变化达成一致性的认同。FICS系统中账本的一致性共识过程包括3个阶段：背书、排序和校验，如图7所示。

系统中工程师站1文件系统上有文件更新，通过客户端发送文件更新提案给背书节点。背书节点对文件更新请求合法性进行审查，如果审查通过背书节点对此提案背书签名后返回给工程师站1。工程师站1将签名的提案发送给排序节点，排序节点对一段时间内收到的提案打包并进行排序生成区块。排序节点通过消息将文件账本分发给系统内所有记账节点，每个记账节点各自验证新账单并提交到本地账本中。经过验证，区块中的更改会被标记为有效或无效，通过事件返回给最先发起更新请求的节点客户端。

文件账本管理功能提供了工控系统中文件账本的强一致性，从而实现了文件数据的不可篡改

3.4　FICS智能合约管理

FICS中智能合约需要实现的业务逻辑主要包括：

1）系统管理员节点可以对整个系统网络层级划分、域划分以及域内操作站、控制站的构建与管理。

2）工程内域的划分，域内操作站、控制站的组建与管理，工程的内不同站点分组与角色设定的权限与配置。

3）控制站下硬件组态管理，如何实现具体模块参数组态管理。其中包括参数分组、参数配置以及参数组态管理。

4）用户程序编译管理和组态下载。根据控制站内部位号、数据结构、功能块等资源自定义用户程序，控制站内软件资源的组态管理，用户程序的下载实现。

FICS系统中的智能合约是对链代码的操作。一次交易可分为部署和调用两个阶段，如图8所示：

1）部署智能合约：由系统管理员创建新的链代码，并且用链代码程序名作为参数调用Fabric SDK API，将链代码上传至区块链，完成智能合约部署。

2）调用智能合约：当FICS中生产运营节点发生文件操作时，自动触发运行链代码。链代码执行并修改相应的状态，返回输出，完成智能合约调用。

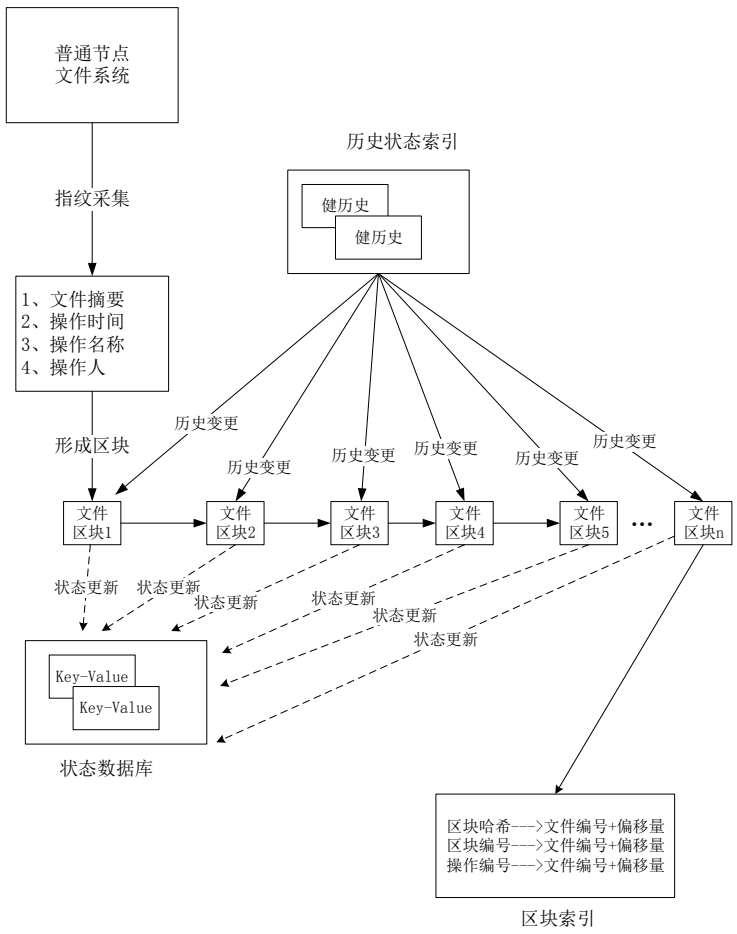


图6　文件账本流程

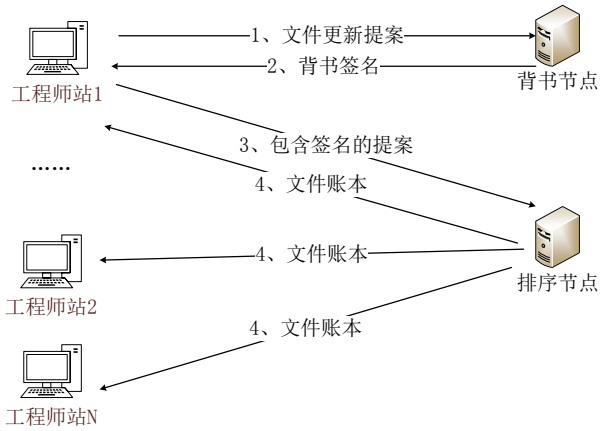


图7　FICS 系统共识流程

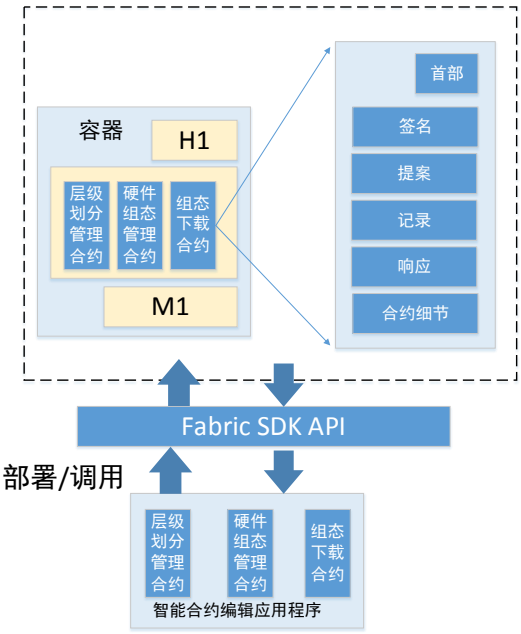


图8　FICS 智能合约管理

4　FICS 分布式文件存储

FICS 需要存储的数据包括生产管理层中的运营数据、过程监控层中的历史监控数据和组态管理数据、现场控制层中的控制程序等。数据集中存储即服务器中心化，一旦服务器遭到网络攻击

或者宕机，整个系统将陷入瘫痪。因此，FICS采用分布式文件存储，来解决数据集中存储带来的安全的风险。

当前，分布式存储有包括Ceph、HDFS、Swift、GFS、Luster等多种实现技术。存储根据其类型，可分为块存储，对象存储和文件存储。在主流的分布式存储技术中，HDFS/GPFS/GFS属于文件存储，Swift属于对象存储，而Ceph可支持块存储、对象存储和文件存储，故称为统一存储。几种主流分布式存储技术的特点比较如表1所示：

表1　几种主流分布式存储技术比较

分布式存储	Ceph	GFS	HDFS	Swift	Luster
平台属性	开源	闭源	开源	开源	开源
系统架构	去中心化	中心化	中心化	去中心化	中心化
数据存储方式	块、文件、对象	文件	文件	对象	文件
元数据节点数量	多个	1个	1个（主备）	多个	1个
数据冗余	多副本/纠删除码	多副本/纠删除码	多副本/纠删除码	多副本/纠删除码	无
数据一致性	强一致性	最终一致性	过程一致性	弱一致性	无
分块大小	4MB	64MB	128MB	视对象大小	1MB
适用场景	频繁读写场景	大文件连续读写	大数据场景	云的对象存储	HPC超算

Ceph相比其它分布式存储技术，其优势点在于：充分利用了存储节点上的计算能力，在存储每一个数据时，都会通过计算得出该数据存储的位置，尽量将数据分布均衡。同时，由于采用了CRUSH、Hash等算法，在分布存储规模扩大增大时，性能并不会下降。目前，Ceph已得到众多云计算和存储厂商的支持，成为应用最广泛的开源分布式存储平台[24][25]。综合考虑数据存储方式、系统架构、数据一致性、系统规模扩展性等，我们采用Ceph来实现FICS的分布式文件存储。FICS中基于Ceph的分布式文件存储如图9所示。

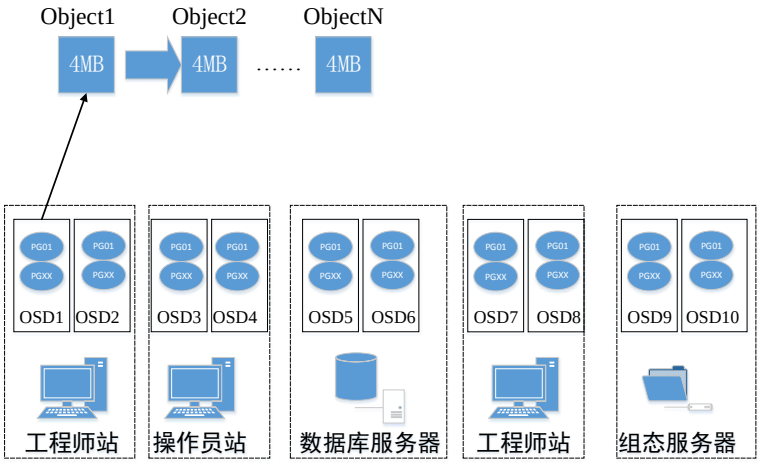


图9　FICS分布式文件存储

FICS中生产运营节点（工程师站、操作员站、组态服务器、数据库服务器等）安装Ceph，调用接口将文件系统上的文件分成多个Object对象以分布式多副本方式存储到其他不同生产运营节点。其分布式文件存储核心组件包括：对象存储设备OSD（Object Storage Device）、监视器Monitor、元数据服务器MDS（Meta Data Server）。OSD主要完成数据存储、数据复制、数据平衡、数据恢复，并与其它OSD间进行心跳检查，将变化情况上报给Monitor。OSD中的放置组PG（Placement Group）用于放置Object的一个载体，在OSD上的存在形式就是一个目录。Monitor负责监视集群，维护集群的健康状态，同时维护集群中的各种Map图，比如OSD Map、Monitor Map、PG Map和CRUSH Map。MDS负责保存文件系统服务元数据。

FICS网络中所有工程师站、操作员站、组态服务器、数据库服务器等生产运营节点每个分区初始化为OSD。文件系统中的文件被分割为多个4MB大小的Object对象，分别以多个副本的形式保存在其他节点的OSD上，每个OSD划分为多个PG。由此，实现FICS中所有生产运营节点文件的分布式存储。

5 实验分析

本文使用云平台技术搭建两套实验系统，分别是基于云平台的ICS实验系统（ICS）和基于云平台的FICS实验系统（FICS），通过对两套实验系统分组攻击，得出实验数据对其安全性能进行量化对比分析。

5.1 实验环境

实验系统逻辑上划分为云计算虚拟化平台和实验仿真平台两部分。云计算虚拟化平台通过云计算虚拟化调度和管理为实验仿真平台虚拟各种实验操作环境，试验人员可用其进行各种计算机、网络设备和安全设备等操作，仿真工控系统运行过程。实验平台的硬件是由管理控制台、计算服务器集群、存储阵列、交换机、网关等设备组成，软件由云计算虚拟化软件和实验仿真平台虚拟机组成如图10所示。

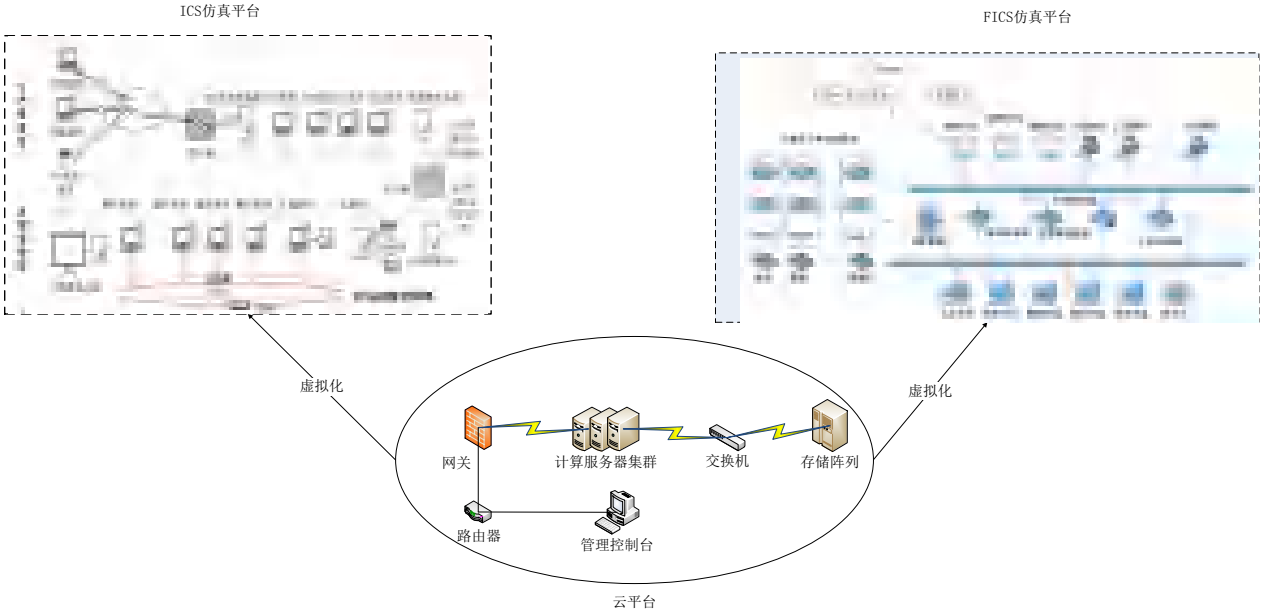


图10 实验环境

5.2 安全测试

按破坏系统安全三要素（完整性、可用性、保密性）为目的分为三个攻击分组如表2所示，建立相应的攻击模型对两套实验系统进行安全测试。

表2 攻击分组

攻击方式	攻击目标	攻击形式	攻击站点
完整性攻击	篡改文件数据,造成系统因数据异常引发故障；	注入攻击、虚假数据、恶意删改；	工程师站、操作员站、组态服务器、数据库服务器
可用性攻击	瘫痪或延迟服务器提供的服务	拒绝服务攻击、物理宕机	组态服务器、数据库服务器
保密性攻击	非法非授权控制系统中计算机	密码破解、身份伪造	组态服务器、数据库服务器、工程师站、操作员站

1） 完整性攻击建模

完整性攻击建模是构建对系统数据进行篡改的一种攻击模型，攻击对象分别为组态服务器上位号参数、工程师站和操作站上用户程序。

位号参数在组态服务器上存储格式如表3所示。

根据位号参数配置格式，采用攻击模型如下：

TagType = （TagType+1）%7；

表3　位号参数

位号参数	参数格式及取值
CtrlUUID	控制站UUID(限定16字节)
TagName	位号名称(最大16字节,只能由英文、数字、下划线组成,英文开头,不区分大小写。控制站下唯一)
TagType	位号类型(1-AI;2-AO;3-DI;4-DO;5-SA; 6-SI; 7-SD)
TagNo	位号序号(AI从10000开始;AO从20000开始;DI从30000开始;DO从40000开始;SA从50000开始;SI从60000开始;SD从70000开始)
ModuleID	IO卡件模块ID
RackAddr	机架号(1~4)
CardAddr	卡件号(1~64)
ChannelAddr	通道号(1~32)
CyclePeriod	周期时间(单位ms)
Descript	描述(最大64字节)

TagNo = （TagNo+10000）%10000；

RackAddr = （RackAddr +1）%4；

CardAddr = （CardAddr +1）%64；

ChannelAddr = （ChannelAddr +1）%32；

CyclePeriod = CyclePeriod +5；

用户程序在工程师站或操作员站上存储格式如表4所示。

根据用户程序存储格式，采用攻击模型如下：

ProgramNo = ProgramNo + 0xff；

CodeLang = （CodeLang + 3）%7；

CycleNum = （CycleNum + 10）%2；

PhaseIndex = （PhaseIndex +7）%10；

PriorityLevel = （PriorityLevel +2）%3；

根据上述攻击模型，采用注入攻击方式将工程师站、操作员站、组态服务器上的数据进行篡改，破坏数据完整性；

2） 可用性攻击建模

可用性攻击模型是构建使攻击对象无法有效提供服务的一种攻击模型，攻击对象分别为组态服务器和数据库服务器。组态服务器内部访问ip地址为192.168.17.102，端口为8080；数据库服务器内部访问ip地址为192.168.17.122，端口为9000。采用多攻击源分时攻击模型如下表5所示：

表4　用户程序参数

用户程序参数	参数格式及取值
Name	用户程序名称(最大45字节)
CtrlUUID	控制站UUID
ProgramNo	程序序号(从1开始)
CodeLang	代码语言(1-ST; 2-IL; 3-FBD; 4-LD; 5-SFC; 6-CFC; 7-C)
Descript	描述(最大128字节)
CycleNum	周期倍数(固定为1,2,5,10)
PhaseIndex	相位序号(限定1~10)
PriorityLevel	优先等级(1-低;2-中;3-高)
Password	密码(明文md5后32字节)

表5　多攻击源分时攻击模型

攻击源	攻击间隔(单位:ms)	攻击方式	攻击目标

192.168.17.n
（n取值101~200）
	0.01*(n%100)	n%10==1: http get;
		n%10==2: UDP DNS Query Flood;
		n%10==3: Connection Flood;
		n%10==4: UDP Flood;
		n%10==5: SYN Flood;
		其他:ICMP Flood
			n%2==1: 组态服务器;
			n%2==2: 数据库服务器;
			其他:组态服务器

根据上述攻击模型，在云平台上创建100个攻击源服务器，每个攻击源服务器运行500个Docker，每个Docker运行1个攻击进程，每个进程运行100个线程。因此，总共$5*10^6$个攻击源分时采用不同的攻击方式发包，平均每个包1000Byte，攻击峰值流量达500GB/s。

3） 保密性攻击模型

保密性攻击模型是构建非授权非法获取信息

的一类攻击模型，攻击对象分别为组态服务器、数据库服务器、工程师站、操作员站。采用基于差分故障分析方法[26]的攻击模型。

5.3 安全分析

在实验中，从数据出错率、平均系统响应时间、异常行为监管识别次数等作为性能指标，对ICS系统和FICS系统的安全性能进行量化对比分析。

1) 数据出错率

ICS系统和FICS系统在完整性攻击安全测试下平均数据出错率对比如表6所示。

表6 注入式攻击下数据出错率

实验系统	SQL注入	恶意删除	数据造假	JSON漏洞劫持	JSON注入
ICS	失败	成功	成功	成功	成功
FICS	失败	失败	失败	失败	失败

2) 平均系统响应时间

在6轮攻击源分时可用性攻击下，ICS和FICS平均系统响应时间如图11所示。

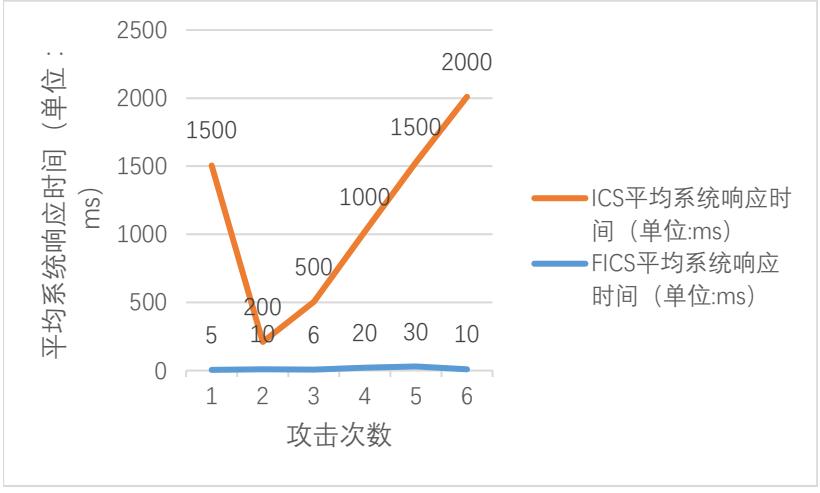


图11 平均系统响应时间

3) 异常行为监管识别次数

分别进行10/100/1000/10000次差分故障分析攻击，系统对异常行为监管识别次数如图12所示。

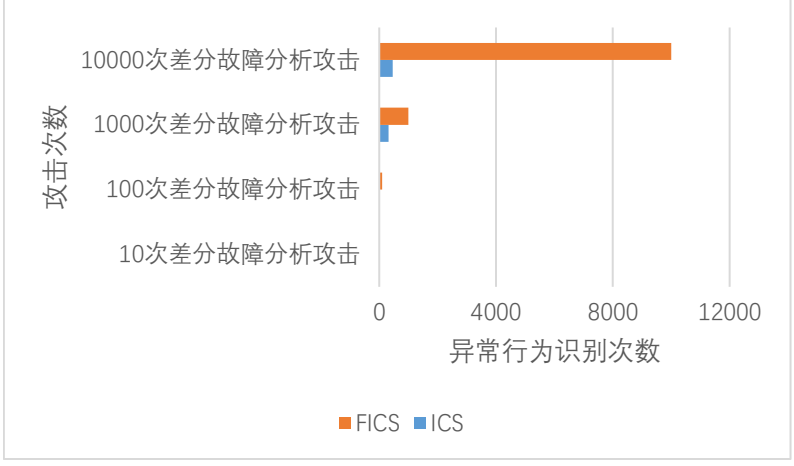


图12 异常行为监管识别次数

通过上述攻击模型的安全测试数据量化对比表明，FICS系统数据出错率为0，系统响应时间波动较小，对异常行为识别达到100%，而ICS虽然采取了杀毒软件和网络防火墙等防护手段，只能在一定概率下有效防止数据篡改，抵御DDOS攻击，监管异常行为。因此，FICS比ICS系统在数

据防篡改、服务器去中心化、异常行为监管识别等方面具有明显的安全优势。

6 结束语

本文分析了工控系统的现状及其中的安全问题，研究了区块链的安全特性。针对工控系统中服务器中心化单点故障、数据恶意篡改等安全问题，提出了基于Fabric联盟区块链技术的工控系统安全解决方案，并结合Ceph分布式文件存储系统实现了工控系统生产运营节点文件的分布式存储，化解了传统工控系统中服务器中心化带来的安全风险，提升了数据安全性和行为可监管的自治性效果。本文中提出的FICS技术方案也可为研究制定基于区块链的工控系统技术实现标准提供参考。

参考文献:

[1] 郑少波,徐伟,石彬．工业控制系统安全现状．网络安全技术与应用,2020年第5期111-113.

[2] 黄容生．工业控制系统信息安全防护研究．网络安全技术与应用，2020年第2期93-94，

[3] Kravchik Moshe and Shabtai Asaf. Detecting Cyber Attacks in Industrial Control Systems Using Convolutional Neural Networks. CPS-SPC@CCS，pp. 72-83，2018.

[4] 陈政熙,张家鹏．工业控制系统安全运维模式研究．自动化仪表，2020年第5期98-102.

[5] MichaelDodson, Beresford R. Alastair and Vingaard Mikael. Using Global Honeypot Networks to Detect Targeted ICS Attacks. 2020 12th International Conference on Cyber Conflict (CyCon)，pp. 275-291，2020.

[6] 屠袁飞,苏清健,杨庚．一种适用于工业控制系统的加密传输方案．电子与信息学报,2020年第2期348-354.

[7] SaranyanSenthivel, ShreyDhungana, HyungukYoo, Ahmed Irfan and Roussev Vassil. Denial of Engineering Operations Attacks in Industrial Control Systems. CODASPY，pp. 319-329，2018.

[8] 赵峰,马跃强．基于等保2.0工业控制系统网络安全技术防护方案的设计．网络安全技术与应用,2020年第5期109-111.

[9] Zheng, Zibin, et al. Blockchain challenges and opportunities: a survey. International Journal of Web and Grid Services 14（2018）：352-375.

[10] 周艺华,李洪明．基于区块链的数据管理方案．信息安全研究，2020年第1期37-45.

[11] Ping Yu, Shufang Zhang and Jiang Zhong. Block-chain Privacy Protection Based on Fully Homomorphic Encryption. Proceedings of the 2019 3rd International Conference on Innovation in Artificial Intelligence，pp. 239-242，2019.

[12] 魏凯,卿苏德,张奕奔,黄胜,徐晓旻,焦丽梅．工业区块链应用白皮书1.0版本，2019年2月．

[13] Li Daming, Cai Zhiming, Deng Lianbing, Yao Xiang and Wang Haoxiang Harry. Information security model of block chain based on intrusion sensing in the IoT environment. Cluster Computing，pp. 1-18，2018.

[14] 李瑾,仵松颀,张森林,陆月明．基于区块链的分布式电能量数据可信存储机制．网络与信息安全学报,2020年4月第6卷第2期．

[15] GeetanjaliRathee, AshutoshSharma, HemrajSaini, Kumar Rajiv and Iqbal Razi. A hybrid framework for multimedia data processing in IoT-healthcare using blockchain technology. Multimedia Tools and Applications，pp. 9711-9733，2019.

[16] 杨敬丽,张瑞洋．区块链应用加速落地呼唤行业标准创新．中国标准化，2019年第1期6-13.

[17] Jian Weng, Jiasi Weng, Jianan Liu, Kaimin Wei and Weiqi Luo. Method for constructing software defined network control layer security mechanism based on block chain.

[18] 工业互联网中区块链应用场景和业务需求,CCSA.

[19] Yu Wanjun and Huang Shiyuan. Traceability of Food Safety Based on Block Chain and RFID Technology. 2018 11th International Symposium on Computational Intelligence and Design (ISCID)，pp. 339-342，2018.

[20] 汪允敏,李挥,王菡等．区块链在工业互联网标识数据管理策略研究[J]. 计算机工程与应用，2020(1)：1-8.

[21] You Barco, MatthiasHub, You Mengzhe, Xu Bo, Yu Mingzhi and Uemlianin Ivan. Block Chain based Intelligent Industrial Network (DSDIN). arXiv：Computers and Society，Volume abs/1809. 06551，2018.

[22] 刘子腾．基于fabric的国际贸易信用证付款系统的设计与实现[D]. 华中科技大学，2018.

[23] ElliAndroulaki, ArtemBarger, VitaBortnikov, ChristianCachin, KonstantinosChristidis, Caro De Angelo, DavidEnyeart, ChristopherFerris, GennadyLaventman, YacovManevich, SrinivasanMuralidharan, ChetMurthy, BinhNguyen, ManishSethi, GariSingh, KeithSmith, AlessandroSorniotti, ChrysoulaStathakopoulou, MarkoVukolic, Cocco Weed Sharon and Yellick Jason. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. EuroSys，pp. 1-15，2018.

[24] 陈阳,王丹．Ceph RadosGW对象存储集群的部署与优化．现代计算机，2020年第14期17-20.

[25] BodonJeong, Khan Awais and Park Sungyong. Async-LCAM: a lock contention aware messenger for Ceph distributed storage system. Cluster Computing，pp. 1-12，2018.

[26] 李玮,谷大武．基于密钥编排故障的SMS4算法的差分故障分析．通信学报，2008年 第10期

基于拟态的MCU设计及应用验证

张明权[1]，于洪[1]，魏帅[1]，崔超[2]，黄丽波[3]

[1]解放军战略支援部队信息工程大学，河南郑州 450002；

[2]天津市滨海新区信息技术创新中心，天津 300450；

[3]92524部队

摘　要：微控制单元MCU（Microcontroller Unit）在工业控制领域，以及物联网应用中占据至关重要的地位，其安全性一直较为薄弱，近年来，越来越多地成为黑客的攻击目标。提高MCU的安全性迫不及待。本文从内生安全的角度考虑MCU安全解决方案，基于“动态异构冗余“的拟态防御模型提出一种拟态MCU架构，在此基础上设计实现了拟态MCU原型板卡，开发了基于拟态MCU的应用程序。最后通过实验模拟多种攻击场景，验证了拟态架构MCU可行性和安全可靠性。

关键词：拟态防御、MCU、网络安全

MCU design and application verification based on minic defence

Zhang Mingquan[1], Yu hong[1], Wei Shuai[1], Cui Chao[2], Huang Libo

1.Information Engineering University , Zhengzhou Henan 450002;

2.Information Technology Innovation Center of Tianjin Binhai New Area, Tianjin 300450. 3. 92524 Troops

Abstract: MCU(Microcontroller Unit) occupies a vital position in the field of industrial control and Internet of Things applications,and its security has been relatively weak. In recent years, it has increasingly become a target of hackers. Improving the safety of the MCU cannot wait. This paper considers MCU security solutions from the perspective of endogenous security, and proposes a mimic MCU architecture based on the mimic defense model of "dynamic heterogeneous redundancy". On this basis, a mimic MCU prototype board is designed and implemented, and Developed the application program based on the mimic MCU. Finally, a variety of attack scenarios are simulated through experiments to verify the feasibility and safety and reliability of the mimic architecture MCU.

Key words: Mimic defense; MCU; Network security

1　引言

MCU在工业控制中的应用非常广泛，低能耗电机控制设备、高精度工业仪器控制设备、特殊工作环境电子控制设备、精细动力控制设备等控制设备的核心器件均为MCU，其应用领域遍布电力、能源、交通、市政等与国计民生息息相关的各种行业。随着物联网的到来，MCU将扩展到生活的方方面面。MCU可以被认为是物联网无数终端节点的中枢神经，负责对传感器捕获的信息进行处理、计算和下达控制指令。

有关研究报告指出，全球8大MCU企业没有中国的企业，中国大陆和中国台湾MCU企业在中国市场各有10%的份额。中国单片机30年，MCU厂商很多，但多数混迹8/16位低端市场，设计能力低、缺少战略发展规划和资金支持，难以改变中国企业在MCU市场的弱势地位。我国的MCU在供应链上受制于人，在安全上面也因此存在很大的安全隐患。除了MCU设计造成的漏洞，还有可能有恶意存在的后门，这些都会对依赖MCU的

基金项目：高安全等级网络基础设施关键装备核心芯片及软件研发（No.2017ZX01030301）

行业和应用带来巨大的安全隐患。

在现在的以微控制器为核心的硬件系统中，都是采用单个MCU作为控制单元。单个MCU可能由于单一器件损坏，外部攻击或内部漏洞导致系统瘫痪，计算结果错误或失效。为了破解MCU存在的安全难题，本文构建了一种基于CMD技术、采用动态异构冗余机制的MCU内生安全技术，搭建拟态MCU原型验证板卡，验证应用设备的安全性。

本文剩余部分的组织如下：第二节主要结合拟态防御思想，介绍拟态架构和拟态MCU架构及其拟态架构相关应用；第三节介绍基于拟态MCU的总体架构和安全机理；第四节介绍应用程序设计及拟态MCU的测试通过应用演示程序验证拟态MCU的可行性和安全性。

2 背景

互联网可能已成为基于MCU的系统中最受欢迎的攻击切入点，但它远非唯一的。无担保或安全性不足的诊断端口一直容易受到攻击。任何无线（或有线）通信链路也可用作入口点。即使是未连接的MCU设备或其相关ROM也会受到篡改和IP盗窃。除了MCU设计造成的漏洞，还有可能有恶意存在的后门，这些都会对依赖MCU的行业和应用带来巨大的安全隐患。

鉴于被动防御存在防护缺陷，许多国家都在开展网络信息安全主动防御技术的研究。美国的移动目标防御（MTD，moving target defenses）技术，针对外部利用未知漏洞的攻击防御，在假设内部安全可信的前提下，主要采用软件技术实现主动防御，应对逻辑层攻击[1-3]。我国邬江兴院士首创的网络空间拟态防御（CMD，cyber mimic defense）技术，采用动态异构冗余（DHR，dynamic heterogeneous redundancy）的系统架构和运行机制，既可防御外部利用未知漏洞的攻击，也可防御利用未知后门的攻击，在允许基本环境一定程度“有毒带菌”的情况下，采用软件技术和系统结构组合应用实现主动防御，可为信息网络基础设施或重要信息服务系统提供一种不依赖传统安全手段（如防火墙、入侵检测、杀毒软件等）的构造化内生安全增益或效应[4]。

随着网络空间拟态防御理论的完善，众多依据CMD技术构建的安全网络设备陆续出现。仝青等[5]设计实现了拟态防御Web服务器；马海龙等[6]设计实现了基于动态异构冗余机制的路由器拟态防御体系结构；魏帅等[7]实现了面向工控领域的拟态安全处理机架构；宋克[8]等实现了基于拟态防御的以太网交换机内生安全体系结构。这些基于拟态防御构建的安全网络设备在测试中均取得了较好的效果。拟态防御Web服务器和拟态防御路由器主要采用虚拟技术，其异构执行体主要在软件层面，底层处理器和操作系统仍为同构非冗余。综上基于拟态架构所设计的应用均取得了良好的安全增益。因此为了提高MCU应用的安全性提出了基于拟态的MCU架构。拟态安全MCU则主要从处理器层面开始实现了异构，包括上层软件和体系架构。

拟态MCU架构是基于拟态安全防御思想进行设计的计算机体系架构，它具有在主动和被动触发条件下动态地、伪随机地选择执行各种硬件变体的特性。使用拟态安全架构的处理器，在程序运行过程中，加上软件的配合，能使得内外部攻击者观察到的硬件执行环境和软件工作状况非常不确定，无法或很难构建起基于漏洞（bug）或后门的攻击链，达成降低系统安全风险的目的。

3 总体设计

3.1 总体架构

基于异构冗余的拟态MCU，通过多个异构MCU和一个调度器的架构设置，使得单个或多个MCU受到外部攻击后系统仍能保持正常工作状态或快速恢复正常工作状态。

其总体架构如图1所示。

系统工作时，上行数据经过调度器进行复制分发给四个异构MCU进行处理。来自MCU的下行数据经过调度器判决后输出唯一结果，判决结果受判决策略的控制。若MCU出现异常，拟态调度器可以通过专有接口对其进行清洗及恢复。拟态MCU能提高系统传统可靠性，和抵抗处理器硬件层面和软件层面未知漏洞/后门能力的特点。

拟态MCU整体结构包括核心板和母板两部分。核心板由异构MCU和拟态调度器组成。

异构MCU模块包括至少三个不同架构的MCU芯片以及对应MCU芯片的外围子系统，该MCU

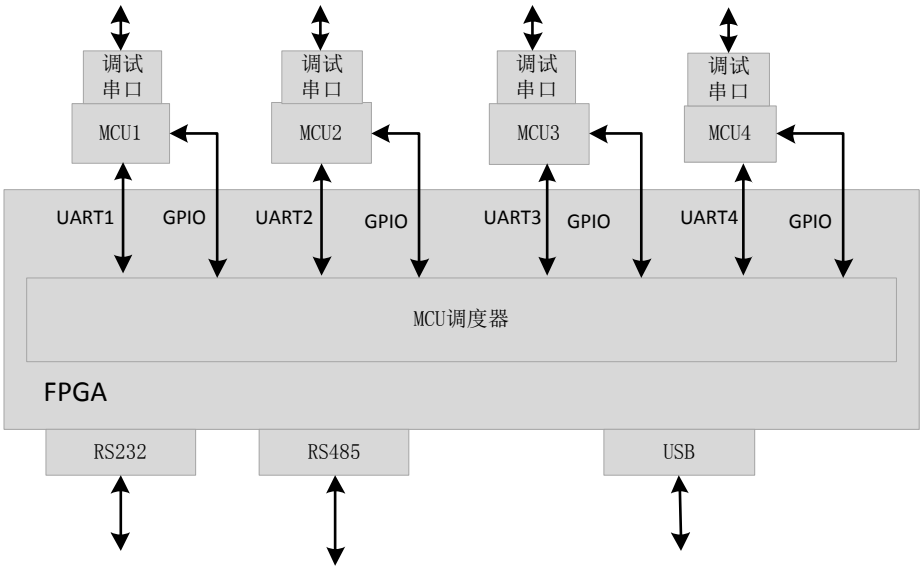


图1　拟态MCU系统结构

芯片中的一个作为备份MCU，其他作为判决MCU。异构MCU在选取时，调研了市面上现存的不同架构的多种MCU，对比了各型号之间的性能，为了在硬件层面达到尽可能高的异构度，最终选取了架构不同、性能相近的四款MCU：PIC32MX775F512L （MIPS）、 STM32F217VET6（ARM）、 AT32UC3C512C （AVR）、 MCF52255（COLDFIREV2）。

拟态调度器由FPGA实现。FPGA和每个异构MCU之间使用串口作为业务数据接口；另有一路SPI，用于调度器对MCU发送复位/清洗/检测信号。四个MCU各自有一路调试串口，用来控制和监测MCU内部程序运行状况。调度器实现的功能有：1、上行数据分发：对外接口的上行数据经过调度器后复制并分发至四个MCU；2、下行数据判决：调度器对来自三个MCU的数据进行判决，并输出唯一结果；3、执行体清洗恢复：根据判决结果和拟态策略，调度器会对被认为异常的MCU发出清洗信号。目前的清洗信号为复位信号，复位后MCU内的程序将恢复到初始状态；4、接口标签添加/解析的功能：由于有多个对外接口，调度器会在上行业务数据前增加标签指示该数据来自哪一路接口。MCU在发送下行数据时也增加标签指示该数据从那一路接口输出。每字节的业务数据前都有一字节的接口标签。标签由FPGA在上行时添加，下行时识别并删掉，因此外部设备看不到标签。

MCU母板上对外业务接口有RS232，RS485，Ethernet。此外，还包含一路USB接口，用于连接调试机上的VIO功能软件，监测MCU的工作状态和包计数。

3.2　拟态MCU安全机理

根据拟态防御理论，基于图1构建的拟态MCU安全机理如下：

（1） 各个异构体MCU独立响应并执行由调度器发送的指令，结果再分别输出到拟态调度器。

（2） 拟态调度器接收多个判决MCU和一个备份MCU的运算结果；将所有MCU芯片的运算结果进行对比，对比结果包括以下三种情况：

（3） （a1）在判决模块预设的阀值内，判决MCU中出现运算结果相同且数量最多的MCU芯片；

（4） （a2）在判决模块预设的阀值内，判决MCU结合备份MCU出现运算结果相同且数量最多的MCU芯片；

（a3）排除以上两种情况的其他情况；

（5） 若对比结果为（a1）或（a2），则将运算结果相同且数量最多的MCU芯片作为正常工作的芯片，并将该芯片的运算结果作为判决模块的唯一结果输出，若存在其他MCU芯片，则同时判决模块向其他MCU芯片输出清洗指令，并重复步骤（I）、（II）和（III），实现其他MCU芯片的全局复

位和状态恢复；

（6）若对比结果为（a3），则判定该拟态MCU工作状态极度不安全，保留其中一个MCU芯片工作，同时判决模块向其他MCU芯片输出清洗指令，并重复步骤（I）、（II）和（III）

通过上述分析，基于拟态MCU面临绝大多数的差模攻击场景时（由于各执行体的处理器架构、执行软件都是同功异构的，绝大多数的攻击只会对其中一种执行体造成影响，因此差模攻击是MCU面临的主要攻击场景），通过调度器的择多判决，均能输出正确结果。而且，这种选择机制仅仅是根据输出结果的异同进行的，不关注到底是何种原因造成的结果不一致。无论是已知的还是未知的安全威胁造成的输出不一致，该架构均能有效处理，不依赖病毒库、木马库、漏洞库等先验数据。因此，该架构可有效应对未知漏洞及后门造成的未知威胁。

4　拟态MCU应用演示

4.1　拟态MCU应用场景设计

应用演示程序用于演示验证拟态MCU的可行性。

在工控网络中，经常会使用到工控网关，其功能是将工业协议转换为以太网协议，以连接上位机，方便管理人员操作。本文中，拟态MCU验证程序设计为可以转换以太网协议和MODBUS-RTU协议的网关。

拟态MCU使用RS485接口与多个PLC终端连接，通讯协议为MODBUS-RTU；使用网络接口与上位机相连，通讯协议为UDP。具体测试场景如下图2所示：

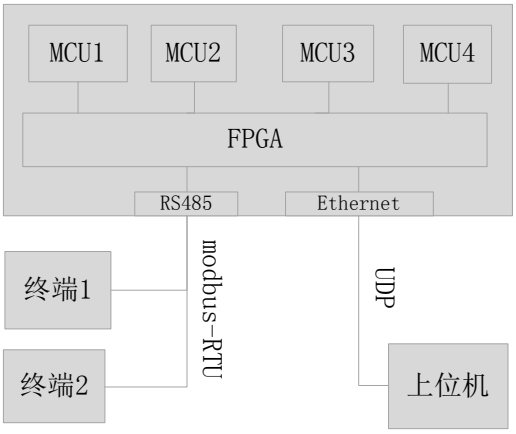


图2　基于拟态MCU的网关

4.2　拟态MCU应用程序设计

应用程序主要完成传感器数据采集及上报、上位机的控制命令接收并将其发送给相关设备。应用程序构成主要包括数据接收模块、数据上报模块、命令接收模块、命令下发模块和数据处理模块。其中数据接收模块和命令下发模块对外接口使用总线通信协议MODBUS-RTU，数据上报模块和命令接收模块对外接口使用以太网通信协议UDP。模块图如图3所示。

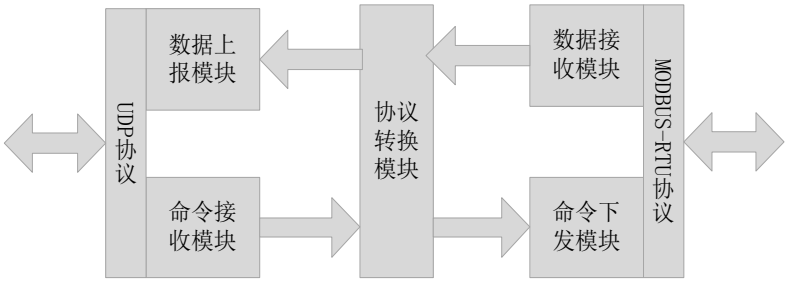


图3　MCU网关应用程序模块图

应用程序处理流程包括：上电，进行硬件的初始化，此过程中要读取硬件配置、检测与外设的连接情况。然后与上位机建立连接，连接建立成功后，应用程序的工作主要分为两部分，一部分为命令处理，一部分为数据处理。命令处理流程主要包括从上位机接收命令，并进行解析，之后按照命令指示进入相关处理。数据处理流程主要包括从MCU终端接收数据包，并封装成以太网协议包，然后发送给上位机。流程图如图4所示：

应用程序是在各异构MCU提供的集成开发环境上分开开发并编译，然后下载到MCU内部的，具有一定的异构性。

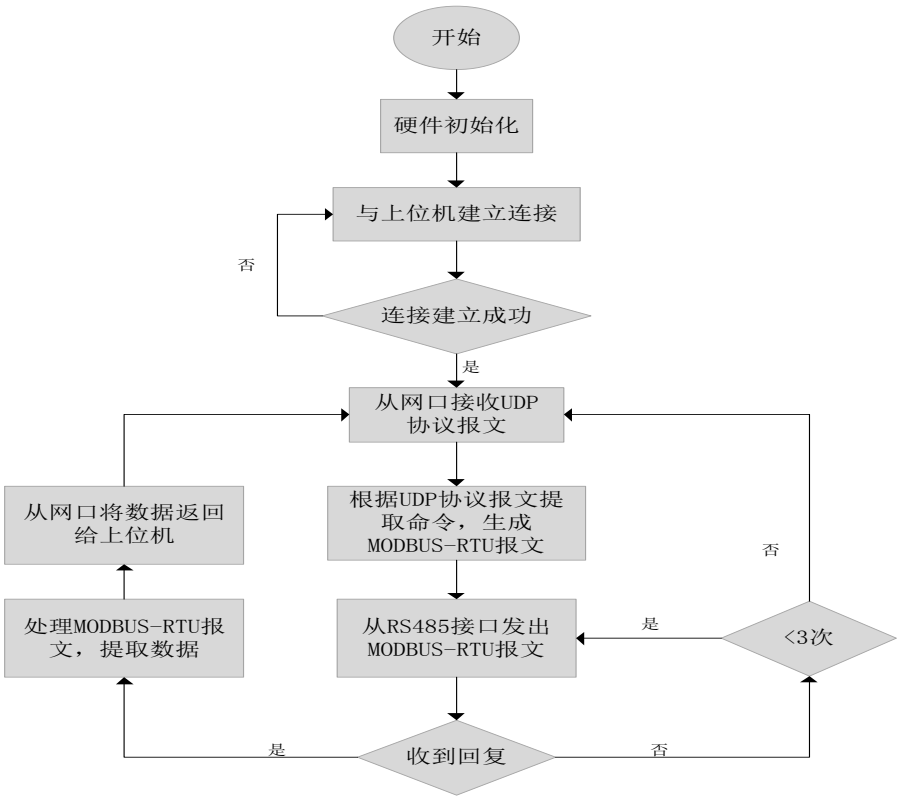


图4　MCU网关应用程序处理流程

为了方便演示，还开发了上位机界面程序，界面程序包括的功能主要有：

a）通过以太网口与基于拟态MCU的网关设备利用UDP协议通信；

b）支持用户管理功能（登录、注册、分销），支持管理员用户与普通用户身份；

c）支持指定的所连接的传感器状态查询，并将查询到的信息记录到数据库中，数据库中的信息支持图表两种方式显示；

d）支持设备管理功能，可以进行当前网关配置、设备增删等操作。

表1　各异构MCU对应的集成开发环境

处理器型号	集成开发环境
PIC32	MPLAB IDE
STM32	Keil MDK
Coldfire	codewarrior IDE for MCU 10.7
AVR	Atmel Studio 7

4.3　拟态MCU应用插桩及测试

为了验证拟态MCU对抗基于未知漏洞/后门攻击的有效性，本文还基于白盒插桩测试[9]理论，设计了MCU应用插桩。利用执行体调试串口，将MCU从调试串口接收到的数据，直接转发给FPGA。利用这种方式，模拟MCU在受到攻击时产生的异常输出，可以验证拟态MCU的安全机理。测试拓扑入如图5所示：

测试时四个MCU运行网关程序：在不开启插桩的情况下，网关正常工作，接受上位机指令，查询RS485接口所连接的传感器状态，并将数据返回上位机界面。为了验证拟态MCU的判决机制和清洗恢复功能，利用插桩模拟了几种可能遭受的攻击情景，包括差模攻击、二模攻击、共模攻击遭遇随机扰动以及正常模式遭遇随机扰动[10]。测试结果显示在上述场景下拟态模组的输出均可恢复到正常输出，且发生异常的MCU被清洗至初始状态。

当前系统上电后默认MCU1，MCU2，MCU3工作，MCU4备份。此时各MCU工作状态正常，VIO软件可以观察到MCU1，MCU2，MCU3绿灯（工作），且无错误计数。拟态模组下行业务数据正确。

（1）差模攻击测试：

构建差模攻击：差模攻击时，只有一个MCU

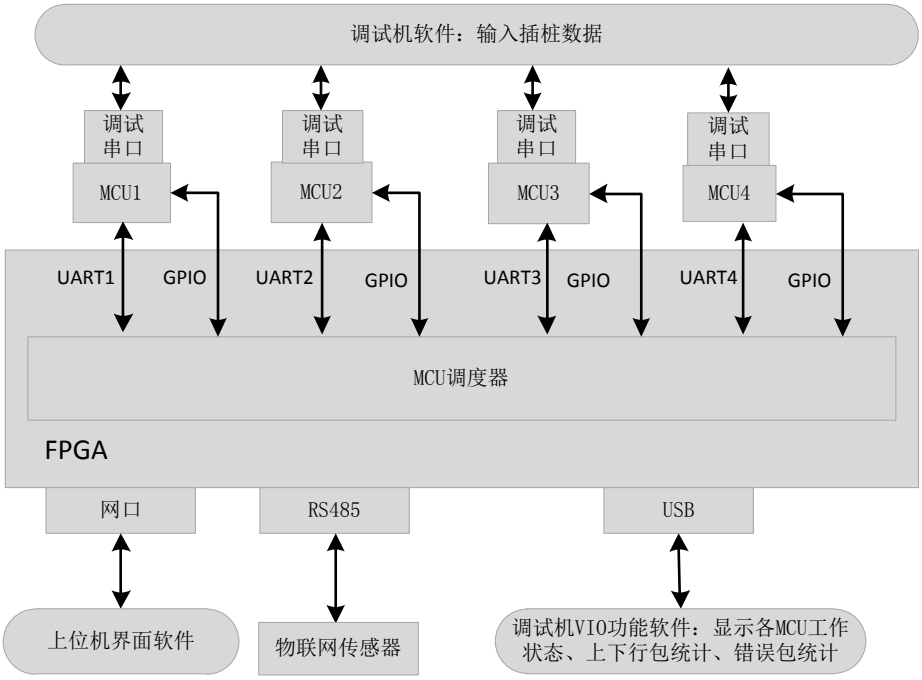


图5　测试拓扑

产生异常输出。以MCU2为例，开启MCU2的插桩，通过调试串口向MCU2发送随机数据。VIO软件可以观察到MCU1，MCU2，MCU3绿灯（工作）。但是MCU2产生错误计数，出现异常指示。拟态模组下行业务数据正确。

当MCU2错误计数达到设定的阈值时，认为该MCU2出现异常。此时清洗MCU2并用备份MCU4进行替换。此时VIO软件可以观察到MCU1，MCU3，MCU4绿灯（工作），MCU2红灯（清洗）。拟态模组下行业务数据正确。

MCU2清洗结束后，拟态模组重新恢复至正常工作状态。此时VIO软件可以观察到MCU1，MCU3，MCU4绿灯（工作），MCU2备份。拟态模组下行业务数据正确。

（2） 二模攻击场景测试

构建二模攻击：二模攻击时，有两个MCU因攻击产生的异常输出相同，对于择多判决策略来说，此时异常的输出能通过判决，对外输出。以MCU1和MCU2为例，通过MCU调试串口向他们发送相同的数据（不同于正常输出）。VIO软件可以观察到MCU1，MCU2，MCU3绿灯（工作）。但是MCU3产生错误计数，出现异常指示。拟态模组下行业务数据错误。

当MCU3错误计数达到设定的阈值时，认为该MCU3出现异常。此时清洗MCU3并用备份MCU4进行替换。此时VIO软件可以观察到MCU1，MCU2，MCU4绿灯（工作），MCU3红灯（清洗）。新替换的MCU4仍产生错误计数，出现异常指示。拟态模组下行业务数据错误。

当MCU4错误计数达到阈值后，启动反向验证策略，清洗MCU1，并用清洗完成的MCU3进行替换。此时VIO软件可以观察到MCU2，MCU3，MCU4绿灯（工作），MCU1红灯（清洗）。MCU2产生错误计数，出现异常指示。拟态模组下行业务数据正确。此时已经退化到差模攻击场景。

（3） 共模攻击遭遇随机扰动测试

构建共模攻击：共模攻击时，所有MCU因攻击产生的异常输出都相同，异常输出可以通过判决。

将MCU1、MCU2、MCU3的插桩都开启，向三个MCU输入相同的数据。VIO软件可以观察到MCU1，MCU2，MCU3绿灯（工作），无错误计数。但是拟态模组下行业务数据错误，并非期望的数据。

当达到设定的时间后，随机扰动机制启动，MCU1清洗，并用MCU4进行替换。VIO软件可以观察到MCU2，MCU3，MCU4绿灯（工作）。但是MCU4产生错误计数，出现异常指示。拟态模

组下行业务数据错误。此时已经退化到二模攻击场景。

（4） 正常模式下随机扰动测试

该场景下MCU拟态模组未遭到攻击（不启用插桩），此时各MCU工作状态正常，VIO软件可以观察到MCU1，MCU2，MCU3绿灯（工作），且无错误计数。拟态模组下行业务数据正确。

当达到设定的时间后，随机扰动机制启动，MCU1清洗，并用MCU4进行替换，系统处于一级工作状态。VIO软件可以观察到MCU2，MCU3，MCU4绿灯（工作），且无错误计数。拟态模组下行业务数据正确。

通过上述分析，基于拟态MCU面临绝大多数的差模攻击场景时（由于各执行体的处理器架构、执行软件都是同功异构的，绝大多数的攻击只会对其中一种执行体造成影响，因此差模攻击是MCU面临的主要攻击场景），通过调度器的择多判决，均能输出正确结果。而且，这种选择机制仅仅是根据输出结果的异同进行的，不关注到底是何种原因造成的结果不一致。无论是已知的还是未知的安全威胁造成的输出不一致，该架构均能有效处理，不依赖病毒库、木马库、漏洞库等先验数据。

5 总结与展望

本文基于拟态防御理论构建出一套动态异构冗余的MCU架构。从底层硬件开始，采用冗余备份。在调度器上采用择多判决的工作模式，使单个或多个MCU收到外部攻击后系统仍保持正常工作状态或快速恢复正常工作状态。根据MCU应用插桩实验，验证测试此MCU架构提高了系统功能和性能的稳定性、安全性和鲁棒性。由于MCU种类繁多应用场景复杂，如何从中选取的执行体来保证之间异构性最大，从而保证安全系数越高，是下一步研究的主要问题。

参考文献:

[1] ZHUANG R, DELOAC H, SCOTT A, et al. A model for analyzing the effect of moving target defenses on enterprise networks［C］. Proceedings of the 9th Annual Cyber and Information Security Research Conference. New York：ACM Press，2014：73-76.

[2] FENG X T, ZHENG Z Z, DERYA C, et al. A signaling game model for moving target defense［C］//IEEE INFOCOM 2017-IEEE Conference on Computer Communications. Piscataway：IEEE Press，2017：1-4.

[3] ZAFFARANO K, TAYLOR J, HAMILTON S. A quantitative framework for moving target defense effectiveness evaluation［J］. Association for Computing Machinery，2015（10）：3-11.

[4] 邬江兴. 拟态计算和拟态安全防御的原意和愿景[J]. 电信科学，2014，30（7）：1-7.
WU J X. Meaning and vision of mimic computing and mimic security defense[J]. Telecommunications Science，2014，30（7）：1-7.

[5] 仝青,张铮,张为华,等. 拟态防御Web服务器设计与实现[J]. 软件学报，2017，28（4）：883-897.
TONG Q, ZHANG Z, ZHANG W H, et al. Design and implementation of mimic defense Web server[J]. Journal of Software，2017，28（4）：883-897.

[6] 马海龙,伊鹏,江逸茗,等. 基于动态异构冗余机制的路由器拟态防御体系结构[J]. 信息安全学报，2017，2（1）：29-42.
MA H L, YI P, JIANG Y M, et al. Dynamic heterogeneous redundancy based router architecture with mimic defenses[J]. Journal of Cyber Security，2017，2（1）：29-42.

[7] 魏帅,于洪,顾泽宇,等. 面向工控领域的拟态安全处理机架构[J]. 信息安全学报，2017，2（1）：54-74.
WEI S, YU H, GU Z Y, et al. Architecture of mimic security processor for industry control system[J]. Journal of Cyber Security，2017，2（1）：54-74.

[8] 宋克,刘琴让,魏帅. 基于拟态防御的以太网交换机内生安全体系结构[J]. 通信学报，2020，41（5）：18-26.
SONG K,LIU Q R,WEI S,et al. Endogenous security architecture of Ethernet switch based on mimic defense[J]. Journal on Communications. 2020，41（5）：18-26.

[9] 邬江兴. 网络空间拟态防御导论[M]. 北京：科学出版社，2017.
WU J X. Introduction to cyberspace mimetic defense[M]. Beijing：Science Press，2017.

[作者简介]

张明权（1996-），男，信息工程大学在读研究生，主要研究方向为计算机技术。

于洪（1989-），女，硕士，信息工程大学助理研究员，主要研究方向网络空间安全，嵌入式技术。

魏帅（1984-），男，博士，信息工程大学助理研究员，主要研究方向计算机软件，网络空间安全。

崔超（1988-），男，硕士，天津市滨海新区信息技术创新中心高级工程师，主要研究方向计算机软件，嵌入式技术。

黄丽波（1981-），女，硕士，工程师，研究方向通信工程，计算机网络。

ASER：基于自适应步长的缓存冲突域生成算法

王崇[1]，魏帅[1]，张钦元[2]，姜海斌[2]，李丹丹[3]

[1]战略支援部队信息工程大学信息技术研究所 郑州 450003；

[2]天津市滨海新区信息技术创新中心 天津 300450；

[3]天津市芯海创科技有限公司 天津 300450

摘　要：在基于争用的缓存侧信道攻击中，攻击者为了完成对目标地址对应缓存的“监听”，需要把较大的备选集合收缩到尽可能小的冲突域，如何能以最小的代价生成尽可能小的冲突域成为研究热点。本文从裁剪集合的大小出发，基于备选集合裁剪不同的数目时的裁剪代价，提出了一种基于自适应步长的冲突域生成算法。首先，对裁剪成功率和裁剪代价进行抽象分析；其次，把整体的裁剪代价作为全局优化目标，采用动态规划求解。实验结果表明，与原始的分组收缩算法相比，本文提出的算法可以减少11.5%的内存访问次数。

关键词：缓存侧信道攻击、冲突域、收缩算法

ASER: Adaptive Step Based Cache Eviction Set Reduction Algorithm

WANG Chong[1], WEI Shuai[1], ZHANG Qinyuan[2], JIANG Haibin[2], LI Dandan[3]

1.Strategic Support Force Information Engineering University Zhengzhou 450003;

2.TianJin Binhai Information Technology Innovation Center, Tianjin Binhai 300450;

3.TianJin Chip-Sea Technology Corporation, Tianjin 300450

Abstract: During the conflict-based cache side channel attack, attacker need to reduce candidate set to an eviction set for monitor the target address, how to create an eviction set in minimum number of memory access has become a research hotspot. This paper started with the size of crop set, proposed an adaptive step based eviction set reduction algorithm based on cropping cost required to crop different numbers of candidate sets. Firstly, this paper uses the model to analysis the crop success rate and crop cost; Then, taking the overall cropping cost as the global optimization goal, and using dynamic programming to get the step sequence. Experimental results show that compared with the original packet shrinking algorithm, the algorithm proposed in this paper can reduce the number of memory accesses by 11. 5%.

Key words: Cache-Based Side Channel Attack; Eviction Set; Reduction algorithm

1　引言

随着计算机处理能力的不断发展，内存的读取速度与CPU的处理速度的差异越来越大。为了弥补内存与CPU在速度上的差异，现代处理器中广泛采用了存储器层次结构的设计方案。一方面，缓存（Cache）可以缩短处理器读取指令和读写数据的时间，缩短了CPU与内存之间的时间差异；另一方面，当数据位于不同的存储器层次结构时，数据的访问时间也会有较大的差异，而这也就为攻击者提供了“接口”。大量研究表明［1-7］，攻击者可以利用缓存的这一特点获取出包括操作系统内核空间分布、用户秘钥、用户浏览器等在内的诸多敏感信息，造成用户信息泄露，严重威胁用户的信息安全。

攻击者实施基于缓存的侧信道攻击时，首先要能够获取到用户对某个或某些特定地址的访问情况，再根据内存访问特征及一定的先验知识就可以分析出用户信息。而在对用户内存访问状态进行探测的时候，攻击者需要根据自己空间内的

基金项目：国家核高基重大专项基金资助项目（No.2017ZX01030301）

地址完成以下几个操作：

通过内存访问等初始化某块缓存到攻击者已知的状态；

能够“探测”到缓存状态是否发生改变；

当攻击者需要“监听”的内存地址在攻击者与用户的共享内存中时，攻击者可以通过*flush*[9]指令就可以满足上述需要。更一般的，当要攻击的目标不在共享内存中时，则只能采用Prime+Probe[10]的方式，通过访问攻击进程内存空间中的数据，初始化缓存状态；对数据的重新访问获取用户状态变化信息。而在Prime+Probe中，一个关键步骤就是攻击者需要确定访问自己内存空间中的哪些数据，这些需要被访问的地址的集合称为冲突域（Eviction Set），当选择的冲突域过小时，会影响“探测”的精准度，而选择的冲突域过大时，又会给攻击过程带来较大的噪声，影响攻击的效率和隐蔽性。如何快速生成一个合适的冲突域是缓存侧信道攻击中的一个热点研究问题，受到了越来越多的关注。

本文把内存访问次数作为选择条件，对现有的冲突域生成算法进行改进，提出基于自适应可变步长的冲突域生成算法，并以一个典型的缓存结构为例，展示了采用自适应步长后冲突域生成过程可以有效减少对内存的访问次数。

本文的组织结构如下所示：第二章介绍相关背景，主要介绍缓存的基本结构和之前的冲突域收缩算法；第三章按照概率模型分析现有冲突域收缩算法的内存放访问次数；第四章介绍自适应步长冲突域收缩算法；最后给出全文的结论并指出未来的研究方向。

2 背景

2.1 缓存结构

现代计算机中，处理器在执行指令访问或数据读写操作的执行过程为：首先，程序把需要访问的地址发送出来，经过操作系统的页表转化等操作转变为物理地址，再使用这个地址在缓存中进行查找比对，如果匹配成功，即缓存命中（Cache Hit）则返回该数据，内存访问结束；若匹配失败，则称为缓存缺失（Cache Miss），处理器需要接着去内存中查找，并将该值返回。访问内存所需要的时间周期大于访问缓存的周期，因此，对程序本身来讲，缓存缺失时所需要的数据读取时间大于缓存命中时的数据读取时间，攻击者通过对这个时间的利用，就可以获取出用户程序的内存访问情况。

在缓存的实际工作过程中，需要解决以下几个问题：1）数据应该放在缓存的哪个位置；2）给定一个地址，怎么索引出数据内容；3）当缓存没有空闲地址时，数据应该替换掉哪个缓存。具体来讲，Cache的结构基本工作原理如图1所示，Cache在内存中的最小存储粒度称为一个块（Cache Line），即当Cache Line中的某个字节发生换入换出时，整个Cache Line都要进行替换。缓存的组织方式一般包括三种：直接映射、全相联和组相联。直接映射的方式为每个地址指明了缓存的位置，查找效率最高；全相联则表示某个地址可以放在缓存中的任何位置，可以对缓存资源实现最大化的利用；而组相联则兼顾两者的优点，某个地址可以放在某一个固定的地址集合中，但是可以存在于这个集合中的任意位置。Cache替换算法主要包括随机替换、最近最少使用（LRU）和先入先出（FIFO）当Cache发生缺失的时候，它决定了当缓存中没有空闲缓存块时，新写入的内容替换哪个缓存块。

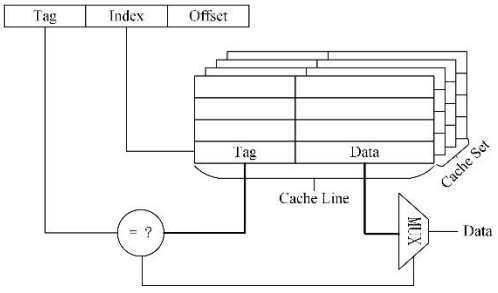


图1 Cache基本工作原理

2.2 冲突域

根据攻击者的监听位置的不同，基于缓存的侧信道攻击可以分为基于争用的攻击（conflict-based attack）和基于复用的攻击（reused-based attack）。基于争用的侧信道攻击是指攻击者需要“监听”的位置不在共享内存中时，攻击者只能通过访问自己用户空间中的地址实现对目标地址的“清除”和“监听”操作，而攻击者需要访问的这些地址称为“冲突域”。在攻击过程中，攻击者首先访问冲突域中的所有地址，保证自己的数据

“填充”目标地址对应的缓存，然后对这些地址进行重新访问，获取用户对“监听”地址的访问情况。

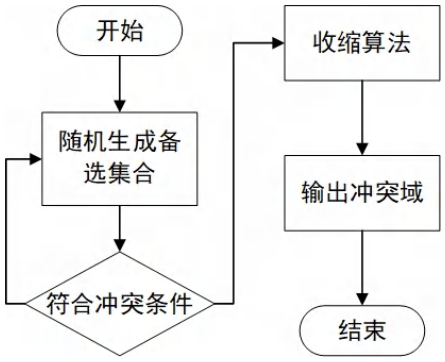


图 2　冲突域生成算法流程

冲突域选择过大时，攻击者对目标地址的“监听”精准度会受到较大的影响，同时对正常用户进程带来较大的干扰，隐蔽性差。因此选择一个合适大小的地址集合作为冲突域就显得尤为重要。如图 2 所示，攻击者在生成冲突域的过程中，首先会随机生成一个备选地址集合（Candidate Set），作为收缩的初始集合，并判断该集合是否与目标地址冲突，若不满足条件，就重新生成一个地址集合；当生成地址集合后，采用收缩算法对原始集合进行“裁剪”，不断缩小集合的大小，直至裁剪后的集合达到最小，即生成冲突域（Eviction Set）。

一般而言，都是采用比较大的集合空间作为初始备选集合，收缩算法收缩地越快，留给攻击者的时间就越多，可以处理的初始集合就越大。大部分的相关研究主要围绕冲突域的收缩算法展开。

a) 原始算法

Liu [10] 等人于 2015 年在针对 LLC 的 prime+probe 攻击中，首次提出了生成冲突域的基本算法。如算法 1 所示，攻击者初始化一个空集合 ***R***，之后从中任选一个地址，检测剩余地址是否仍能构成冲突域，若剩余集合不能构成冲突域，表示该地址与目标地址映射于同一个缓存集合中，故把这个地址添加到冲突域中，并从备选集合中删除。通过对备选集合中的地址不断地筛选，直至集合 ***R*** 中的地址可以把目标地址从缓存中“清除”。

b） **分组收缩（threshold group）算法**

算法 1 原始收缩算法

输入：目标地址 v，备选集合 S，缓存关联度 a
输出：冲突域 R

```
R←[]
while |R| < a do
    if ¬TEST(R∪(S \ {c}),v) then
        R ← R∪{c}
    S ← S \ {c}
return R
```

算法 1

虽然完成了初始备选集合的收缩，实现了访问一个较小集合从而实现对目标地址监控的目的，但是在算法的裁剪过程中，每次只能去除一个地址，算法需要完成的内存访问次数符合$O(N^2)$复杂度，其中 N 代表备选集合的大小。这样的收缩算法会导致内存访问次数过多，不利用侧信道攻击的实施。因此，Vila [11] 等人根据群试理论，把整个集合分为（a+1）个组，遍历所有组直至找到可以被清除的集合。如算法 2 所示，分组收缩的方式，可以增加每次收缩过程中的裁剪的地址数目，有效提高收缩效率，减少集合收缩过程中内存访问次数。

算法 2 分组收缩算法

输入：目标地址 v，备选集合 S，缓存关联度 a
输出：冲突域 S

```
while |S| > a do
    {T_1,⋯,T_{a+1}} ← SPLIT(S,a+1)
    i ← 1
    while ¬TEST (S \ T_i,v) do
        i ← i + 1
    S ← S \ T_i
return S
```

c) 随机筛选算法

攻击者在生成冲突域的过程中，可能受到用户程序的影响，导致算法 2 中找不到可以被清除的集合，从而导致算法失败，Song [12] 等人提出了一种更具有鲁棒性的随机筛选算法，如算法 3 所示，攻击者在裁剪原始集合时，不采用分割元素集合为多个组的方案，而是从中随机选择出一个组大小的地址集合作为裁剪对象，一旦裁剪失败，则重新选择一个组大小的地址集合进行重新裁剪，避免由于用户程序运行过程中的噪声带来的算法

失败的情况，提高算法可靠性。

3 模型分析

在之前的冲突域收缩算法中，大多是从算法的执行过程的角度对算法的复杂度进行分析，把待收缩集合分为固定的组数，每次裁剪一个组的数目，而没有考虑每次裁剪的代价以及整个收缩过程的内存访问次数。

算法3 随机筛选算法

输入：目标地址 v，备选集合 S，缓存关联度 a，筛选参数 l

输出：冲突域 S

```
while |S| > a do
    G ← RANDOM_SPLIT(S,l)
    if TEST(S \ G,v) then
        S ← S \ G
    end
return S
```

本文从这一角度出发，首先分析在不同的初始集合S大小的前提下，裁剪不同大小的集合K后，剩余集合仍能与目标地址冲突的概率，以及不同裁剪集合K对应的内存访问数目。

3.1 裁剪成功率

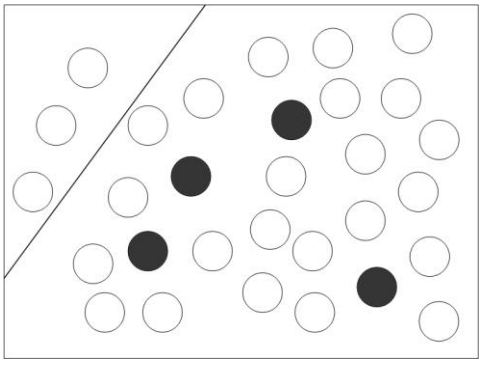


图3 地址裁剪示意图

如图3所示，对于一个初始地址集合$\boldsymbol{S}$，从中任意筛选出k个地址构成集合$\boldsymbol{K}$，把集合$\boldsymbol{K}$中的地址从初始集合$\boldsymbol{S}$中去除，若剩余集合仍能与目标地址产生冲突，则称为集合$\boldsymbol{S}$裁剪$\boldsymbol{K}$成功，否则称为裁剪失败。发生裁剪失败后，需要将集合$\boldsymbol{K}$重新添加到集合$\boldsymbol{S}$，重新进行裁剪。裁剪成功率是指从初始集合$\boldsymbol{S}$中一次就可以裁剪掉集合$\boldsymbol{K}$的概率。

记与目标地址x冲突的地址集合为［x］，原初始集合$\boldsymbol{S}$中包含冲突域集合中的地址数目为m，从初始集合中任选出k个地址，构成集合$\boldsymbol{K}$。

在集合$\boldsymbol{S}$中，任选一个地址t属于目标地址冲突集合［x］的概率为：

$$P(t \in [x]) = \frac{|[x]|}{|S|} \tag{1}$$

由此可得，从集合$\boldsymbol{S}$中任选出k个地址构成的集合$\boldsymbol{K}$中包含冲突域中地址数目为b的概率为：

$$P(|K \cap [x]| = b) = \frac{C_k^b \, C_m^{m-b+1} \, C_{N-m}^{N-m-k+b}}{C_N^{N-b+1} \, C_{N-b}^{N-k}} \tag{2}$$

也就是说，在一个集合中随机去除k个地址后，剩余地址仍能构成一个冲突域的概率，即裁剪成功率为：

$$P(S \cap K \setminus [x] \geq a) = 1 - \sum_{i=0}^{m-a} P(|K \cap [x]| = i) \tag{3}$$

3.2 裁剪代价

从原始集合$\boldsymbol{S}$中去除集合$\boldsymbol{K}$后，需要访问集合$\boldsymbol{S}\backslash\boldsymbol{K}$，判断该集合能够与目标地址集合构成冲突域，选择不同大小的集合$\boldsymbol{K}$，对应的裁剪成功率不同，需要完成的内存访问也不同。因此我们用集合$\boldsymbol{S}$中选择$\boldsymbol{K}$所需要的内存访问的期望值作为裁剪代价。

记裁剪成功率为p，则每次裁剪代价为：

$$C(S \cap K) = \sum_{i=1}^{\infty} (Test(|S| - |K|))p(1-p)^{i-1} \tag{4}$$

图4展示了对于一个初始大小为4000，且其中包含16个冲突域地址的初始备选集合，缓存的关联度a=16，当选择不同大小的裁剪集合$\boldsymbol{K}$时，所对应的不同的裁剪成功率和裁剪代价。从中可以看出，当初始集合$\boldsymbol{K}$选择的越大，即每次从备选集合中裁剪的步长越大，裁剪成功率会有所下降，从而导致裁剪代价增加。

4 自适应步长生成算法

从初始备选集合向冲突域收缩的过程中，内存访问次数可以反映算法的收缩速率。裁剪集合$\boldsymbol{K}$的大小会影响裁剪的步长，同样也会影响裁剪成功率和裁剪代价。裁剪集合$\boldsymbol{K}$越大，收缩速率越大，而收缩步长的增大会带来裁剪代价的增加，又会导致收缩速率降低；裁剪集合$\boldsymbol{K}$越小，收缩速率越小，但是相应的裁剪代价相对较低。因此，在收缩算法中，需要选择合适的裁剪集合$\boldsymbol{K}$，以达到最高的收缩速率。

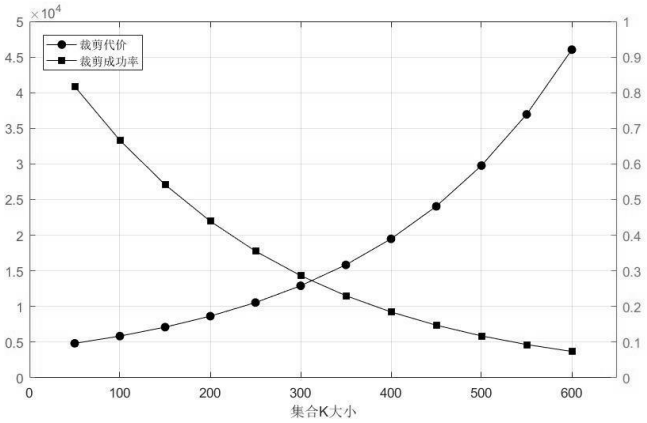


图 4 裁剪集合 K 的大小与裁剪成功率和裁剪代价关系图

4.1 自适应步长收缩算法

文献［10］中提到的收缩算法，即步长为 1 的收缩算法，虽然每次都可以保证一个较大的裁剪成功率，但是由于每次收缩的步长过小，并不能实现一个较高的收缩速率。文献［11］和文献［12］中的算法类似，采用分组的方式，每次把原始集合分割为（$a+1$）个组，或者从中任选出（$N/(a+1)$）个地址作为筛选集合，即每次收缩的步长为（$N/(a+1)$）。但是从理论角度考虑算法的复杂度，没有讨论裁剪成功率和裁剪代价。

算法 4 自适应步长收缩算法

输入：目标地址 v，备选集合 S，缓存关联度 a，裁剪集合序列 $\mathcal{K}$

输出：冲突域 S

1. *Generate* $\mathcal{K} = \{k_0, k_1, k_2, \cdots, k_n\}$
2. ***while*** $|S| >$ a ***do***
3. $K \leftarrow$ *RANDOM_SEL(S,*k_0*)*
4. ***if*** TEST($S \setminus K$,v) ***then***
5. $S \leftarrow S \setminus K$
6. ***end***
7. ***return*** S

裁剪步长和裁剪成功率对收缩算法的影响是两个相反的趋势，需要从整体的角度考虑算法的收缩速率，即优化目标为：

$$T_{access\_time} = \min_{|K|=step}(C(S \cap K)) \tag{5}$$

如算法 4 所示，对于一个初始备选集合 S，选择不同大小的裁剪集合 K，其对应的裁剪代价为 $C(S \cap K)$，则下一次的备选集合大小更新为（$S \setminus K$），选择一个新的裁剪集合 K'，完成下一步的收缩，直至集合收缩到冲突域大小。根据不同集合大小下不同裁剪集合对应的裁剪成功率和裁剪代价，使用动态规划获得裁剪集合序列 $\mathcal{K}$，并由此可以计算得到全局最优的 T_{access_time}。

4.2 仿真验证

在本文中，仿真验证的参数表 1 所示。对于攻击者而言，缓存的大小、缓存的关联度以及缓存组的数目会影响攻击者初始备选集合的大小，本实验中，选择一个典型的缓存架构作为验证对象，初始备选集合的大小为 4000，更进一步地，我们假定对于初始备选集合中有且只有与缓存关联度相同数目的目标冲突域集合地址，即 $|S \setminus [x]| = a$，这一条件对攻击者而言是成立的，且更为严格的，因此后续对内存访问次数的讨论都基于此开展。

表 1 缓存结构及参数选择

缓存结构	关联度	缓存大小	缓存组数目
	16	8MB	8192
初始集合		4000	

图 5 展示了当选择不同的分组数目时，分组收缩与自适应收缩的收缩过程比较图，其中横坐标表示了共需收缩的步数，而纵坐标表示随着收缩过程的进行，备选集合的大小不断收缩，直至达到最小冲突域。从图中可以看出，分组的数目越多，每次可以收缩的步长越大，但是由于初始集合的大小以及分组数目的细微差别，分组数目对每次收缩前进的步长影响不大；由于自适应步长收缩算法根据当前集合的大小灵活地选择了不同的步长，可以在每次收缩过程中实现较大的收缩

步长，有利于提高整体的收缩效率。

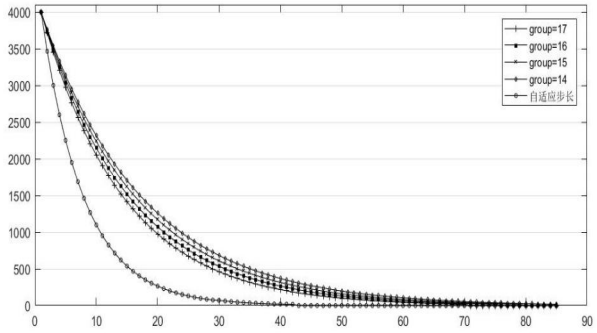


图 5　分组收缩与自适应收缩比较

虽然在收缩过程中选择较大的裁剪集合可以提高收搜步长，但是也会给裁剪成功率和裁剪代价带来负面影响。表 2 展示了不同的分组数目带来的不同的内存访问次数，以及自适应步长算法的内存访问次数。虽然采用了不同的分组方式，但由于每次裁剪集合的大小相差不大，因此对于不同的分组数目而言，在裁剪过程中所需要的内存访问次数相差不大；对于自适应步长收缩算法，可以兼顾整体的访问次数，根据全局内存访问次数自适应地选择每次裁剪集合，与分组收缩算法相比，自适应步长收缩算法可以减少 11.5% 的内存访问次数，从而提高攻击者的攻击速率，降低缓存侧信道攻击的隐蔽性。

表 2　不同算法内存访问次数

收缩算法		内存访问次数
分组数目	g = 14	170053
	g = 15	169393
	g = 16	169266
	g = 17	169650
自适应步长		149743

4.3　讨论

攻击者在生成冲突域的过程中，会受到其他程序等其他因素的干扰，影响收缩算法的裁剪过程，本节对这些因素进行讨论。

1）缓存 TLB：攻击者在生成冲突域的过程中，需要把某个地址的访问时间作为该地址是否在缓存中的重要依据。对于最后一级缓存而言，现代处理器中普遍采用的是使用物理地址的索引和物理地址的标记域，即 PIPT（Physical Index Physical Tag），而地址转换需要 TLB 完成，会影响对目标地址的判断，把原处于缓存中的地址错误地认为不在缓存中。

2）缓存替换算法：当缓存中没有空闲位置后，缓存替换算法会决定选择把哪个块从缓存中删除，因此攻击者需要根据替换算法的不同选择一个比较合适的 *TEST*（）方式，从而提高收缩算法的成功率。

3）其他用户程序：由于缓存被处理器中的不同用户共享，别的用户程序在执行过程中，也可能产生影响目标地址缓存位置读写操作等，影响算法对目标地址状态的判断。

4）新型缓存架构：改变缓存架构，是近年来实现针对缓存侧信道攻击的一种新的解决方案[13]，如何应对这种新型缓存架构，生成冲突域，将是后续完成侧信道攻击所面临的重大挑战。

这些因素都可以看作是攻击者在生成冲突域过程中的“噪声”，这些噪声对收缩算法产生不同程度的影响，甚至导致收缩算法收缩失败，因此需要选择一个合适的抗噪声“检测方式”，提高检测精准度，从而缓存冲突域的生成成功率，这将是我们的后续研究工作。

5　结束语

在基于争用的缓存侧信道攻击中，攻击者需要以最快的速度生成一个冲突域，从而提高攻击的速度和隐蔽性。冲突域的生成过程可以分为两个关键步骤：1）生成备选集合；2）收缩至最小冲突域。本文对收缩算法的收缩步长进行讨论，根据不同备选集合大小下的裁剪成功率和裁剪代价动态选择裁剪的步长，与原始分组收缩算法相比，可以减少 11.5% 的内存访问次数，提高攻击者的攻击效率。下一步将继续完善本文的研究，考虑 TLB、替换算法等“噪声”的情况下完善监测方式和裁剪步长选择方式。

参考文献：

[1] Kocher P, Horn J, Fogh A, et al. Spectre attacks: Exploiting speculative execution[C]//2019 IEEE Symposium on Security and Privacy (SP). IEEE，2019：1-19.

[2] Lipp M, Schwarz M, Gruss D, et al. Meltdown: Reading kernel memory from user space[C]//27th {USENIX} Security Symposium ({USENIX} Security 18). 2018：973-990.

[3] Van Bulck J, Minkin M, Weisse O, et al. Foreshadow: Extracting the

keys to the intel {SGX} kingdom with transient out-of-order execution [C]//27th {USENIX} Security Symposium ({USENIX} Security 18). 2018：991－1008.

[4] Schwarz M, Lipp M, Moghimi D, et al. ZombieLoad: Cross-privilege-boundary data sampling[C]//Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. 2019：753-768.

[5] Gruss D, Maurice C, Wagner K, et al. Flush+ Flush: a fast and stealthy cache attack[C]//International Conference on Detection of Intrusions and Malware，and Vulnerability Assessment. Springer，Cham，2016：279-299.

[6] Kim H, Yoon H, Shin Y, et al. Cache Side-Channel Attack on Mail User Agent [C]//2020 International Conference on Information Networking (ICOIN). IEEE，2020：236-238.

[7] Ge J, Gao N, Tu C, et al. More Secure Collaborative APIs Resistant to Flush+ Reload and Flush+ Flush Attacks on ARMv8-A[C]//2019 26th Asia-Pacific Software Engineering Conference (APSEC). IEEE，2019：410-417.

[8] Gülmezoğlu B, Inci M S, Irazoqui G, et al. A faster and more realistic flush+ reload attack on AES [C]//International Workshop on Constructive Side-Channel Analysis and Secure Design. Springer，Cham，2015：111-126.

[9] Yarom Y, Falkner K. FLUSH+ RELOAD: a high resolution, low noise, L3 cache side-channel attack[C]//23rd {USENIX} Security Symposium ({USENIX} Security 14). 2014：719-732.

[10] Liu F, Yarom Y, Ge Q, et al. Last-level cache side-channel attacks are practical[C]//2015 IEEE symposium on security and privacy. IEEE，2015：605-622.

[11] Vila P, Köpf B, Morales J F. Theory and practice of finding eviction sets [C]//2019 IEEE Symposium on Security and Privacy (SP). IEEE，2019：39-54.

[12] Song W, Liu P. Dynamically Finding Minimal Eviction Sets Can Be Quicker Than You Think for Side-Channel Attacks against the {LLC} [C]//22nd International Symposium on Research in Attacks，Intrusions and Defenses ({RAID} 2019). 2019：427-442.

[13] Qureshi M K. CEASER: Mitigating conflict-based cache attacks via encrypted-address and remapping[C]//2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO). IEEE，2018：775-787.

[14] Ramkrishnan K, Zhai A, McCamant S, et al. New Attacks and Defenses for Randomized Caches [J]. arXiv preprint arXiv: 1909. 12302，2019.

[15] Dai C, Adegbija T. CONDENSE: A Moving Target Defense Approach for Mitigating Cache Side-Channel Attacks [J]. IEEE Consumer Electronics Magazine，2020，9(3)：114-120.

[作者简介]

王崇（1995-），男，信息工程大学博士研究生，主要研究方向为拟态防御，缓存侧信道防御。

魏帅（1984-）男，博士，信息工程大学信息技术研究所助理研究员，主要研究方向为信息安全、嵌入式系统、高性能与分布式计算

张钦元（1984-），男，学士，天津滨海新区信息技术创新中心助理工程师，主要研究方向为拟态交换机硬件架构。

姜海斌 （1987-），男，硕士，天津市滨海新区信息技术创新中心中级工程师，主要研究方向为集成电路设计。

李丹丹（1992-），女，学士，天津市芯海创科技有限公司助理工程师，主要研究方向为集成电路设计。

工业控制器软件拟态化改造方法研究

吴立其，邹涛，杨汶佼，王延松，张汝云
之江实验室 310023

摘　要：在网络工业控制器中引入拟态防御技术，可以有效解决传统工业控制器缺少内生安全设计、难以抵御各类未知攻击等问题。工业控制器软件的拟态化改造是工业控制器拟态化改造的重要组成部分。本文在对拟态安全技术原理进行分析的基础上，结合工业控制器硬件部分的拟态化改造方案，给出了工业控制器软件部分拟态化改造中的关键技术，并设计了包括多执行体同步运行、多执行体动态随机调度、实时择多裁决等在内的具体实现方案。该软件改造方案不但能与硬件方案有效适配，共同为工业控制器提供异构、动态、冗余的拟态防御功能，同时还可提供多异构执行体之间的精确运行同步。

关键词：工业控制器、拟态改造、内生安全、拟态防御

Study on Mimicry Modification Method of Industrial Controller Software

WU Liqi, ZOU Tao, YANG Wenjiao, WANG Yansong, ZHANG Ruyun
Zhejiang Lab, Hangzhou, 310023, China

Abstract: The introduction of mimicry defense technology into network industrial controllers can effectively solve the problems such as lack of endogenous security design and difficulty in resisting unknown attacks in traditional industrial controllers. Mimicry modification of industrial controller software is an important part of mimicry modification of industrial controller. This paper first introduces the principle of mimicry safety technology and the mimicry modification scheme of industrial controller hardware. And then analyses the key technology in mimicry modification of industrial controller software. At last designed a concrete implementation scheme which including multiple executive body synchronous operation, multiple executive body dynamic stochastic scheduling and real-time majority ruling. The software modification scheme can not only be effectively adapted to the hardware scheme, but also provide heterogeneous, dynamic and redundant mimicry defense function for the industrial controller, and at the same time provide accurate operation synchronization between multiple heterogeneous actuators.

Key words: industrial controller; mimicry modification; endogenous security; mimic defense

1　引言

工业控制系统（Industrial Control Systems，ICS）是指工业部门与关键基础设施中的控制系统[1]。近年来，随着工业化与信息化的深度融合，工业控制系统向着网络化迅猛发展，开放和标准化的互联网技术越来越多地应用于工业控制系统[2]。随之而来的，是所面临的更加严峻的安全威胁。2010年，成功入侵伊朗核电站的“震网病毒”震惊全球[3]，其后发生的重大安全事件有：

2012年超级火焰病毒席卷全球事件[4]，2014年Havex病毒入侵能源系统事件[5]，2015年BlackEnergy病毒攻击乌克兰电网事件[6]等。

目前，工业控制器的安全防护与传统的防御方法一样，大多是通过不停地封堵漏洞和发现后门来进行亡羊补牢式的修补。从最底层采用自主可控硬件，其上使用国产自主操作系统，到应用层的数据加密、入侵检测、防火墙、应急响应等安全技术。然而，这些传统的防御技术大多是建立在“已知风险”的前提条件上，并不能有效应对基于未知漏洞或未知后门的“未知的未知”安全威胁，因而需要研究新型的主动防御技术逆转该安全态势。

2 拟态安全技术原理分析

以大自然界生物自我防御的拟态现象作为启迪，邬江兴院士提出了一种网络空间的新型主动防御技术——拟态防御[7]。一种生物在色彩、纹理和形状等特征上模拟另一种生物或环境，从而使一方或双方受益的生态适应现象，称为拟态现象。生物体如果不仅在色彩、纹理和形状上，而且在行为和形态上也能模拟另一种生物或环境的拟态伪装，我们称之为“拟态防御”。

拟态防御以软硬件多样性为基础，以异构性最大化为主要目标，综合了冗余、裁决、主动重构、动态迁移、重配置等技术，构建了动态异构冗余结构（Dynamic Heterogeneous Redundancy，DHR）[7]。由系统级设计的软硬件模块联合作用，搭建起可配置的异构元素池，然后组建包含m个等价功能的异构构件的集合｛E1，E2，E3...E_m｝，由动态选择算法RS构造出包含n个异构构件的执行体集｛A1，A2，A3...A_n｝，这些执行体集当中的异构构件并行处理输出代理转发的系统输入，同步生成输出，再由表决器对构件的输出进行裁决，最终得到系统的输出。DHR结构如图1所示：

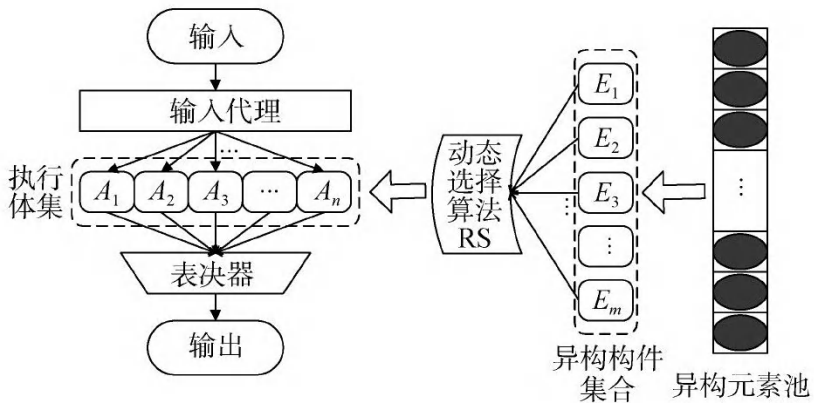


图1 DHR结构

在防御效果上，拟态防御依靠表决输出的方式，扰乱传递至攻击者的攻击效果信息，从而模糊攻击者对攻击结果的判断，进而阻断攻击链；同时表决技术本身为系统的高可用性提供保障[8]。具体的，可以从以下两个方面提高系统安全：

A. 增大攻击植入难度

假设图1中执行体集中包含三个异构执行体A_1、A_2和A_3，其被攻击成功的概率分别为P_1、P_2和P_3，那么在拟态防御结构中，攻击成功的概率可以记为$P'=P_1\times P_2\times P_3\times V_i$，其中$V_i$是3个执行体的输出结果在裁决时的一致性，即：

$$V_i=\begin{cases}1,\text{表决结果一致}\\0,\text{表决结果不一致}\end{cases} \quad (1)$$

由于$0\leqslant P_1、P_2、P_3\leqslant 1$，即$P'\leqslant\min\{P_1、P_2、P_3\}$，定义交互成功率下降幅度为：

$$D=(\min\{P_1,P_2,P_3\}-P')/\min\{P_1,P_2,P_3\} \quad (2)$$

从上述理论分析可以看出，采用拟态防御方法后攻击的成功率将被大幅抑制[9]。

B. 增大攻击维持难度

从各个功能相同的异构冗余体池中取出几个元素构成当前的服务集，构建出异构冗余体，或者对原先使用的异构冗余体做修复性质的清洗，从而使得攻击者难以复现攻击现场[10]。例如攻击

者使用木马病毒，意图获取被攻击主机的文件目录。而在主动攻击的架构体系下，相互异构的主机之间文件目录并不一致，从而可以扰乱反馈给攻击者的信息，使攻击难度加大。

3　工业控制器软件拟态化改造关键问题

工业控制器拟态化改造包括硬件的拟态化改造和软件的拟态化改造。本文的重点是研究工业控制器软件的拟态化改造，软件的拟态化改造方案应与硬件部分的拟态化改造相适配，软硬协同，共同实现拟态安全功能。我们的工业控制器拟态化改造方案硬件部分，采用的是“双裁决调度器FPGA+四异构冗余主控单元”共同构成工业控制器拟态界，以下简称拟态工业控制器。拟态工业控制器硬件架构如图2所示：

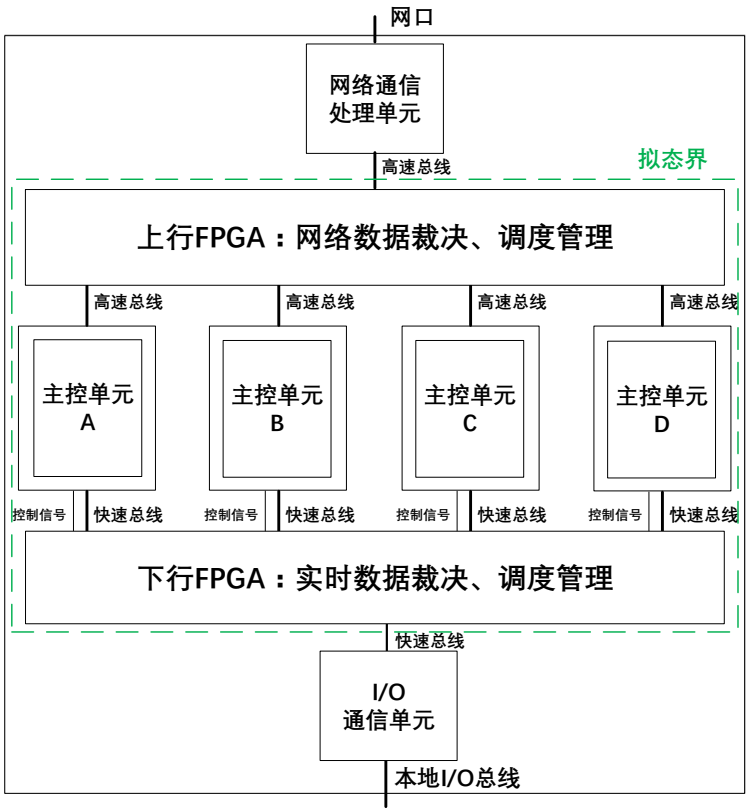


图2　拟态工业控制器硬件架构

这种“双裁决调度器FPGA+四异构冗余的主控单元”的硬件方案在提供“异构”、“动态”、“冗余”的硬件基础资源的同时，也对工业控制器软件部分提出了拟态化改造需求。比如如何让四异构冗余的主控单元同步工作，采用什么调度策略才能使四异构冗余的主控单元更好的相互协调，主控单元之间的数据混合裁决怎么解决等等问题。

具体说来，工业控制器软件拟态化改造中涉及的关键问题主要包括三个方面：

1） 多执行体同步运行：拟态安全要求多个异构冗余执行体为在功能上完全等价，首先意味着拟态工业控制器中的四个异构主控单元需要在运行控制方面实现拟态等价功能体的同步运行控制。但实际上，由于各异构冗余主控单元的内部实现机理和控制逻辑各不相同，要实现同步运行控制，需要在工业控制器软件中增加多执行体同步运行控制的拟态化改造。

2） 多执行体动态调度：动态随机调度通常是指在调度环境和任务存在不可预测扰动情况下所进行的调度。在拟态工业控制器中，会存在差模攻击和共模攻击（N-1模攻击、N模攻击），差模攻击指的是同一时刻有一个执行体被攻击，共模攻击指的是同一时刻有两个及以上的执行体被攻击。常规的择多裁决往往会把N-1模攻击误判为差模攻击，而N模攻击根本无法感知，而且系统无法从共模攻击中恢复。拟态工业控制器的多执行

体调度机制需要能够对共模攻击和差模攻击都具有较好的防御能力。注：本论文中的N-1模攻击即为2模攻击，N模攻击即为3模攻击。

3） 多执行体裁决：裁决技术是在冗余技术的基础上衍生出来的，是对多个同构或者异构的执行体的输出结果的选择过程。在工业控制器的实际应用中，需要裁决的数据往往既包含数字量又包含模拟量。需要对拟态工业控制器软件进行拟态化改造，既解决多执行体的择多裁决问题，同时也解决数字量和模拟量的混合裁决问题。

4 工业控制器软件拟态化改造设计与实现

针对前述问题，本文中的工业控制器软件拟态化改造提出了包括基于主控单元“控制周期同步”的多执行体同步运行控制、抗共模攻击的多执行体动态随机调度以及多执行体的实时择多混合裁决等在内的完整设计和实现方案。

4.1 基于主控单元“控制周期同步”的多执行体同步运行

通过对拟态工业控制器硬件实现方案的分析可知，四个主控单元的异构性体现在各主控单元的主频不同，运行代码也不相同。我们提出了一个基于主控单元“控制周期同步”的多执行体同步运行控制方案，通过为多执行体设计相同的控制周期且保证每个执行体的控制周期起始时刻完全相同，从而实现多执行体在系统运行步调上的完全一致。

主控单元之间的控制周期同步流程如下：在开始阶段，裁决调度器FPGA通过硬件信号线同时通知四个主控单元开始控制周期运行，四个主控单元接收到控制周期开始信号后同时开始控制周期运行。当某一个主控单元结束控制周期之后，这个主控单元通过硬件信号通知裁决调度器FPGA“我的控制周期结束”。然后开始等待裁决调度器FPGA的控制周期开始信号。当裁决调度器FPGA接收到四个主控单元发送的控制周期结束信号且控制周期时间到达或者超时，则裁决调度器FPGA给四个主控单元发送控制周期开始信号，让四个主控单元重新同时开始新的控制周期运行。主控单元之间的控制周期同步示意如图3所示：

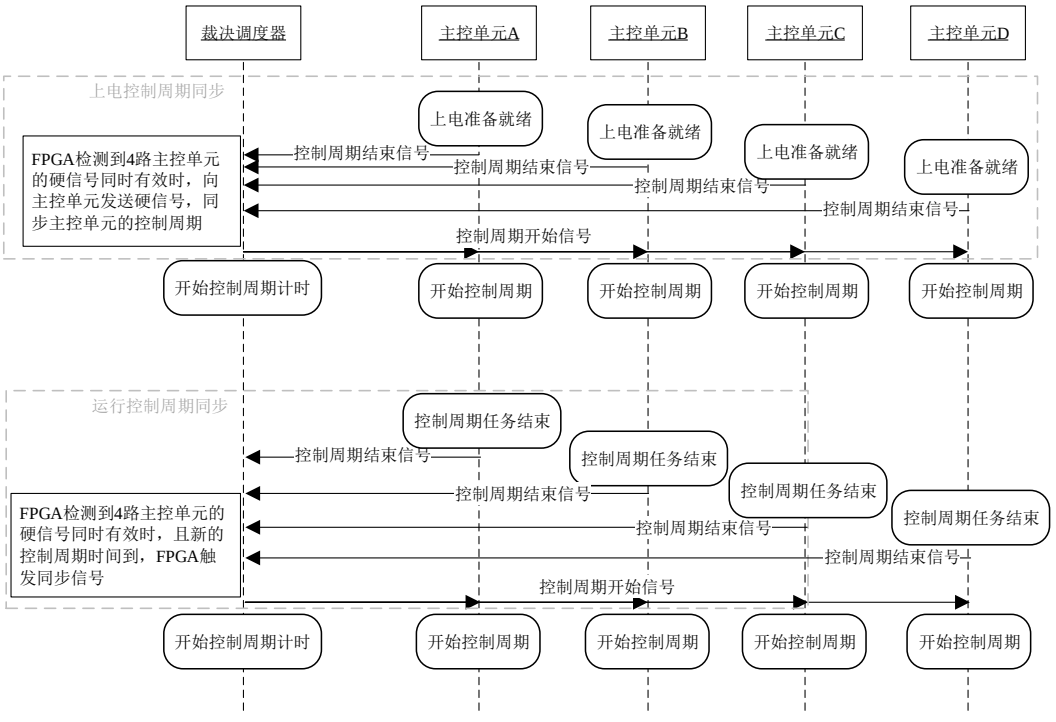


图3 主控单元控制周期同步示意图

4.2 抗共模攻击的多执行体动态随机调度

拟态化改造后的工业控制器包含四个主控单元和两个裁决调度器FPGA。裁决调度器FPGA负责对四个主控单元进行统筹调度。裁决调度器FPGA首先把四个主控单元放进调度池，然后使用随机函数从调度池中选出三个主控单元到裁决池，

最后对三个主控单元的输出结果进行择多裁决。当某个主控单元（记为E_i）的输出结果出现异常时，裁决调度器FPGA对这个主控单元进行清洗操作，然后从调度池中选取另一个主控单元（记为E_j）到裁决池中。待E_i完成清洗操作并恢复之后，裁决调度器FPGA再把E_i放入调度池中。如果E_i的输出结果还是和其他两个主控单元不一致，则可能是其他两个主控单元被N-1模攻击了。这时裁决调度器FPGA会对那两个主控单元中的其中一个进行清洗，然后再对另一个进行清洗，以此逐渐把N-1模攻击退化为差模攻击，最终再把差模攻击移除。如果三个裁决的主控单元的输出结果在比较长一段时间内都是一致的，则裁决调度器FPGA会选择加入一个随机扰动的策略，即随机对其中一个主控单元进行清洗，把另一个主控单元加入到裁决池中。随机扰动能有效对抗三个主控单元同时被攻击的N模攻击场景：随机扰动首先把N模攻击退化为N-1模攻击，然后再把N-1模攻击退化为差模攻击，最终把所有攻击都移除。动态随机调度的示意如图4所示：

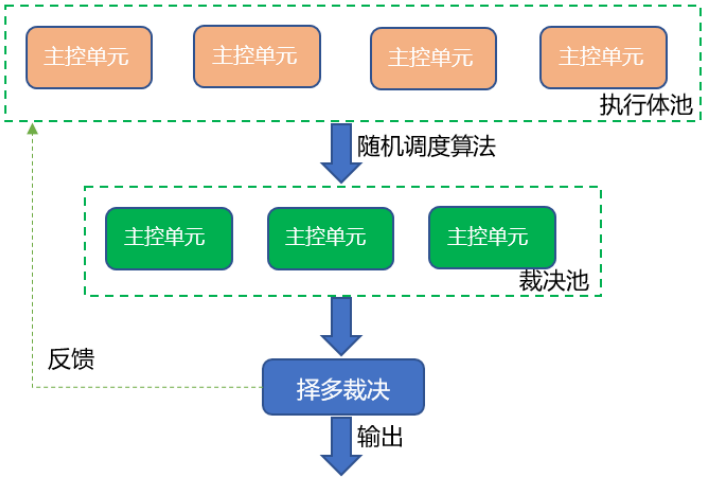


图4　动态随机调度示意图

4.3　多执行体的实时择多混合裁决

考虑到拟态化改造后的工业控制器包含四个主控单元，所以在择多裁决算法上选择三取二的裁决策略，即裁决调度器FPGA实时对裁决池中的三个主控单元进行裁决，选取多数相同的结果作为最终的输出结果。如果某个主控单元与其他两个主控单元的输出结果不一致，裁决调度器FPGA会在第一时间感知到这个情况，然后通过择多裁决把多数相同的结果作为最终的输出结果，另外再反馈一个信号告诉调度器FPGA有个主控单元的输出结果有异常。

为了支持数字量模拟量同时存在的工业控制应用场景，多执行体实时择多混合裁决算法中主控单元首先要对模拟量模块中的模拟量数值位号进行剥离，单独组合成包，并在原模拟量数值位置填0。这样，主控单元将实时输出数据拆分成两个包（一个纯数字量，一个纯模拟量）发送给裁决调度器FPGA进行裁决。裁决调度器FPGA在收到输出数据后实施裁决动作：针对数字量包，采用逐字比较的方式；针对模拟量包，采用逐个通道比较的方式，首先对3个值进行比较得出一个中值，然后判断3个值与中值的偏差，超出预设范围的值则认为是异常。在完成所有裁决后，FPGA得出一路主控单元作为可信单元，将其发送的输出数据进行输出。FPGA仍然以两个包（一个纯数字量，一个纯模拟量）的形式发送给I/O通信处理模块，由I/O通信处理模块将每个模拟量位号的值重新填回到模块完整的输出数据包中，然后将这些数据发送给现场。混合裁决算法中数据包的处理如图5所示：

5　拟态防御基准功能实验

从拟态防御理论预期效果出发的测试称为基准功能实验，也可称之为从拟态防御定义出发的效果测试。可以采用“白盒插桩”实验，要求在不依赖实验者经验和技巧的情况下，依据产品测

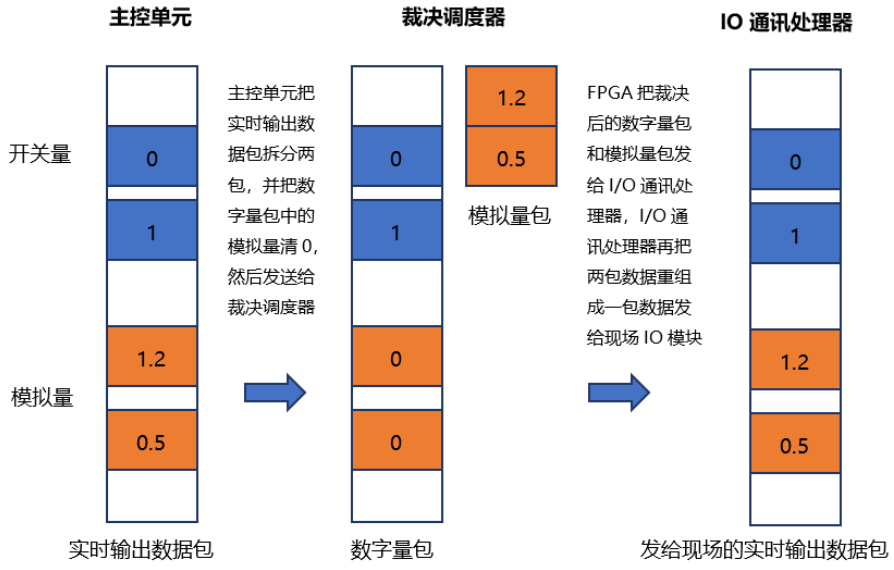


图5　混合裁决算法中数据包处理示意图

试规范用例就能准确甄别被测对象是否具备基本的拟态防御功能。测试例主要有三种：差模测试、N-1模测试（在本文中，N-1模即2模）、N模测试（在本文中，N模为3模）。

5.1　差模测试注入实验

通过白盒插桩方式，首先对拟态工业控制器进行差模测试注入实验，使主控单元A的输出数据与其他主控单元不一致。测试结果表明拟态工业控制器的差模测试注入实验不会对拟态工业控制器的输出造成影响，同时主控单元A的攻击事件被裁决调度器FPGA感知到后清洗恢复正常。随后通过白盒插桩方式，分别使主控单元B和主控单元C的输出数据与其他主控单元不一致，测试结果与主控单元A的差模测试注入实验现象一致。即拟态工业控制器能对差模攻击进行无感的清除。差模测试注入实验效果如图6所示：

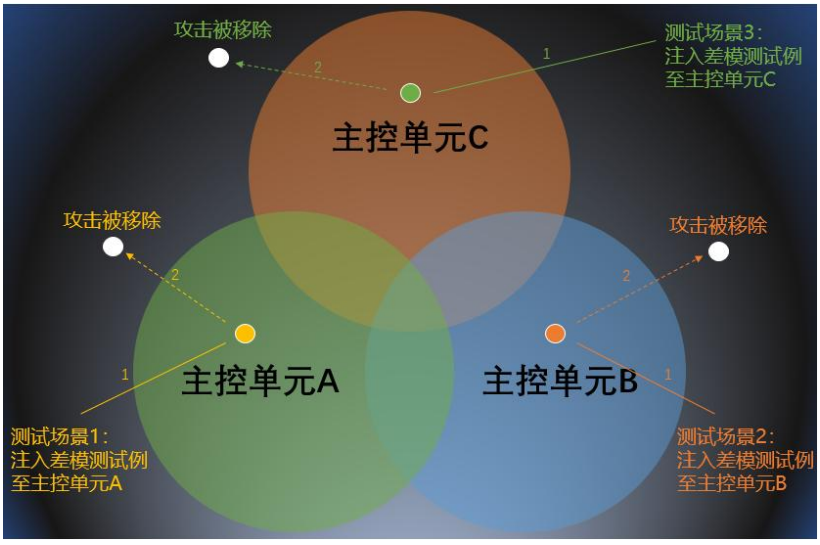


图6　差模测试注入实验效果图[7]

5.2　N-1模测试注入实验

通过白盒插桩方式，对拟态工业控制器进行N-1模测试注入实验，同时使主控单元A、B的输出数据与其他主控单元不一致。测试结果表明N-1模测试注入实验会对拟态工业控制器的输出造成短暂的攻击逃逸，随后主控单元A、B的攻击事件

被裁决调度器FPGA感知到后分别进行清洗恢复正常。随后分别通过白盒插桩方式，使主控单元B、C的输出数据与其他主控单元不一致以及使主控单元A、C的输出数据与其他主控单元不一致两种场景，测试结果与主控单元A、B的N-1模测试注入实验的现象一致。即拟态工业控制器在N-1模攻击下会出现短暂的攻击逃逸，但是随后会逐渐的把N-1模攻击退化为差模攻击，最后把攻击彻底清除。实验结果表明攻击逃逸时间（T_t）跟主控单元的清洗恢复时间（T_r）以及择多裁决周期（T_c）有关：

$$T_t = T_r + T_c$$

N-1模测试注入实验效果如图7所示：

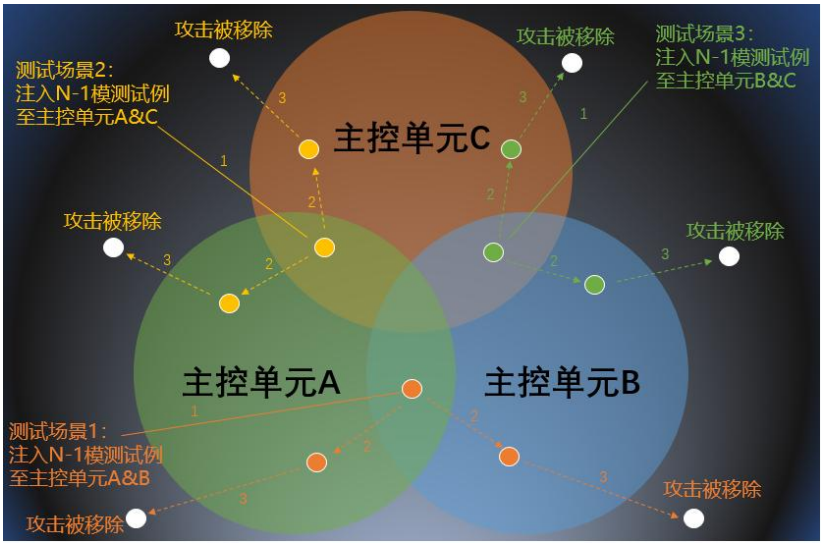


图7　N-1模测试注入实验效果图[7]

5.3　N模测试注入实验

通过白盒插桩方式，对拟态工业控制器进行N模测试注入实验，同时使主控单元A、B、C的输出数据与主控单元D不一致。测试结果表明N模测试注入实验会对拟态工业控制器的输出造成比较长时间的攻击逃逸，但是最后拟态工业控制器仍能把所有攻击移除之后恢复正常。即拟态工业控制器在N模攻击下会出现比较长时间的攻击逃逸，但是随后逐渐的把N模攻击退化为N-1模攻击、再把N-1模攻击退化为差模攻击，最后把攻击彻底清除。实验结果表明攻击逃逸时间（T_t）跟主控单元的清洗恢复时间（T_r）、择多裁决周期（T_c）、随机扰动周期（T_{cc}）有关：

$$Tt <= Tr + 2Tc + Tcc$$

N模测试注入实验效果如图8所示：

6　结束语

本文针对工业控制器软件的拟态化改造问题，分析了其中的多执行体同步运行控制、执行体动态调度、拟态裁决等关键技术，并设计了包括基于主控单元“控制周期同步”的多执行体同步运行控制、抗共模攻击的多执行体动态随机调度、多执行体实时择多混合裁决等在内的完整技术方案。该方案不但能与拟态工业控制器硬件方案有效适配、支持模数混合等复杂工业应用场景。最后通过对拟态工业控制器进行拟态防御基准功能实验，实验测试结果表明拟态改造后的拟态工业控制器具有明显的内生安全属性，能对差模攻击进行无感清除，对共模攻击（N-1模攻击、N模攻击）进行逐级退化并最终清除，使系统恢复正常运行。

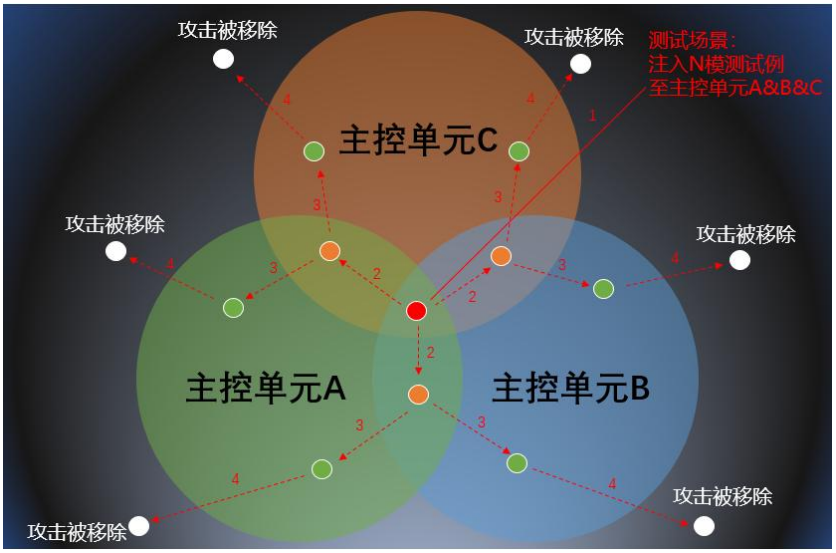


图8　N模测试注入实验效果图[7]

参考文献:

[1] Guide to Industrial Control Systems(ICS) Security: Supervisory Control and Data Acquisition(SCADA) systems, Distributed Control Systems(DCS), and other control system configurations such as Programmable Logic Controllers(PLC)[J]. Stouffer K A，Scarfone K A. Guide to Industrial Control Systems Security，2011.

[2] 李兴．工业控制系统加密控制器实验平台及方法研究[D]. 浙江大学，2018.

[3] FarwellJamesP, RohozinskiRafal. Stuxnet and the Future of Cyber War[J]. Survival，2011(1).

[4] 张敏, 张五一, 韩桂芬．工业控制系统信息安全防护体系研究[J]. 工业控制计算机，2013(10).

[5] 李鸿培, 忽朝俭, 王晓鹏．工业控制系统的安全研究与实践[J]. 保密科学技术，2014(04).

[6] 郭庆来, 辛蜀骏, 王剑辉, 孙宏斌．由乌克兰停电事件看信息能源系统综合安全评估[J]. 电力系统自动化，2016(05).

[7] 邬江兴．网络空间拟态防御原理-广义鲁棒控制与内生安全(第二版). 科学出版社，2018. 11.

[8] 仝青, 张铮, 邬江兴．基于软硬件多样性的主动防御技术[J]. 嘻嘻安全学报，2017(01).

[9] 仝青, 张铮, 张为华．拟态防御Web服务器设计与实现[J]. 软件学报，2017(4).

[10] 邬江兴．网络空间拟态防御研究[J]. 信息安全学报，2016(04).

[11] Pewny J, Schuster F, Bernhard L, et al. Leveraging semantic signatures for bug search in binary programs. Proceedings of the 30th Annual Computer Security Applications Conference，2014.

[12] 杨学军, 廖湘科, 卢凯, 胡庆丰, 宋君强, 苏金树．The TianHe-1A Supercomputer: Its Hardware and Software[J]. Journal of Computer Science & Technology，2011(03).

[13] 吴明桥, 陈香兰, 张晔, 龚育昌．一种基于服务体/执行流的新型操作系统构造模型[J]. 中国科学技术大学学报，2006(02).

[14] 高明, 罗锦, 周慧颖, 焦海, 应丽莉．一种基于拟态防御的差异化反馈调度判决算法．电信科学2020年第5期73-82.

[15] 刘宝旭, 马建民, 池亚平．计算机网络安全应急响应技术的分析与研究[J]. 计算机工程，2007(10).

[16] OkhraviHamed, ComellaAdam, RobinsonEric, HainesJoshua. Creating a cyber moving target for critical infrastructure applications using platform diversity [J]. International Journal of Critical Infrastructure Protection，2012(1).

[17] 邬江兴．"网络安全再平衡战略"之抓手:拟态防御[J]. 中国信息安全，2018(06).

[18] WuJ. X.. "Cyberspace Mimic Defense". Technical report，National Digital Switching System Engineering & Technological R&D Center，2015.

拟态防御调度算法研究综述

梅波[1]，赵博[1]，郭乔羽[1]，王苏南[2]，郭雷[3]

[1]信息工程大学信息技术研究所，河南 郑州 450002；

[2]深圳职业技术学院电子与通信工程学院，广东 深圳 518000；

[3]中国人民解放军31131部队

摘　要：当前网络面临的安全问题日益突出，随着新型复杂多变网络攻击手段的出现，传统的被动防御技术难以维护系统安全。拟态防御理论作为一种新型的主动防御理论，旨在利用攻防信息的不对称性，改变当前网络对抗格局中“易攻难守”态势。通过其自身架构的DHR特性来提高系统的安全性。本文首先简要介绍了网络安全的现有问题以及网络防御技术。然后对拟态防御基本理论进行论述。其次重点对现有的拟态防御中的调度关键技术研究进行归纳。最后对未来的研究和发展方向进行展望。

关键词：拟态防御、网络安全、动态异构冗余、调度

Summary of Research on Mimic Defense Scheduling Algorithm

Mei Bo[1], Zhao Bo[1], Guo Qiaoyu[1], Wang Sunan[2], Guo Lei[3]

1.Institute of Information Technology, Information Engineering University, Zhengzhou 450002, Henan;

2.School of Electronics and Communication Engineering, Shenzhen Polytechnic, Shenzhen 518000, Guangdong;

3.31131 Unit of the Chinese People's Liberation Army

Abstract: The current security issues facing networks are becoming increasingly prominent. With the emergence of new, complex and changeable network attack methods, traditional passive defense technologies are difficult to maintain system security. As a new type of active defense theory, the mimic defense theory aims to use the asymmetry of offensive and defensive information to change the "easy to attack and difficult to defend" situation in the current network confrontation pattern. Improve the security of the system through the DHR feature of its own architecture. This article first briefly introduces the existing problems of network security and network defense technology. Then the basic theory of mimic defense is discussed. Secondly, it focuses on the research of key scheduling technologies in the existing mimic defense. Finally, the future research and development directions are prospected.

Key words: mimic defense; network security; DHR; scheduling

1　前言

随着科学技术突飞猛进和高速发展，我们在享受着的网络带给我们的便利的同时，也潜在地陷入了网络攻击之中。国家安全建立在网络安全之上，现代化以信息化为基础。信息化和现代化促进了社会经济繁荣与社会进步，与此同时也带来了隐患[1]。网络安全已成为全球关注的热点话题。勒索病毒Wannacry的爆发和“棱镜门”的曝光也进一步表明了网络安全中存在着问题[2]。首先，在全球化产业链分工的趋势下，科技产品供应链难以实现漏洞后门零污染。而且在设计过程中难以实现零缺陷，因此漏洞和后门无法从根本上消除，成为网络中的不确定威胁。第二，现有的防御技术大多是基于已知攻击和漏洞后门采取的防御技术。但是，面对未知的威胁如Oneday漏洞，这些被动防御技术很难在网络安全中发挥作用[3]。

传统的网络博弈中的防御技术从以人们所熟悉的杀毒软件或者防火墙为代表的被动防护，发

基金项目：国家核高基科技重大专项：高安全等级网络基础设施关键装备核心芯片及软件研发（编号：2017ZX01030301）

展到以入侵检测[4]，蜜罐[5]，沙箱[6]，入侵容忍[7]等为代表的网络主动防御技术。然而，上述防护技术最大的弊端就是作为攻防模型中的防御方所采取的防御措施必须在知道攻击的类型或者自身的漏洞后门条件下才能有效，具有一定的特异性。当面对未知类型的攻击或漏洞后门时，防御者精心设计用来针对已知类型的攻击的防御屏障形同虚设。另外针对某漏洞的补丁的下发相对于攻击者对漏洞的利用具有一定的滞后性[8]。正是由于上述问题的存在，从攻击者的视角来看，信息系统具有确定性静止性[9][10]，从而表现出脆弱性，使得网络攻防长期处于一种易攻难守的态势。

为了打破网络攻防的固有态势，人们开始探索新型的防御技术，主动防御是网络空间安全领域的研究热点。移动目标防御（MTD）是主动防御中极具代表性的技术。它通过引入了攻击面概念[11]，旨在希望通过动态移动原来系统的一些关键特征（即攻击面），通过这种“移动”使得系统对于攻击者呈现不可测状态。然而，MTD很难选择有效的策略来应对攻击覆盖范围、攻击及时性和攻击不可预测性的挑战[12]。另外，一方面针对已经入侵成功的网络攻击，MTD不能有效防御，另一方面由于其变化攻击面的低效性，使得MTD中存在的问题逐渐显现。

网络空间拟态防御（Cyber Mimic Defense，CMD）技术在自身的漏洞和后门被攻击者利用的情况下，凭借其自身的异构冗余加上动态反馈的特性，依旧能够有效的抵御攻击者的入侵。作为一种新型的防御思路，CMD在网络攻防过程中体现出拟态系统所具备的自适应内生安全特性，有着广阔的应用前景。本文的组织结构如下：第二部分简要介绍了CMD的基础理论知识。第三部分对现有的主要的拟态调度技术做总结介绍。第四部分对未来拟态防御调度技术的发展做了展望。

2　拟态防御基础理论

2.1　概述

移动目标防御通过不断的改变自己暴露给外界的攻击面，使得攻击者在本时刻所掌握的目标系统信息在下一时刻失效，从而增大攻击成本，但是如上所述，移动目标防御具有变换低效性的缺点。N变体系统[13]试图通过多个系统实例同时在线，使攻击者必须同时攻击多个实例才能完成攻击，加大了攻击者攻击难度，但是在现有的N变体系统研究中，各个变体之间大多是同构，攻击者只要掌握了各个变体之间的共有内生漏洞，N变体的防御效果将会大大折扣。邬江兴院士通过将移动目标防御的动态性和N变体系统的异构冗余特性有机结合，创造性地提出了拟态防御基础理论。通过引入N（N≥3）个功能等价的异构执行体，并且根据自己的裁决结果和调度策略进行工作中的执行体集变换，使得攻击者无法简单的进行一次有效的攻击。另外对下线的执行体进行清洗等一系列操作。基于拟态防御，文献［14］从攻击链模型分析，尽管攻击者针对系统的一次攻击有效，但是由于裁决和调度策略以及下线清洗机制的存在，模型中的攻击维持阶段被破坏，从而达到增强系统安全性的目的。

2.2　DHR架构

1) 理论模型

拟态防御的基本模型[3]如图1所示：

在DHR架构中，输入代理在接收到输入端的输入序列之后，对输入序列进行复制，并下发给处于“活跃”状态的执行体。其中每个执行体之间功能等价，输入序列经由执行体处理完成之后，多模裁决模块接收到执行体的输出结果，对输出结果进行仲裁，将仲裁后的结果交由输出代理，生成输出序列。同时仲裁的结果也会反馈到负反馈模块，负反馈模块主要负责控制执行体的调度，以及对执行体执行的下线、清洗等操作。

2） 架构特性

异构性：是指各个执行体之间在功能等价的前提下，结构相异。异构性可以基于硬件实现，也可以基于软件从多角度提高异构性[14]。

冗余性：指对某一输入序列，有n（n≥3）个执行体去响应该序列。在保证异构度的基础上，增加冗余度能够改变攻击依赖的环境，增大了攻击者实施攻击的攻击难度，提高了系统的安全性[15]。

动态性：主要体现在引入闭环的负反馈机制，基于此种机制，异构冗余执行体可以实现动态调度，在保证冗余度和异构度的基础下调度，系统呈现不可测特性。

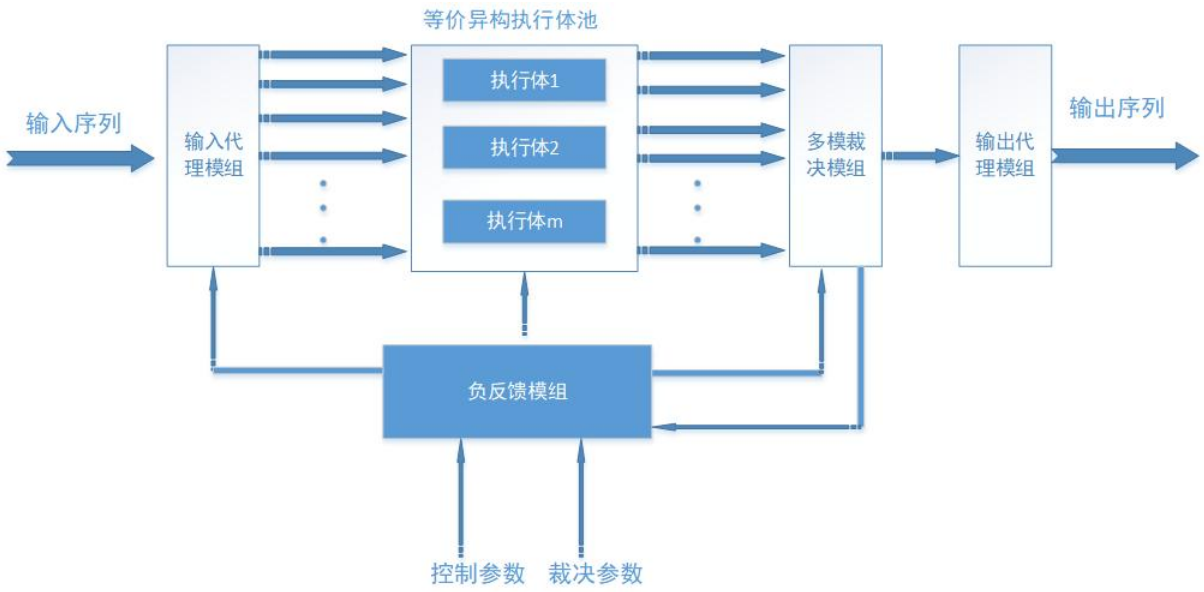


图 1　DHR 模型

3　调度算法研究

CMD 架构从设计到实现并最后应用到具体场景，涉及到多个关键技术、关键模块之间的协同合作，如调度策略、裁决策略、负反馈策略和清洗下线策略等。其中调度策略作为 DHR 架构的关键环节，本节我们将主要总结拟态防御调度策略的已有研究。这些研究占据整个拟态防御领域研究成果的主要部分。目前针对于拟态防御理论中调度策略的研究根据不同的划分标准会有不同的分类，从研究领域可以将算法划分为四个类别：SDN 控制层[21][22][23]、web 服务器[14]、拟态交换机[31]、DNS 服务器[32]。从研究出发点可以将调度算法所研究的问题分为三类：1）调度几个执行体，即调度数量问题。2）调度哪些执行体，即如何确定调度对象问题。3）什么时候调度，即如何确定调度时机问题。从调度算法所考虑的系统特性可以分为三个方面：1）动态性。2）负反馈特性。3）异构冗余特性。其中的算法所体现的负反馈特性可以根据防御方的信息对算法进行负反馈，也可以根据所捕获到的攻击方的信息进行负反馈，而且具体算法并不一定只突出表现一种系统特性，可以是三者的任意组合。拟态防御调度策略，首先要满足自己的调度约束（如调度下线之后的恢复清洗时间、系统开销等），然后从时空或者系统的异构度等多个维度来寻求最优解，可以从一个维度入手，也可以多个维度结合。

目前对拟态调度算法的前沿研究中，都体现了拟态架构中的动态性。笔者将从调度算法所研究的问题出发，对调度算法进行分类介绍。

3.1　调度数量

动态弹性调度算法

李等提出了一种基于效用的动态弹性调度算法[16]。其弹性体现在调度的动态性程度（对执行体进行调度、下线、清洗等操作的频率）和系统的执行体的数量即执行冗余度根据当前时间段网络环境动态改变。具体来说：若当前处于弱攻击场景（攻击较少）则可以减少调度的执行体数量和动态性；反之，若处于强攻击场景，则需要增加调度的执行体的数量和动态性。

差异化反馈调度数量算法

黄认为可以从两个方面去设计拟态调度策略[17]：1）调度的具体对象，关于调度对象的选择是目前的主要研究方向。2）调度的个数。基于异构度从调度数量出发，黄设计了一种差异化反馈调度策略。在该算法中利用 MOSS（measure of software similarity）[18] 得到系统中各个等价执行体之间的异构度。

该调度算法的主要思想及过程：首先将判决结果中处于异常的执行体（如果有）进行下线清洗，然后根据此次调度的个数 m，对可供调度的执行体遍历所有的排列组合，选出组合异构度最大

即调度函数最小的作为调度对象。该算法体现了拟态系统的动态性和负反馈特性以及异构冗余性。

3.2 调度对象

目前对调度对象的拟态调度算法研究中，多数是在调度个数为1的情况下去确定具体的调度对象。因为目前主流的大数一致判决算法会判决占少数的结果为错误结果，加之在实际拟态系统中，考虑系统开销和资源的消耗，所以执行冗余度一般为3，因此只会判决一个执行体出错，然后对其进行下线清理操作，调度入一个新的执行体在线运行。

基于正态分布调度算法

张等人以计算机操作系统调度算法为例引入了先来先来先换调度（FCFC）算法[19]，简要介绍了FCFC调度算法，然后从内部和外部角度分析先来先换算法的不足（内部视角即防御者角度，外部视角即攻击者角度）：当攻击者不计时间和攻击成本，持续多个周期对系统进行探测，便能发现系统的调度规律，从而从攻击者视角而言，目标系统呈现一种伪动态。基于伪动态原因，张等在FCFC调度算法的基础上，引入了人工加权调度算法，给系统中的异构执行体人工地赋予一定的权值，然后异构执行体按照自身所被赋予的权值进行排序，拟态调度策略根据的排序结果选择调度到运行状态的执行体。基于人工加权的拟态防御调度算法虽然解决了FCFC调度算法的伪动态问题。但是由于其一方面每次依靠人工来赋予权值，无形中增加了系统运营成本，另一方面还会有社会工程学方面的风险，不能保证完全可控。最后，分析了完全随机调度算法会使得系统不可控，基于系统可控性的分析，从拟态系统的动态性入手，提出了一种基于正态分布的拟态防御调度算法。

表1给出三种算法的对比：

表 1　算法对比

算法	是否具有动态性	是否可控	安全性
先来先到算法	否(伪动态)	是	低
人工加权算法	是	否	中
正态分布调度算法	是	是	高

保证系统异构度的调度算法

李、任等，提出一种基于拟态防御的SDN服务部署架构[20]，李从SDN服务部署系统的异构度入手，认为在一次调度之中，调度完成之后的处于运行状态恶的执行体集之间的异构度应该大于一个阈值，避免由于异构度过低而出现共模逃逸的概率。同时李指出SDN服务部署系统中可供选择的异构执行体较少，如果采用随即调度，不添加任何限制条件，则会降低系统安全性。

最后，从拟态防御的异构度约束出发，结合上述分析，提出了一种当整体异构度系数取值小于（（K－1）~） 0.4时满足系统异构度。式中的K为执行集中执行体的个数。该算法考虑了拟态系统的动态性和异构冗余性。

自适应调度算法

自适应调度，指系统指根据所获取的信息，负反馈给调度策略模块，更新相关参数，从而动态改变调度策略。其中所获取的信息可以是各个执行体（即防御方）的信息如历史置信度也可以是攻击方的信息如攻击场景或者攻击类型等。这些自适应调度算法充分考虑了DHR架构的负反馈特性、动态性和异构冗余特性。

1) 基于信誉度和相异度的自适应调度算法

沈等提出了一种基于信誉度和相异度的自适应调度算法[21]。其核心思想是根据每一次裁决结果，通过负反馈模块，动态更新信誉度即更新下一次裁决调度的相关参数。

该算法加强了调度模块与多模裁决模块之间的交互，使裁决结果影响调度算法中的信誉度。当裁决算法便是执行体异常时，相对应的异常执行体的信誉度会下降，并实时更新，用与下一次的调度之中。

2） 基于Q-learning的动态调度学习算法

顾提出了一种基于Q-learning的动态调度学习算法[22]，将机器学习与拟态防御结合，其目标为通过算法的一系列步骤，来决定是否采取防御动作即进行调度。其具体流程如下：

1） 根据裁决结果，判断是否收到攻击，若有，则执行调度，并转至4)。若无，则下一步。

2） 随机生成一个参数a，若a＞b，则随机决定是否调度，并转至4)。若a≤b，则下一步。

3） 根据反馈函数选择最优动作，即是否调度以及怎么调度。

4） 更新反馈函数和回报函数。

其中a为随机算法产生的随机数，b为贪婪参

数，表示防御者以b的概率不依据反馈函数进行动作的选取（随机动作：随机选择调度或者不调度），以1-b的概率来根据反馈函数来决定采取的防御动作。b越大，随机动作可能性越大。初始化b大于0，随着时间的推移已经算法的迭代，b越来越小并趋近于0。

3) 基于历史行为的自学习调度算法

基于进一步增强控制层的可靠性的目的，李提出了一种SDN多控制器的动态自学习调度方法[23]。该调度方法以高可靠性（高历史置信度）为目标，首先根据各个执行体的历史行为，更新执行体的可靠性属性，在调度时，充分考虑各个执行体基于历史行为的可靠性，以筛选出更高安全的控制器组合。具体来说，在满足相关约束条件的情况下，被调度上线执行任务的控制器应该满足：1）可靠性高，即其历史统计中的故障率较低；2）避免将由于相同攻击而导致故障的执行体在后续任务中同时上线。从而增强拟态架构下SDN网络的抗攻击效果。 并且在可行性方面，从复杂度的角度去分析验证算法的实际可用性。

该算法相对于文献［21］所提出的算法，有相同之处，但是本算法创造性的提出了避免出现共模故障的执行体下次同时执行任务的观点，减少了共模逃逸的可能性。

4） 基于攻击信息的动态负反馈调度算法

吕认为攻击类型可以被分为具有针对性的非一致性攻击和无针对性的非重复性攻击两类[24]。对于防御方而言，可以通过自身所设置的监测模块，来获取攻击的信息，判断攻击类型。具有针对性的攻击往往对某类控制器有效，该类控制器为非一致性攻击的目标控制器，非重复性攻击则无特异性。如果基于先前监测判断出此次攻击的类型为非一致性攻击，则将调度该类型非一致性攻击的目标控制器的概率降低或者不调用。否则，执行随机调度策略。

基于上述分析，可以将控制器做分类，并且记录好每次监测的详细信息：何种控制器被攻击，被攻击次数。从而根据监测信息，检验为何种攻击，从而执行调度算法。

3.3 调度时间

基于滑动窗口模型的调度序列控制算法

文献［12］指出拟态调度过程应该着重两个角度：目标和时间。第一个角度是如何选择合适的执行体，关于该方向目前已经有大量的研究。第二个角度是执行体切换时机的选择（调度时机）。目前虽然有一些研究[25][26]提到调度时机的选择。但是，它们没有明确指出调度时机选择过程中的具体模式和值，也没有设计具体的算法去实现。郭针对拟态化架构中的元数据服务DHR结构的调度机制，提出了一种基于滑动窗口的调度序列控制方法[27]。

该算法的主要思想：将调度窗口定义为具有时间和大小（出现错误的执行体个数）属性的模型，调度触发的条件有两个：1）时间到达滑动窗口的时间属性所设置的阈值。2）出现异常的执行体到达滑动窗口的大小属性的所设置的阈值。当满足上述两个条件时，根据拟态调度策略执行调度。然后根据拟态调度时间算法的反馈信息动态更改调度窗口的时间和大小。若是因为到达时间阈值而触发调度，则从侧面反应系统处于一种弱攻击场景，下一次的调度窗口的时间阈值则会根据公式相应增大，大小阈值相对应缩小，弥补安全性问题。如果是因为到达大小阈值而触发调度，说明处于较强的攻击场景（攻击频率高），则窗口的时间阈值相应减小，大小阈值相应增加，增强鲁棒性。

该算法能够根据不同的攻击场景动态的改变窗口的时间和大小参数，提高了系统的安全性、鲁棒性。

基于更新收益理论的调度时间算法

安全领域中攻击和防御双方的探测模型通常使用贝叶斯-斯坦科尔伯格博弈[28]模型来描述。卢[29]在前人研究的基础上，将安全领域的攻防模型建模为贝叶斯-斯坦科尔伯格博弈模型，在该模型中，攻击方和防御方的博弈目标都是通过各自的策略来获得最大的收益，其中收益由收益函数编表示。将其应用在拟态防御攻防模型中：防御方的策略即调度，而选择众多攻击类型中的一种攻击是博弈模型中的攻击方的策略。算法站在博弈模型中的防御方，从防御者的角度分析，在一轮博弈之后，防御者可依据环境以及攻击信息更新收益函数，并且假设防御方可以或者攻击者的能力和攻击成功的概率，基于此假设，从而求得均衡策略即最佳的调度策略。

Lu等提出了一种基于更新收益理论的调度时间算法[30]。该算法认为遭受攻击给系统带来的损失和调度所带来的开销会影响收益函数，然后基于博弈模型，计算最佳的调度时间。

上述算法分类总结如表2所示：

表2 拟态调度算法总结

算法	应用场景	划分粒度	系统特性	出发点
动态弹性调度算法	SDN	控制器	动态性 负反馈特性 异构冗余性	调度数量
差异化反馈调度数量算法	SDN	控制器	动态性 负反馈特性	调度数量
基于正态分布调度算法	无具体应用场景	/	动态性	调度对象
保证系统异构度的调度算法	SDN	控制器	异构冗余性 动态性	调度对象
自适应调度算法	SDN	控制器	动态性 负反馈特性 异构冗余性	调度对象
基于滑动窗口模型的调度序 列控制算法	无具体应用场景	/	动态性 负反馈特性	调度时间
基于更新收益理论的调度时间算法	SDN	控制器	动态性 负反馈特性	调度时间

由总结表不难看出，由于SDN控制与转发平面的分离以及控制层面对于高安全性的需求，使得目前拟态防御主要的研究领域集中在SDN，执行体的划分粒度为控制层面的控制器。拟态调度算法都体现了拟态系统的动态性，从而使得攻击者很难掌握系统的变化规律。负反馈特性使得每一次的调度更准确高效。异构冗余特性则是相对正确公理的前提保证，能够保证拟态判决的准确性和系统的安全性。

4 总结与展望

由上一章节分析我们可以看出，目前关于拟态的防御的前沿调度算法的研究还没有广泛应用到诸多领域，应用最多的还是SDN的控制层，在拟态交换机、拟态Web服务器等领域所采用的算法任然没有应用前沿研究。目前对拟态防御调度算法的研究任然具有以下不足：

1）大多数算法任在理论阶段，没有和具体领域相结合。

2）对于执行体单元的划分粒度不够，如SDN领域的执行体大都为控制器。

3）调度算法的出发点都是从防御者入手，即考虑防御方的属性，而忽略了攻击方的攻击属性。

因此，对于后续的研究，研究者可以下述这四点入手：1）在算法设计阶段，以更细的粒度在系统的关键位置去构建异构冗余执行体，可以减少不必要的开销，也可以增强系统的安全性（一般认为结构越简单，系统漏洞越少）。2）在设计算法的时候，也可以考虑攻击属性，如攻击类型等，这些信息都是可以被防御系统所嗅探到，我们应该充分利用。针对不同类型的攻击，系统中的执行体的历史执行度以及防御能力都不相同，在调度的时候引入攻击类型参数，使得调度更有针对性，更加安全可靠。3）考虑在系统开销允许的情况下，实行多算法融合，多层调度。4）后续的研究也应该更加注重将理论应用在具体的工业领域，权衡好系统开销和安全性问题。

参考文献:

[1] 刘彩霞．专题:网络空间安全[J]. 无线电通信技术，2020，46（04）:377.

[2] Ma B , Zhang Z . Security research of redundancy in mimic defense system[C]// 2017 3rd IEEE International Conference on Computer and Communications (ICCC). IEEE，2017.

[3] 邬江兴．网络空间拟态防御导论: 下册[M]. 科学出版社，2017.

[4] National Institute of Standards and Technology, Guide to Intrusion Detection and Prevention System(IDPS), Feb，2007.

[5] Spitzner L . Honeypots: Catching the insider threat[C]// Computer Security Applications Conference. IEEE，2003.

[6] Jana S , Porter D E , Shmatikov V . TxBox: Building Secure, Efficient Sandboxes with System Transactions [C]// Security & Privacy. IEEE，2011.

[7] Wang F, Gong F, Trivedi K, et al. A scalable intrusion-tolerant architecture for distributed services [C]//2nd Annual IEEE SMC Information Assurance Workshop（New York，2001). 2001.

[8] 蔡桂林. 移动目标防御技术若干关键问题研究[D]. 国防科学技术大学，2016.

[9] NITRD C. IWG: Cybersecurity game-change research and development recommendations[J]. 2013.

[10] Executive Office of the President. Trustworthy cyberspace: strategic plan for the federal cybersecurity research and development program [J]. 2011.

[11] Zhuang R, Zhang S, DeLoach S A, et al. Simulation-based approaches to studying effectiveness of moving-target network defense [C]//National symposium on moving target research. 2012，246.

[12] Guo W, Wu Z, Zhang F, et al. Scheduling Sequence Control Method Based on Sliding Window in Cyberspace Mimic Defense[J]. IEEE Access，2019，8：1517-1533.

[13] Kc G S, Keromytis A D, Prevelakis V. Countering code-injection attacks with instruction-set randomization [C]//Proceedings of the 10th ACM conference on Computer and communications security. 2003：272-280.

[14] 仝青, 张铮, 张为华, 等. 拟态防御 Web 服务器设计与实现[J]. Journal of Software，2017，28(4).

[15] 邬江兴. 网络空间拟态防御研究[J]. 信息安全学报，2016 (4)：1-10.

[16] 李军飞. 软件定义网络中拟态防御的关键技术研究[D]. 战略支援部队信息工程大学，2019.

[17] 黄前淼. 基于拟态防御的 SDN 服务路径配置及其验证机制研究[D]. 浙江工商大学，2020.

[18] Qiu D H, Li H, Sun J L. Measuring software similarity based on structure and property of class diagram[C]//2013 Sixth International Conference on Advanced Computational Intelligence (ICACI). IEEE，2013：75-80.

[19] 张震骁. 拟态防御动态调度策略研究[D]. 郑州大学，2018. 震骁. 拟态防御动态调度策略研究[D].

[20] 李传煌, 任云方, 汤中运,等. SDN 中服务部署的拟态防御方法[J]. 通信学报，2018，039(0z2)：121-130.

[21] 沈丛麒, 陈双喜, 吴春明,等. 基于信誉度与相异度的自适应拟态控制器研究[J]. 通信学报，2018，39(S2)：177-184.

[22] 顾泽宇,张兴明,魏帅. 基于增强学习的自适应动态防御机制[J]. 小型微型计算机系统，2019，40(02)：401-406.

[23] Li J, Wu J, Hu Y, et al. DSL: Dynamic and Self-Learning Schedule Method of Multiple Controllers in SDN[J]. ETRI Journal，2017，39(3)：364-372.

[24] 吕迎迎. 拟态 SDN 控制器架构安全关键技术研究[D]. 战略支援部队信息工程大学，2018.

[25] Qi C, Wu J, Hu H, et al. An intensive security architecture with multi-controller for SDN [C]//2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE，2016：401-402.

[26] Qi C, Wu J, Hu H, et al. Dynamic-scheduling mechanism of controllers based on security policy in software-defined network[J]. Electronics letters，2016，52(23)：1918-1920.

[27] 郭威. 分布式存储系统拟态化架构与关键技术研究[D]. 战略支援部队信息工程大学，2019.

[28] Paruchuri P, Kraus S, Pearce J P, et al. Playing games for security: An efficient exact algorithm for solving Bayesian Stackelberg games[J]. 2008.

[29] 卢振平. SDN 控制器主动防御关键技术研究[D]. 战略支援部队信息工程大学，2017.

[30] Lu Z, Chen F, Cheng G, et al. Towards a dynamic controller scheduling-timing problem in software-defined networking[J]. China Communications，2017，14(10)：26-38.

[31] 宋克, 刘勤让, 魏帅, 等. 基于拟态防御的以太网交换机内生安全体系结构[J]. 通信学报，2020，41(5)：18-26.

[32] 任权, 邬江兴, 贺磊. 基于 GSPN 的拟态 DNS 构造策略研究[J]. 信息安全学报，2019，4(2)：37-52.

[作者简介]

梅波（1998—），男，硕士，主要研究方向为网络空间安全、拟态防御。

赵博（1981-），男，博士，副研究员，主要研究方向：拟态防御架构，芯片设计。

郭乔羽（1997-）男，硕士，主要研究方向为网络安全。

王苏南（1984-），男，博士，副教授，研究方向为计算机网络体系结构。

郭雷（1981-），男，硕士，工程师，主要研究方向为传输通信。

SDN网络QoS流量调度和路径更新方法

陈立水[1]，左宇飞[2]，唐亚哲[2]

[1]通信网信息传输与分发技术重点实验室，石家庄050081；

[2]西安交通大学 计算机科学与技术学院， 西安710049

摘　要：随着网络流量的显著增加，网络带宽利用率越来越受到重视。传统网络受限于其分布式架构，流量调度技术不能处理网络长时间运行后出现的带宽碎片化等问题。本文设计了SDN网络环境下的基于QoS的流量调度方法。该方法能够整合资源碎片，提高带宽利用率。同时，本文提出一致性更新算法解决流表动态迁移的问题。功能和性能测试表明，论文提出的方法能够尽可能多的容纳业务流，提升带宽利用率，并在流量调度的过程中保持业务流的丢包率维持在一个较低的水平。

关键词：软件定义网络、流量调度、一致性更新

QoS-aware Traffic Scheduling and Route Updating in SDN networks

CHEN Lishui[1], ZUO Yufei[2], TANG Yazhe[2]

1.Science and Technology on Communication Networks Laboratory, Shijiazhuang 050081;

2.School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049

Abstract: With the significant increase of network traffic, more and more attention has been paid to network bandwidth utilization. Traditional networks are limited by their distributed architecture, hence traffic scheduling technology cannot deal with issues such as bandwidth fragmentation in long-term running of networks. This paper designs a QoS-based traffic scheduling method in SDN networks. The method can integrate bandwidth fragments and improve bandwidth utilization. At the same time, this paper proposes a consistent update algorithm to solve the problem of dynamic migration of flow tables. Experiments show that the method proposed can accommodate as many flows as possible, improve bandwidth utilization, and maintain the packet loss rate of service flows at a low level during the flow re-scheduling process.

Key words: Software Defined Networking; Traffic Scheduling; Consistent Update

1　引言

随着网络流量的增加，网络管理部门期望在有限的网络资源条件下为更多的用户提供服务。因此，如何充分整合网络资源，实现流量合理调度，提高网络整体带宽利用率，容纳更多的流已经成为热点问题。特别地，网络长时间运行情况下，会导致网络出现链路带宽碎片。进而出现即使网络的总体带宽足够，也不能接纳新流的问题。

我们看一个例子。图1所示网络拓扑中，有表1所示的4条流f1到f4正在传输（我们称之为热流）。假设某一个时刻4条流的路由配置是：f1：S12-S2-S1- S4； f2：S13-S2-S3-S6； f3：S17-S8-S9-S11-S15； f4：S18-S8-S7-S10-S11-S16。此时来了一个预约新流（我们称之为冷流）。在流f1和流f2占用部分带宽的情况下，路径和路径带宽剩余分别为20M和10M，均不能满足预约业务流的带宽需求。但是，如果我们把f5和f1，f2，f3和f4重新一起进行QoS路由计算，我们会发现解是存在的。5条流情况下，新的路由配置是：f1：S12-

基金项目：通信网信息传输与分发技术重点实验室基金课题资助项目（SXX18641X024）

S2-S3-S4； f2： S13-S2-S1-S4-S3-S6； f3： S17-S8-S7-S10-S11-S15； f4： S18-S8-S7-S10-S9-S11-S16； f5：S14-S2-S1-S4-S5-S8-S9。这样的场景在数据中心[1]中是很常见的，因为目前数据中心常见的拓扑结构都是类似于胖树（Fat tree）的结构，端到端之间都存在多条路径可以选择。显然，我们可以通过整合链路带宽碎片，重新规划流传输路径从而容纳更多的流。这里的挑战是：1）有带宽需求的流的路径规划问题，已经证明是NP问题，如何快速得到精确的解？ 2）f1到f4是4条有带宽需求的热流，如何在不中断这4条热流传输的情况下，把它们从旧的路径更新到新的路径上？

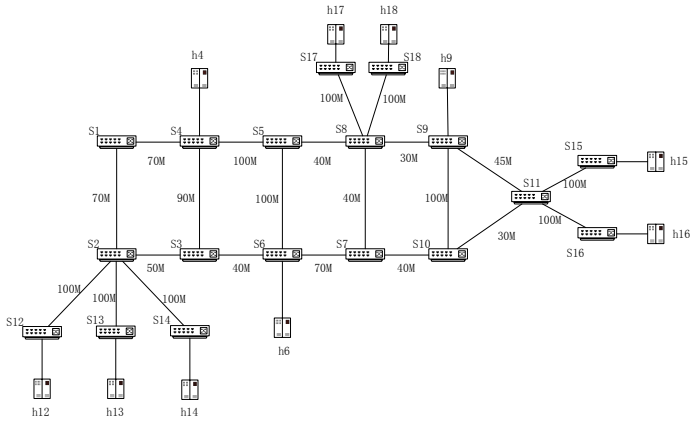


图1 测试拓扑图

表 1 网络中现有流信息

数据流	源目的交换机	带宽需求
f1		50M
f2		40M
f3		15M
f4		25M

现有的基于SDN技术的流量调度[4][5][6]，主要考虑了在当前全局网络状态下，提出一个或者多个约束条件，为到来的业务流寻找一条符合需求的路径，同时利用SDN特性尽量使流均衡地利用全局网络带宽资源，提高带宽的利用率。所应用的场景为带宽资源较为充足的情况下，如何尽可能地避免局部链路拥塞，同时保证业务流带宽需求。目前为止，还没有看到解决前述的带宽碎片化等问题的相关研究。

本文提出新的SDN网络QoS流量调度方法。基本思想是：将新的用户流量（称为冷流）与正在网络中运行的用户流量（称为热流）统一进行QoS路由计算。根据热流的新旧路由路变化情况，进行路由的热更新。更新完成后，新的用户流量将能够找到满足其QoS要求的路由，由此将能够增加整个网络中容纳的流数量。具体来讲，本文主要工作包括：1）设计和优化QoS路由算法；2）针对流表动态更新时引发的流量黑洞、链路拥塞、环路和死锁等一致性等问题，基于Petri网对热流路径变化进行建模，求解热流的路径变化顺序。

2 相关工作

如前文所述，本文的主要工作包括QoS路由算法以及根据新旧路由的变化对热流进行路径切换。因此，下面就从上述两个方面介绍目前的相关研究工作。

QoS路由算法，通常研究在最短路径的基础上，加入额外的资源约束后如何寻找合适的路由。论文［4］利用SDN南向接口协议采用主动的方式实时采集网络链路上的流量信息，在SDN环境下基于Dijkstra算法设计了一个具有带宽约束的路由算法，选择多个可行路径中剩余带宽最多的路径，但是并没有考虑尽可能多地容纳更多有带宽需求的流。论文［5］优化了基于SDN南向接口协议的保障带宽路径选择算法，结合全局链路带宽信息，同时在选择多条可传输路径中的最优路径的时候，考虑链路剩余带宽方差，提高传输路径的健壮性，另外，结合交换机出端口队列设置为不同优先级的业务流提供相应的带宽。但是对于进入同一出

端口队列的业务流之间的优先级没有做限制，可能会导致该队列出现流量拥塞。论文［6］通过使用链路发现协议监控模块和Echo监控模块来设计链路延迟的监测机制，实时获取链路交换机之间、控制器和交换机之间的传输时延，并将该时延作为链路权值，利用Dijkstra算法为业务流计算最短路径，选择一条链路时延最小的路径作为最优传输路径，同时计算一条次优路径增加容错性。该工作也只考虑了时延这一个QoS参数作为服务质量的评估标准，无法满足其他约束。

现有的研究在选择路径时即使采取最佳适应的方式，也会在一定程度上出现链路资源碎片，或者一些小流传输完成后释放所占用的小部分链路带宽，同样会出现链路资源碎片。因此，在网络较为拥塞，链路资源碎片化较多的场景中，采取上述流调度算法，业务流可能无法直接加入网络，这正是本文着重解决的问题。

SDN架构下流的传输路径发生改变时，控制器需要指示数据平面进行路径的更新，然而，更新SDN交换上的流表通常是任意的选择交换机的顺序进行更新，可能会产生数据平面的不一致性问题，从而导致流量黑洞、流量环路、死锁、链路拥塞等问题。因此，SDN环境下有很多基于流表更新过程中保持一致性属性的研究。

为了避免流量黑洞和环路的出现，Wang等人[7]提出一种倒序更新流表的机制，控制器首先倒序下发新的流表然后正序删除旧的流表。Mahajan R等人[8]提出可以采用创建简单依赖树的方式，即将网络拓扑中根据新的传输路径中的数据包的流向确定交换机之间依赖关系，将目的交换机设为根节点，创建该依赖树，子节点的流表的更新顺序仅仅取决于其直接关联的父节点的流表更新，因此可以同步更新一些交换机流表且保持一致性。但是这样的依赖树会降低交换机流表更新的并发性，因此，该论文又提出一个缩小节点依赖性的算法，利用环路检测算法创建一个极小依赖的流表更新拓扑，最终权衡流表更新的一致性和更新速度。以上两个研究主要解决了环路和黑洞的问题，但是没有关注更新过程中链路拥塞问题，在链路资源较少的情况下，任意顺序下数据流的迁移可能会造成某条链路的拥塞。Wu等人[9]结合倒序更新流表机制，着重关注了更新的顺序，解决了链路拥塞的问题。它将需要更新流表的业务流新旧路径按照一定规则划分成段，创建每两个不重合流段和相关联的链路之间的依赖关系，通过依赖图确定更新顺序，更新流表的时候，各个分段在没有链路依赖的情况下，可以并行更新，提高更新速率。当出现死锁的时候，降低某条流的带宽需求，在死锁问题解决之后，恢复带宽需求。该论文的研究工作很好的解决了上述一致性更新问题，但是在死锁问题的解决上还是会造成某些流的带宽降低。

3 系统设计

3.1 基于QoS的路由

网络中每一条链路的带宽资源都是有限的，在网络带宽占用率较高的情况下，为了使网络中能够容纳更多数目的业务流，应该尽量整合网络中碎片化的带宽资源，进一步提高链路带宽利用率。本节主要介绍基于QoS的路由算法。

1） 问题建模

将网络拓扑抽象为一个无向联通图$G=(V,E)$，其中V表示网络中所有的交换机节点集合，E表示转发设备之间的链路集合。设网络包括用户预约的新流在内共有m条流，每条流都有其带宽需求，对于第i条流，其带宽需求为b_i，在无向连通图中，与该流源主机直接相连的交换机节点u_i和与目的主机直接相连的交换机节点v_i之间共有n_i条路径，对这条路径按照其所经过的交换机跳数从小到大进行排序，构建的路径矩阵M如公式1所示。

$$M=\begin{bmatrix}\lambda_{11} & \cdots & \lambda_{1n}\\ \vdots & \ddots & \vdots\\ \lambda_{m1} & \cdots & \lambda_{mn}\end{bmatrix}\quad 其中\ n=\max_{i=1,2,...m} n_i \quad （公式1）$$

路径矩阵中的λ_{ij}表示第i条流，且选择第j条传输路径，由于不同流可选择传输的路径数目不同，所以，将$\left\{\lambda_{ij}|i\in\{1,2,...,m\}\ j\in\left\{n_i+1,n_i+2,...,n\right\}\right\}$视为不可选路径。

本系统希望网络能够容纳包括用户预约的新流在内的所有业务流，即在给定的网络拓扑下，部署所有流的可传输路径，因此，系统目标即为业务流数目最大，目标函数如公式2所示。

$\max \sum_{i=1}^{m} x_i$ 其中 $x_i \in \{0,1\}$（公式 2）

这里的可传输，指的是任意一条链路上，所传输的流占用的带宽都不应该超出该链路所能提供的最大带宽。形式化表示为设 $x_{ij}=1$ 对应路径 λ_{ij}，表示第 i 条流可以进行传输且选择其第 j 条路径，$x_{ij}=0$ 表示第 i 条流不选择第 j 条路径，根据路径矩阵可知，x_{ij} 满足公式3和公式4约束。

$$\sum_{j=1}^{n} x_{ij} \in \{0,1\}, i \in \{1,2...m\}, j \in \{1,2...n\} \quad \text{（公式 3）}$$

$$\sum_{j=n_i+1}^{n} x_{ij} = 0 \quad \text{（公式 4）}$$

设 $e(u,v)$ 表示两个相邻的交换机节点 u 和 v 之间的链路，其中 $e(u,v) \in E$，$u \in V$，$v \in V$。$C_{e(u,v)}$ 表示链路 $e(u,v)$ 当前可用带宽。满足公式5和公式6约束。

$$r_{ij,e(u,v)} = \begin{cases} 1 & e(u,v) \in \lambda \\ 0 & e(u,v) \notin \lambda \end{cases} \quad \text{（公式 5）}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} b_i x_{ij} r_{ij,e(u,v)} \le C_{e(u,v)} \quad \text{（公式 6）}$$

2） 算法设计

该问题的抽象描述为：给定一个网络拓扑，需要将所有的业务流加入到该网络中，并且每一条链路的带宽占用都不会超过该链路带宽初始值上限，每一条新流加入计算最短路径时都会受到已经在网络中部署的流的影响。对于 m 条流来说，部署方案数目即为 m 条流的全排列 $m!$，解决该问题最差需要在 $O(m!)$ 的时间复杂度内，不能在多项式时间内求解，但是如果给定某种方案，即 m 条流的传输路径，可以在多项式时间内验证该方案，因此该问题是一个 NP 问题。对于 NP 问题，启发式算法能够给出优化问题的每一个可行解，但是该可行解与问题最优解的偏离程度无法被估计，极易陷入局部最优解。因此，该算法使用确定性解法，保证在链路资源足够的情况下一定能找到所有业务流的部署方式，该算法的算法流程图如图2所示。

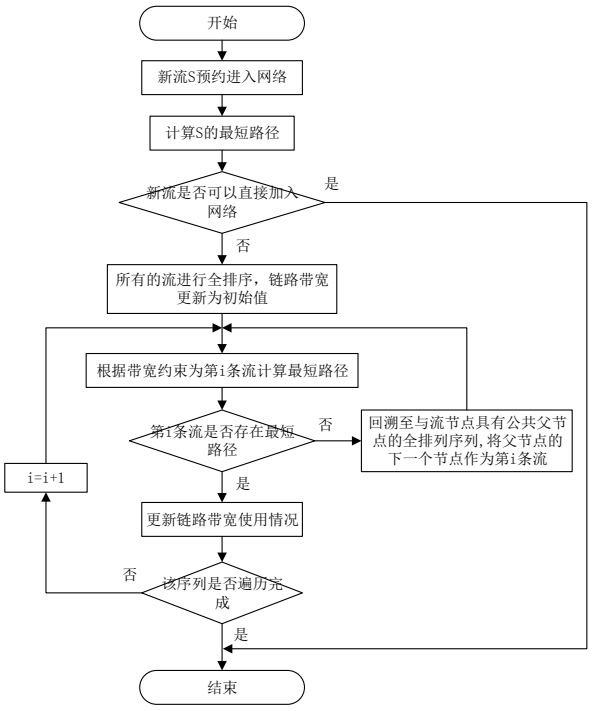


图2　基于QoS的路由确定性算法流程图

3） 算法复杂度

对于带权有向图 $G(V,E)$，每个节点被访问的次数为 $(v-1)$，因此对于 v 个节点，最短路径算法的复杂度为 $O(V^2)$，按照上述算法流程，最差情况下迭代的次数为流数目的全排列 $m!$，总体复杂度

为$O(|V|^2|m!|)$。在业务流数目较多的情况下，复杂度较高，因此，本文对算法进行优化。按照贪心策略，首先将所有流按照由大到小的顺序排序，规定每一条业务流在多条代价相同的最短路径中，按照最佳适应的方式选择最优路径，即选择该流加入后剩余带宽最小的链路。当$m! \leq m^3$时，按照全排列的计算方式求可行解，即时间复杂度为$O(|V|^2|m^3|)$，当$m! > m^3$的时候，设置修枝策略，如果某个子序列首个业务流所占用带宽较小，不再计算该序列，因为小流首先加入会使网络碎片化较多，该算法认为无法计算出部署所有流的方案。此算法将中位数作为判断流大小的依据，即每个序列首先考察前$\frac{m}{2}$条流的全排列，对于m条流，复杂度为$O\left(\frac{m!}{2^m}\right)$，因此总体复杂度为$O\left(|V|^2|\frac{m!}{2^m}|\right)$。

下面对剪枝原理进行分析。算法在执行过程中需要遍历所有可能的流量排序，复杂度较高且算法的目的是找到一个可行的流量规划方案。因此，可以通过两种方法降低算法执行代价，分别是提高有解时给出解的效率，和降低无解时的代价。由于算法的目的是找到一个规划方案，因此该问题本质是一个求解问题而非优化问题，所以在算法执行过程中找到解时立刻返回结果，可以有效提高在问题有解时的求解效率；为了降低在问题无解时的复杂度，基于贪心策略，在算法执行前对流量根据其占用带宽大小由大到小排序，使得在流量规划过程中倾向于优先规划带宽占用较大的流量，因此当小带宽流量被优先规划的时候意味着之前优先规划大流量的情况失败，根据贪心策略，优先规划小带宽流量的情况相比较于优先规划大带宽流量更容易失败，这意味着该问题无解，因此通过剪枝策略，去掉由较小带宽流量优先规划的情况可以有效提高算法在遇到问题无解的情况下的执行效率。

3.2 一致性更新

在计算出所有流新的传输路径后，为了保证业务流的服务质量，流量调度不中断已有业务流的传输，SDN控制器会下发新的流表，指导底层的转发设备动态地完成流迁移。在流迁移的过程中，控制器需要仔细协调跨多个交换机的规则更新[7]，确定各条流迁移的顺序，保证流更新一致性。所谓的一致性，包括避免流量黑洞、环路、链路拥塞和死锁。关于一致性的定义和解释，请参看参考文献［7］。

为了避免流量黑洞和环路，要从整体上按照一定的顺序更新交换机的流表，可以先倒序安装新流表，然后正序删除旧的流表，即在计算出某条业务流对应的新的转发路径后，从路径末尾的交换机开始下发优先级较高的新流表，然后再删除旧的流表，这样就可以保证流到达更新了流表的交换机后，一定能够找到转发端口和转发路径到达目的节点，不会出现流量黑洞和环路。因此在流表更新的过程中，需要重点处理的是因流表更新顺序而产生的链路拥塞问题和两条流互相等待对方释放链路资源而导致的死锁问题。本文基于链路和数据流的传输路径变迁，在Petri网的基础上建模，设计一种决策网络中流更新顺序的决策方法。

在满足给定条件或者约束的前提下，Petri模型将会自动地进行状态转换，因此，可以使用此模型分析系统的可达性[10]。结合本文提出的问题场景，创建对应的模型，将拓扑中每条链路表示为S元，链路当前剩余带宽为库所的容量，二者结合即为Petri网的一个标识。网络中每条流的变化表示为Petri网中的变迁。在实际网络拓扑中，链路的总带宽是一定的，也就是说仓库的容量最大值总是由设定的初始值决定，且在变迁的过程中无论流如何迁移变化，总不会超过其最大值。因此，变迁是否能够发生只需要考虑该流迁移目的链路是否能够提供足够的带宽。举一个简单的例子，如图3所示，对于图中流的迁移，如果随机按照某个顺序更新流表，例如（F1，F2，F3，F4），那么链路（s1，s5）和（s2，s5）会在更新的过程中出现链路拥塞，这可能会造成丢包或者增大时延，无法保证业务流的服务质量。通过Petri网模型可以找到一个合理的更新顺序从而避免上述问题的发生。

利用Petri网表示上述网络拓扑图，如图4所示，该Petri网的可达性状态分析图，如图5所示。

根据Petri网建立的模型，封装链路节点和流迁移节点，创建一个基于流量迁移和相关链路的

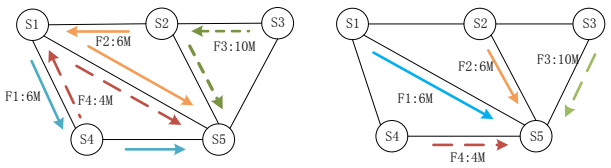


图3　流迁移示例图

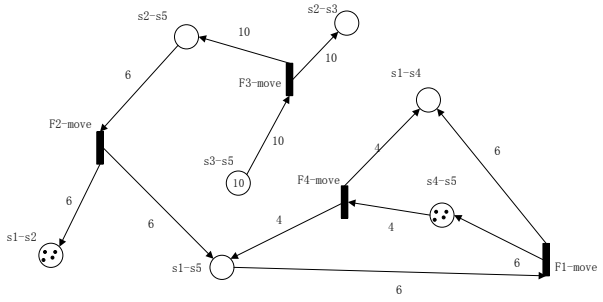


图4　Petri网表示

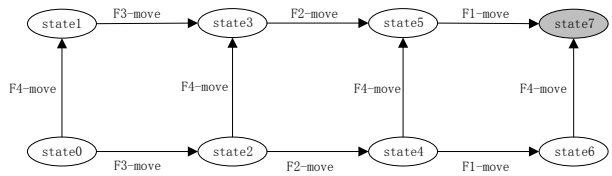


图5　可达性状态分析图

依赖图。在依赖图中，用(s_i, s_j)代表两个交换设备之间的链路，$f_i move$代表某条流的迁移，$(s_i, s_j) \rightarrow f_i move$表示该流的变迁将会消耗链路资源，$f_i move \rightarrow (s_i, s_j)$表示该流的变迁将会释放对应的链路资源，每个流迁移节点的前置链路节点为流迁移之后传输路径上的各条链路，后置链路节点为流原始传输路径上的各条链路。对于图3所示的拓扑以及流迁移的情况，创建的依赖图如图6所示。通过该依赖图得知，流F3和流F2具有链路资源上的依赖关系，F2-move事件的发生需要F3-move释放(s_2, s_5)上的带宽，同理，F1-move等待F2-move释放资源，F4-move等待F1-move释放资源，一致性更新算法需要根据上述依赖关系找出一个合理的更新顺序。

4　系统测试

本文面向网络中的资源碎片化问题，结合QoS路由和流表更新技术，提出解决方案。在QOS路由方面，主要关注可接受的时间内求出可行解；在流表更新方面，关注的是通过Petri网建模，完成流表更新顺序的求解。因此，测试时更多地关

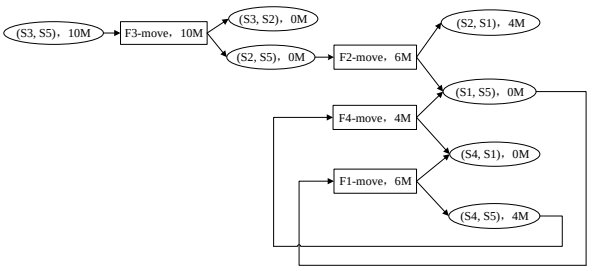


图6　流迁移依赖图

注方案的可行性和方法的可用性。

为了使数据流路径切换之后仍能找到一条到达目的节点的路径，选择的网络拓扑任意两个节点之间应该尽可能有多条路径。参考美国骨干网对等互联的网状网模型，本实验采用18个虚拟交换机和10台虚拟主机构成测试拓扑图，图中包括五个环路。为方便区分主机，环路连接多个主机的交换机节点与每个主机之间增加一台交换机，拓扑结构如图1所示，为方便计算，分别为每条链路设置初始带宽值。

4.1　模块功能测试

4.1.1　基于QoS的路由模块测试

在网络带宽占用率较高的情况下，系统接收用户预约的新流，重新部署所有的业务流，在带宽约束下进行选路的测试，验证本系统可以使网络容纳更多数目的流，提高带宽利用率。设定网络中已经存在四条业务流，其流信息如表2所示。

表2　网络中现有流信息

数据流	源目的交换机	源IP	目的IP	带宽需求
f1	$(s_{12} \rightarrow s_4)$	10.0.0.12	10.0.0.4	50M
f2	$(s_{13} \rightarrow s_6)$	10.0.0.13	10.0.0.6	40M
f3	$(s_{17} \rightarrow s_{15})$	10.0.0.17	10.0.0.15	15M
f4	$(s_{18} \rightarrow s_{16})$	10.0.0.18	10.0.0.16	25M

首先利用QoS路由模块初始化四条流的传输路径，并下发相应的流表，业务流传输路径及交换机端口信息如图7所示。

当四条流进行传输的时候，用户预约的新流到来，本实验新流源主机为S14，目的主机为S9，带宽需求为30M，由图1可知，在流f1和流f2占用部分带宽的情况下，路径(s_2, s_1)和路径(s_2, s_3)带宽剩余分别为20M和10M，均不能满足预约业务流的带宽需求。因此，该模块将重新计算传输路

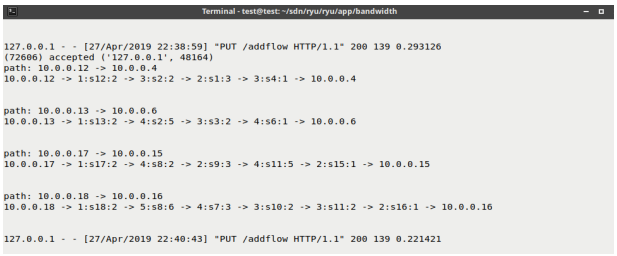


图7　流传输路径图

径，结果如图8所示，由输出可知四条已经存在的业务流传输路径改变，系统新增了预约业务流的传输路径，验证该模块可以在计算路径的时候，寻找到一个尽可能容纳所有流的解，整合网络带宽资源碎片，提高了带宽利用率。

4.1.2　一致性更新模块测试

基于QoS的路由模块测试结果表明，由于新的业务流的加入，网络中现有的四条流的传输路径发生了变化，需要对路径中的交换机进行规则更新。此时，一致性更新模块将根据业务流的带宽以及相关联的链路创建依赖图，主要依赖关系如图9所示。

由上述依赖图可知，f4-move所需带宽可以得到满足，首先进行迁移，释放链路带宽之后f3-move可以进行迁移。对于f1-move和f2-move来说，出现死锁，需要将小流提交至控制器协助转发，一致性更新模块决策结果如图10所示，与依赖图分析结果一致。

4.1.3　冲突处理模块测试

一致性更新给出的结果表明在更新流表的过程中出现了死锁，需要冲突处理模块协助，将f2上传至控制器，使用wireshark工具抓包，显示在协助转发期间控制会一直收到Packet-in数据包和Packet-out数据包，Packet-out数据包格式如图11所示，可以看到Openflow数据包类型为Packet-out，且Data字段是控制器协助转发的完整数据包，源目的IP分别为10.0.0.13和10.0.0.6，在这个过程中，两个主机的通信没有中断，验证该模块可以利用控制器协助转发死锁环中的流量。

4.2　系统性能测试

业务流服务质量主要测试数据流在流量调度系统中传输时的丢包率，验证网络中现有数据流的传输质量不会因为流量调度、传输路径的改变而下降。实验仍然采用上述拓扑和数据流，实验中使用到的八台主机的命令如表3所示。

首先初始化网络状态，为四条流下发流表，八台主机同时开始传输数据包，稳定传输后，发起新流的预约。传输结束后，四台server主机的丢包率分别如图12~14和图15所示。验证该系统可以在进行业务流调度的同时保证其丢包率维持在一个较低的水平，分别为0.36%，1.7%，0.01%和0.00%，保证了业务流的服务质量。

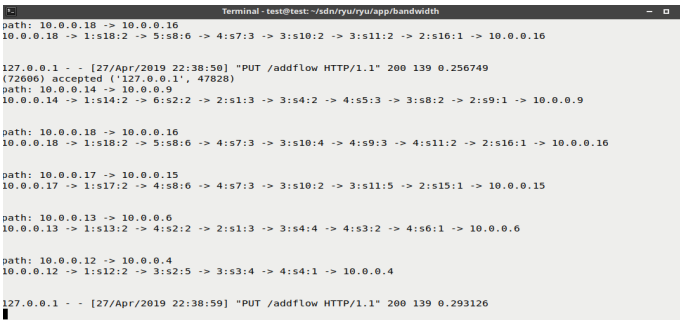


图8　加入新流后流传输路径图

5　结束语

本论文提出了SDN环境下基于QoS的流量调度方法，能够整合资源碎片，提高带宽利用率。同时，本文提出一致性更新算法和冲突处理算法解决流表动态迁移的问题。实验表明，本系统可以在有效地提高网络带宽利用率，且保证流表迁移过程中的低丢包率。

下一步工作考虑深入研究QoS路由算法和流表更新技术，提升整个方案的性能，更好地实现网络资源碎片的整合和利用。

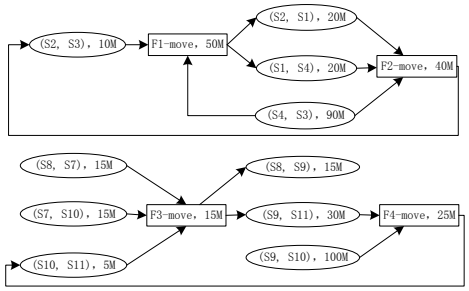


图9　流路径依赖图

```
remove: ('f4_old', 'f4_new')
remove: ('f3_old', 'f3_new')
controller: ('f2_old', 'f2_new')
remove: ('f1_old', 'f1_new')
```

图10　流更新顺序结果图

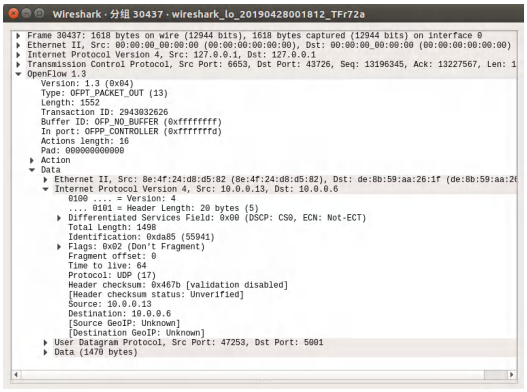


图11　Packet-out数据包

表 3　丢包率测试主机命令

主机	命令	说明
h12	iperf -c 10.0.0.4 -i 1 -t 50 -b 50M	流f1 client端
h4	iperf -s -u -i 1	流f1 server端
h13	iperf -c 10.0.0.6 -i 1 -t 50 -b 45M	流f2 client端
h6	iperf -s -u -i 1	流f2 server端
h17	iperf -c 10.0.0.15 -i 1 -t 50 -b 15M	流f3 client端
h15	iperf -s -u -i 1	流f3 server端
h18	iperf -c 10.0.0.16 -i 1 -t 50 -b 25M	流f4 client端
h16	iperf -s -u -i 1	流f4 server端

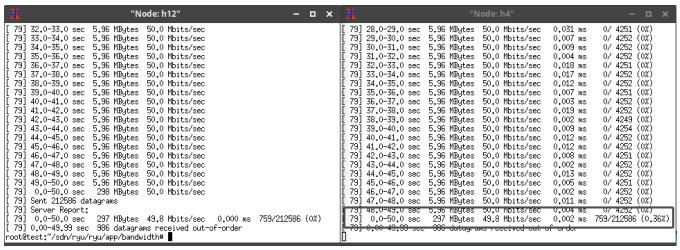


图12　流f1丢包率测试结果

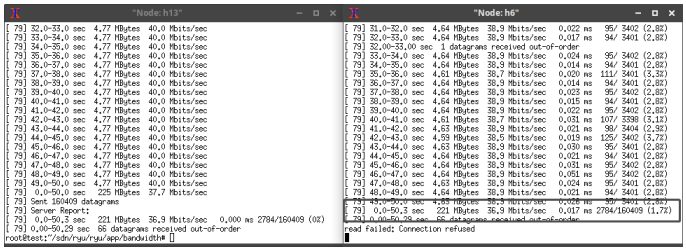


图13　流f2丢包率测试结果

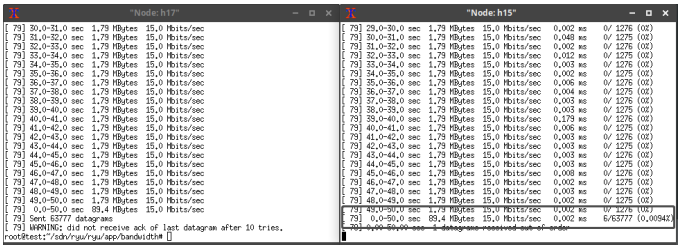


图14　流f3丢包率测试结果

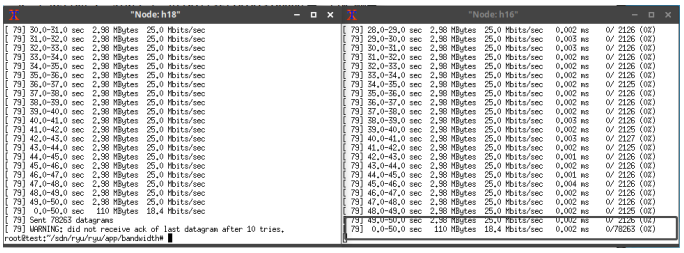


图15　流f4丢包率测试结果

参考文献:

[1] Fortz B , Thorup M . Internet Traffic Engineering by Optimizing OSPF Weights[J]. Proceedings of IEEE Infocom Mar，2000，2(3)：519-528 vol. 2.

[2] 罗雨佳, 欧亮, 莫志威等. 基于BGP增强的流量调度技术[J]. 电信科学，2016，32(3).

[3] 孔祥彬. 软件定义联网的路由选择技术研究[D]. 南京邮电大学，2017.

[4] Xiao J, Chen S, Sui M. The Strategy of Path Determination and Traffic Scheduling in Private Campus Networks Based on SDN[J]. Peer-to-Peer Networking and Applications，2019，12(2)：430-439.

[5] Zhang X, Hou W, Guo L, et al. Joint Optimization of Latency Monitoring and Traffic Scheduling in Software Defined Heterogeneous Networks [C]//International Conference on Heterogeneous Networking for Quality，Reliability，Security and Robustness. Springer，Cham，2017：104-113.

[6] Wang W, He W, Su J, et al. Cupid: Congestion-free Consistent Data Plane Update in Software Defined Networks[C]//IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications. IEEE，2016：1-9.
Van Sprundel I. Fuzzing: Breaking software in an automated fashion [C]，2005.

[7] Mahajan R, Wattenhofer R. On Consistent Updates in Software Defined Networks[C]//Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks. ACM，2013：20.

[8] Wu K R, Liang J M, Lee S C, et al. Efficient and Consistent Flow Update for Software Defined Networks[J]. IEEE Journal on Selected Areas in Communications，2018，36(3)：411-421.

[9] 李绪凯. MPLS TE技术应用示例[J]. 计算机与网络，2018，44

（8）：62-64.

[10] Balan D G, Potorac D A. Linux HTB Queuing Discipline Implementations［C］//2009 First International Conference on Networked Digital Technologies. IEEE，2009：122-126.

[作者简介]

陈立水（1981—），男，博士，高工，主要研究通信网络系统。

左宇飞（1993—），女，硕士，主要研究软件定义网络。

唐亚哲（1970—），男，博士，副教授，主要研究计算机网络。

支持QoS的虚拟网络映射算法及在OpenVirteX上的应用

齐琪[1, 2]，陆晓兵[1]，唐亚哲[1]

[1]西安交通大学 计算机科学与技术学院，西安 中国 710049；

[2]通信网信息传输与分发技术重点实验室，石家庄 中国河北 050081

摘　要：随着新型网络业务的快速发展，网络虚拟化技术得到广泛应用。虚拟网络应用方面相关研究成果很多，但在QoS保障方面存在不足：1）支持QoS的虚拟网络映射算法性能有待提升；2）多数算法仅模拟仿真，并未在实际网络虚拟化平台部署。同时，已有开源网络虚拟化平台不支持差异化的QoS需求。针对以上问题，本文首先提出了一种支持QoS保障的虚拟网络映射算法，并通过拓展开源的SDN网络虚拟化平台OpenVirteX的功能、部署虚拟网络映射算法以支持差异化的QoS需求。测试表明，本文的算法在2个性能指标上均有最优效果；通过模拟用户请求虚拟网络，测试虚拟网络的创建和流的带宽限制效果，证明了拓展后的平台具备QoS保障功能。

关键词：网络虚拟化、虚拟网络映射算法、SDN、OpenVirteX、QoS

A QoS-aware Virtual Network Embedding Algorithm and its application on OpenVirteX platform

Qi Qi[1,2], Lu Xiaobing[1], Tang Yazhe[1]

1.School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049;

2.Science and Technology on Communication Networks Laboratory, Shijiazhuang 050081

Abstract: With the rapid development of new network services, network virtualization technology has been widely used. Although there are lots of researches in this area, weaknesses are existing in terms of QoS supporting: 1) The performance of virtual network embedding algorithm needs to be improved; 2) Almost all the work are based on simulations, there is no deployment on the actual platform. Meanwhile, the existing open SDN virtualization platforms do not support differentiated QoS requirements. In this paper, A QoS-guaranteed virtual network embedding algorithm is proposed. This algorithm is further added to the well-known virtual network platform OpenVirteX, to actually provide differentiated QoS support of network flows. The results show that the proposed algorithm achieves the best results about two evaluation metrics. By simulating the process of creating and using virtual network, we test the capacity of traffic bandwidth limitation of virtual network and verify the effect of QoS guarantee.

Key words: Network virtualization; virtual network embedding algorithm; SDN; OpenVirteX; QoS

1　引言

近年来，网络虚拟化技术［1］已经成为解决传统网络僵化问题的有效方法。但随着网络服务种类增多，不同业务的网络资源需求也不尽相同，而用户对网络的服务质量需求日益提高。如何保证多种业务的差异化的服务质量（QoS），是网络虚拟化中的一个关键问题。

目前在虚拟网络映射算法和虚拟网络支撑平台方面，都有大量的研究和进展［2］。具体地，支持QoS的虚拟网络映射算法方面，文献［3］首次引入了粒子群算法来解决虚拟网络映射问题。由于该方案中可行解的寻找是随机的，丧失了算法寻优过程中的方向性，虚拟网络映射质量不高。文献［4］基于临近原则，当某个物理节点被映射后其相连的物理节点权重就会提升，从而提高其

基金项目：通信网信息传输与分发技术重点实验室基金课题资助项目（SXX18641X024）

相邻节点被后续的虚拟节点映射的概率。文献[5]在节点映射阶段考虑带拓扑聚合信息的节点，目的也是尽可能将虚拟网络映射到物理拓扑聚合性高的区域，以减少底层物理资源的开销。文献[6]提出一种支持在线处理、路径分割、路径迁移的虚拟网络映射算法。但该算法会将相邻的虚拟节点映射到物理间隔较远的物理节点上，导致过多的物理链路消耗，对QoS支持不够。文献[7]提出一种链路优先的随机映射算法。先计算出待映射的虚拟网络拓扑的最大独立链路集，然后将该链路集中每一条虚拟链路都随机映射到满足资源限制的单条物理链路上，最后再将还未被映射的虚拟节点和虚拟链路基于一定的规则映射到物理网络中。但是该算法只考虑到带宽需求，且可能会导致邻接的虚拟链路映射到相距较远的物理链路上，故还有很大的优化空间。

在网络虚拟化平台方面，也涌现了一批开源网络虚拟化平台。FlowVisor[8]是由斯坦福大学开发的首款基于OpenFlow协议的网络虚拟化研究平台，采用切片技术将一个底层OpenFlow网络分割成多个切片。每个切片都由底层物理网络的部分交换机和链路构成，拥有自己的网络控制器。但多个切片共享一个地址空间且无法创建任意的虚拟拓扑。HyperFlex[9]也是SDN网络虚拟化管理平台，它将平台的功能拆解为各个必要的功能模块，功能模块可以在软件或SDN网络元素上灵活地执行。但其只专注于数据平面虚拟化，忽略了SDN网络控制平面的虚拟化。OpenVirteX[10]是斯坦福大学基于FlowVisor进一步开发的功能更为强大的网络虚拟化平台，通过虚拟网络元素和物理网络元素之间的解耦合实现了地址虚拟化、拓扑虚拟化和控制功能虚拟化的特性，使得每一个虚拟网络都能具备自己的地址空间，还可以指定任意拓扑。

通过对比分析现有的SDN网络虚拟化平台和虚拟化映射算法可以发现，网络虚拟化应用，在支持QoS方面存在两个不足：一是现有的虚拟网络映射算法在支持QoS方面稍显不足，算法的性能还有待提升；二是多数算法仅模拟仿真，并未在实际网络虚拟化平台部署。同时，已有开源网络虚拟化平台不能支持差异化的QoS需求。比如，网络虚拟化平台不能识别用户的QoS需求，缺少限制网络资源（带宽）配置方式和消息机制，因此无法直接支持QoS映射算法在平台上应用。虚拟化效果最好的OpenVirteX，在QoS保障上也有欠缺。无法对用户申请到的虚拟网络做进一步的带宽限制，不能提供差异化服务质量。

针对以上问题，本文研究支持QoS的高效虚拟映射算法，对OpenVirteX进行多方面的拓展以搭建具有QoS保障能力的SDN网络虚拟化实验平台。主要工作有：

1） 提出了一个支持QoS的虚拟网络映射算法，并根据此算法完整实现了OpenVirteX的虚拟映射器。

2） 设计改进的OpenVirteX的网络虚拟化平台框架。

3） 基于OpenVirteX设计了QoS保障机制，对该平台进行了三个方面的拓展，使其能接收用户的QoS需求，为用户在交换机中的流设置不同带宽限制。

4） 平台测试验证。

2 方法和设计

2.1 QoS虚拟网络映射算法

本文关注支持QoS的虚拟网络映射，从算法的角度分析是以减少虚拟链路的平均物理链路条数为目标。首先，单条物理链路出故障的概率比多条物理链路中任一条物理链路出故障的概率小，所以虚拟映射的可靠性受单条物理链路故障的影响低于受多条物理链路故障的影响。在同样条件下，只映射到单条物理链路上的可靠性最好，或者组成虚拟链路映射后的物理链路条数越少可靠性越好。其次，虚拟网络消耗的链路带宽资源与映射后物理路径所含的物理链路条数有关，物理链路越少占用的带宽资源也越少。再者，如果物理链路时延相同，虚拟映射后的物理路径所包含的物理链路越少，链路时延也越小。综上所述，虚拟网络映射方案中虚拟链路的平均物理链路条数越少，越有利于提供QoS保障。

从优化的角度，本文的映射方案优先映射带宽需求大的虚拟链路，并将其映射到单条物理链路上。同时，避免邻接节点和邻接链路映射的不一致性问题，尽可能地保证被映射到的物理链路聚合在一起，保证虚拟链路映射到的物理路径所

含的链路条数尽可能少。

1） 算法目标

本文提出的基于最大加权匹配的链路优先映射算法（Maximum-Weighted-Match-Link-First Algorithm）VNE-MWMLF，优化目标是最小化虚拟链路的平均物理链路条数，如公式（1）所示。

$$AL(G^V)=\frac{\sum_{l^V\in L^V}hop(M(l^V))}{|L^V|} \tag{1}$$

式中：$hop(M(l^V))$表示虚拟链路l^V所映射到的物理路径的长度，即该物理路径所包含的物理链路条数；$|L^V|$表示虚拟链路的总数。

2） 算法流程

算法优先映射虚拟网络拓扑的最大加权匹配中的所有虚拟链路，这些链路的映射确定后能有效减少问题的搜索空间，然后按照消耗链路资源最小的原则确定剩余未被映射的虚拟节点的映射方案，最后采用搜索K条最短路径的KSP算法对剩余的虚拟链路进行映射。

首先，给定虚拟网络拓扑图中的最大加权匹配。匹配是图论中的概念，是一个图的子集，这个子集中任意两条边都没有公共的顶点，故匹配又称为图的独立边集。若M_L是图G的一个匹配，而且G中不存在匹配M'，使得$|M'|>|M_L|$，则称M_L是G的一个最大匹配。一个图的最大匹配有很多种，其中各个匹配边的权重之和最大的匹配称之为最大加权匹配。一般图求解最大加权匹配需要使用带花树算法，本文使用Python自带的网络包NetworkX中的方法求解。

接着，对最大加权匹配中的所有链路进行映射。原则是只映射到单条物理链路上，以此保证重要的虚拟链路的映射质量最高。对于首条待映射的虚拟链路的侯选物理链路，选择链路带宽资源最丰富的物理链路。本文提出了链路综合带宽衡量指标*cb*，*其定义如下*：

$$cb(l_i)=(1-\gamma)nbw(l_i)+\gamma\sum_{l_j\in N(l_i)}\frac{nbw(l_j)}{\sum_{l_m\in N(l_i)}nbw(l_m)}cb(l_j) \tag{2}$$

式中：$cb(l_i)$表示链路l_i的综合带宽度量；$N(l_i)$表示l_i的邻接链路，即与l_i有共同节点的链路；权值$\gamma\in$（0，1），调节当前链路和邻接链路之间的相对权重，当γ=0时，式子即退化为链路带宽，仅用链路自身的带宽值作为链路重要性的指标。γ越大，则邻接链路的带宽值影响越大，$\frac{nbw(l_j)}{\sum_{l_m\in N(l_i)}nbw(l_m)}$表示邻接链路$l_j$在链路$l_i$的所有邻接链路中所占的比重。从公式（2）中可知，指标*cb*考虑链路的带宽能力不仅与链路本身的带宽相关，也与邻接链路的带宽有关。选择*cb*值最大的链路意味着选择了资源最丰富的物理链路，从而提高后续的映射成功概率。

然后，对于最大加权匹配中的其它虚拟链路，映射目标是尽可能与已映射虚拟链路所对应的物理链路聚合在一起，为了衡量物理链路之间的聚合程度，本文提出了链路距离的概念：

$$dis(l_{a,b},l_{c,d})=\frac{sp(a,c)+sp(a,d)+sp(b,c)+sp(b,d)}{4} \tag{3}$$

式中：$dis(l_{a,b},l_{c,d})$表示链路$l_{a,b}$与链路$l_{c,d}$的距离；$sp(node1,node2)$表示节点node1到节点node2之间的最短路径的长度。两条链路距离越短，聚合度越高。

最后，对虚拟网络中剩余未被映射的虚拟节点和虚拟链路进行映射。虚拟节点的映射方案完全确定之后就能确定虚拟链路的映射，所以先对剩余未映射的虚拟节点进行映射。此处采用文献[7]中的映射方法，细节不再赘述。

2.2 改进的OpenVirteX平台架构

OpenVirteX是基于SDN的网络虚拟化平台[11]，它位于SDN控制器和底层物理网络之间相当于代理者的中间位置，向下面对物理网络设备，收集底层物理网络拓扑信息；向上面对SDN控制器，负责回应SDN控制器发出的LLDP包以提供虚拟拓扑信息。OpenVirteX在目前的开源SDN网络虚拟化平台中效果最好，但没有提供差异的QoS服务。具体表现在：

1） 在QoS网络映射方面，OpenVirteX自带虚拟网络映射器没有实现可用的虚拟网络映射算法，只提供了一个模板。

2） 不能对用户申请的虚拟网络做进一步的带宽限制。

3） 没有提供一套机制保障QoS消息的识别、存储、下发、转发等一系列操作。

针对以上问题，本文设计了改进的OpenVir-

teX平台如图1所示，包括虚拟网络映射器、平台管理模块、OpenVSwitch队列管理模块、拓展的OpenVirteX模块。该平台能够根据用户自定义的虚拟网络配置QoS需求、自动创建虚拟网络、识别用户的虚拟网络流量并通过流表使数据包入相应的出端口队列，实现虚拟网络流量的带宽限制。

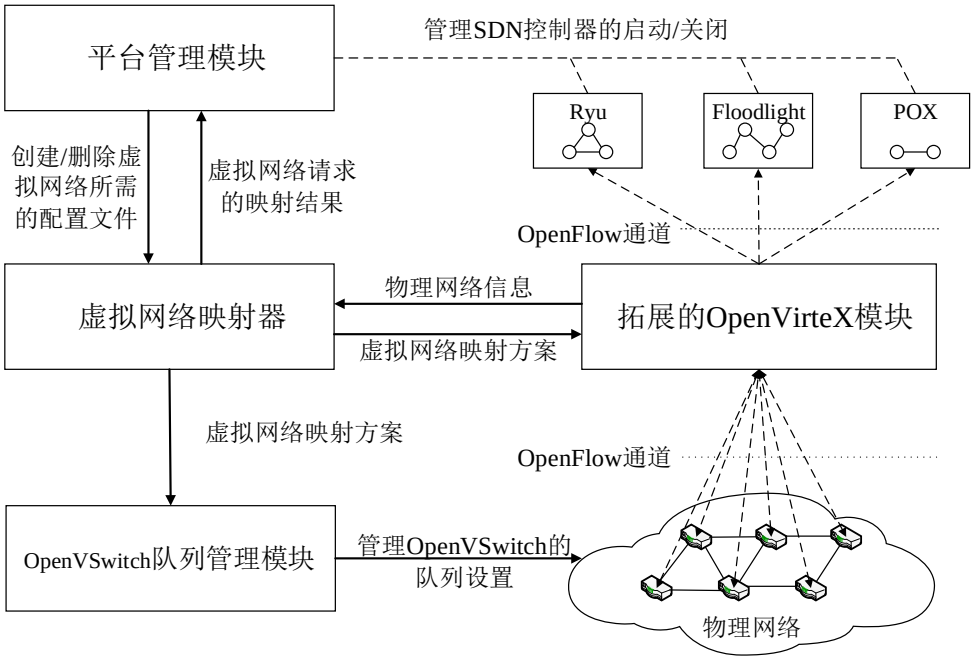


图1　改进的OpenVirteX平台架构图

1） 虚拟网络映射器

用于对虚拟网络请求进行映射的模块，实现了虚拟网络和物理网络的映射。为了便于虚拟网络映射算法的升级，本文设计的虚拟网络映射器与OpenVirteX平台解耦。OpenVirteX平台只负责创建、管理虚拟网络。虚拟网络映射器负责判断是否接受虚拟网络请求、制定具体的映射方案。

本文在OpenViretX模板的基础上，完整实现了接收用户创建虚拟网络请求，调用虚拟网络映射算法，自动根据映射方案创建对应虚拟网络的功能。虚拟网络映射器的内部结构如图2所示。运行时开启遵循JSON-RPC 2.0规范的服务端，其作用是接收外部发送过来的json配置文件，在json文件中指明虚拟网络的各项配置。OVXEmbedder-Handler接收OVXEmbedderServer转发来的请求，并做出相应的处理并回复。OVXClient通过API来调用OpenVirteX所提供的创建虚拟网络的各项功能并进行封装。在创建虚拟网络时，需要使用OVXClient所封装好的各种创建、管理虚拟网络元素的方法。

2） 平台管理模块

解决了现有OpenVirteX的两大问题：未记录剩余资源导致无法进行资源约束和不支持SDN控制器的动态管理。具有以下功能：

a） 限制、更新底层物理网络剩余资源。设置物理节点和物理链路的资源限制并保存在数据库。当用户申请或移除虚拟网络资源时，更新数据库中的物理网络资源信息。

b） 管理SDN控制器的启动和关闭。为用户分配并启动一个SDN控制器，利用开源的事件驱动网络引擎Twisted监听某个端口，根据收到建立/移除请求，启动/关闭控制器的运行。

c） 与虚拟网络映射器通信。生成配置文件发送给虚拟网络映射器。接收网络映射请求的结果。

3） 拓展的OpenVirteX模块

实现了OpenVirteX上的QoS拓展功能，它能

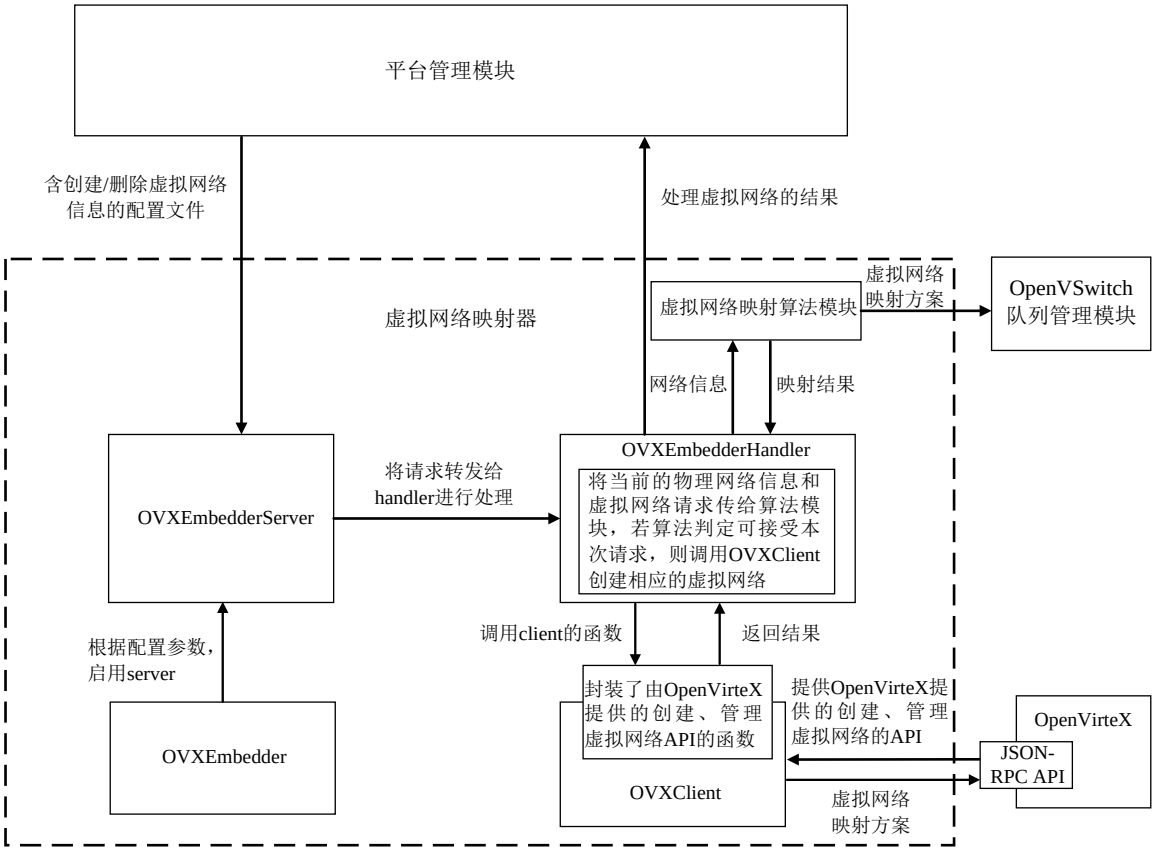


图2　虚拟网络映射器内部结构

够识别不同用户的QoS需求，记录相应的QoS参数，并根据带宽限制需求管理OpenVSwitch交换机队列，最终实现差异化的QoS需求。该模块设计了一套基于OpenVirteX平台的从QoS消息识别、存储、下发、转发的机制，从而保障了本文提出的虚拟化映射算法在开源平台的实现。此部分的关键技术在2.3中展开。

4） OpenVSwitch队列管理模块

本文的数据层面使用OpenVSwitch软件交换机，它支持两种QoS机制。一种是在入端口上配置限速策略，将高于配置值的报文直接丢弃，实现入端口的流量限制。另一种是在出端口设置一条或多条队列，对于不同队列的数据包采取不同的出队优先级，实现对数据流的差异化控制。由于入端口的流量限制机制过于粗糙，且效率较低，所以此处采用出端口队列流量整形。具体功能是：

a） 封装OpenVSwitch管理配置队列的指令，如创建队列、删除队列等功能。

b） 区分不同用户虚拟网络的QoS需求，配合2.3中的FlowMod消息的改写机制，为交换机的转发端口添加不同的优先级队列并为数据流设置带宽限制。

2.3　基于OpenVirteX的QoS保障机制设计

为了实现不同用户的QoS需求，OpenVirteX平台需要识别用户QoS需求并记录到全局映射表中，在虚拟网络运行过程中可以通过查询全局映射表获取记录，在改写FlowMod消息的时候写入不同的动作，配合OpenVSwitch队列管理模块以实现虚拟网络流量的带宽限制。因此，需要对OpenVirteX平台进行三方面的拓展：

1） 新增OpenVirteX的JSON-RPC API

解析从虚拟网络映射器中传来的用户ID与QoS参数之间的对应关系并记录在全局映射表中， OpenVirteX遵循的是JSON-RPC 2.0规范，所以OpenVirteX新增的设置用户QoS参数的API也要遵循JSON-RPC 2.0规范。OVXClient新增setTenantQoS API，负责将用户设置的QoS参数传给拓展后的OpenVirteX。具体的流程为：虚拟网络映射器在收到用户的创建虚拟网络请求之后，读取到该用户的QoS级别和需求，通过OVXClient

调用 setTenantQos API，将用户的 QoS 参数信息传给 OpenVirteX，OpenVirteX 在 OVXMap 新增相应的记录并维护。

2） 新增全局映射表的记录项

OpenVirteX 接收到用户 ID 和 QoS 参数之后，需要记录该对应关系，这样才能在后续改写 FlowMod 消息的时候根据用户 ID 查询到 QoS 参数，将入队动作 enqueue 写入 FlowMod 消息的动作列表。记录保存在全局映射表 OVXMap 中，所以需要对全局映射表进行扩展。其中新增的 tenantQosMap 和 tenantBandwidthMap 分别记录用户对应的 QoS 优先级和用户需求的带宽保证，如表 1 所示。

3） FlowMod 消息改写机制

要实现对虚拟网络流量的带宽限制，就需要将相应的数据包转发入 OpenVSwitch 对应的队列，这需要流表项的配合。在 OpenFlow 1.0 协议中，需要使用 enqueue 动作使数据包入队列。交换机上的流表项是由下发的 FlowMod 生成的，而 OpenVirteX 原先改写的 FlowMod 消息并不能生成带有 enqueue 入队动作的流表。

表 1　拓展的全局映射表

Name	Key	Value
tenantQosMap	TenantID	TenantQos
tenantBandwidthMap	TenantID	TenantBandwidth

因此本文设计了新的 FlowMod 消息的改写机制，如图 3 所示。将发送给边缘入口交换机的 FlowMod 消息中的动作改写为 enqueue，将该虚拟网络的数据包指定入 OpenVSwitch 队列管理模块为该虚拟网络创建的队列中。

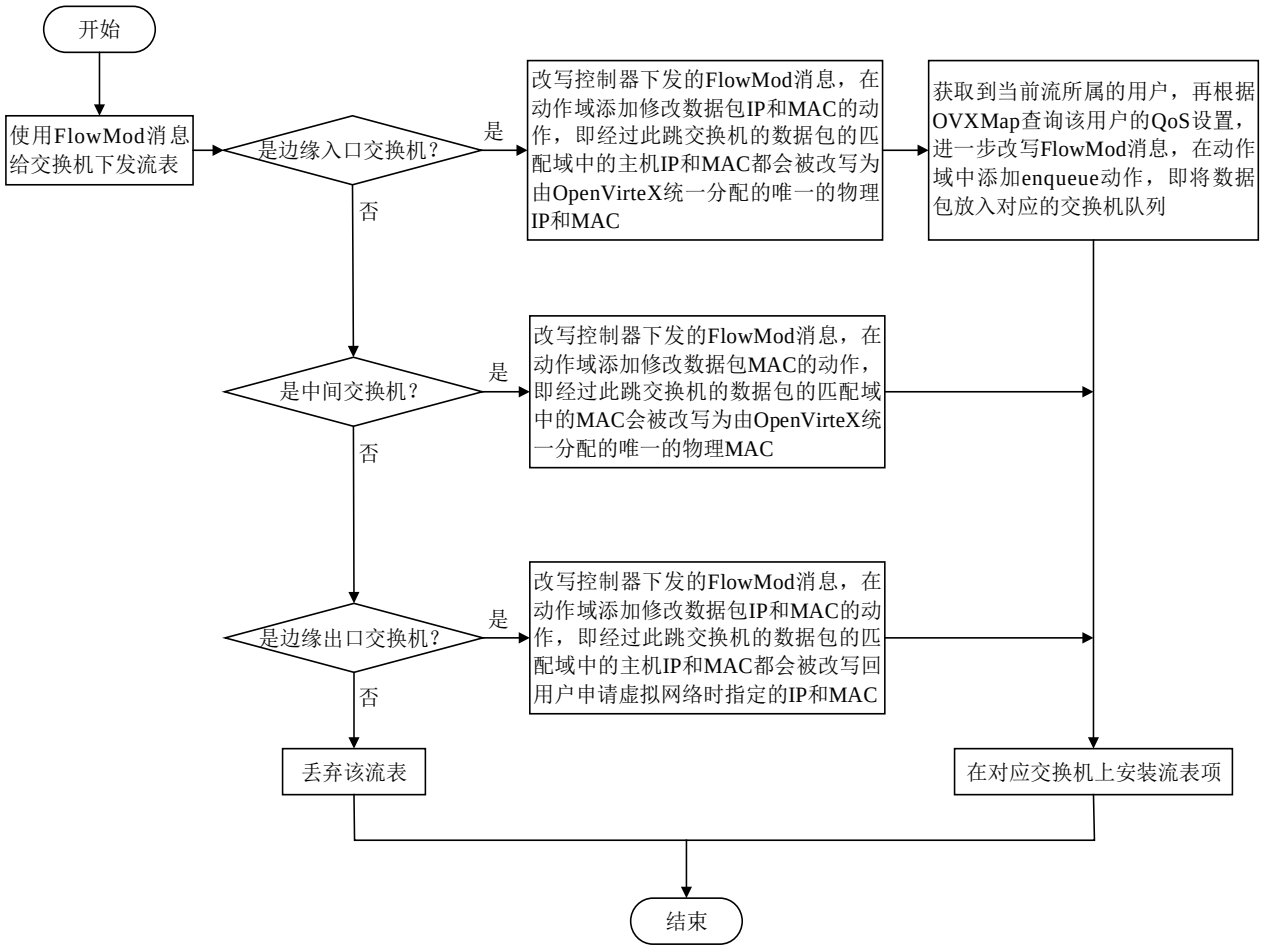


图 3　改进的 FlowMod 消息改写机制

经过以上拓展后，用户的虚拟网络中的流量添加带宽限制的流程如下：

1）虚拟网络映射器在接收到用户的 QoS 参数信息后，通过新增的 API 传给拓展的 OpenVirteX

平台，由OpenVirteX中的全局映射表OVXMap保存并维护用户ID和QoS参数之间的关系。

2） 创建好虚拟网络之后，调用OpenVSwitch队列管理模块所封装的队列管理方法，根据虚拟网络映射方案在相对应的物理交换机的端口上创建OpenVSwitch队列。

3） 当该用户的虚拟网络流量到达边缘入口交换机时，通过查询全局映射表，在改写用户控制器给相应交换机下发的FlowMod消息时，将其转发动作改写为enqueue动作并指定队列，边缘入口交换机根据改写后的FlowMod消息生成带有enqueue动作的流表项。数据包匹配该流表项后，将进入由OpenVSwitch队列管理模块所创建的OpenVSwitch队列中，按照队列规定进行调度转发。

3 实验与测试

本文搭建的测试系统选择Floodlight作为SDN控制器，搭建基于OpenVSwitch和OpenFlow1.0软件的OpenFlow交换机。主要工作有：首先，通过仿真实验验证本文提出的支持QoS的虚拟网络映射算法性能。接着，在改进的OpenVirteX平台模拟用户创建虚拟网络请求，获取映射方案，测试物理网络的连通性。最后，通过xterm连接到用户对应的主机，通过iperf发包测试其带宽，测试平台所实现的带宽限制功能。

3.1 虚拟网络映射算法验证

本文参照大多数文献的实验参数设置，利用GT-ITM分别生成物理网络拓扑和虚拟网络拓扑，详细的实验参数设置如表2所示，U表示均匀分布。每个算法的仿真实验在处理完所有的虚拟网络请求后结束。在此期间每到达100个虚拟网络请求统计一次结果，算法的每组指标共有20条数据。对比算法包括经典的基准算法G-SP，和本文映射场景相近的算法VNE-PSO、VNE-PA、VNE-MCD、VNE-ILS。五种对比算法均具有良好的代表性，算法说明见表3。虚拟网络映射问题建模和评价指标在多篇文献中都有详细介绍，具体请参照文献［12］。根据本文提出的映射算法的设计目标，选取接受率和虚拟链路的平均物理链路条数作为评价指标。

表2 仿真实验参数设置

参数	数值范围
物理网络节点个数	100
物理网络链路条数	650
物理节点CPU资源	U[50,100]
物理链路带宽资源	U[50,100]
虚拟节点个数	U[2,10]
虚拟节点CPU需求	U[1,50]
虚拟链路带宽需求	U[1,50]
虚拟网络请求到达时间	事件平均到达率为0.05的泊松过程
虚拟网络请求持续时间	均值为1000的指数分布

表3 对比算法介绍

算法名称	算法简要说明
G-SP[6]	在节点映射阶段使用贪心算法将资源需求大的虚拟节点映射到剩余资源多的物理节点；链路映射阶段使用KSP算法选择链路。
VNE-PSO[3]	使用粒子群智能算法，每个粒子的位置向量代表一种虚拟节点映射方案，粒子群算法的适应度函数值是物理路径的带宽开销，粒子群算法的优化目标是使适应度函数值最小，粒子群算法运行结束后就确定了节点映射方案；在链路映射阶段KSP算法选择链路。
VNE-PA[4]	在节点映射阶段采用就近原则，节点映射阶段优先考虑加权剩余资源度量指标最大的底层物理节点，链路映射节点采用KSP算法。
VNE-MCD[5]	在节点映射阶段选择拓扑聚合信息的资源度量指标最大的底层物理节点;链路映射阶段采用KSP算法。
VNE-ILS[7]	从虚拟网络拓扑中计算出一个最大独立链路集，随机将该链路集中的每一条虚拟链路映射到满足资源限制条件的单条物理链路上，再映射剩余未被映射的虚拟节点和虚拟链路。

1） 接受率

接受率表示成功映射到底层物理网络的虚拟网络请求数占总共到达的虚拟网络请求数的比例。实验结果如图4所示。从图中可以看到，VNE-MWMLF算法具有最高的接受率，这是因为在VNE-MWMLF算法中，链路带宽需求大的链路都以单条物理链路的形式映射下去，这样大大降低了物理链路资源开销，使得底层物理网络能够尽

可能多地剩余链路资源，后续的虚拟网络映射具有更高的成功率。VNE-PA算法的映射方案中物理节点尽可能地临近，从而使虚拟链路两端节点所映射到的物理节点不会相距太远，链路映射不会占用过多的资源，因此效果次之。随着时间的推移，虚拟网络请求到达数开始和虚拟网络请求离开数形成动态平衡的态势，虚拟网络映射的接受率慢慢趋于平稳的数值。

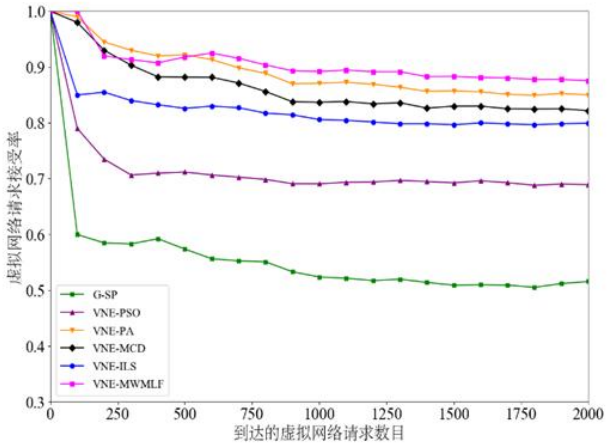


图4 各算法的虚拟网络请求接受率

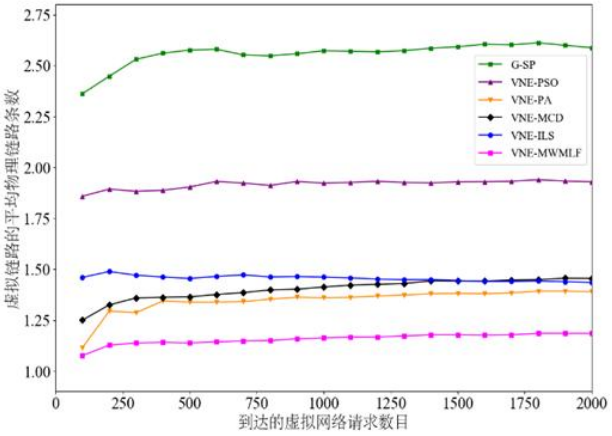


图5 各算法虚拟链路的平均物理链路条数

2） 虚拟链路的平均物理链路条数

虚拟链路的平均物理链路条数指的是成功被映射的虚拟网络中一条虚拟链路映射到的物理路径所含物理链路条数的平均值。实验结果如图5所示。VNE-MWMLF算法具有最小的虚拟链路的平均物理链路条数。当处理完全部的2000个虚拟网络请求后，可以看出VNE-MWMLF算法比效果次好的VNE-PA算法减少了14.7%的虚拟链路平均物理链路条数。本文算法的优化目标是减少虚拟链路的平均物理链路条数，通过分别对首条链路和候选链路做了优化，取得了最优的效果。也再次证明了高质量的虚拟链路映射结果有利于对虚拟网络进行QoS保障。

3.2 创建虚拟网络

用户申请的虚拟网络请求的拓扑随机生成，如图6所示。物理网络拓扑使用有14个交换机和23根链路的德国电信Deutsche Telecom （DT）[13]拓扑，物理网络的拓扑如图7（a）所示，每个物理节点上都连接有主机。底层物理环境由Mininet创建，Mininet可以模拟主机、链路、交换机等网络设备。

虚拟网络映射器接收到用户的虚拟网络请求后，通过运行日志可知虚拟节点1、虚拟节点2、虚拟节点3、虚拟节点4映射到物理节点分别是S12、S3、S4、S7，那么对应的h1，h2，h3，h4分别就是S12、S3、S4、S7所连接的主机。虚拟网络映射到底层物理网络的结果如图7（b）所示。

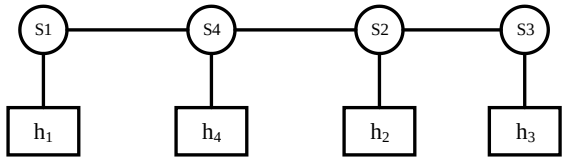
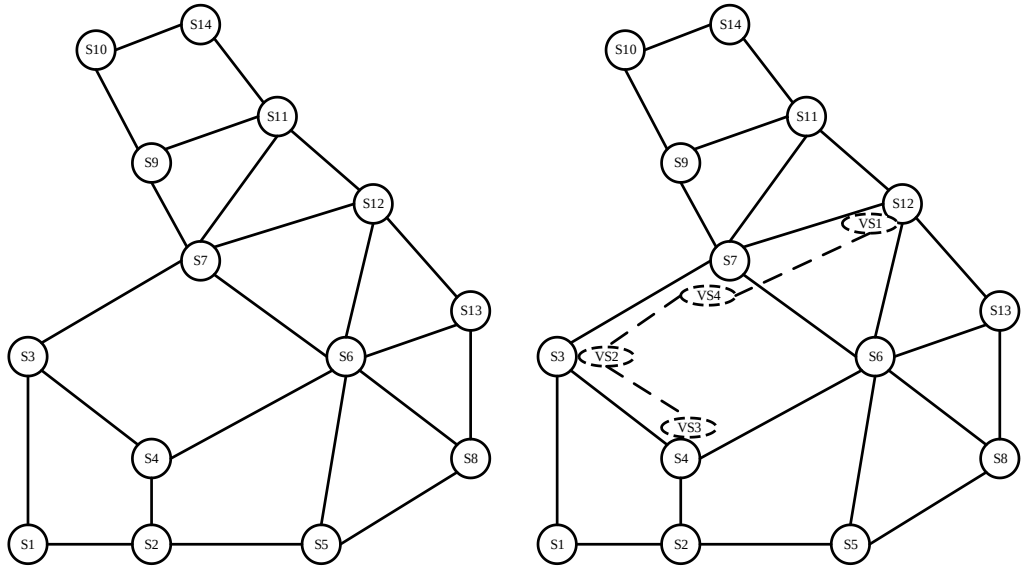


图6 虚拟网络请求的拓扑示意图

3.3 带宽限制测试

本节主要测试改进的OpenVirteX平台新增的虚拟网络带宽限制能力。测试使用UDP协议在发送端以恒定的速率发送数据，记录接收端接收的带宽以验证带宽限制能力。用户向SDN网络虚拟化平台申请的虚拟网络的带宽需求见表4。用户提交创建虚拟网络的申请后，创建出对应的虚拟网络，如图8所示。OpenVSwitch队列管理模块也接收到虚拟网络的映射方案，并在对应的OpenVSwitch交换机出端口上设置了队列，出端口上的队列将该用户虚拟网络中的流量带宽限制在10Mbps。

此时IP为10.0.0.4的主机向IP为10.0.0.12的主机发送UDP数据流，设置数据流发送带宽为



(a) 物理拓扑示意图　　(b) 虚拟网络映射关系示意图

图7　实验拓扑图

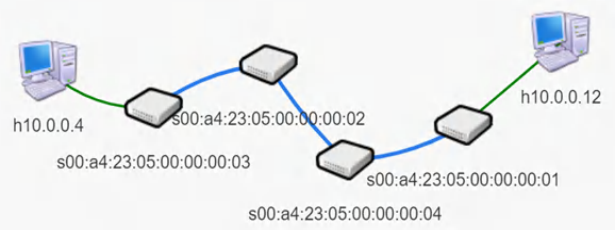


图8　用户控制器所看到的虚拟网络拓扑图

20Mbps，测试时间30秒，在两个主机上分别执行iperf命令。

发送端和接收端每秒的发送速度和接收速度

表4　用户申请的虚拟网络带宽需求

匹配域(五元组)					带宽
源IP	目的IP	源端口	目的端口	协议类型	
10.0.0.4	10.0.0.12	*	5001	UDP	10Mbps

的统计结果如图9所示。发送端以20Mbps的速度发送数据包，但是由于OpenVSwitch队列管理模块进行了队列设置，限制了队列的最大带宽，接收端的接收速度被限制在10Mbps左右，验证了队列机制成功将数据流的速率限制在规定的带宽内。

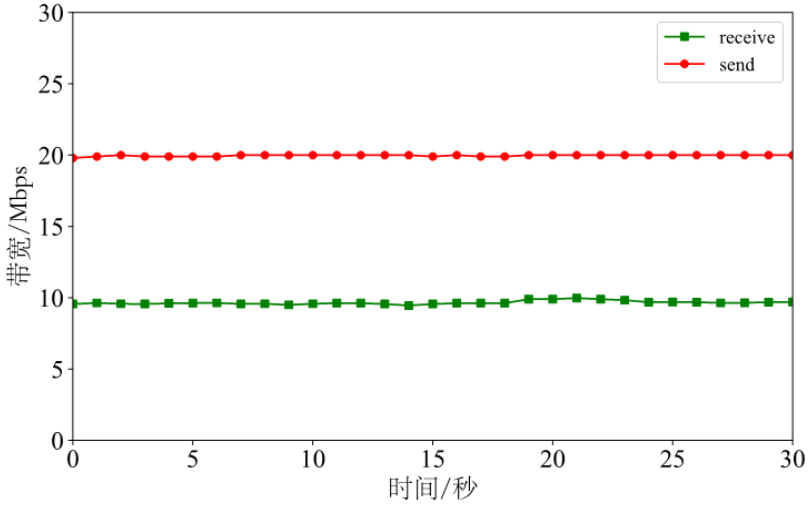


图9　限速验证图

通过以上的实验，验证了改进的OpenVirteX平台实现了带宽限制功能，并且可以根据不同的用户的需求创建不同带宽限制和优先级的队列，从而实现不同的QoS保障。

4 结束语

本文研究了SDN网络虚拟化平台和虚拟网络映射算法。对OpenVirteX进行多方面拓展以搭建具有QoS保障能力的SDN网络虚拟化实验平台。本文设计了一种基于最大加权匹配的链路优先映射算法，仿真实验表明该算法有最高的接受率和最少的虚拟链路的平均物理链路条数。接着，基于此算法完整实现了虚拟网络映射器。拓展OpenVirteX的接口使之能接收用户的QoS需求，改进OpenVirteX的消息改写机制，最终在对应的物理交换机上创建相应的出端口队列，为不同用户提供不同的流量控制以实现QoS保障能力。

参考文献:

[1] 温涛, 虞红芳, 李乐民. 网络虚拟化的过去,现在和未来[J]. 中兴通讯技术，2014，000(003)：1-7.

[2] 李小玲, 王怀民, 丁博, 等. 虚拟网络映射问题研究及其进展[J]. 软件学报，2012，23(11)：3009-3028.

[3] 程祥, 张忠宝, 苏森, 等. 基于粒子群优化的虚拟网络映射算法[J]. 电子学报，2011，39(10)：2240-2244.

[4] Liu J, Huang T, Chen J Y, et al. A new algorithm based on the proximity principle for the virtual network embedding problem[J]. Journal of Zhejiang University SCIENCE C，2011，12（11）：910-918.

[5] 唐少华. 虚拟网映射及预测机制下的资源分配研究[D]. 北京邮电大学，2014.

[6] Yu M, Yi Y, Rexford J, et al. Rethinking Virtual Network Embedding: Substrate Support for Path Splitting and Migration[J]. ACM SIGCOMM Computer Communication Review，2008，38（2）：17-29.

[7] 王颖, 熊文成, 李文璟. 基于最大独立链路集的随机虚拟网络映射算法[J]. 北京邮电大学学报，2014，37(S1)：8-11.

[8] Sherwood R , Gibb G , Yap K K , et al. Can the Production Network Be the Testbed?[C]. OSDI，2010：365-378.

[9] Blenk A, Basta A, Kellerer W. HyperFlex: An SDN virtualization architecture with flexible hypervisor function allocation[C]. IFIP/IEEE International Symposium on Integrated Network Management (IM)，IEEE，2015：397-405.

[10] Al-Shabibi A, De Leenheer M, Gerola M, et al. OpenVirteX: make your virtual SDNs programmable[C]. Proceedings of the third workshop on Hot topics in software defined networking，ACM，2014：25-30.

[11] 张歌. OpenVirtex网络虚拟化平台虚网识别机制的改进与实现[D]. 北京邮电大学，2017.

[12] Chowdhury M, Rahman M R, Boutaba R. ViNEYard: Virtual Network Embedding Algorithms With Coordinated Node and Link Mapping[J]. IEEE/ACM Transactions on Networking (TON)，2012，20(1)：206-219.

[13] Azodolmolky S, Perello J, Angelou M, et al. Experimental Demonstration of an Impairment Aware Network Planning and Operation Tool for Transparent/Translucent Optical Networks[J]. Journal of Lightwave Technology，2011，29（4）：439-448.

[作者简介]

齐琪（1985—），女，博士研究生，主要研究方向：软件定义网络，确定性网络。

陆晓兵（1994—），男，硕士研究生，主要研究方向：网络虚拟化。

唐亚哲（1970—），男，副教授，博士生导师，主要研究方向：计算机网络，软件定义网络。

基于操作码可视化的深度学习恶意软件分类

魏书宁[1, 2]，陈小寒[1]，唐勇[2]，覃正泽[1]
[1]湖南师范大学 信息科学与工程学院，长沙市 410006；
[2]国防科技大学 计算机学院，长沙市 410006

摘　要：近年，恶意软件增长速度越来越快。为了有效检测并减少恶意软件对计算机带来的威胁，本文提出一种基于操作码可视化的深度学习恶意软件分类算法。该方法将递归神经网络跟卷积神经网络相结合，首先使用递归神经网络对恶意软件操作码序列进行处理，生成的预测码序列反映了原始代码与时序特征之间相关联的能力，同时也提高了模型的泛化能力；然后，利用Simhash将原始编码与预测编码融合生成RGB图像，增强图像信息密度，减少信息碰撞；最后，针对传统分类模型无法解决的特征自动提取和传统神经网络参数过多的问题，使用轻量级卷积神经网络对RGB图像进行分类。通过实验验证，该方法分类精度达到99.28%，分类速度较传统模型也有显著提升。

关键词：恶意软件分类、操作码可视化、递归神经网络、轻量级卷积神经网络、Simhash

Deep Learning Malware Classification Based on Opcode Visualization

WEI Shuning[1,2], CHEN Xiaohan[1], TANG Yong[2], QIN Zhengze[1]
1.College of Information Science and Engineering, Hunan Normal University, Changsha 410006, China;
2.College of Computer Science, National University of Defense Technology, Changsha 410006, China

Abstract: In recent years, the growth of malware has been accelerating. In order to effectively detect and reduce the threat of malware to computers, this paper proposes a deep learning malware classification algorithm based on deep opcode visualization. This method combines the recursive neural network with the convolutional neural network. Firstly, the recursive neural network is used to process the malware opcodes sequence, and the generated predictive code sequence reflects the ability of correlation between the original code and the timing characteristics, and also improves the generalization ability of the model. Then, the original encoding and predictive encoding are fused to generate RGB images using Simhash to enhance the image information density and reduce the information collision. Finally, lightweight convolutional neural network is used to classify RGB images to solve the problems of feature automatic extraction which cannot be solved by traditional classification models and too many parameters of traditional neural network. Through experimental verification, the classification accuracy reaches 99. 28%, and the classification speed is also significantly improved compared with the traditional model.

Key words: Malware classification; Opcode visualization; Recurrent neural network; Lightweight Convolutional neural networks; Simhash

1 引言

恶意软件是威胁当今互联网安全的主要原因之一。国家互联网应急中心的互联安全威胁报告[1]（2020年3月）显示，一月内中国境内感染网络病毒的终端数近151万，较上月增长了13.7%。恶意软件数量的不断增长使其带来的威胁也在不但增加。恶意软件通过操作系统或应用软件的安全漏洞潜入受害者的电脑，从而施行恶意操作。因此，恶意软件分析与检测是现阶段安全人员面临的一大挑战。

主流恶意软件检测技术可分为静态和动态两大类。静态分析指在不执行软件的前提下进行分析。Schultz等[2]第一次在恶意软件检测中引入数据挖掘的概念，使用三种不同的静态特征进行分类。Kong等[3]提出了一种基于恶意软件结构信息（功能调用图）的自动分类框架，使用分类器从成对的恶意软件的距离中进行学习。罗世奇等[4]融合了恶意软件的纹理特征和指令频率特征，用栈式自编码器模型进行分类。静态分析方法快速安全且功能强大，但在分析前必须要对文件进行解包和解密，导致它容易受到混淆技术的影响。动态分析指恶意软件在沙箱、虚拟机等环境下执行时，系统提取动态特征进行分析检测。动态分析能更好更全面的检测已知和未知的恶意软件，但是分析过程中多数在虚拟环境中进行，导致系统资源消耗过大，耗时且计算密集。

随着机器学习算法的出现，传统的机器学习算法如支持向量机、决策树等被用来作为恶意代码检测模型。机器学习算法依赖于特征向量，可结合静态分析、动态分析、静动态分析进行分析。Nataraj等[5]提出了一种利用可视化图像处理和分类恶意软件的方法，将二进制文件可视化为灰度纹理图像进行分类。实验结果表明，同一家族图像在布局和纹理上非常相似；Han等[6]通过纹理分割的方法将其分割成块，采用灰度共生矩阵提取恶意代码的特征。刘亚姝等[7]在此基础上将恶意代码逆向分析与可视化结合起来，提取反汇编文件的“.txet”段操作码，提高有效信息密度。这些方法避免静动态分析时所需的解包或执行操作，因此，具有更快的分析速度，并且可以处理打包的恶意软件，不受操作系统的限制。

因深度学习技术的日益成熟，恶意软件原始图像也被作为输入直接传递到各种深度学习模型中。Kolosnjaji等[8]将系统调用序列作为特征，构建一个有卷积层和循环层的深度神经网络进行恶意软件分类。赵炳麟等[9]使用卷积神经网络（Convolutional neural networks，CNN）处理恶意代码应用程序接口（Application Programming Interface，API）调用图，解决了子图同构算法复杂度的问题。ZHAO等人[10]提出一种基于纹理可视化的深度学习恶意分类框架MalDeep。近年，递归神经网络（Recurrent Neural Network， RNN）作为强大的神经网络，也被使用到恶意软件检测和分类中。2015年，Pascarnu等人[11]首次将RNN运用到恶意软件分析中。Eui等人[12]将RNN用于识别反汇编文件。Tobiyama等[13]提出了基于过程行为的深度神经网络恶意软件检测。

为了能更快更准确的对恶意软件进行分类，本文提出了一种基于操作码可视化的深度学习的恶意软件分类方法RSBM（R-RNN、 S-Simhash、R-RGB images、M-MobileNet v2）。该方法使用RNN生成预测代码，未使用隐藏层中的信息提高RNN使用可解释性；然后使用Simhash可视化方法将恶意软件映射到尺寸相同的RGB图像中，同时降低有效信息碰撞程度；最后将轻量级CNN（MobileNet v2）作为分类器，在保证精确度的前提下，减少模型资源消耗、降低计算量、提高自动化程度。

2 基于操作码可视化的深度学习恶意软件分类方法

2.1 概述

与传统的静态和动态分析不同，可视化分类方法不需要复杂的反汇编和耗时的执行过程，难度也不受软件数量的影响，很大程度上提高了恶意软件的处理速度。因此本文提出一种基于操作码可视化的深度学习恶意软件分类方法。如图1所示，RSBM由3部分组成：数据预处理，恶意软件特征提取、描述、可视化和CNN分类。其中，数据预处理阶段对一个可执行文件使用反汇编工具IDA Pro进行处理，生成反汇编文件。恶意软件特征提取阶段对反汇编文件以函数块为单位进行操作码序列的提取，将操作码序列处理后送入RNN中生成预测码序列，接着使用Simhash将原始的操

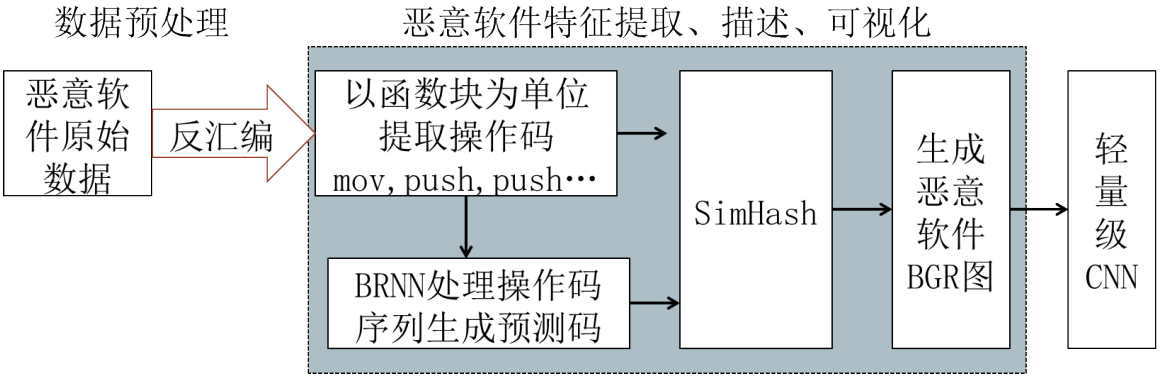


图1　基于操作码可视化的深度学习恶意软件分类方法流程图

作码序列跟预测码序列转换成相应的哈希值，随后转化成为可以用于训练的CNN的RGB图像。最后，使用轻量级CNN（MobileNet v2）模型对这些图像进行迭代训练，以识别其家族。

代码语言是上下文敏感的，而经过反汇编后生成的代码块内部更是进行了序列化。RNN模型在处理时序数列上很有优势，因此将其引入到恶意软件分类中来，方便更好的提供更具代表性的时序序列。原始代码反应了恶意代码的独特特征，而预测序列则反映了恶意软件中的共同特征。与单一信息特征相比，两者信息融合后，能更加精准和全面的判断。

2.2　提取操作码

在Windows操作系统中，可移植可执行文件（Portable Executable，PE）是封装必要信息的数据结构。PE文件由许多头和节组成，这些头和节中保存着如何将文件映射到内存中的信息。一个可执行文件由几个不同的区域组成，每个区域需要不同的内存保护，因此每个部分的开头必须与页面边界对齐。一个简单的PE文件如图2所示。因为RSBM算法选择代码段的操作码作为特征，所以先将恶意软件可执行文件反汇编，获得反汇编文件。反汇编代码中，“proc near”表示一段函数的开始，“endp”表示该段函数的结束，依据这两个标记可以将汇编代码拆分成多个函数块，方便我们进行特征提取。

本文实验数据来自2015年Kaggle微软恶意软件分类挑战[14]。本实验从中收集到846种不同类

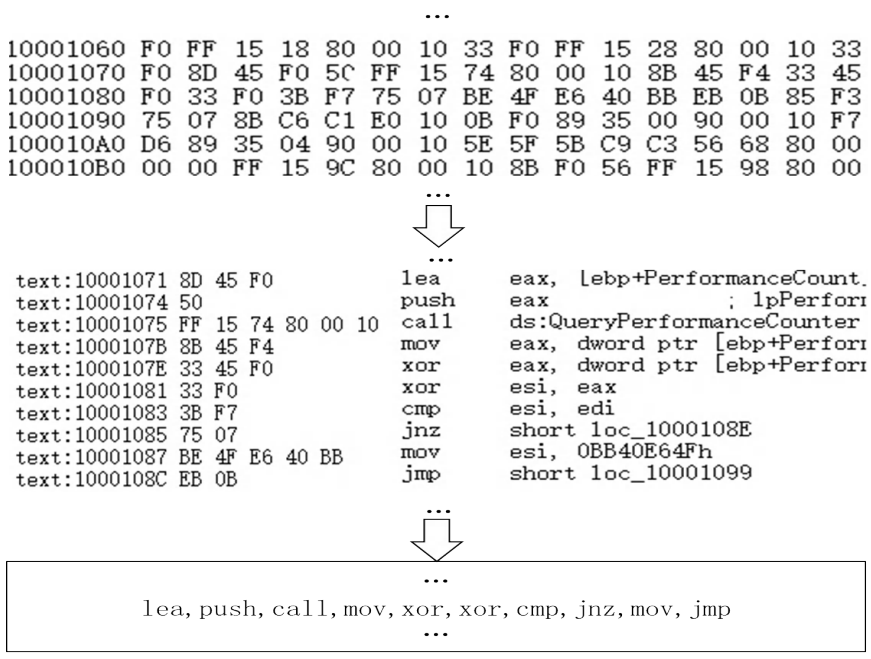


图2　PE文件操作码提取过程

型的操作码，常用的255个类型出现比例达到99.97%，余下的种类型仅占0.03%。因此将余下类型全部划分到第256种类型中。操作码序列经过处理后共有256种。因操作码序列不能直接作为神经网络的输入，所以本方法对操作码进行One-Hot编码，编码长度为256。One-Hot编码又称为一位有效编码，每一个编码中仅包含一个1，向量中的其它位置全部为零，互不干涉，从而提高了在BRNN中操作码之间的差异性。操作码的类型确定了1在One-Hot中的位置，如图3所示。

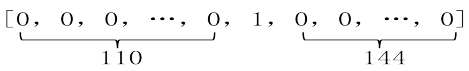


图3　操作码111的one-hot编码

2.3　RNN特征训练

递归神经网络（RNN）对处理时序数列很有优势，RNN不仅将当前隐藏层的输出传递到下一层，还将该输出作为了下一时段当前隐藏层的输入。但是，有的序列信息不仅仅依赖先前信息，而且依赖于未来甚至整体序列。因此，研究员提出双向递归神经网络（Bidirectional RNN，BRNN）解决这一局限性（图4）。本文考虑到传统RNN训练中存在着梯度消失和爆炸的问题，所以选择门控循环单元网络（Gated recurrent unit，GRU），它包含着更少的训练参数，所需要的时间与空间资源也更少。本文构造一个具有一个输入层，两个隐藏层（每个具有288个GRU）和一个softmax输出层的双向RNN。

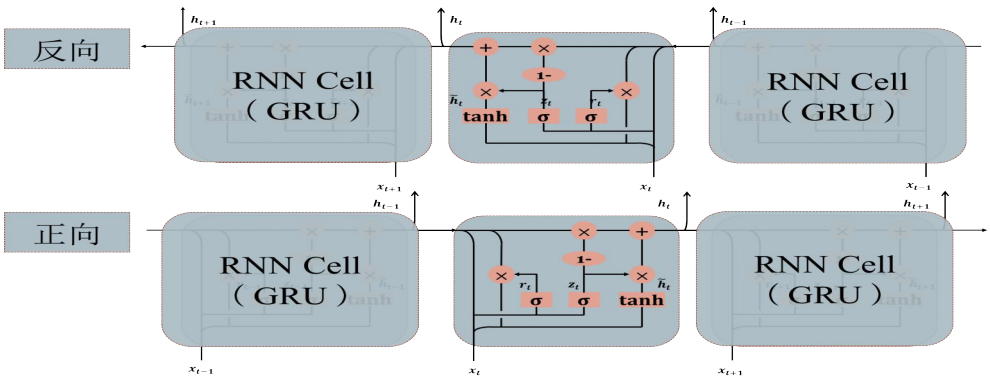


图4　BRNN结构图

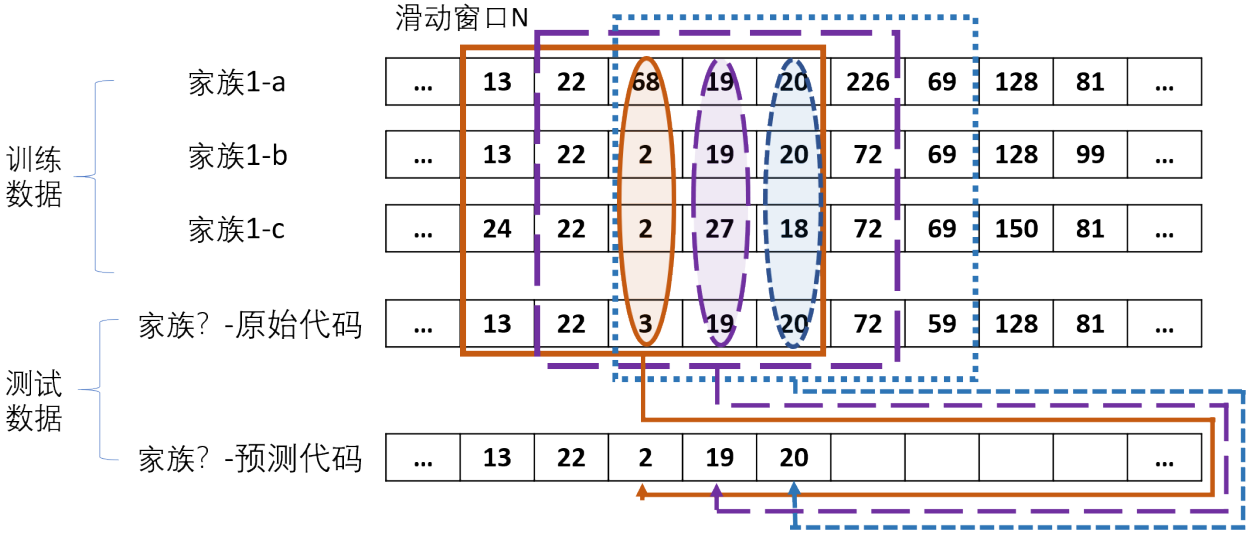


图5　带滑动窗口的BRNN训练，并生成预测代码

与传统RNN不同，BRNN不仅取决于先输入的信息，还能在特定的时间段内使用先前输入与未来信息进行训练。因此，在训练BRNN时，我们需要设置一个滑动窗口N。参数N决定神经网络

的学习速率跟效果，如N太小，分类特征不足，难以正确分类；如N过大，长期依赖信息处理不全，降低处理速度与准确率。使用矩形窗中的操作码序列来预测滑动窗口中第M个圆形中的操作码时，要考虑前（M-1）个操作码和后（N-M）个操作码。参数M取决于当前代码受前后代码影响的程度。如果预测信息依赖较少的先前信息，则可设置较小的M值；相反，则设置较大的M值。处理好当前滑动窗口后，下滑一个操作码的距离，并继续预测当前窗口中的第M个操作码。

同一家族的恶意代码都存在着相似的且与其它家族不同的特征，二进制可执行文件在进行反汇编后生成了经过时序特征序列化的函数块。如图5所示，恶意代码家族中的相似特征是通过使用RNN来学习的。RNN通过在滑动窗口中使用操作码来预测圆中的操作码（圆中没有操作码）。神经网络具有强大的泛化能力。即使窗口内操作码稍有不同，RNN也会很快的学习到类似的序列特征。在反向传播阶段，参数沿损失函数梯度的相反方向更新。为了使损失函数尽快下降，滑动窗口的测试结果为相似窗口的最常见的操作码。因此，即使原始代码序列的某些部分不同，RNN也能生成相似的预测序列，使得同一家族的恶意代码更加相似。

2.4 基于操作码的SimHash可视化

样本可视化阶段中，因为恶意代码长度不同，若直接生成特征图像也会导致尺寸大小不一，不能直接使用CNN进行分类，若转化为同一尺寸也会在图像伸缩过程中产生信息丢失的情况。所以，本文使用局部敏感哈希（locality sensitive Hashing，LSH）提取操作码中对位置敏感的信息，解决特征图像大小不相同的问题。SimHash是一种局部敏感哈希算法，主要用于计算两集合之间的相似性。首次在网络去重中提出，后被运用到克隆代码的检测中[15]，与倒排索引表相结合后，在恶意代码复用性溯源问题中表现出了优越性[16]。

本文对提取的每个函数块操作码序列和经过RNN生成的预测码进行SimHash，经过分词、Hash、加权、合并以及降维五个过程。本实验中，操作码序列被看成是一段语句，进行分词，得到有效的具有代表性的特征向量（单个操作码）。每个特征向量都是同样重要的，因此设置都相同权重，权值为1；随后通过Hash算法获得每个操作码序列的160位Hash值；然后根据Hash值是为“1”还是“0”，来判断该位是加上还是减去权值，得到加权数列；接着将函数块的所有操作码加权序列叠加起来得到新的序列；之后判断新的序列上的值是否大于“1”，大于则为“1”，否则为“0”，从而或得到了该操作码序列的SimHash值；最后将SimHash值映射到RGB图上。详细步骤如下所示，图6表示将SimHash值映射到RGB图像上的过程。

SimHash算法实现过程如下：

输入：恶意代码样本

输出：256*256的RGB图

1. 读取恶意代码样本，获取函数块及其操作码序列

2. 对恶意软件样本的函数块操作码序列进行分词，并赋予权值为1

3. 对每个操作码进行Hash计算，得到160为的Hash值

4. 判断P是否为1，是则加上权值，否则减掉权值，得到新的序列

5. 操作码加权序列叠加

6. 判断T是否大于1，是则设置为1，反之则为0，获得SimHash值S

7. 将SimHash值划分为5个8位的十六进制数

8. 判断十六进制数是否大于7，是则为1，反之为0，得到五个8位的二进制数，产生大小为0-255的数值

9. 将前两个大小在0-255间的数值作为横纵坐标标记在256*256的图上，后三个数值表示该点的RGB值，如在同一点，则RGB值叠加

同时，本文对预测码也进行SimHash，得到预测码SimHash值，然后使用同样的方法得到预测码的特征图像，将原始代码特征图像与预测码特征图像融合在一起（如图7）。RNN的预测码会添加一些有助于我们分类的信息，过滤一些原始干扰。但在以往的RNN的研究中会存在丢失一些有效信息的问题。因此，本文将原始特征与预测特征一起映射到同一图像中，这样即使预测特征在某一位置遗失像素，原始数据在该位置依然会有特征信息，从而提高恶意软件分类的准确性。

2.5 轻量级CNN分类

随着恶意软件可视化的发展，研究员发现来

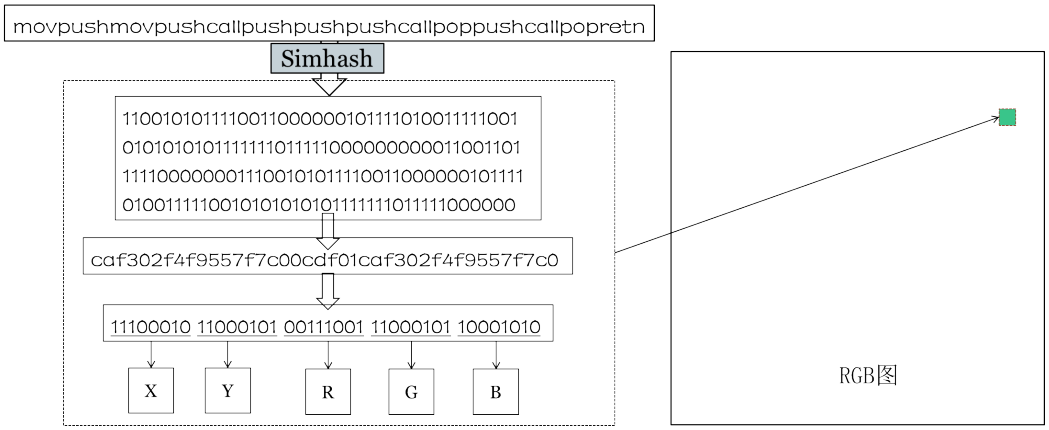


图6　SimHash值映射得到RGB特征图像过程

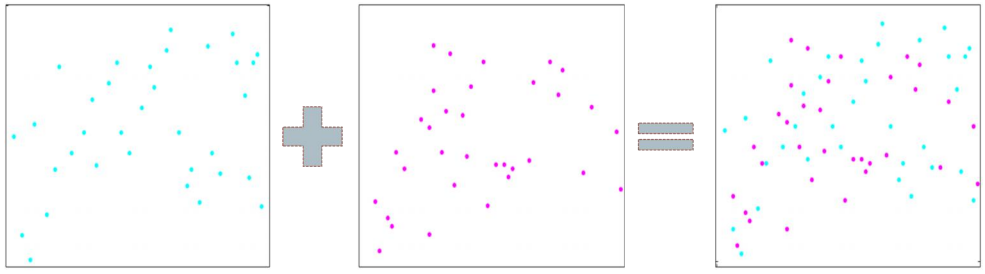


图7　原始代码特征与预测代码特征融合的特征图像

(a)原始特征图像 (b)预测特征图像 (c)融合特征图像

自相同恶意家族的特征图像非常相似，并且不同的恶意家族也仅包含少数共同特征。因此，擅长处理图像信息的卷积神经网络被引入到恶意软件家族分类中。但随着CNN网络层数的增加，参数数量也在不断增加，进而导致了计算量的增加。因此，本文采用深度可分离卷积结构MobileNet v2模型进行分类。较传统CNN而言，MobileNet v2模型同样能够自动提取特征进行学习，并且在相同精度条件下，轻量级CNN比传统CNN拥有更少的参数和计算量，从而提高分类学习速度。如图8所示，为了避免过拟合问题，本文在原来的MobileNet v2模型上去除了最后两个Conv2d和Bottleneck操作，并将原来的平均池化层操作的大小改为3*3，最后使用Softmax分类器将恶意软件划分到属于自己的恶意软件家族中去。

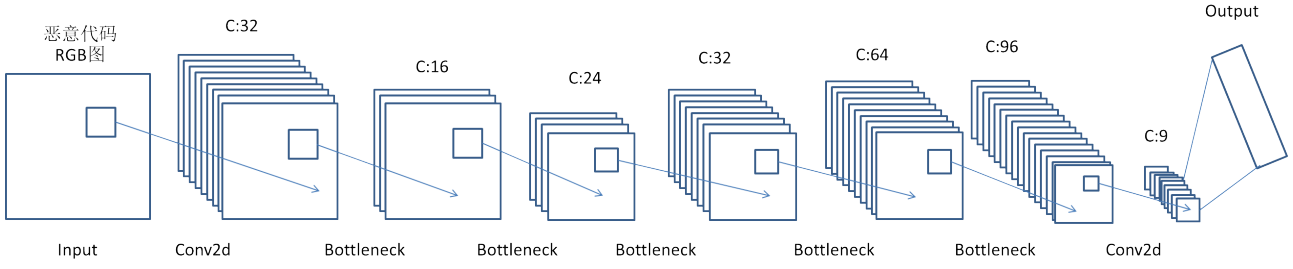


图8　轻量级CNN结构图

3　实验结果与分析

3.1　实验数据及分析方法

本文实验数据来源于2015年的Kaggle Microsoft恶意软件分类挑战，实验数据集中的恶意软件样本分布如表1所示。此数据集中包含着9个家族共10868个标记恶意软件样本的二进制文件和反汇编文件，因此我们实验时不需要再额外对恶意软

件进行反汇编。其中70%的数据用于训练，其余用于测试。实验程序用Python和MATLAB编写，硬件环境为Intel Core i7-9700处理器，主存32GB。

表1　用于实验的恶意软件数据集

家族	数量	家族	数量
Ramnit	1541	Tracur	751
Lollipop	2478	Kelihos_ver1	398
Kelihos_ver3	2942	Obfuscator.ACY	1228
Vundo	475	Gatak	1013
Simda	42		

为了便于分析恶意软件分类效果，本文使用统一指标来评论分类结果：准确率、误报率。具体来说，正确预测样本属于恶意软件家族A为真阳率（TP），不属于家族A的样本被预测到家族中为假阳率（FP），属于家族A的样本没有被预测到A中为假阴率（FN），不属于家族A的样本没有被预测到家族A中为真阴率（TN）。准确率、误报率定义如下：

$$准确率 = \frac{TP + TN}{TP + +TN + FP + FN} \tag{1}$$

$$误报率 = \frac{FP}{FP + TN} \tag{2}$$

3.2　RNN训练结果

在RNN网络中，使用RNN滑动窗口时，我们需要考虑滑动窗口大小（N的值）和预测位数（M的值）对结果的影响。如图9所示，当滑动窗口的大小固定（N固定），M接近N / 2时，RNN能够更好的学习恶意家庭的特征。因此，本文将要预测的操作码设置在滑动窗口的中心位置左右。同时，为了验证加入预测代码能够提高图像分类进度，我们将原始RGB图像与加入预测码后的RGB图像放入同样的轻量级CNN中进行分类，同时对恶意软件进行LBP特征提取，并用K-最近邻算法（K-Nearest Neighbor，KNN，K=2）进行分类。所得结果如图10所示，实验结果证明加入预测代码能够提高模型的分类效果。

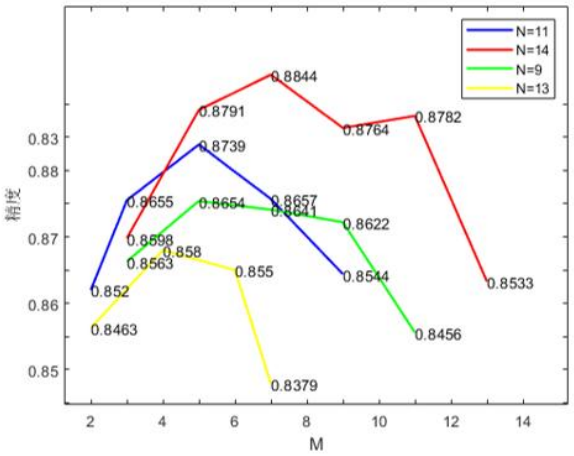


图9　RNN滑动窗口N和预测码M的选择结果

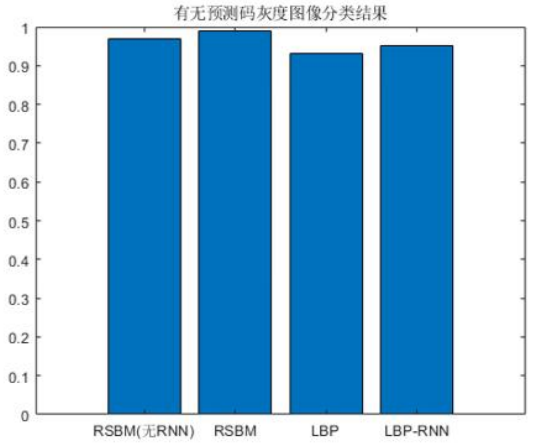


图10　有无预测码灰度图像分类结果

3.3　不同分类方法对比

本文数据按照测试集与训练集为7：3的固定比例，使用轻量级CNN（MobileNet v2）模型进行分类，并与其它四种分类模型进行对比，其中包括基于特征提取的传统分类算法、基于纹理图像的CNN分类算法等。GIST方法[5]是经典的恶意软件二进制文件可视化分类方法之一，通过提取灰度图的GIST特征进行聚类分析。LBP方法[7]在GIST的基础上，通过提取“.text”段的操作码使用敏感哈希，提高了分类信息密度，随后提取LBP特征进行分类。MCSC方法[17]采用一种敏感哈希对全部操作码进行处理并生成灰度图像，然后利用改经的卷积神经网络对恶意软件家族进行分类分析。CNN-image[18]将恶意软件映射成为彩色图像，并使用传统的CNN模型对恶意软件进行分类。各个模型的精度（准确率、误报率）表现如下表所示，实验结果表明本文提出的分类模型拥有更好的分类效果。

表2　不同方法的分类准确率

方法	准确度	误报率
RSBM	99．28%	0.057
GIST	97.03%	0.079
LBP	95.20%	0.071
MCSC	98.86%	0.064
CNN-image	91.75%	0.070

4　总结与展望

本文提出了一种基于静态分析的RSBM算法，将恶意代码转化为RGB图像，同时引入RNN和CNN两种深度学习网络进行预测和分类功能。RNN生成的预测序列，增加信息之间的相关性，增强了分类过程中的抗干扰能力，提升了模型的分类效果。随后使用SimHash对操作码序列进行可视化处理，生成大小相同的RGB图像，解决了因恶意代码长短不同造成的图像大小不一的问题。最后使用轻量级的CNN对图像进行分类，取得了很好的分类效果并在一定程度上提高了分类速度。将来，我们将继续收集数据，将该方法在更大的数据集上进行验证；改进或考虑其它编码方法对操作码进行处理（如dummy编码、one-hot编码+正则化、TF-IDF等），解决one-hot编码中存在的信息稀疏问题，进一步提高实验效率；同时，结合其它方法，提高无监督学习能力，减少分类时间。

参考文献:

[1] 国家互联网应急中心．CNCERT互联网安全威胁报告-2020年3月［EB/OL］（2020-04-30）［2020-07-24］https://www. cert. org. cn/publish/main/45/2020/20200430101311844493731/20200430101311844493731_. html

[2] Schultz MG , Eskin E , Zadok F , Stolfo SJ . Data mining methods for detection of new malicious executables. Proceedings of the IEEE symposium on security and privacy，2001. S&P 2001. Oakland，CA，USA；2000. p. 38－49 . May

[3] Kong D , Yan G . Discriminant malware distance learning on structural information for automated malware classification. Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining，Chicago，Illinois，USA；2013. p. 1357－65 . August

[4] 罗世奇,田生伟,孙华,禹龙．栈式自编码的恶意代码分类算法研究［J］. 计算机应用研究，2018，35(01)：261-265.

[5] Nataraj L, Karthikeyan S, Jacob G, et al. Malware images: Visualization and automatic classification［C］. In Proc. 8th International Symposium on Visualization for Cyber Security (VizSec'11)，Pittsburg，July 20，New York，ACM，2011：21-29.

[6] HAN KS, LIM JH, KANG B, et al. Malware analysis using visualized images and entropy graphs［J］. International Journal of Information Security，vol. 14，no. 1，pp:1-14，2015.

[7] 刘亚姝,王志海,侯跃然,严寒冰．信息密度增强的恶意代码可视化与自动分类方法[J]. 清华大学学报(自然科学版)，2019，59(1)：9-14.)

[8] Kolosnjaji B, Zarras A, Webster G, et al. Deep Learning for Classification of Malware System Call Sequences［C］. LNCS 9992：Australasian Joint Conference on Artificial Intelligence. Cham，Nov 29，Berlin，Springer，2016：137-149.

[9] 赵炳麟,孟曦,韩金,等．基于图结构的恶意代码同源性分析[J]. 通信学报，2017. 38(S2)：86-93.

[10] ZHAO YZ, XU CY, BO B, et al. MalDeep: A Deep Learning Classification Framework against Malware Variants Based on Texture Visualization［J］. Security and Communication Networks，2019，2019(8)：1-11.

[11] Pascanu R, Stokes JW, Sanossian H, et al. Malware classification with recurrent networks［C］. 2015 IEEE International Conference on Acoustics. Brisbane，Apr 19-24，Piscataway，IEEE，2015：1916－1920.

[12] Eui CRS, Dawn S, Reza M. Recognizing functions in binaries with neural networks［C］. the 24th USENIX Conference on Security Symposium. Washington，Aug 12-14，Berkeley，USENIX，2015：611－626.

[13] Tobiyama S, Yamaguchi Y, Shimada H, et al. Malware Detection with Deep Neural Network Using Process Behavior［C］. 2016 IEEE 40th Annual Computer Software and Applications Conference，Atlanta，Jun 10-14，Piscataway，IEEE，2016：577－582.

[14] Kaggle. Microsoft malware classification challenge (big2015）[DB/OL]（2015）［2020-04-24］ https://www. kaggle. com/c/malware-classification/datxiu.

[15] Uddin MS, Roy CK, Schneider KA, alet, On the effectiveness of simHash for detecting near-miss clones in large scale software systems［C］. 18th Working Conference on Reverse Engineering，Limerick，Oct 17-20，Piscataway，IEEE，2011：13-22.

[16] 乔延臣．恶意代码同源判断技术研究[D]. 北京：中国科学院大学，2016.

[17] NI S, Qian Q, Zhang R. Malware identification using visualization images and deep learning[J]. Computers & Security. 2018，77(6)：871-885

[18] Kim HJ, Image-Based Malware Classification Using Convolutional Neural Network［J］. Advances in Computer Science and Ubiquitous Computing(CSA-CUTE 17). 2017，474(1)：1352-1357

[作者简介]

魏书宁（1979—），女，博士，副教授，研究领域为智能信息处理、智能控制、网络信息安全等，E-mail：weishuning@hunnu. edu. cn；

陈小寒（1995—），女，硕士研究生，研究领域为网络信息

安全；

唐勇（1979—），男，博士，副研究员，研究领域为网络安全、软件安全等；

覃正泽（1994—），男，硕士研究生，研究领域为模式识别与智能系统。

SHFuzz: A Hybrid Fuzzing Method Assisted by Static Analysis for Binary Programs

WANG Wenjie[1], TIAN Donghai[1,2], MA Rui[1], WEI Hang[1], YING Qianjin[1], JIA Xiaoqi[1,2,3]

1.School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China;

2.Key Laboratory of Network Assessment Technology, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100049, China;

3.School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Key words: hybrid fuzzing; static analysis; concolic execution; binary programs

Abstract. Fuzzing is an effective technique to find security bugs （or vulnerabilities） in programs. It can quickly explore the input space of programs by generating a large number of different test cases. However， existing fuzzers are still limited in discovering vulnerabilities that are hidden in the deep execution paths of a program. To address this problem， the hybrid fuzzing is proposed. This approach combines fuzzing and concolic execution for going through the complex branch conditions quickly. In general， we observe that the execution path which comes across complex basic blocks may have a higher chance of containing a security bug. Based on this observation， we propose a hybrid fuzzing method assisted by static analysis for binary programs. The basic idea of our method is to prioritize the seed inputs according to the complexity of their associated execution paths. For this purpose， we take advantage of static analysis to extract each basic block’s complexity weights. Then， we accumulate the weights of all basic blocks associated with an execution path extracted by Intel Processor Trace as the corresponding seed weights. Finally， based on these seed weights， we perform the seed prioritization and power scheduling. The key advantage of our method is that our system can test binary programs efficiently by using Intel Processor Trace. To evaluate the effectiveness of our method， we design and implement a prototype system， namely SHFuzz. The evaluation results show SHFuzz discovers more unique crashes on several real-world applications and the LAVA-M dataset when compared to the previous solutions.

1 Introduction

Fuzzing is an automated software testing technique to find security bugs （or vulnerabilities） in programs. Due to the high speed and simplicity， this technique can quickly explore the input space of programs. In recent years， fuzzing has been widely adopted in both academia ［1， 2， 3］ and industry ［4， 5， 6］ for program testing. However， this technique still suffers from discovering vulnerabilities hidden in the deep execution paths of a program. The main reason is that fuzzing blindly mutates the input values， which is not good at passing complex branch conditions ［7］. To address this issue， many solutions have been proposed， and they often utilize some program analysis techniques to assist fuzzing， such as static analysis ［8， 9］， taint tracking ［10， 11］， and symbolic execution ［7，

12， 13］. Hybrid fuzzing is one of these techniques. It combines fuzzing and concolic execution. Since the concolic execution is good at solving complex branch conditions， the hybrid fuzzing has a potentially good effect on vulnerability discovery.

Intuitively， the seed inputs that access more complex basic blocks may have a higher possibility to trigger bugs. These seeds should be selected first and allocated more mutation energy when performing fuzzing and concolic execution. However， many hybrid fuzzers successively select seeds in the order of their generation. This strategy may be ineffective for testing as it may take a long time to select the seed inputs with a higher possibility to trigger program bugs. Moreover， many fuzzers utilize a uniform power scheduling policy when calculating the mutation score. However， a different execution path may have a different chance of containing a bug. Therefore， the uniform power scheduling strategy may not find a bug in a targeted manner.

To address the above problem， we propose a hybrid fuzzing method assisted by static analysis. The basic idea of our method is to prioritize the seed inputs according to the complexity of their associated execution paths， which is based on an observation that the execution path which comes across complex basic blocks may have a higher chance of containing a bug. Accordingly， our fuzzer will allocate more mutation energy for the associated seed inputs. To this end， we take advantage of static analysis to extract each basic block’ s complexity weights at first. Then， we utilize the Intel PT （Intel Processor Trace） ［14］ mechanism to collect the basic block sequence of each execution path efficiently. Thanks to the Intel PT， our system can test binary programs efficiently without using instrumentation. Moreover， we calculate the complexity weights of each execution path by accumulating weights of all basic blocks on this path. Since fuzzing and concolic execution are good at exploring different execution paths， we use two different weights， namely the fuzzing weight （denoted by “FW” for short） and the concolic execution weight （denoted by “CEW” for short）. The FW mainly focuses on the complexity of basic blocks while the CEW focuses on the instructions related to path constraints. When fuzzing a program， we first calculate the FW and the CEW of each seed input dynamically with the help of the Intel PT. Then， we perform the seed prioritization and power scheduling based on the weights of each seed. In general， a seed input with higher weights will be selected first and allocated more mutation energy.

Additionally， we mark a seed as a stuck state when the fuzzer generates fewer useful inputs based on this seed. This state indicates a seed may get stuck due to some complex branch conditions that prevent it from finding new paths. These stuck seeds will be processed by the concolic execution preferentially. On the other hand， to avoid the concolic execution generating seeds that cover the duplicated execution paths reached by the fuzzer， we import the fuzzer’ s code coverage bitmap to the concolic execution. When the concolic execution tries to generate an input， it checks this bitmap at first. In this way， our method can improve the synergy between fuzzing and concolic execution.

To evaluate the effectiveness of our method， we design and implement a prototype system， namely SHFuzz. We conduct experiments on several real-world programs and the LAVA-M dataset. The experiment results show SHFuzz can discover more unique crashes than previous works like PTfuzz ［15］. Moreover， one of the bugs which are found by our system has been assigned a CVE ID （i.e.， CVE-2019-20352）.

Overall， our contributions can be summarized as follows：

We propose a hybrid fuzzing method assisted by static analysis. The basic idea of our method is to combine hybrid fuzzing and static analysis to prioritize seed inputs with higher complexity and vulnerability risk.

We employ the fuzzing weight （FW） and the concolic execution weight （CEW）， respective-

ly. A seed input with a higher FW will be selected first and allocated more mutation energy by the fuzzer. A seed input with a higher CEW will be preferentially processed by the concolic execution.

We record the stuck seed inputs and offer them to the concolic execution first. Moreover, we import the fuzzer's code coverage bitmap to the concolic execution to avoid generating seed inputs that cover the duplicated execution paths.

We design and implement our hybrid fuzzing system, namely SHFuzz. The evaluation results show SHFuzz can discover more unique crashes when compared to the previous solutions.

The rest of this paper is organized as follows. Section 2 analyzes the background and related work of this paper. Section 3 and Section 4 describes the general architecture and the design of SHFuzz, respectively. In Section 5, we describe the implementation of SHFuzz and then evaluate its effectiveness by conducting various experiments on real-world programs and the LAVA-M dataset. Section 6 discusses SHFuzz's limitations and possible solutions. Section 7 concludes this paper.

In this section, we introduce the background and related work of the coverage-based fuzzing and the symbolic execution.

2.1 Coverage-based Fuzzing

Coverage-based fuzzing solutions try to traverse as many programs running states as possible [16]. It becomes popular especially since AFL [17] made good progress in vulnerability discovery. Thus, we take AFL as an example to show the workflow of coverage-based fuzzing. Generally, AFL employs the evolutionary algorithm to mutate seeds and then generate new useful inputs. Fig. 1 shows the basic workflow of AFL. In the beginning, AFL receives initial seed inputs and maintains a seed queue. Then, it selects a seed input from the seed queue and mutates it to generate new inputs. To determine whether a new input is useful or not, it executes the target program and collects run-time coverage information. If an input reveals new branches, AFL will add it to the seed queue. Otherwise, this input will be discarded. AFL continues the above operations until the fuzzing is completed. Based on the above workflow, there are two critical steps in AFL: how to select a seed input from the seed queue (seed selection) and how to mutate it to generate new inputs (seed mutation)?

Seed Selection. Given the same applications, a good seed selection strategy could significantly find more crashes or paths [16]. In particular, AFL simply prioritizes seeds that likely reveal new paths. To improve the efficiency of AFL, AFLFast [18] utilizes Markov Chain to prioritize seeds that being fuzzed fewer times. AFLGo [19] defines some dangerous code locations and selects seeds that likely reach these locations. NeuFuzz [20] utilizes the deep neural network to classify vulnerable paths and then prioritizes seed inputs that cover these paths. In this paper, we take advantage of static analysis to evaluate the complexity of each basic block. Then, we calculate the weights of each seed by analyzing the associated execution path extracted by Intel PT. Finally, we prioritize seed inputs with higher complexity for triggering more crashes.

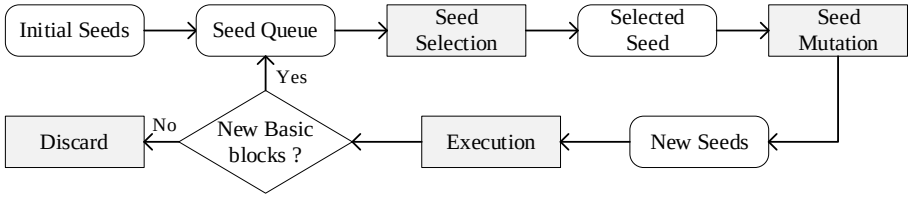


Fig. 1 The basic workflow of AFL.

Seed Mutation. The crashes and paths found by fuzzers may vary significantly depending on their seed mutation strategies. Currently, AFL employs some random and seed splicing mutation opera-

tions， such as *bitflip*， *interest*， *insertion*， etc. However， only the mutation on a few key positions would affect the control flow of the execution. To locate these positions and then adopt optimal mutation operations， VUzzer ［10］ employs static and dynamic analysis to infer values that may affect the control flow. Then， it mutates these recognized values with appropriate offsets to meet branch conditions. Similarly， TIFF ［11］ adopts bug-directed mutation by inferring the type of the input bytes. FairFuzz ［21］ identifiers and mutates rare branches with lightweight program analysis. In this paper， we simply allocate more mutation energy to seed inputs with high complexity so that there is a higher probability to trigger more vulnerabilities.

Code Coverage. Code coverage that guides the fuzzer to execute uncovered paths is an important indicator of the coverage-based fuzzing. To measure the code coverage， AFL instruments a random ID for each basic block at compile time and utilizes a 64KB shared memory （namely bitmap） to record edge coverage information. Fig. 2 shows how AFL updates the bitmap. For each executed edge， AFL updates the corresponding bitmap position that is calculated by the XOR operation with the source ID and the destination ID. To fuzz binary programs and avoid the hash collision caused by random IDs， PTfuzz employs Intel PT to extract the execution path. Then， it takes the memory address of each basic block as a unique ID to update the bitmap. Inspired by PTfuzz， in this paper， we adopt Intel PT to extract the execution path and then measure the code coverage as well. Moreover， we import the fuzzer’s bitmap to the concolic execution to avoid generating seed inputs that cover the duplicated execution paths.

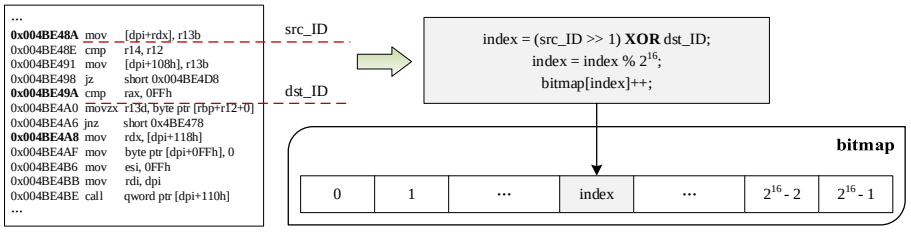


Fig. 2 Bitmap in AFL.

2.2 Symbolic Execution

Although fuzzing has made good progress in vulnerability discovery， it still suffers from limitations in discovering bugs that are hidden deep in the execution path due to some complex branch conditions. Fortunately， symbolic execution ［22， 23］ could construct inputs that satisfy the path constraints by taking symbolic values as program inputs and then representing branch conditions as symbolic expressions ［24， 25］. After solving these symbolic expressions， the symbolic execution can generate new seed inputs that explore multiple execution paths.

Unfortunately， classical symbolic execution fails to explore all feasible execution states and some dynamic libraries. To cope with this issue， the concolic execution combines symbolic execution and concrete execution. When the concrete execution takes a branch， the symbolic execution extracts the corresponding constraints and then negates and solves the constraints for exploring a different branch. Moreover， to take the respective advantages of the concolic execution and the fuzzing， the hybrid fuzzing combines these two techniques. The concolic execution is used to generate seeds that satisfy branch conditions while fuzzing is used to mutate and verify these seeds quickly. With the help of hybrid fuzzing， Driller ［13］ identifies the same number of vulnerabilities with the winning team of the DARPA Cyber Grand Challenge. QSYM ［7］ implements a fast concolic execution engine by exploiting the Intel PIN to analyze each instruction at runtime. Moreover， it utilizes some heuristics to solve the path constraints.

By using the hybrid fuzzing， QSYM finds more bugs than the state-of-the-art fuzzers.

However， QSYM employs a similar seed selection strategy with AFL. It prioritizes seed inputs with new branches and smaller file sizes. In this paper， we consider the CEW of each seed input on the top of QSYM. On the other hand， inspired by Driller， we prioritize seed inputs that may get stuck due to the complex branch conditions， which could help the fuzzer go through branch conditions faster and explore more execution paths.

3 Overview

Fig. 3 presents a high-level overview of SHFuzz. It consists of three components： static analyzer， fuzzer， and concolic execution.

Static analyzer. Static analysis is the first step of SHFuzz to evaluate the complexity and vulnerability risk of each basic block in the target binary program. Our method utilizes the IDA Pro to get the assembly code of the target program and then calculates the weights of each basic block. Since the fuzzing and the concolic execution are good at exploring different execution paths， we employ the fuzzing weight （FW） and the concolic execution weight （CEW）， respectively. The FW focuses on the complexity of basic blocks while the CEW focuses on the instructions related to path constraints.

Fuzzer. Our main fuzzing loop follows the basic workflow of PTfuzz. It starts with initial seed inputs and tries to generate new useful inputs that could increase the code coverage. For each new seed input， the fuzzer executes it and exploits Intel PT to extract the associated execution path. This path can help the fuzzer to update the code coverage and decide whether a seed input should be kept or discarded. On the other hand， it can be used to calculate the seed weights by accumulating all basic blocks’ weights on this path. Based on the seed weights， the fuzzer can perform seed prioritization and power scheduling when choosing the next input. In general， the seed inputs with higher weights will be selected first and allocated more mutation energy.

Concolic execution. The concolic execution is used to drive a program to go through complex branch conditions. It first takes seed inputs with higher CEWs or stuck states from the seed queue. Then， it tries to generate new seed inputs by extracting and solving the associated path constraints. To improve the synergy between the fuzzing and the concolic execution， we import the fuzzer’s bitmap to the concolic execution.

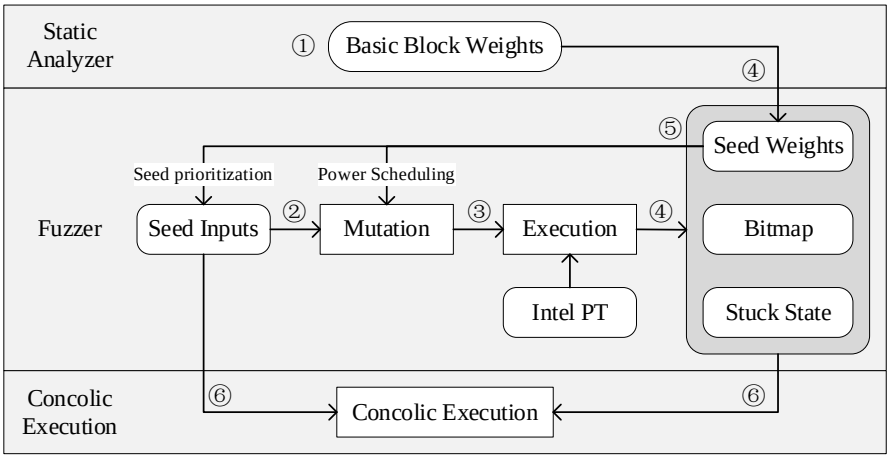


Fig. 3 An overview of SHFuzz.

In general， the basic workflow of our method can be summarized as follows：

 We take advantage of static analysis to calculate the weights of each basic block by scanning the assembly code of the target program.

The fuzzer takes seed inputs from the seed

queue and then generates new seed inputs by various mutations.

For each new seed input, the fuzzer executes it and then exploits Intel PT to extract the associated execution path.

Based on the execution path, the fuzzer updates the coverage bitmap and then calculates the FW and CEW of the seed related to this path. Moreover, the fuzzer marks a seed as a stuck state when it generates fewer useful seed inputs based on this seed.

The fuzzer performs seed prioritization and power scheduling based on the seed weights. A seed input with higher FWs will be selected first and allocated more mutation energy.

The concolic execution prioritizes seeds with higher CEWs or stuck states from the seed queue. Moreover, we import the fuzzer's bitmap to the concolic execution for improving the synergy between fuzzing and concolic execution.

4 Design

In this section, we present the design details of our hybrid fuzzing system SHFuzz from three parts: static analyzer, fuzzer, and concolic execution.

4.1 Static Analyzer

The static analyzer disassembles the target binary program and then scans the assembly code to evaluate the complexity and vulnerability risk of each basic block. Since fuzzing and concolic execution are good at exploring different execution paths, we employ the fuzzing weight (FW) and the concolic execution weight (CEW), respectively.

Fuzzing Weight. To evaluate the FW, we take advantage of two complexity metrics inspired by Cerebro [8], namely McCabe's Cyclomatic Complexity (CC) [26] and Halstead Complexity Measures (H.B) [27]. CC is calculated by the number of edges and basic blocks in a function, and it reflects the structural complexity. H.B is calculated by the number of operators and operands in a function. This metric reflects the operation complexity. In general, only certain basic blocks in a function will be executed. Therefore, CC and H.B metrics are coarse-grained. To improve the accuracy, different from the previous method [8], we divide the CC and H.B metrics by the number of basic blocks in the associated function. Then, we take the result of the division as each basic block's complexity weight.

Moreover, we consider the Vulnerable Library Function (VLF) metric, which is calculated by counting the times that some vulnerable library functions (e.g., *strcpy*, *strcat*, *memcpy*) are invoked in a function. Inspired by the method [28], all banned functions listed in the Microsoft secure development lifecycle concept [29] will be considered as vulnerable functions. Similar to the H.B and CC, we divide the VLF metric by the number of basic blocks in the associated function and then take the result as the VLF score of each basic block.

After calculating the CC, H.B, and VLF score of each function, we use the following formula to normalize each metric for eliminating the effects of different magnitudes:

$$norm(f_m) = \frac{f_m}{t_m} \tag{1}$$

where f_m denotes the score of a function about the metric m, t_m is the sum of all function scores about the matric m.

After that, we calculate the FW of each basic block with the following equation:

$$bbl_{fw} = \frac{norm(f_{H.B}) + norm(f_{CC}) + norm(f_{VLF})}{bbl_{count}} \tag{2}$$

where bbl_{fw} denotes the FW of a basic block in the function f, bbl_{count} is the number of basic blocks in this function.

Concolic Execution Weight. The CEW mainly focuses on instructions associated with path constraints, such as conditional jump instructions and assignment instructions. Specifically, we exploit the ABC complexity metric [28] to evaluate the CEW. The ABC metric is calculated by the number of assignment instructions, jump instructions, and call instructions. Since the concolic execution is not good at handing library function calls, we only consider

the assignment and jump instructions when calculating the ABC metric. After getting the ABC score of each function， we adopt the formula 3 to calculate the CEW of each basic block：

$$bbl_{cw} = \frac{f_{ABC}}{bbl_{count}} \tag{3}$$

where bbl_{cw} denotes the CEW of a basic block in the function f， bbl_{count} is the number of basic blocks in this function.

4.2 Fuzzer

Algorithm 1 shows the main fuzzing loop of SHFuzz. It follows the basic workflow of PTfuzz， which starts with initial seeds and tries to generate new useful inputs. In general， our main fuzzing loop can be divided into four parts： path extraction （Line 4， 14）， weight calculation （Line 6， 17）， seed prioritization （Line 10）， and power scheduling （Line 11）.

Path Extraction. When fuzzing a program， we need to extract the complete execution path for updating the bitmap and calculating the FW and the CEW. Inspired by PTfuzz， we utilize Intel PT to recover the execution path for each execution. When the Intel PT is enabled， the CPU can record the execution control flow of a target program with a minimum performance overhead. Specifically， it collects the runtime control flow information and stores into the specified memory space in the form of compressed PT packets. By decoding these PT packets， we can extract the complete execution path that is comprised of a series of basic blocks （Line 4， 14）. Based on the execution path， SHFuzz can update the code coverage bitmap for evaluating whether a seed is useful or not （Line 5， 15， 16）. On the other hand， the execution path can be used to calculate the weights of each seed by accumulating all basic blocks' weights （Line 6， 17）.

Weight Calculation. Similar to the weights of each basic block， there are two kinds of weights for each seed： FW and CEW. The FW is calculated by accumulating all basic blocks' FWs associated with an execution path （Line 30）. The CEW is calculated by accumulating all basic blocks' CEWs associated with the same execution path （Line 31）. Based

Algorithm 1 The Main Fuzzing Loop.

Input: (i) Target binary program: *P*, (ii) Initial seeds: *initial_seeds*, (iii) Basic blocks' weights extracted by static analysis: *bbl_weights*.

```
1: function main_fuzzing_loop(P, initial_seeds, bbl_weitghts)
2:    Initialize the code coverage bitmap (bitmap) and the seed queue (Q)
3:    for each seed ∈ initial_seeds do
4:       path = path_extraction(P, seed)
5:       bitmap = update_bitmap(path, bitmap)
6:       fuzzing_weight, concilic_weight = weight_calculation(path, bbl_weitghts)
7:       put_into_queue(Q, seed, fuzzing_weight, concolic_weight)
8:    end for
9:    while In the fuzzing loop do
10:      seed = seed_prioritization(Q)
11:      energy = power_scheduling(seed)
12:      for i form 0 to energy do
13:         new_seed = mutation(seed)
14:         path = path_extraction(P, new_seed)
15:         if has_new_coverage(path, bitmap) then
16:            bitmap = update_bitmap(path, bitmap)
17:            fuzzing_weight, concilic_weight = weight_calculation(path, bbl_weitghts)
18:            put_into_queue(Q, new_seed, fuzzing_weight, concolic_weight)
19:         end if
20:      end for
21:   end while
22: end function
23:
24: function weight_calculation(path, bbl_weitghts)
25:   fuzzing_weight = 0
26:   concolic_weight = 0
27:   execution_times = []
28:   for each basic_block ∈ path do
29:      if is_power_of_two(execution_times[basic_block]) then
30:         fuzzing_weight += bbl_weights[basic_block].fuzzing_weight
   //The fuzzing weight of basic_block, and it is extracted by static analysis.
31:         concolic_weight += bbl_weights[basic_block].concolic_weight
   //The concolic execution weight of basic_block, and it is extracted by static analysis.
32:      execution_times[basic_block] += 1
33:      end if
34:   end for
35:   return {fuzzing_weight, concolic_weight}
36: end function
```

on the FW of a seed， the fuzzer performs seed prioritization and power scheduling when choosing the next seed to mutate. Similarly， the concolic execution engine first selects seed inputs with higher CEWs. Moreover， to avoid the basic blocks in loop structures being frequently calculated， we employ the exponential back-off algorithm to prune these frequent basic blocks. This algorithm only calculates basic blocks whose execution times are a power of two （Line 29）. Moreover， since the weight calculation is time-consuming， and it could slow down the fuzzing speed， we only calculate the weights for the seed inputs which trigger new branches （Line 15）.

Seed Prioritization. In general， the execution path which comes across complex basic blocks may have a higher chance of containing a bug. Therefore， we need to prioritize seed inputs with higher FWs for triggering more bugs. To this end， after getting the weights of seed inputs， the fuzzer places them into the seed queue in the descending order of their FWs （Line 7， 18）. By doing so， the fuzzer will first pick the seed input with higher FW from the seed queue （Line 10）.

Power Scheduling. For a seed input ready to be mutated， the fuzzer needs to assign mutation energy to it. This energy is used to control the execution time of mutation operations. A seed with higher FW should be mutated more frequently since it is more likely to trigger new paths or crashes （Line 11）. As a result， we utilize the following formula to assign mutation energy for each seed based on its FW：

$$energy' = energy \times (1 + \frac{fuzzing\_weight}{average\_fuzzing\_weight}) \quad (4)$$

where *energy* is the mutation energy calculated by the original PTfuzz strategy， *fuzzing_weight* is the FW of the current seed input， *average_fuzzing_weight* is the average FW calculated by all seed inputs.

Moreover， we also count the number of useful seed inputs generated by mutating a seed input. If this number is less than half of the average calculated by all seeds， we mark this seed as a stuck state. This state indicates the mutations may be less useful to this seed. In this case， the concolic execution may help generate more interesting seeds by solving the associated path constraints. As a result， we prioritize seeds with stuck states or higher CEWs when performing the concolic execution.

4.3 Concolic Execution

Our concolic execution engine is implemented on the top of QSYM ［7］. It first utilizes the Intel PIN tool to capture the execution path constraints. Then it employs the Z3 solver ［30］ to solve these constraints for generating new seed inputs that satisfy the complex branch conditions. With the help of the concolic execution， the fuzzer could quickly go through complex branch conditions by executing these generated seeds. When performing the concolic execution， the seed prioritization is the first step for selecting an appropriate seed input from the fuzzer’s seed queue.

Seed Prioritization. As described in Section 4.1， the CEW indicates the complexity of instructions associated with path constraints. Accordingly， we employ the CEW as an indicator of seed selection. This weight can be extracted during the fuzzing process. On the other hand， we utilize the stuck state as another indicator. The stuck state shows a seed may get stuck during fuzzing due to some complex branch conditions. SHFuzz offers seeds that get stuck or have higher CEWs to the concolic execution engine at first. Then， the concolic execution engine tries to extract path constraints associated with this selected seed. After that， it generates new seed inputs that satisfy branch conditions by solving the above path constraints. Finally， the fuzzer will execute these new seed inputs and then update the code coverage bitmap. In this way， the fuzzer has more chances to go through complex branch conditions and then trigger more paths or crashes.

Coverage Bitmap. Listing 1 is a simple code snippet taken from *nasm*. We take this code as a motivating example to elaborate on why we need to import the fuzzer’s bitmap to the concolic execution. The control flow graph of this code is shown in Fig.

4. It contains 7 basic blocks （without considering the called functions）. The green color indicates a basic block has been executed by the fuzzer. This execution information is recorded in the code coverage bitmap. If the concolic execution does not check this bitmap， it needs to extract and solve path constraints associated with the basic blocks A， C and D. However， the basic blocks A， C， and their child nodes have been executed. As a result， some seed inputs generated by the concolic execution may cover the same execution paths with the fuzzer.

To address the above issue， we import the fuzzer’s bitmap to the concolic execution. When the concolic execution engine tries to generate a seed input， it checks this bitmap at first. Specifically， for each basic block that contains the condition jump instruction， it exploits Intel PIN to extract two possible basic blocks following the condition jump. Then， we use the method shown in Fig. 2 to check the corresponding bitmap indexes based on the memory addresses of the current and two next basic blocks. If there is any edge not being executed， the concolic execution engine will solve the path constraints associated with the current basic block. In this way， only new seed inputs will be retained， and the execution time can be saved.

Listing 1： Sample code snippet from *nasm*.

```
1: static const char *unquote_token(Token *t){
2:   if (t->type != TOK_STRING)   //Basic Block A
3:     return tok_text(t);   //Basic Block B
4:   t->type = TOK_INTERNAL_STRING;   //Basic Block C
5:
6:   if(t->len > INLINE_TEXT){
7:     char *p = t->text.p.ptr;   //Basic Block D
8:     t->len = nasm_unquote(p, NULL);
9:     if (t->len <= INLINE_TEXT) {
10:       nasm_zero(t->text.a);   //Basic Block E
11:       memcpy(t->text.a, p, t->len);
12:       nasm_free(p);
13:       return t->text.a;
14:     } else {
15:       return p;   //Basic Block F
16:     }
17:   } else {
18:     t->len = nasm_unquote(t->text.a, NULL);   //Basic Block G
19:     return t->text.a;
20:   }
21: }
```

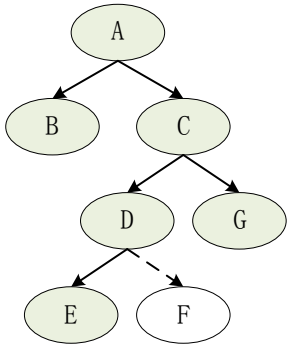


Fig. 4 The control flow graph of Listing 1.

5 Implementation and Evaluation

SHFuzz is implemented on the top of PTfuzz［15］ and QSYM ［7］， which are the state-of-the-art fuzzer and concolic execution engine， respectively. Moreover， our static analyzer utilizes the disassembly tool （i. e.， IDA Pro） to calculate the weights of each basic block. To evaluate the weights of each seed input at runtime， we accumulate all basic blocks’ weights associated with an execution path extracted by the Intel PT. Moreover， we also add the support on PTfuzz to perform seed prioritization and power scheduling based on the seed weights. To prioritize an appropriate seed input for the concolic execution， we add the support on QSYM to first select seed inputs with stuck state or higher CEW. Moreover， we also import the bitmap of the fuzzer to the concolic execution engine for improving the synergy between fuzzing and concolic execution.

5.1 Experiment Setup

Baseline Fuzzers. To highlight the effectiveness of our method， we compare the experimental results with PTfuzz， which is a state-of-the-art fuzzer that supports fuzzing binary programs by utilizing Intel PT to extract the execution flow of programs.

Benchmarks. First， we evaluate SHFuzz on 6 real-world applications as shown in Table 1. These applications cover a wide range of functionality and input formats， including GNU Binutils （i.e.， *size*， *nm*）， image processing libraries （i. e.， *jhead*，

libtiff), and PDF processing tool (i.e., *pdftotext*). On the other hand, all these applications are well-tested by the state-of-the-art fuzzers [7, 12]. Therefore, they are good benchmarks to demonstrate the effectiveness of a fuzzer.

Second, we choose the LAVA-M dataset [31] as another benchmark. LAVA-M is a test suite for bug detection. It consists of four buggy programs (i.e., *who*, *uniq*, *base64*, and *md5sum*). Each bug is injected artificially and identified by a unique ID. Since the LAVA-M dataset contains several well-designed bugs, many fuzzers choose it as their benchmarks. The programs contained by LAVA-M have been widely evaluated as well.

Evaluation Metrics. We select unique crashes and paths as major evaluation metrics to compare the efficiency of each fuzzer. The number of unique crashes represents the fuzzer's ability to trigger vulnerabilities. The number of unique paths represents the code coverage captured by a fuzzer. A fuzzer is more effective if it could trigger more unique crashes or paths.

Experiment Environment. For each program under test, we employ PTfuzz and SHFuzz on it for 24 hours. Then, we compare the effectiveness of different strategies. Moreover, all experiments are conducted with the same initial seed inputs on a machine with one Intel i7-9700K CPU and 32GB RAM running 64-bit Ubuntu 16.04 LTS.

Table 1 Real-world applications used in the evaluation.

Applications	Version	Input Format
nasm	2.15rc0	assembly code
nm+ binutils	2.23	binary
size+ binutils	2.23	binary
jhead	2.97	jpg
tiff2ps+ libtiff	3.9.7	tiff
pdftotext+ xpdf	4.02	pdf

5.2 Evaluation on Real-world Applications

To show the effectiveness of our hybrid fuzzing strategy assisted by static analysis, we first conduct experiments on 6 real-world applications. For each application, we count the number of unique paths and crashes found by PTfuzz and SHFuzz with the same initial seeds. To show the benefits of static analysis and concolic execution respectively, we conduct experiments on two cases: SHFuzz without concolic execution and SHFuzz with concolic execution.

Table 2 shows the experiment results of PTfuzz and SHFuzz on 6 real-world applications. For the case of SHFuzz without concolic execution, SHFuzz discovers 15521 unique paths and 1047 unique crashes while PTfuzz finds 15156 unique paths and 778 unique crashes in total. SHFuzz outperforms PTfuzz on all applications in terms of the number of crashes, which increases by 34.58% over PTfuzz in total. Specifically, SHFuzz discovers 150% and 120% more unique crashes on *size* and *tiff2ps*, respectively. Since SHFuzz pays more attention to more complex basic blocks, it finds fewer paths on *nasm* and *tiff2ps*. Moreover, the run-time weight calculation is time-consuming, which may slow down the speed of finding unique paths. However, SHFuzz can discover more unique crashes on all applications and more unique paths on the other 4 applications. As a conclusion, with the help of static analysis, SHFuzz can discover more unique crashes than PTfuzz, which demonstrates its effectiveness in vulnerability discovery.

For the case of SHFuzz with concolic execution, SHFuzz finds more unique paths and crashes than PTfuzz and SHFuzz (without concolic execution) in total. Furthermore, SHFuzz could find more unique paths and crashes than PTfuzz on all applications and more unique crashes than the case of SHFuzz without concolic execution on 4 applications (but except for *tiff2ps* and *pdftotext*). In particular, with the help of concolic execution, SHFuzz finds 53 unique crashes on *jhead* while the other two fuzzers do not find any crashes. Specifically, compared with PTfuzz, SHFuzz finds 141.11% more unique paths on *jhead* and 450% more unique crashes on *size*. Compared with the case of SHFuzz without concolic execution, SHFuzz discovers 131.06% more unique paths on *jhead* and 120% more unique crashes

Table 2 The unique paths and crashes found by PTfuzz and SHFuzz on 6 applications.

Applications	PTfuzz		SHFuzz (without concolic execution)		SHFuzz (with concolic execution)	
	paths	crashes	paths	crashes	paths	crashes
nasm	2565	442	2386	503	3031	512
nm	3615	26	3658	37	4013	47
size	2291	12	2326	30	3299	66
jhead	506	0	528	0	1220	53
tiff2ps	352	10	305	22	374	11
pdftotext	5827	288	6318	455	6363	386
total	15156	778	15521	1047	18300	1075

on *size*. Moreover， one of our finding bugs has been assigned a CVE ID （i. e.， CVE-2019-20352）. However， SHFuzz finds fewer unique crashes on *tiff2ps* and *pdftotext* since the concolic execution engine may drive the fuzzer to reach more execution paths rather than the code locations that cause crashes.

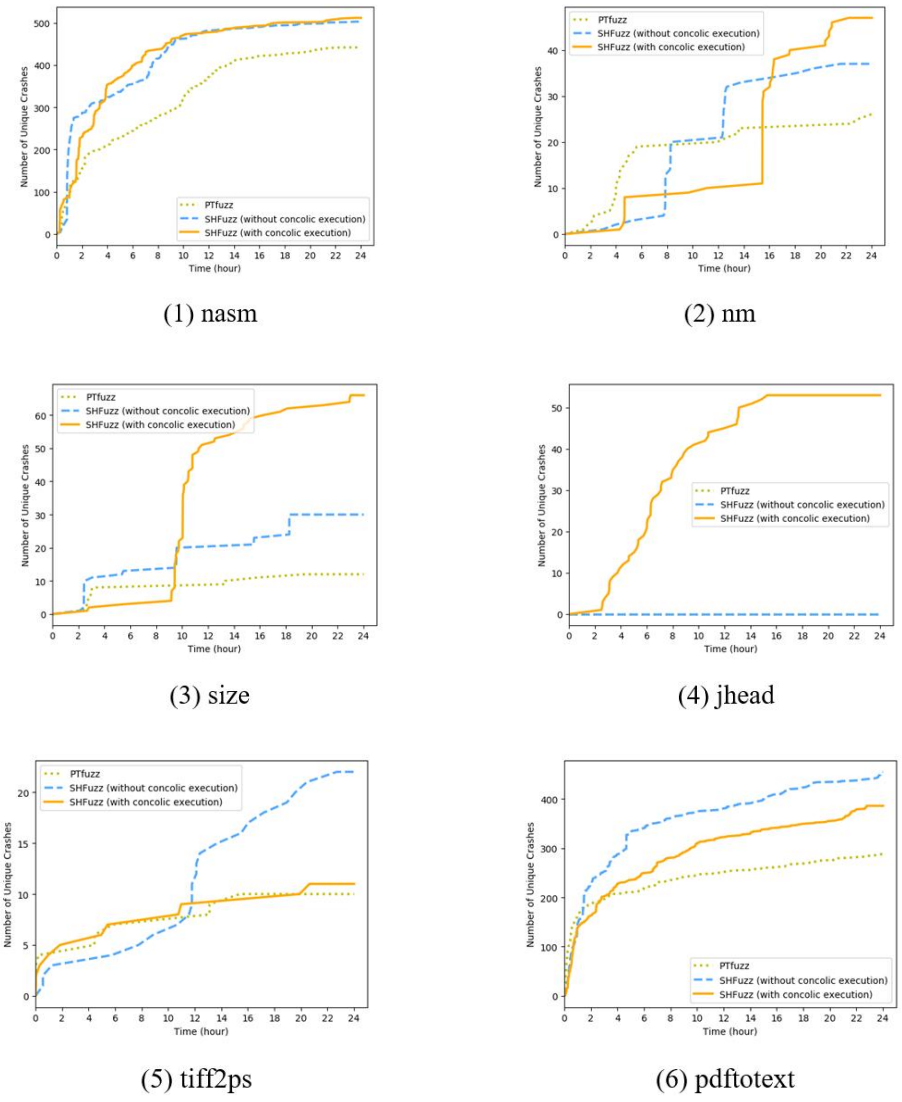


Fig. 5 Number of unique crashes discovered by PTfuzz and SHFuzz over time in real-world applications in 24 hours.

To further demonstrate the effectiveness of SHFuzz, we track the growth trend of the number of unique crashes explored in 24 hours, which is presented in Fig. 5. From this Fig., we can see that SHFuzz outperforms PTfuzz on all applications eventually. The unique crashes found by PTfuzz may grow rapidly at first but it gets stuck or slows down after a while, especially on *nm*, *size*, and *tiff2ps*. Moreover, the number of unique crashes discovered by SHFuzz with concolic execution sharply increases in *nm* and *size* at a time. The reason is that the concolic execution engine generates some new interesting seed inputs. These inputs could sanctify the path constraints and drive the fuzzer to go through deep paths and trigger more crashes. Furthermore, compared with the case of SHFuzz only assisted by static analysis, the concolic execution is ineffective on *tiff2ps* and *pdftotext* since it drives the fuzzer to find more execution paths rather than crashes.

5.3 Evaluation on LAVA-M

To further evaluate the effectiveness of our proposed SHFuzz, we also compare it with PTfuzz using the LAVA-M dataset [31], which consists of four buggy programs (i. e., *who*, *uniq*, *base64*, and *md5sum*). We run PTfuzz, SHFuzz (without concolic execution), and SHFuzz (with concolic execution) on these four buggy programs for 24 hours with the same settings. Then, we count the number of unique crashes and paths found on each program.

As shown in Table 3, with the help of static analysis, SHFuzz (without concolic execution) finds more unique paths than PTfuzz on *who*, *md5sum*, and *uniq*. Specifically, SHFuzz (without concolic execution) discovers 1009, 645, and 262 unique paths on *who*, *md5sum*, and *uniq* while PTfuzz finds 907, 574, and 250 unique paths, which increases around 11.25%, 12.37%, and 4.80%, respectively. Particularly, SHFuzz (without concolic execution) discovers 29 unique crashes on *who* while PTfuzz finds 3 crashes. However, PTfuzz outperforms SHFuzz on *base64* since SHFuzz pays more attention to more complex basic blocks rather than tries to improve the code coverage.

Table 3 The unique paths and crashes found by PTfuzz and SHFuzz on LAVA-M.

Applications	PTfuzz		SHFuzz (without concolic execution)		SHFuzz (with concolic execution)	
	paths	crashes	paths	crashes	paths	crashes
who	907	3	1009	29	1029	92
md5sum	574	0	645	0	682	3
uniq	250	0	262	0	271	0
base64	467	0	437	0	470	89
total	2198	3	2353	29	2452	184

With the help of concolic execution, SHFuzz finds 2452 unique paths and 184 unique crashes in total. It discovers 4.21% more unique paths and 534.48% more unique crashes when compared to SHFuzz without the concolic execution. In particular, SHFuzz finds 3 and 89 unique crashes on *md5sum* and *base64* respectively while SHFuzz (without concolic execution) and PTfuzz do not find any crashes. Moreover, SHFuzz finds 92 unique crashes on *who* while SHFuzz (without concolic execution) and PTfuzz finds 29 and 3 unique crashes, respectively. Furthermore, SHFuzz discovers more unique paths than PTfuzz on all programs. However, SHFuzz does not discover any crashes on *uniq*. A possible reason is that the fuzzer could not provide appropriate seed inputs to the concolic execution to generate useful inputs.

6 Discussions and Limitations

SHFuzz has some limitations that need to be improved. First, since we need to calculate the seed weights at runtime, it could slow down the fuzzing speed. To reduce this overhead, SHFuzz only calculates weights for seed inputs that trigger new branches. Second, we calculate the seed weights by accu-

mulating the weights of all basic blocks on the associated execution path. This calculation method favors larger seed inputs because they may contain more basic blocks. To balance the weight and file size of a seed input， a possible solution is to prioritize seed inputs based on more comprehensive metrics. Moreover， we just allocate more mutation energy for seeds with higher fuzzing weights， which can be improved by utilizing some heuristic algorithms to choose an optimal mutation operation. Since the symbolic execution is limited to solve the complex path constraint in large-scale applications， a more advanced combination scheme between the fuzzing and concolic execution is needed for hybrid fuzzing.

7 Conclusion

In this paper， we present SHFuzz， a hybrid fuzzing system assisted by static analysis for testing binary programs. The basic idea of SHFuzz is to perform seed prioritization and power scheduling based on the fuzzing and concolic execution weights of seed inputs. Additionally， SHFuzz records the stuck seed inputs and imports the fuzzer’s coverage bitmap to the concolic execution to improve the efficiency and effectiveness of the hybrid fuzzing. Our evaluation results show SHFuzz could discover more unique crashes on several applications and the LAVA-M dataset when compared to the previous solutions.

References:

[1] Woo M, Cha S K, Gottlieb S, et al. Scheduling black-box mutational fuzzing[C]//Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. 2013：511-522.

[2] Cha S K, Woo M, Brumley D. Program-adaptive mutational fuzzing [C]//2015 IEEE Symposium on Security and Privacy. IEEE，2015：725-741.

[3] Han H S, Cha S K. Imf: Inferred model-based fuzzer [C]// Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017：2345-2358.

[4] Adobe. A Basic Distributed Fuzzing Framework for FOE. https：//blogs. adobe. com/security/2012/05/a-basic-distributed-fuzzing-framework-for-foe. html.

[5] Google. ClusterFuzz. https：//google. github. io/clusterfuzz/.

[6] Google. OSS-Fuzz. https：//google. github. io/oss-fuzz/.

[7] Yun I, Lee S, Xu M, et al. QSYM: A practical concolic execution engine tailored for hybrid fuzzing [C]//27th USENIX Security Symposium (USENIX Security 18). 2018：745-761.

[8] Li Y, Xue Y, Chen H, et al. Cerebro: context-aware adaptive fuzzing for effective vulnerability detection[C]//Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 2019：533-544.

[9] Du X, Chen B, Li Y, et al. Leopard: Identifying vulnerable code for vulnerability assessment through program metrics [C]//2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE). IEEE，2019：60-71.

[10] Rawat S, Jain V, Kumar A, et al. VUzzer: Application-aware Evolutionary Fuzzing [C]//In Proceedings of the 2017 Annual Network and Distributed System Security Symposium (NDSS). 2017，17：1-14.

[11] Jain V, Rawat S, Giuffrida C, et al. TIFF: using input type inference to improve fuzzing [C]//Proceedings of the 34th Annual Computer Security Applications Conference. 2018：505-517.

[12] Huang H, Yao P, Wu R, et al. PANGOLIN: Incremental Hybrid Fuzzing with Polyhedral Path Abstraction[J]. 2020.

[13] Stephens N, Grosen J, Salls C, et al. Driller: Augmenting Fuzzing Through Selective Symbolic Execution [C]//In Proceedings of the 2016 Annual Network and Distributed System Security Symposium (NDSS). 2016，16(2016)：1-16.

[14] CorporationIntel. Intel Processor Trace. Accessed: Jun. 1，2020. [Online]. Available: https：//software. intel. com/en-us/blogs/2013/09/18/processortracing.

[15] Zhang G, Zhou X, Luo Y, et al. Ptfuzz: Guided fuzzing with processor trace feedback[J]. IEEE Access，2018，6：37302-37313.

[16] Li J, Zhao B, Zhang C. Fuzzing: a survey[J]. Cybersecurity，2018，1(1)：6.

[17] ZalewskiMichal. American Fuzzy Lop. http://lcamtuf. coredump. cx/afl/，2015.

[18] Böhme M, Pham V T, Roychoudhury A. Coverage-based greybox fuzzing as markov chain [J]. IEEE Transactions on Software Engineering，2017，45(5)：489-506.

[19] Böhme M, Pham V T, Nguyen M D, et al. Directed greybox fuzzing [C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017：2329-2344.

[20] Wang Y, Wu Z, Wei Q, et al. Neufuzz: Efficient fuzzing with deep neural network[J]. IEEE Access，2019，7：36340-36352.

[21] Lemieux C, Sen K. Fairfuzz: Targeting rare branches to rapidly increase greybox fuzz testing coverage [J]. arXiv preprint arXiv: 1709. 07101，2017.

[22] Clarke L A. A program testing system[C]//Proceedings of the 1976 annual conference. 1976：488-491.

[23] King J C. Symbolic execution and program testing [J]. Communications of the ACM，1976，19(7)：385-394.

[24] Cadar C, Sen K. Symbolic execution for software testing: three decades later [J]. Communications of the ACM，2013，56(2)：82-90.

[25] Baldoni R, Coppa E, D'elia D C, et al. A survey of symbolic execution techniques[J]. ACM Computing Surveys (CSUR)，2018，

51(3): 1-39.
[26] McCabe T J. A complexity measure [J]. IEEE Transactions on software Engineering, 1976 (4): 308-320.
[27] Maurice H H. Elements of software science (operating and programming systems series)[J]. 1977.
[28] Shudrak M O, Zolotarev V V. Improving fuzzing using software complexity metrics [C]//ICISC 2015. Springer, Cham, 2015: 246-261.
[29] Secure Development Lifecycle. List of banned syscalls: https://msdn. microsoft. com/en-us/library/bb288454. aspx.
[30] De Moura L, Bjørner N. Satisfiability modulo theories: introduction and applications [J]. Communications of the ACM, 2011, 54(9): 69-77.
[31] Dolan-Gavitt B, Hulin P, Kirda E, et al. Lava: Large-scale automated vulnerability addition [C]//2016 IEEE Symposium on Security and Privacy (SP). IEEE, 2016: 110-121.

About the authors

WANG Wenjie received a B. E. degree in software engineering from Beijing Institute of Technology, China, in 2018, where he is currently pursuing an M. S. degree with the School of Computer Science and Technology. His research interests include software security and vulnerability analysis. (Email: bit_wen@163. com)

TIAN Donghai [corresponding author] received the Ph. D. degree from Beijing Institute of Technology, China, in 2012. He is an associate professor with the School of Computer Science and Technology, Beijing Institute of Technology, China. His current research interests include software security, malware analysis and detection, Android security, and cloud security. (Email: donghaitad@gmail. com)

MA Rui received the Ph. D. degree from Beijing Institute of Technology, China, in 2004. She is an associate professor with the School of Computer Science and Technology, Beijing Institute of Technology, China. Her current research interests include software security, Internet of things, neural network, and data mining. (Email: mary@bit. edu. cn)

WEI Hang received a B. E. degree in software engineering from Beijing Institute of Technology, China, in 2019, where he is currently pursuing an M. S. degree with the School of Computer Science and Technology. His research interests include software security and malware analysis. (Email: weimarinata@gmail. com)

YING Qianjin received the B. E. degree from Zhengzhou University, China, in 2020. He is currently pursuing an M. S. degree with the School of Computer Science and Technology at the Beijing Institute of Technology, China. His research interests include software security and malware analysis.

JIA Xiaoqi received the Ph. D. degree from the University of Chinese Academy of Sciences, China, in 2010. He is a professor at the State Key Laboratory of Information Security, Institute of Information Engineering in the Chinese Academy of Sciences, China. His research interests include operating system security, cloud security, and virtualization technologies.

面向武器装备的内生安全控制计算机设计

霍立田，余新胜，罗论涵，解维
中国电子科技集团公司第三十二研究所，上海，201808

摘　要：本文针对当前网络空间安全的严峻态势和武器装备安全问题，提出一种面向武器装备的内生安全计算机设计思路。首先，该设计基于拟态防御的DHR架构，在不改变系统操作流程的情况下，基于C/S架构实现TCP数据收发，并通过嵌入式异构执行体的不同配置实现计算机的内生安全设计，包括分发代理、裁决服务、反馈控制等关键模块。其次，通过研究面向武器装备地面系统功能和信息安全的需求背景，将计算机的显控、业务模块进行分离，通过裁决服务进行输出控制，实时切换异常信息的显示，提高系统清洗恢复的效率。最后，对于内生安全计算机的分发代理、裁决服务和反馈控制模块均采用嵌入式实时处理方式，很好的保证计算机服务处理的实时性和时间确定性。

关键词：内生安全计算机、武器装备、分发代理、裁决服务、反馈控制

Design of Endogenous Safety Control Computer for Weaponry

Huo Litian, Yu Xinsheng, Luo Lunhan, Xie Wei
The 32nd Research Institute of China Electronics Technology Group Corporation,Shanghai, 201808

Abstract: In this paper, aiming at the current severe situation of cyberspace security and the security problems of weapon equipment, a design idea of endogenous security computer for weapon equipment is proposed. First, the design is based on the DHR architecture of mimic defense. Without changing the system operation process, the C/S architecture realizes TCP data transmission and reception, and realizes the computer's endogenous security design through the different configurations of embedded heterogeneous executive bodies. This computer includes key modules such as distribution agent, voting service, feedback control, etc. Second, by studying the background of the requirements for weaponry ground system functions and information security, the computer display control and business modules are separated, the output control is performed through the voting service, the display of abnormal information is switched in real time, and the efficiency of system cleaning and recovery is improved. Finally, the distribution agent, voting service and feedback control module of the endogenous security computer all adopt embedded real-time processing methods, which can guarantee the real-time and time certainty of computer service processing.

Key words: endogenous security computer; weaponry; distribution agent; voting service; feedback control

1　引言

网络空间作为继陆、海、空、天之后的第五维空间，已成为信息时代国家间博弈的新舞台和战略利益拓展的新疆域。[1] 习总书记指出“网络安全和信息化是一体之两翼、驱动之双轮，必须统一谋划、统一部署、统一推进、统一实施”。“新基建”作为服务于国家长远发展和“两个强国”

建设的战略部署，以5G、数据中心/云平台、人工智能、物联网、工业互联网等为代表的新一代信息基础设施，软硬件产品丰富，软硬件代码数量庞大，复杂性高，不可避免地会出现各种各样的设计缺陷。另外，由于产业链整体不可控，以信息系统为依托构建的新一代基础设施被安装后门或预置恶意代码不可避免，基于漏洞后门问题（称之为“内生安全问题”）的蓄意攻击依然是“新基建”，特别是武器装备信息系统面临的最重要的网络安全威胁。

针对目前主动防御的不足，我国科学家提出拟态防御的思想，它是一种有效的新型主动防御技术[2-3]，能通过系统内生的机制对网络攻击达成事前的有效防御。拟态防御基于DHR（Dynamic Heterogeneous Redundancy，DHR）架构[4]可以实现期待的内生安全，旨在提高攻击成功难度，解决系统在“有毒带菌”环境下仍能正确有效运行的问题，深入研究面向信息系统的革命性主动防御创新体系架构和机制，对于确保国家重要信息系统安全具有重大的意义。

本文从拟态技术应用的角度出发，综合当下微妙复杂的国际形势对自主可控、安全可信技术的迫切需求，以及武器装备控制系统计算机功能需求和改善挑战，提出了一种面向武器装备的内生安全计算机设计方案。其中关键模块设计包括分发代理模块、控制执行体模块、裁决服务模块和反馈控制模块。通过各个模块的设计和最终集成，实现武器装备控制计算机的内生安全，为武器装备计算机的自主可控和高可靠发展提供借鉴思路。

2　研究背景与现状分析

2.1　国际形势分析

近年来，美国等西方国家已经把实施先发制人的网络战、夺取网络空间优势乃至制网权作为制胜的关键，并以此展开网络战演习和部署。我国网络空间面临着挑战，隐患就在身边，威胁近在眼前，安全形势十分严峻。

世界范围内发生过多起由于对武器装备控制系统安全考量不完善引发的恶性事故。2019年《纽约时报》报道，美国网络司令部针对伊朗发动网络攻击，主要目标是伊朗情报部门和导弹发射系统，成功的瘫痪了伊朗导弹发射控制系统，严重威胁伊朗国家安全。英国皇家战略研究所对全球武器系统进行了调研，指出目前的武器系统指令、控制和交流设施逐渐实现数字化，但是部分系统由于设计缺陷，以至于系统存在许多明显的安全漏洞，这类系统对于黑客等攻击基本没有抵御能力，由于部分武器装备基地使用的还是固网同性和局域网通信协议等，曾有消息称战机数据连接曾被黑客劫持。根据美国国防部监察长的说法，美国弹道导弹防御系统技术信息所在设施的机密网络易受大量内部和外部网络攻击，其中安全问题包括身份验证不统一、网络漏洞潜伏、缺乏对可移动存储媒介的机密数据监控、缺乏入侵检测和防病毒功能等多重因素。

网络安全是信息产业全球化背景下武器装备系统面临的巨大挑战，当前的基本态势是“攻易守难”，漏洞的普遍性、后门的易安插、网络空间构架基因的单一性和攻防双方的不对称性使得安全漏洞无法被根除且容易被利用，从而导致武器装备系统由于漏洞引发故障。中国科学家邬江兴院士先后提出了“结构适应应用”的拟态计算技术、“结构适应安全”的拟态防御技术以及以软件定义互连和软件定义节点为特征的新一代软件定义体系结构[5]，即拟态技术。其在普适性网络安全架构、高效能处理以及异构网络互联等方面从理论、技术和技术应用等方面均取得了突破性的进展。拟态技术以融合先进性与可信性、安全性与开放性、高可靠与高效能为一体，作为我国可控的颠覆性技术，成为网络空间安全新型防御技术中的新期待。

2.2　研究现状分析

美国研究的基于动态特征的移动目标MTD技术[6]是一种典型的主动防御方法。它作为一项重要的网络空间主动防御技术，其主要防御机制则体现为随机化机制[7][8]、多样化机制[9]、动态化机制[10-12]和共生机制[13]。MTD技术虽然能够在不依赖或较少依赖攻击特征的情况下增加漏洞利用难度或者使攻击效果不确定，但无法给出防护等级的量化参考，而且对基于目标对象后门或恶意代码的攻击在理论上就无效。因此MTD技术对于基于安全的武器装备计算机设计并没有很好的解决方案。

国内武器装备安全最初以保护武器装备指令安全为主要目标，武器装备安全指令接收机接收地面安控设备发出的安全指令，经解码、识别后将指令送到安全控制器，由控制器输出执行指令，完成地面信息指令任务。信息指令一般为二进制数字编码指令， 以增加抗干扰能力，而且信息系统具有较强的变换指令功能，通常备有数百种指令，在武器临射前进行选择。指令选定后装订入信息指令接收机，然后和信息发射机对接，进行指令产生、传输和检出的检验。

目前在基于安全防护的计算机设计方面的研究有：

参考文献［14］提出了一种计算机结构被公开用于实现防黑客计算装置。其设计所提的计算机结构主要是将各种计算机设备的计算装置设计成两个分区。通过物理隔离实现防黑客攻击。

参考文献［15］ 提出了一种内生安全的工业控制网络动态防御方法，其特征在于，通过对加密算法进行动态可重构，并结合密钥及认证口令的动态变化，在网络传输层对IP报文进行安全重构，以在节点设备之间建立内生安全的专属数据传输信道。

拟态防御技术是我国科学家提出的新型主动防御技术，其核心是从架构层面实现动态、异构、冗余的内生安全设计，从而保证系统在收到后门、漏洞威胁攻击时能够正常运行，不受影响。

本文以武器装备地面系统功能和信息安全为需求背景，以传统武器装备安全方法为基础，将内生安全的思想融入到计算机架构设计中，研究面向武器装备的内生安全操控计算机，从架构层面解决了面向武器装备控制计算机的内生安全问题，通过分发、裁决模块和反馈控制模块的协同作用实现即便出现异常情况下依然不影响武器装备控制指令的收发，为武器装备系统提供强有力的保障，对现代战争提供高可靠支持。

3　内生安全控制计算机设计方案

3.1　内生安全控制计算机架构设计

针对当前国际形势和网络安全研究现状，结合传统武器装备系统实时性差和安全性不高等问题，本文提出了一种基于C/S架构的内生安全计算机设计方案。

武器装备控制系统的总体框架如下图所示。其系统由指挥调度分系统、数据分发分系统、内生安全控制计算机、地面密码分系统、卫星通信系统多个模块组成。其中，指挥调度台生成作战任务，并以文件形式下发给内生安全控制计算机。数据分发设备作为信息交换、中转和分发的枢纽，与指挥调度、控制计算机、密码控制等分系统进行交互，为各分系统和设备之间提供了信息和文件交换的高速通道。内生安全控制计算机作为武器装备状态控制的终端，提供武器装备状态监视、轨迹查看等功能，并能接收来自指挥调度台的指令，对武器装备进行操控（改变航迹、取消等）。因此控制计算机是整个武器装备控制系统中不可或缺的关键模块，如图1所示为武器装备系统总体框架图，其中虚线框内部分为本文研究内容。对其进行内生安全改造设计，以保证武器装备的高安全性和命令执行实时性，对于整个武器装备控制系统的高可靠高可信具有关键意义。

本文便是基于武器装备控制系统的需求和拟态防御的思路，采用C/S架构和嵌入式技术，在内生安全防御体系设计上，利用分发代理、裁决服务与反馈控制，支持武器装备的异构控制执行体请求分发、多模裁决、清洗恢复等内生安全防御机制。本文设计的面向武器装备控制的内生安全计算机包括分发代理、异构执行体、裁决服务、反馈控制等模块，其通过交换机部署在同一集成环境中。如下图所示为该计算机关键模块的物理框图：

分发代理、多模裁决服务及反馈控制模块框架图如下图所示。

3.2　分发代理

分发代理软件包括网络数据收发、帧数据签名、数据缓冲和帧数据转发等功能模块。网络数据接收模块接收来自上游的数据报，并提交至数据缓冲队列进行缓冲，通过帧数据签名进行统一编码和签名后，复制为三份分别提交给各控制执行体。

分发代理模块负责维护系统内所有设备的编码、域名、地址、转发通道、转发端口等信息，形成系统的整体路由表单。其次，在接收到来自端口的数据报文/文件后，分析报文头检索出目的用户编码，同时查询路由表检查目的用户地址、

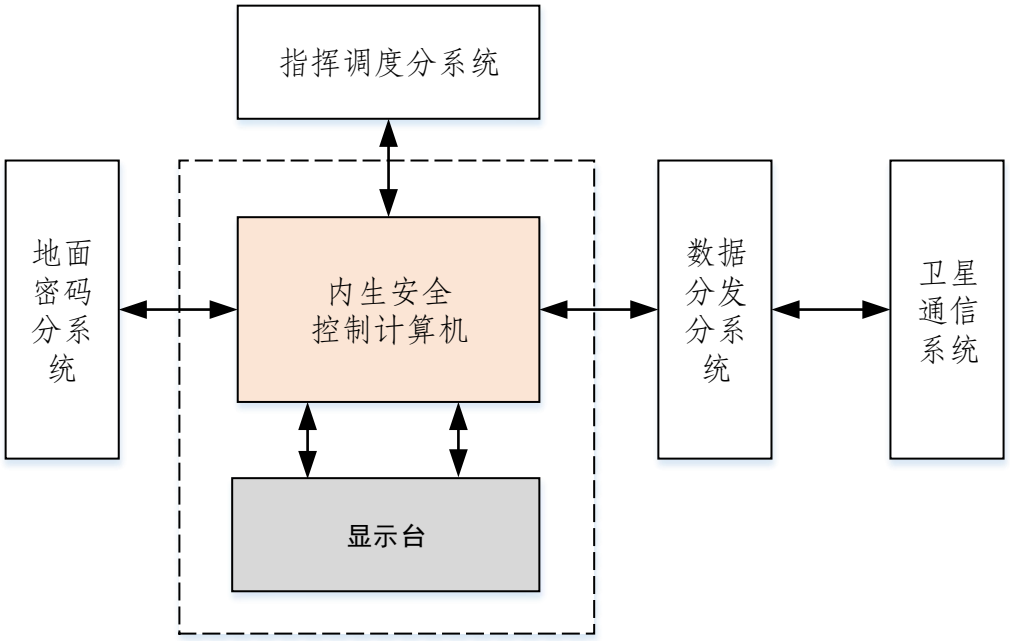


图1　武器装备控制系统总体框架图

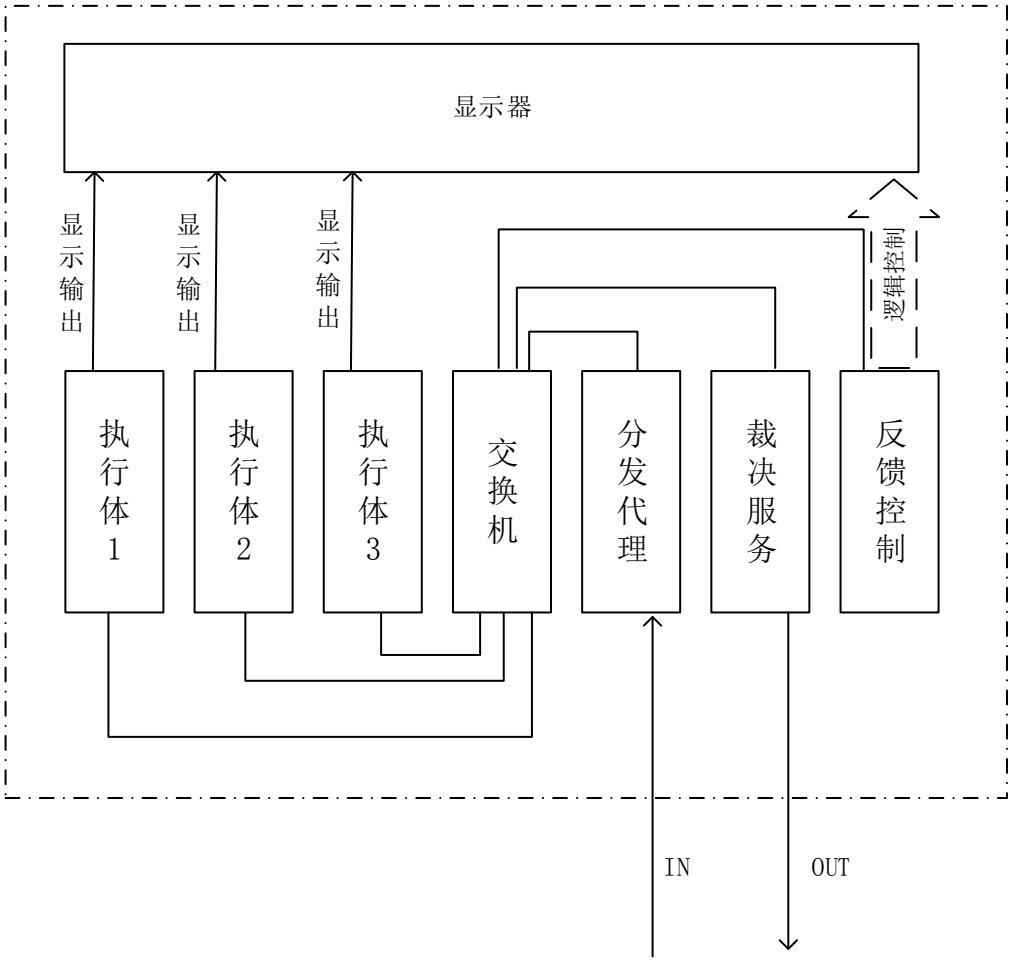


图2　内生安全控制计算机系统物理框图

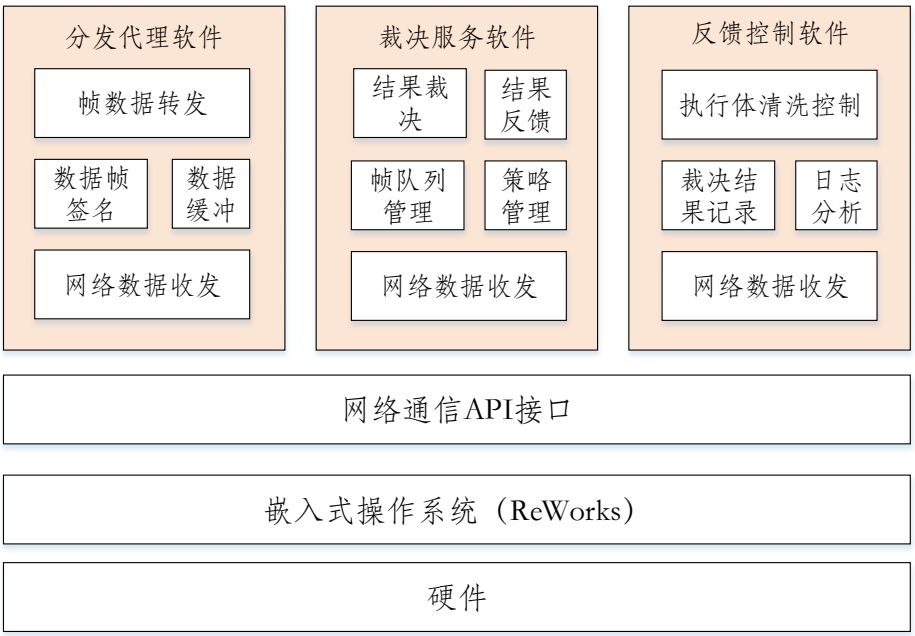


图3　内生安全控制计算机系统架构图

转发通道和转发端口，并按照要求转发出去。除此之外，分发代理能够针对转发报文的内生安全防护，通过应用软件的异构化设计形成多副本，在接收到需要转发的报文时独立执行数据解析，完成路由表查询。

3.3　控制执行体

顾名思义，控制执行体主要提供组成内生安全控制计算机异构特征的控制执行体。分发代理可将来自分发设备、外接设备的战争计划、武器装备控制数据与指令以一转多的形式传递给多个武器装备控制执行体，经过数据加解密、数据解析等过程后，异构执行体将数据响应、指令响应状态、上下文状态转发至裁决服务，裁决服务基于多数表决机制输出正确的结果。

控制执行体是内生安全控制计算机实现内生安全的关键模块，通过异构构造的执行体并行工作同时出现问题的概率十分小的理论现实来实现面向武器的计算机内生安全控制。

3.4　裁决服务

裁决服务软件包括网络数据收发、帧队列管理、策略管理、结果裁决和结果反馈等功能模块。裁决服务软件接收到来自应用执行体的数据并提交至帧队列进行缓冲，分析签名，得到统一签名的三个数据报后，依据裁决策略进行结果比对并输出正确结果，同时将裁决记录上报给反馈控制服务软件。若裁决服务发现系统存在多个执行体响应数据不一致情况，将依据裁决设备预设策略，调用反馈控制软件进行异步清洗恢复处理。

对于本文提出的分发代理和裁决服务，均为软硬一体化的模块，并非纯软件实现，本身具有较高的可靠性。

3.5　反馈控制

反馈控制软件包括网络数据收发、裁决结果记录、日志分析、执行体清洗控制等功能模块。反馈控制软件主要记录在运行过程中由裁决服务软件上报的结果，并记录为日志，通过日志比对和分析判断应用执行体是否为异常，并针对异常应用执行体发起清洗控制指令，为内生安全应用系统提供自愈反馈能力。同时，本发明的一大特点是通过将显控、业务等不同模块分离，大大提高系统的安全性。

对于反馈控制模块，本文设计的面向武器装备内生安全计算机，针对武器装备控制计算机多路显示与控制的情况，特别设计了显示切换功能，提供多路屏幕显示到单路显示的控制，输出控制有裁决服务根据判决结果选择。用户针对屏幕的操控输入由显示切换发送至分发代理，并提交至目前正在运行的执行体，以保证所有执行体在运行过程中的状态一致性；如遇到目前在显示的执行体存在输出结果异常情况，则自动执行显示切

换工作，保证武器装备控制计算机在执行过程中正确的显示与操控。

3.6 内生安全计算机工作流程

本文设计的面向武器装备控制计算机信息流图如下所示：

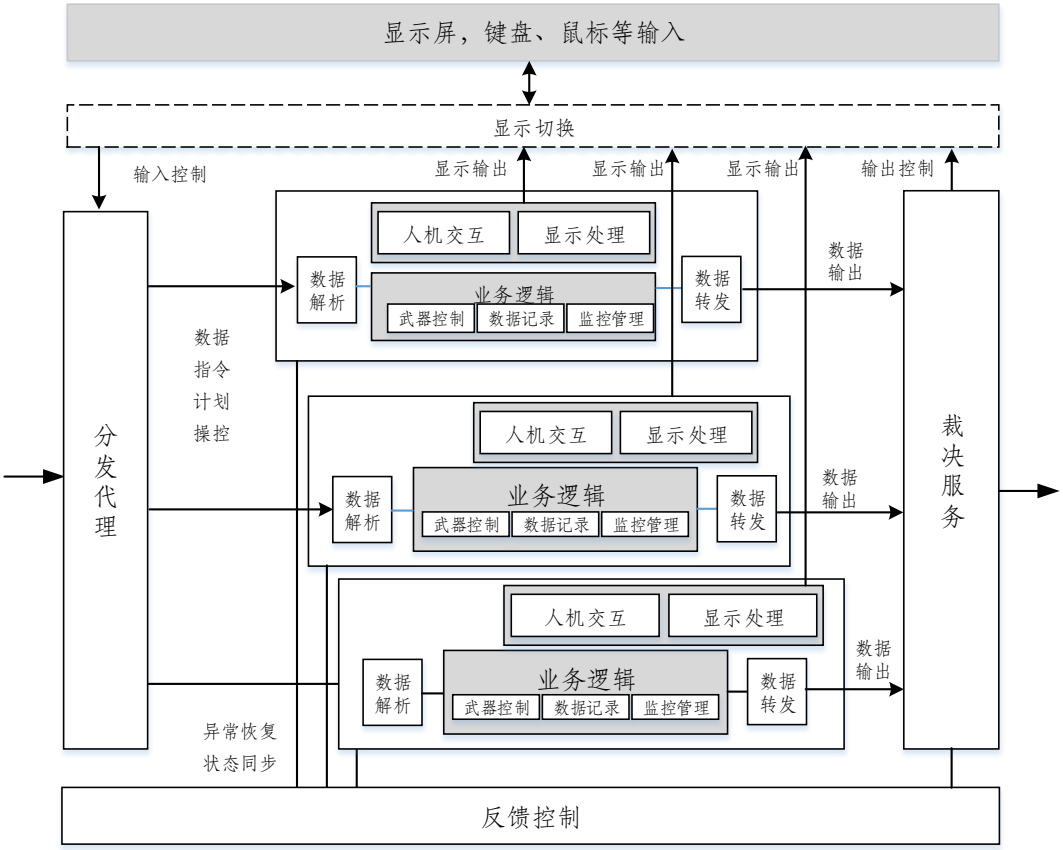


图4 武器装备控制内生安全计算机信息流图

如上图所示的面向武器装备控制内生安全计算机的一个完整流程可概述为如下步骤：

（1）通过显示器、键盘、鼠标等输入发送控制命令给分发代理模块；

（2）分发代理接收到相应的武器装备控制命令请求后对系统的路由表单进行更新维护，准备数据指令的计划操控工作；

（3）在接收到来自端口的数据报文/文件后，分析报文头检索出目的用户编码，并查询路由表检查目的用户地址、转发通道和转发端口，并按照要求转发出去；

（4）针对转发报文的内生安全防护，通过应用软件的异构化设计形成多副本；

（5）各执行体在接收到需要转发的报文时独立执行数据解析，按照各自的执行逻辑进行武器装备控制和数据记录及监控管理；

（6）各武器装备控制执行体经过数据加解密、数据解析等过程后，第一时间将数据响应、指令响应状态、上下文状态转发至裁决服务；

（7）裁决服务软件接收到来自应用执行体的数据并提交至帧队列进行缓冲，分析签名，得到统一签名的三个数据报；

（8）对于分析签名后得到的三个相同格式的数据报，依据裁决策略进行结果比对并输出正确结果，同时将裁决记录上报给反馈控制服务软件；

（9）若裁决服务裁决一致，则系统正常输出武器装备控制命令；显示器显示正常情况下的各执行体状态及武器装备控制信息。若裁决服务发现系统存在多个执行体响应数据不一致情况，则转入（10）；

（10）若裁决服务发现系统存在多个执行体响应数据不一致情况，将依据裁决设备预设策略，

调用反馈控制软件进行异步清洗恢复处理。此时将异常执行体情况输出显示于显示器；并针对其实行实时的清洗恢复。

对于上述内生安全控制计算机，能够支持处理器和操作系统的3种异构；同时支持多异构体的部署、运行、判决、协同执行和资源调度能力。能够提供硬件、操作系统及应用系统各层未知漏洞后门的安全防御能力。另外，初步验证其分发裁决引入的时延开销不大于100毫秒；自愈能力也在可接受范围内。单执行体输出异常时，结果检测到在不计算应用执行体启动时间的情况下，清洗恢复时间不大于200毫秒。在白盒测试条件下，单执行体受攻击情况下的成功防御概率100%。说明该方案能够在时延开销可接受范围内实现防御效果。

4 结束语

本文提出的面向武器装备的内生安全计算机设计，基于武器装备控制系统中武器装备控制台的监视、调度等功能需求，结合拟态防御新型安全思路展开研究。每台内生安全计算机由分发代理模块、裁决服务模块、反馈控制模块、控制执行体以及接口等组成，各模块间功能清晰，相互独立，被赋予了良好的通信和管理功能并提供友好的接口，其中分发代理、裁决服务和反馈控制均采用嵌入式方式实现模块集成，同时反馈控制模块多路显示与控制的情况，设计显示切换功能，提供多路屏幕显示到单路显示的控制，输出控制有裁决服务根据判决结果选择。用户针对屏幕的操控输入由显示切换发送至分发代理，并提交至目前正在运行的执行体，以保证所有执行体在运行过程中的状态一致性；如遇到目前在显示的执行体存在输出结果异常情况，则自动执行显示切换工作，保证武器装备控制计算机在执行过程中正确的显示与操控。

参考文献:

[1] 陈森, 肖寿高. 美国网络空间战略之"变"[J]. 环球军事，2015(20):55-57.

[2] 邬江兴. 网络空间拟态防御研究[J]. 信息安全学报,2016,1(04):1-10.

[3] 邬江兴. 网络空间拟态安全防御[J]. 保密科学技术,2014(10):4-9+1.

[4] 邬江兴. 网络空间拟态防御原理[M]. 北京：科学出版社，2018.

[5] 吕平,刘勤让,邬江兴,陈鸿昶,沈剑良. 新一代软件定义体系结构[J]. 中国科学:信息科学,2018,48(03):315-328.

[6] Lei C, Zhang H Q, Tan J L, et al. Moving target defense techniques: A survey[J]. Security and Communication Networks，2018，2018.

[7] Lu K, Song C, Lee B, et al. ASLR-Guard: Stopping address space leakage for code reuse attacks[C]//Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM，2015：280-291.

[8] Wang Q, Wang C, Li J, et al. Enabling public verifiability and data dynamics for storage security in cloud computing[C]//European symposium on research in computer security. Springer，Berlin，Heidelberg，2009：355-370.

[9] Wang L, Zhang M, Jajodia S, et al. Modeling network diversity for evaluating the robustness of networks against zero-day attacks[C]//European Symposium on Research in Computer Security. Springer，Cham，2014：494-511.

[10] Sengupta S, Vadlamudi S G, Kambhampati S, et al. A game theoretic approach to strategy generation for moving target defense in web applications[C]//Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems. International Foundation for Autonomous Agents and Multiagent Systems，2017：178-186.

[11] Lei C, Ma D H, Zhang H Q. Optimal strategy selection for moving target defense based on Markov game[J]. IEEE Access，2017，5：156-169.

[12] Zangeneh V, Shajari M. A cost-sensitive move selection strategy for moving target defense[J]. Computers & Security，2018，75：72-91.

[13] Cui A, Stolfo S J. Defending embedded systems with software symbiotes [C]// International Workshop on Recent Advances in Intrusion Detection. Springer，Berlin，Heidelberg，2011：358-377.

[14] 弗兰克·N·纽曼;丹·纽曼. 防黑客计算机设计:中国. CN110337651A[G06F21/71]. 20191015.

[15] 中国工程物理研究院计算机应用研究所. 内生安全的工业控制网络动态防御方法:中国. CN107065750A [G05B19/05]. 20170818.

[作者简介]

霍立田（1994—），女，硕士，工程师，主要研究方向：网络安全防御。

余新胜（1979—），男，硕士，高级工程师，主要研究方向：网络安全防御，信息安全，信息系统架构。

罗论涵（1985—），女，博士，中级工程师，主要研究方向：网络安全防御，信息安全。

解维（1986—），女，硕士，中级工程师，主要研究方向：网络安全防御，信息系统架构。

基于区块链的 FICS 工控系统安全研究

薛镭，邹涛，吴少勇，林会肖，杨汶佼，王延松，张汝云
之江实验室，310023

摘　要：针对工业控制系统中组态服务器和数据库服务器中心化面临的单点故障、数据恶意篡改等问题， 利用区块链技术去中心化、防篡改、身份认证、数据加密等特性，提出一种基于 Ceph 分布式文件存储与基于 Fabric 联盟区块链技术相结合的工控系统安全解决方案 FICS（Fabric based Industry Control System）。利用 Ceph 分布式文件存储解决服务器中心化问题，利用 Fabric 联盟区块链技术解决数据恶意篡改、身份认证、数据加密等安全性问题，同时使用 FICS 智能合约对节点行为实行监管。FICS 解决方案有效地解决了工业控制系统中服务器中心化、数据可篡改、行为不可追溯及缺乏合约监管等问题。

关键词：联盟区块链、工业控制安全、分布式文件存储、智能合约

Research on FICS industrial control system security based on block chain

——XUE Lei, ZOU Tao, WU Shaoyong, LIN Huixiao, YANG Wenjiao, WANG Yansong, ZHANG Ruyun (Zhejiang Lab, Hangzhou, 310023, China)

Abstract: Aiming at the problems of single point of failure and malicious data tampering in the centralized configuration server and database server of industrial control system, a security solution FICS (Fabric) based on the combination of Ceph distributed file storage and fabric alliance block chain technology is proposed by using the characteristics of decentralized block chain technology, anti tampering, identity authentication and data encryption based Industry Control System). Ceph distributed file storage is used to solve the problem of server centralization, and the Fabric alliance block chain technology is used to solve the security problems such as malicious data tampering, identity authentication and data encryption. At the same time, FICS smart contract is used to supervise node behavior. FICS solution can effectively solve the problems of server centralization, data tampering, behavior traceability and lack of contract supervision in industrial control system.

Key words: Alliance Block Chain; Industrial control safety; Distributed File System; Smart contract

1　引言

工业控制系统作为包括电力、军工、水电、石油、天然气、化工、交通运输、药品制造、加工厂（食品、饮料、纸质）和高端制造业（汽车、航空和耐用品）等行业广泛使用的信息化控制系统，是涉及国民经济及公共安全的关键信息基础设施，同时也是非法入侵者恶意攻击的重点目标。因此，工业控制系统的信息安全防护尤为重要，增强工业控制系统的安全性要求也日益迫切。

区块链是以比特币为代表的数字加密货币体系的核心支撑技术。区块链的核心技术优势是去中心化，

通过运用数据加密、分布式记账、共识机制等手段，实现基于点对点网络的数据交易。

本文首先分析区块链技术的安全性、自治性、可追溯、不可篡改等特性，并进一步探讨区块链技术在工业控制领域中的应用。目的是探索解决传统工业控制系统中的网络安全、数据可篡改、服务器中心化等问题，从而使工业控制系统升级成为网络更安全、行为可追溯、数据更可信的新型工业控制系统，更好支撑制造业数字化、网络

基金项目：浙江省市场监督管理局“数字经济标准化试点项目”（项目编号：ZJCT5-2019063）。

化、智能化发展。

2　工控系统现状与区块链分析

工业控制系统是由各种自动化组件、过程监控组件共同构成的以完成实时数据采集、工业生产流程监测控制的管控系统，包括过程控制、数据采集系统（SCADA）、分布式控制系统（DCS）、程序逻辑控制（PLC）以及其他控制系统等。

2.1　工控系统现状

工业控制系统应用于现代大型企业规模生产与运营控制，经过多年积累和优化，系统层级划分成熟，典型的工业控制系统网络结构如图 1 所示：

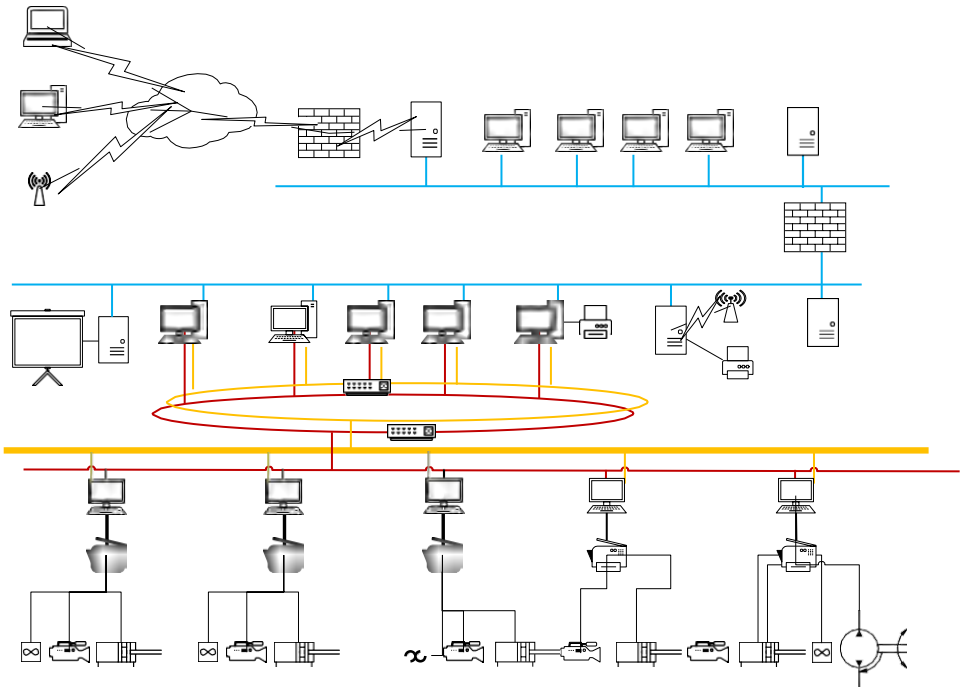


图 1　工业控制系统网络架构图1）生产管理层（MES 系统）：网络集中式管理；

0M/100

生 远程维护**产**管**理** 远程监控**层** WEB 信息服务器 RMIS 系统 ERP 系统 MES 系统 MCAS 系统 数据库服务器以太网 10M/100 防火墙 M/1000MGPS 无线通讯过程监控操作员站操作员站 操作员站 操作员站 工程师站 记录站防火墙 GPS 无线通讯以太网 10M/10 层大屏幕显示器现场**控** 控制站制控制站交换机控制站交换机 WEB 信息服务器双冗余容错光纤环网控制站 控制站**层** 现场采集卡现场设备**层** 传感器 传送器现场采集卡传感器传送器现场采集卡现场采集卡 现场采集卡现场总线执行器

0MTE1

2）过程监控层（DCS 系统、PLC 系统、SCADA 系统）：网络集中式监控；

3）现场控制层（RTU、DCS 控制器、PLC 控制器）：集中操作，分级管理，分布式控制；

4）现场设备层（变送器、执行器、传感器）：通过现场总线，工业协议现场控制。

组态管理软件是工业控制系统中的重要组成部分，既是过程监控层中的数据收集处理中心，也是生产管理层远程监视中心和数据转发中心。在组态管理软件的支持下，操作人员可完成：查看生产现场实时数据及流程画面；自动打印各种实时 / 历史生产报表；自由浏览各个实时 / 历史趋势画面；及时得到并处理

各种过程报警和系统报警；需要时，人为干预生产过程，修改生产过程参数和状态；与管理部门的计算机联网，为其提供生产实时数据等。

组态管理软件为用户提供了高可靠性实时运行环境和功能强大的开发工具。使用者可以利用组态软件，将各种功能软件进行适当的“组装连接”（即组态），便可极为方便地生成满足控制系统要求的应用系统，如图 2 所示。

2.2　工控系统安全性分析

工业控制系统安全风险主要包括工业协议隐患、操作系统漏洞、服务器中心化风险、组态管

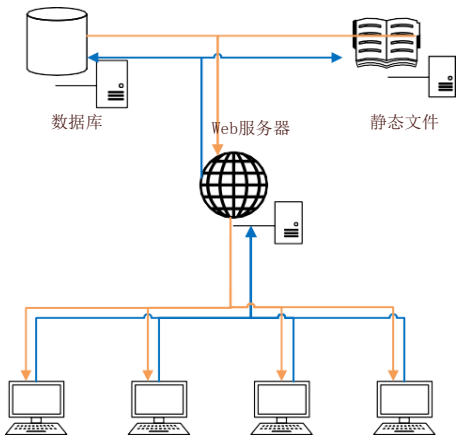


图 2　组态管理软件信息流图

数据流控制流操作员站 工程师站 操作员站 工程师站

理软件风险、安全策略风险等五个方面：1）目前 Modbus、DNP、ProfiBus 等主流的工业控制协议在设计时还都是主要考虑通讯的可用性和实时性，对安全性普遍考虑不足，缺乏身份认证、数据加密保护等机制；2）数量庞大的操作员站、工程师站安装 Windows 操作系统和 Linux 系统，网络管理员出于对保障企业日常生产的稳定运营考虑，担心影响效率，基本不对操作系统进行版本和补丁升级，导致系统中各类站点存在大量的安全漏洞；3）大量的运营数据和现场控制文件集中保存在服务器上，一旦服务器遭到网络攻击，数据被篡改，将导致整个工业控制系统瘫痪，给企业带来无法估量的损失。另外，地震、强风、暴雨等自然灾害也是影响工业控制系统物理安全的重大威胁，易造成设备损毁、网络瘫痪、数据丢失等工业事故；4）组态管理软件中，很多登录界面用户密码安全强度低，服务器端缺乏身份认证措施。SQL 注入式、DoS 攻击、伪装身份等攻击方式极大地威胁着组态管理软件及后台数据库服务器数据的安全。且数据被篡改或丢失后的恢复能力不足，数据可信度不高。另一方面，工业控制系统中由于组态管理软件很少有安全测试和审计，系统出现故障除了更新很难进行取证；5）安全策略及技术标准不够完善，信息管理员及系统操作人员安全意识不足也是很大风险。

当前，工业协议隐患、操作系统漏洞等工控系统风险可以采用异构冗余、执行裁决等技术实现工控系统内生安全拟态防御思想，增强系统安全性；安全策略也可以根据企业相关安全管理规范制订相关规章制度，对企业生产运营人员进行安全培训，提高安全意识。而工控系统服务器中心化及数据可篡改、组态管

理软件缺乏行为监管等几种安全风险有效解决方案尚需进一步探索与实践。**2.3 区块链分析**

区块链是源自比特币，由网络中不同节点共同维护，使用密码学保证传输和访问安全，能够实现数据一致存储、难以篡改的分布式账本技术。按照系统是否具有节点准入机制，区块链可分类为联盟链、私有链、公有链。在联盟链和私有链中，节点加入退出需要系统的许可；而在公有链中，节点可以随时自由加入和退出。从技术角度看，区块链技术以 P2P 网络技术、分布式账本技术、非对称加密、共识机制技术和智能合约技术等五大技术作为技术支撑：

1） P2P 网络技术，根据其路由查询结构可以分为四种类型，分别是集中式、纯分布式、混合式和结构化模型。这四种类型也代表着 P2P 网络技术的四个发展阶段。比特币采用的是混合式模型，而现今公链大多采用的是结构化模型。在结构化网络的具体实现上，大都采用 DHT（Distributed Hash Table，分布式哈希表）算法的思想。基于 DHT 算法思想的具体实现方案有 Chord、Pastry、CAN 和 Kademlia 等算法。

2） 分布式账本技术，本质上是一种可以在多个网络节点、多个物理地址或多个组织构成的网络中进行数据分享、同步和复制的去中心化数据存储技术。

3） 非对称加密，采用椭圆曲线加密算法（Elliptic Curve Cryptography，简称 ECC），应用场景主要包括信息加密、数字签名和登录认证等：

4） 共识机制技术，目前较为主流的共识算法有 POW （Proof Of Work）、POS（Proof of Stake）、DPOS （Delegated Proof of Stake）、PBFT（Practical Byzantine Fault Tolerance）等，每种算法在实际应用中时有各自的优缺点。因此，在不同的应用场景中，区块链将会采用不同的共识算法。

5） 智能合约技术，智能合约程序不只是一个可以自动执行的计算机程序，它本身就是一个系统参与者，对接收到的信息进行回应，可以接收和储存价值，也可以向外发送信息和价值。这个程序就像一个可以被信任的人，可以临时保管资

产，总是按照事先的规则执行操作。

从安全优势上看，区块链技术使用全新的加密认证技术和去中心化共识机制去维护一个完整的、分布式的、不可篡改的账本，让参与者在无需相互认知和建立信任关系的前提下，通过一个统一的账本系统确保信息安全。区块链在点对点网络上由许多分布式节点和计算机服务器来支撑，任何一部分出现问题都不会影响整体运作，而且每个节点都保存了区块链数据副本。所以区块链可提供极高的业务连续性、可靠性、容错性，能够有效预防故障与网络攻击。此外，由于所有文件都能够以代码或分类账的形式体现，通过对区块链上的数据处理程序进行设置，智能合约及自动交易就可能在区块链上实现，从而提升业务自动化水平。

3　FICS 工控系统安全解决方案

针对现有工控系统中的服务器中心化及数据可篡改、组态管理软件缺乏行为监管等问题，综合考虑联盟链弱中心化、强可控性、强拓展性、交易速度较快等特点和 Fabric 可插拔实现各种功能的模块化架构，

具有强大的容器技术来承载各种主流语言的技术优势，我们设计了一个基于 Fabric 的工控系统安全解决方

案 ——FICS （Fabric based Industry Control System），其网络架构如下图 3 所示。

区块链

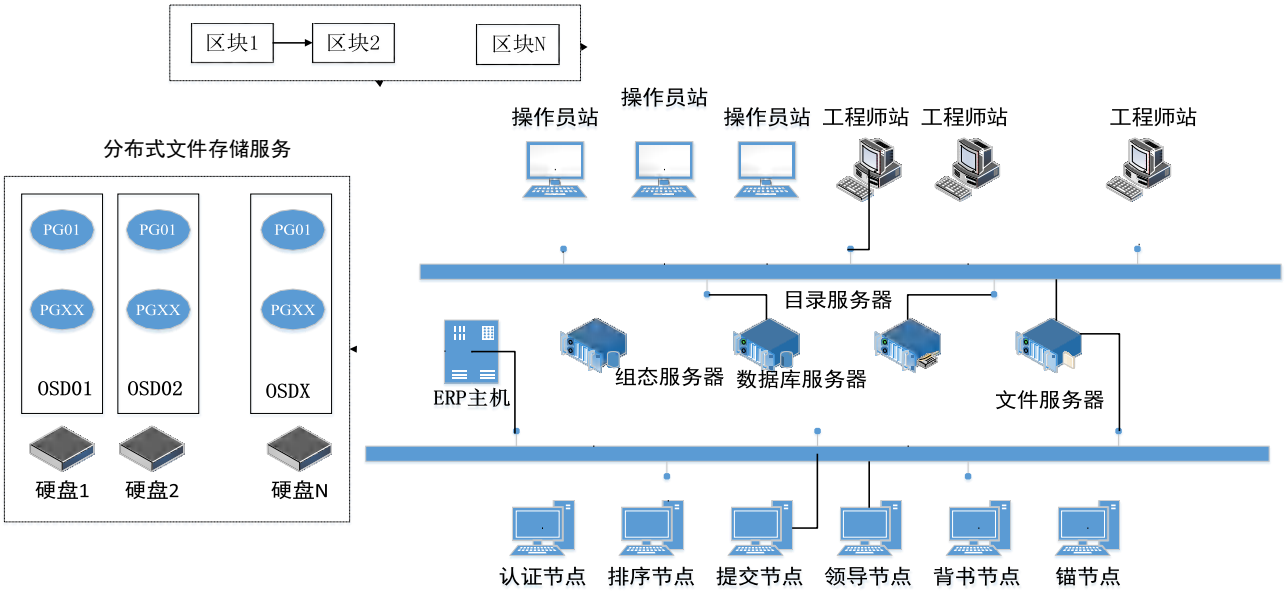


图 3　FICS 系统网络结构图

FICS 系统中包括了传统工业控制系统中的普通节点、承担区块链系统监管角色的特殊节点、分布式文件系统及区块链组成

其中，普通节点包括：组态服务器、数据库服务器、文件服务器、工程师站、操作员站、目录服务器等；承担区块链系统监管角色的特殊节点包括以下类型：

背书节点（Endorser）：主要提供 Process Proposal 方法供客户端调用，完成对交易提案的背书处理；每一个安装了智能合约的节点，都可以成为背书节点。

提交节点（Committer）：负责维护区块链和账本结构；

排序节点（Orderer）：负责区块链排序；

认证节点（CA）：负责网络中所有证书的管理，实现标准的 PKI（Public Key Infrastructure）架构。

领导节点（Leader）：一个机构可能在某个通道中有多个节点，其中只需要一个领导节点来接收交易，然后由它负责将交易分发给其他节点。

锚节点（Anchor）：用于机构之间的通信，它使得不同机构间的对等节点了解彼此的存在。

上述特殊节点是 FICS 系统实现网络监管功能的必要构成节点。

3.1 FICS 安全访问机制

FICS 普通节点成员身份认证基于标准的 X.509 证书，采用 PKI 体系为每个成员生成数字证书以标识用户身份。FICS 利用 PKI 体系发布数据证书，结合 MSP（Membership Service Provider）组件进行身份认证和权限控制。认证节点提供对工程师站、操作站、组态服务器、数据库服务器等节点用户登录和注册的数字证书管理功能。系统通过 MSP 标识检查身份证书有效性、证书路径检查是否存在用户证书到认证节

点的有效路径及 CRL 检查证书是否被吊销完成身份认证。

工程师站、操作站、组态服务器、数据库服务器等运营生产节点成员必须被许可才能加入网络，通过实体注册来获得长时间的，根据实体类型生成的身份凭证。在用户使用过程中，这样的证书允许交易证书颁发机构 TCA（Transaction Certificate Authority）颁发匿名证书。交易证书被用来对提交交易授权。交易证书存储在区块链中，并对审计集群授权；否则交易是不可链接的。FICS 系统中各节点必须注册到系统中，获得证书后才能进行区块链操作，如图 4 所示。

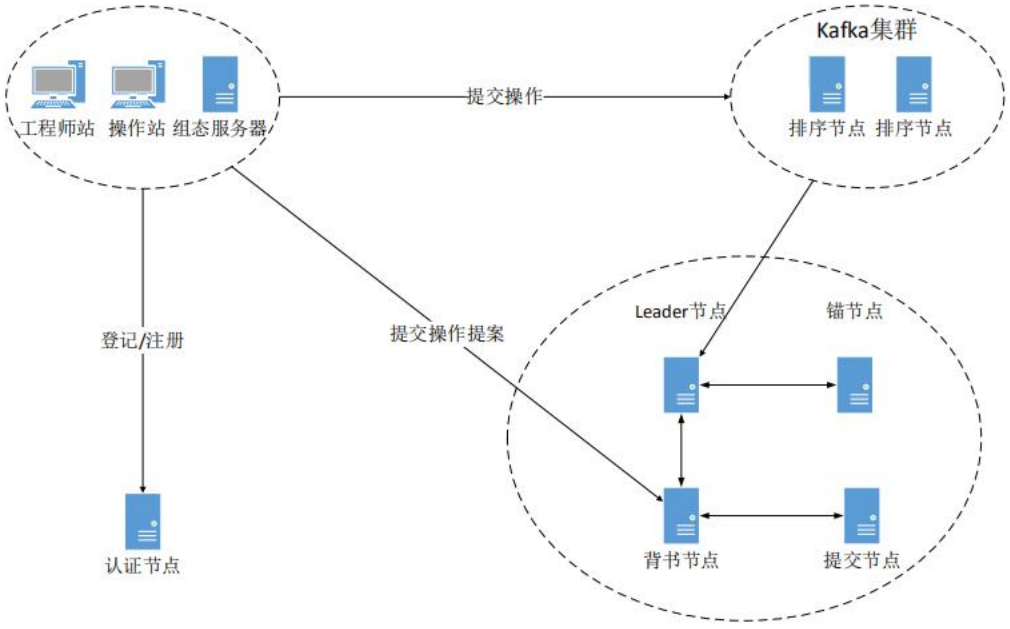


图 4　FICS 节点安全访问

3.2 FICS 监管可信机制

为了提高 FICS 系统的整体安全可信度，需要其中承担系统监管功能的特殊节点在接入网络时具备不可篡改的身份可信。为此，FICS 在认证节点、背书节点、排序节点、领导节点、锚节点等监管节点设备上加入了可信根，通过集成专用微控制器在软件栈初装或重启时对其进行静态度量和验证。先启动的软件对后一级启动的软件进行度量，实现基于硬件的物理可信的传递，如图 5 所示。

特殊节点启动时检测 BIOS 和操作系统的完整性和正确性，保障其硬件配置和操作系统没有被篡改过，所有系统的安全措施和设置都不会被绕过；在节点设备启动后，对所有的软件应用可进行实时监控，若发现应用被篡改立即采取保护措施。

3.3 FICS 文件账本管理

FICS 系统中每个对文件记账的节点都是互相复制的状态机，节点之间需要保持相同的账本状态。为了实现系统中节点状态的一致性，每个节点需要通过共识过程对账本状态的变化达成一致性的认同。

系统中账本的一致性共识过程包括 3 个阶段：背书、排序和校验，如图 7 所示。

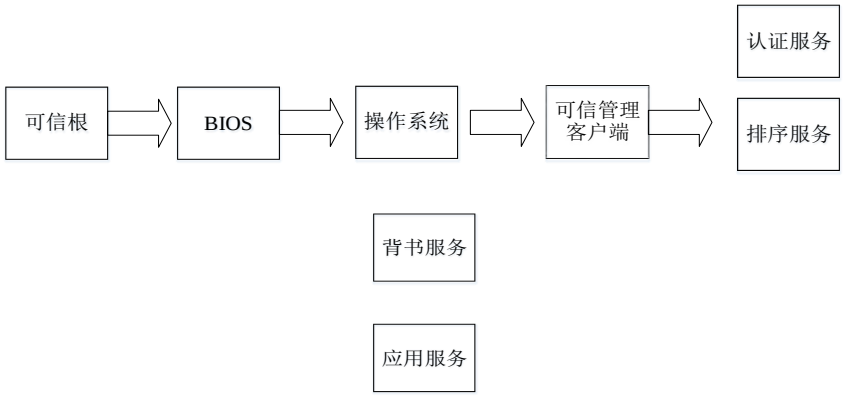


图 5　物理可信链

系统中工程师站 1 文件系统上有文件更新，通过客户端提交更新请求给背书节点。背书节点对文件更新请求合法性进行校验，如果校验通过背书节点对此提案签名后返回给工程师站 1。工程师站 1 将签名后的提案发送给排序节点，排序节点对一段时间内收到的提案打包并进行排序生成区块。排序节点通过消息分发给系统内所有记账节点，每个记账节点各自验证新账单并提交到本地账本中。经过验证，区块中的更改会被标记为有效或无效，通过事件返回给最先发起更新请求的节点客户端。

3.4　FICS 智能合约管理

FICS 中智能合约需要实现的业务逻辑主要包括：

1） 系统管理员节点可以对整个系统网络层级划分、域划分以及域内操作站、控制站的构建与管理。

2） 工程内域的划分，域内操作站、控制站的组建与管理，工程的内不同站点分组与角色设定的权限与配置。

3） 控制站下硬件组态管理，如何实现具体模块参数组态管理。其中包括参数分组、参数配置以及参数组态管理。

4） 用户程序编译管理和组态下载。根据控制站内部位号、数据结构、功能块等资源自定义用户程序， 控制站内软件资源的组态管理，用户程序的下载实现。

FICS 系统中的智能合约是对链代码的操作，如图 8 所示，交易分为部署和调用：

1） 部署智能合约指的是创建新的链代码，并且用一个程序作为参数，当一个部署交易成功执行时， 链代码就被安装到区块链上了。

2） 调用智能合约指的是运行链代码，链代码执行时可能会修改相应的状态，并返回输出。

4　FICS 分布式文件存储

图 8 FICS 智能合约管理

FICS 需要存储的数据包括生产管理层中的运营数据、过程监控层中的历史监控数据和组态管理数据、现场控制层中的控制程序等。数据集中存储即服务器中心化，一旦服务器遭到网络攻击或者宕机，整个系统将陷入瘫痪。因此，FICS 采用分布式文件存储，来解决数据集中存储带来的安全的风险。

当前，分布式存储有包括 Ceph、HDFS、Swift、GFS、Luster 等多种实现技术。存储根据其类型，可分为块存储，对象存储和文件存储。在主流的分布式存储技术中，HDFS/GPFS/GFS 属于文件存储，Swift 属于对象存储，而 Ceph 可支持块存储、对象存储和文件存储，故称为统一存储。几种主流分布式存储技术的特点比较如表 1 所示：

Ceph 相比其它分布式存储技术，其优势点在于：充分利用了存储节点上的计算能力，在存储每一个数

据时，都会通过计算得出该数据存储的位置，尽量将数据分布均衡。同时，由于采用了 CRUSH、Hash 等算法，避免了单点故障，且随着规模的扩大，性能并不会下降。目前 Ceph 已得到众多云计算和存储厂商的支持，成为应用最广泛

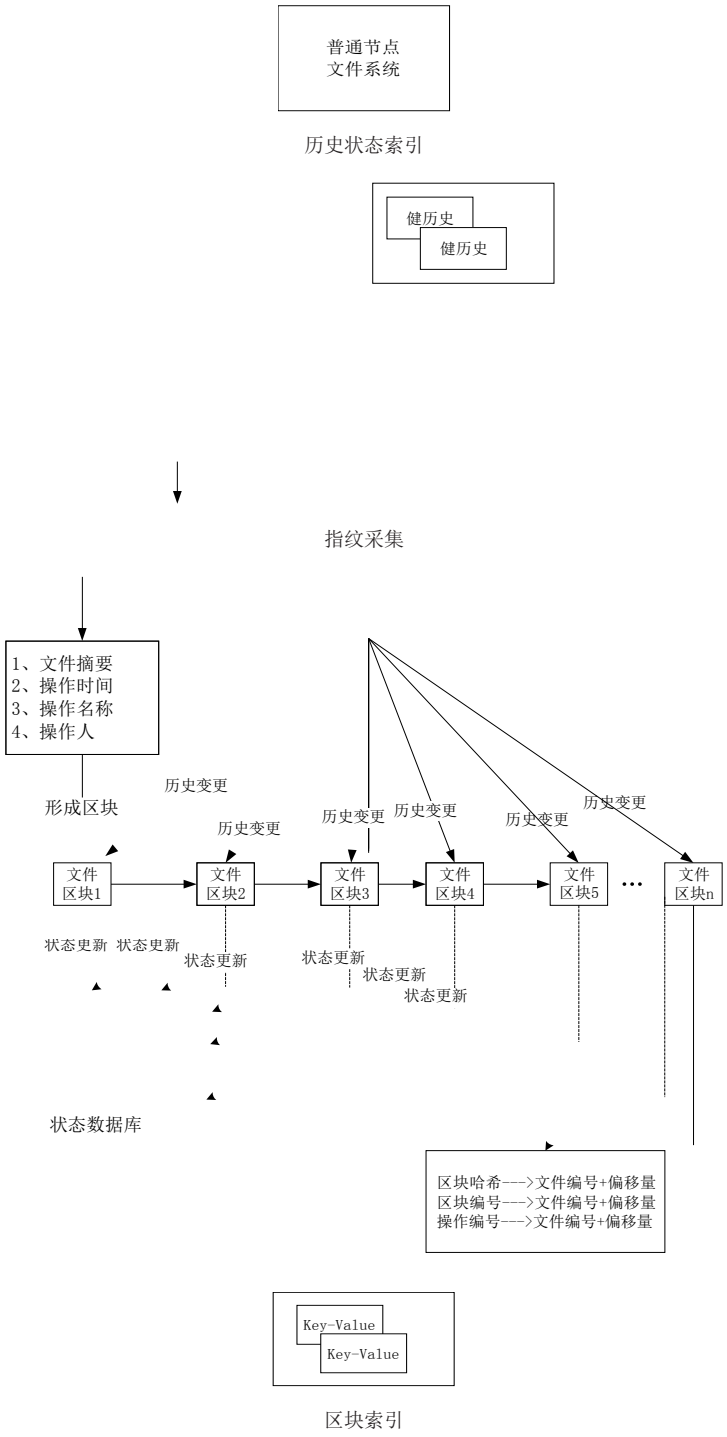


图 6　文件账本流程

的开源分布式存储平台。因此，由于 Ceph 支持多种数据存储方式，系统架构去中心化及数据强一致性存储等特点，FICS 系统采用基于 Ceph 构建分布式文件存储。FICS 中基于 Ceph 的分布式文件存储如图 9 所示。

Object1 Object2 ObjectN

FICS 中生产运营节点（工程师站、操作员站、组态服务器、数据库服务器等）安装 Ceph，调用接口将文件系统上的文件分成多个 Object 对象以分布式多副本方式存储到其他不同生产运营节点。

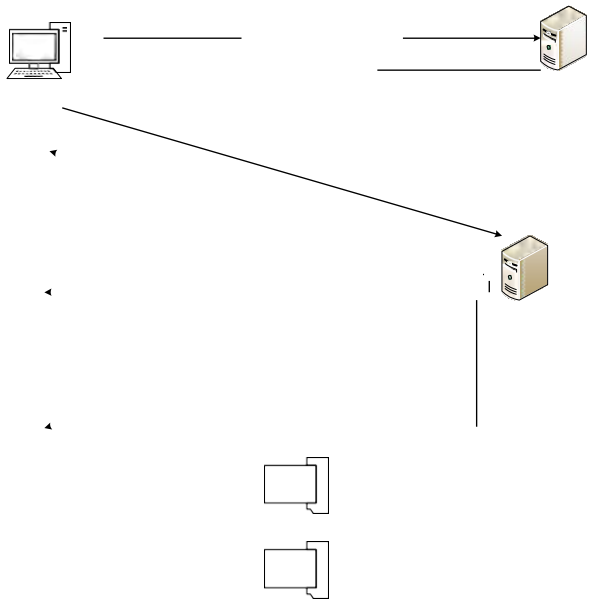
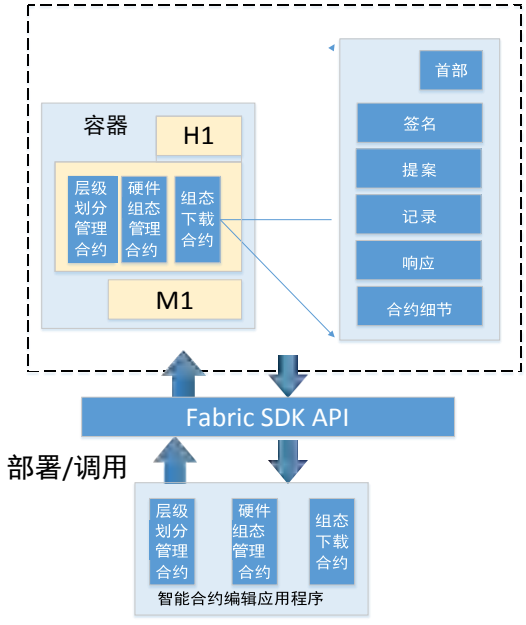


图 7　FICS 系统共识流程

工程师站N



FICS 分布式文件存储核心组件包括对象存储设备 OSD（Object Storage Device）、监视器 Monitor、元数据服务器 MDS（Meta Data Server）。OSD 主要完成数据存储、数据复制、数据平衡、数据恢复，并与其它 OSD 间进行心跳检查，将变化情况上报给 Monitor。OSD 中的放置组 PG（Placement Group）用于放置 Object 的一个载体，在 OSD 上的存在形式就是一个目录。Monitor 负责监视集群，维护集群的健康状态，同时维护集群中的各种 Map 图，比如 OSD Map、Monitor Map、PG Map 和 CRUSH Map。MDS 负责保存文件系统服务元数据。

FICS 网络中所有工程师站、操作员站、组态服务器、数据库服务器等生产运营节点每个分区初始化为OSD。文件系统中的文件被分割为多个 4MB 大小的 Object 对象，分别以多个副本的形式保存在其他节点的 OSD 上，每个 OSD 划分为多个 PG。由此，实现 FICS 中所有生产运营节点文件的分布式存储。

5　结束语

本文分析了工控系统的现状及其中的安全问题，研究了区块链的安全特性。针对工控系统中服务器中心化单点故障、数据恶意篡改等安全问题，提出了基于Fabric联盟区块链技术的工控系统安全解决方案，并结合Ceph分布式文件存储系统实现了工控系统生产运营节点文件的分布式存储，解决了服务器中心化问题，提升了数据安全性和行为可监管的自治性效果。此外，FICS技术方案可为研究制定基于区块链的工控系统技术实现标准提供参考。

表 1　几种主流分布式存储技术比较

分布式存储	Ceph	GFS	HDFS	Swift	Luster
平台属性	开源	闭源	开源	开源	开源
系统架构	去中心化	中心化	中心化	去中心化	中心化
数据存储方式	块、文件、对象	文件	文件	对象	文件
元数据节点数量	多个	1 个	1 个(主备)	多个	1 个
数据冗余	多副本/纠删除码	多副本/纠删除码	多副本/纠删除码	多副本/纠删除码	无
数据一致性	强一致性	最终一致性	过程一致性	弱一致性	无
分块大小	4MB	64MB	128MB	视对象大小	1MB
适用场景	频繁读写场景	大文件连续读写	大数据场景	云的对象存储	HPC 超算

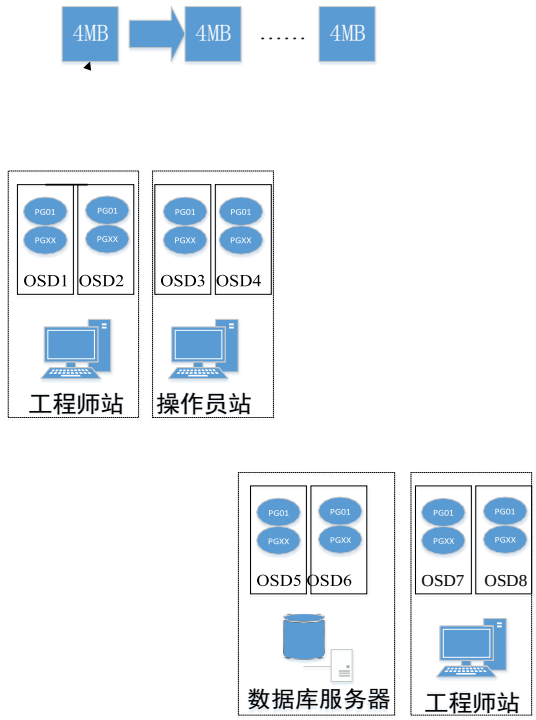


图 9　FICS 分布式文件存储

参考文献:

[1] 刘威,李冬,孙波. 工业控制系统安全分析. 第 27 次全国计算机安全学术交流会论文集,2012 年第 08 期.

[2] 王小山,杨安,石志强,孙利民. 工业控制系统信息安全新趋势. 信息网络安全,2015 年第 01 期.

[3] 赵勇,廖建华,沈昌祥. 基于访问验证的工业控制系统安全保障方法. 北京工业大学学报,2013 年 12 月,第 39 卷第 12 期.

[4] 伍锦荣. 工业控制系统网络安全现状及解决方案. 石油化工自动化,2017 年第 53 卷第 4 期.

[5] 张盛杰,顾昊旻,李祉岐,应欢. 电力工业控制系统信息安全风险分析与应对方案. 电力信息与通信技术,2017 年第 15 卷第 4 期.

[6] 邵诚,钟梁高. 一种基于可信计算的工业控制系统信息安全解决方案. 信息与控制,2015 年第 44 卷第 5 期.

[7] 陈昱镔,陈思,程楠. 工业控制系统信息安全防护研究. 信息网络安全. 2016 年第 9 期.

[8] 赖英旭,熊增辉,蔡晓田,杨凯翔. 工业控制系统入侵检测研究综述. 通信学报. 2017 年 2 月,第 38 期第

[9] 卷.

[10] 邹春明,郑志千,刘智勇,陈良汉,陈敏超. 电力二次安全防护技术在工业控制系统中的应用。电网技术,2013 年 11 月第 37 卷第 11 期.

[11] 彭勇,江常青,谢丰,戴忠华,熊琦,高洋. 工业控制系统信息安全研究进展. 清华大学学报«自然科学版»2012 年第 52 卷第 10 期.

[12] 李佳玮,郝悍勇,李宁辉. 工业控制系统信息安全防护. 中国电力,第 48 卷第 10 期.

[13] 周江建,周运森. 中间件 OPC 技术在工业控制系统中的应用. 计算机工程,2004 年 12 月,第 30 卷第

[23] 期.

[15] 于立业,薛向荣,张云贵,赵永丽,赵华,卢永明,张秀明. 工业控制系统信息安全解决方案. 冶金自动化,2013 年 1 月,第 37 卷第 1 期.

[14] 褚健. 重中之重 工业控制系统安全的盛世危言. 中国信息安全，2012 年 8 月.

[17] 周小锋,陈秀真. 面向工业控制系统的灰色层次信息安全评估模型. 信息网络安全，2014 年第 01 期.

[16] 夏春明,刘涛,王华忠,吴清. 工业控制系统信息安全现状及发展趋势. 信息安全与技术，2013 年 2 月.

[19] 杜伟平,王平,王浩. 工业控制系统安全威胁分析与策略. 重庆邮电学院学报(自然科学版)，2015 年

[20] 月，第 17 卷第 5 期.

[18] 张敏,张五一,韩桂芬. 工业控制系统信息安全防护体系研究. 工业控制计算机，2013 年第 26 卷第 10 期.

[22] 李杰,柴焰明,杨燕,白梵,龚华健,彭西阳,南峰涛等. 区块链智能合约技术的原理与应用. 云南电力技术，2018 年 06 期第 46 卷.

[23] 倪蕴帷. 区块链技术下智能合约的民法分析、应用与启示[J]. 重庆大学学报(社会科学版)，2019(3)：
170-181.

[21] 胡健,肖鹏,尹君. 基于区块链技术的电网安全研究. 云南电力技术，2018 年 12 月第 46 卷.

[25] 魏凯,卿苏德,张奕奔,黄胜,徐晓旻,焦丽梅. 工业区块链应用白皮书 1. 0 版本，2019 年 2 月.

[26] 徐元清,卓蔚. 区块链技术在烟草系统工控安全中的应用. 微型电脑应用，2019 年第 35 期第 3 卷.

[24] 李瑾,仵松颀,张森林,陆月明. 基于区块链的分布式电能量数据可信存储机制. 网络与信息安全学报，2020 年 4 月第 6 卷第 2 期.

[28] 工业互联网中区块链应用场景和业务需求,CCSA.

[29] 汪允敏, 李挥, 王菡, 等. 区块链在工业互联网标识数据管理策略研究[J]. 计算机工程与应用，2020(1)：1-8

基于网络安全的流量分析进展研究

祁正伟[1]，石灏苒[2]，卫红权[2]，李海涛[2]，朱宇航[2]

[1]国防科技大学信息通信学院 陕西 西安710106；

[2]战略支援部队信息工程大学 河南 郑州 45000

摘　要：随着互联网快速发展，网络流量呈现爆发式增长，网络中信息蕴含的价值也与日俱增。但由于网络架构设计的滞后性，网络传输过程中存在的漏洞也常被挖掘出来以用于网络信息的窃听与劫持。如何根据网络流量特征进行分析，合理规划设计网络结构达到在合理成本条件下最大化网络传输速率的同时，保证信息传递的安全性，成为了近年来网络流量分析的研究重点。该文综述了该领域提出的各类方法模型，并提出了未来发展的展望。

关键词：流量分析、网络安全、网络传输、特征分析

A Review of Traffic Analysis based on Network Security

Qi Zhengwei[1], Shi Haoran[2], Wei Hongquan[2], Li Haitao[2], Zhu Yuhang[2]

1.School of Information and Communication, National University of Defense Technology, XiAn 710106,China;

2.People's Liberation Army Strategic Support Force Information Engineering University, Zhengzhou 450001,China

Abstract: With the rapid development of the Internet, network traffic has shown explosive growth, and the value of information contained in the network is also increasing day by day. However, due to the lag of network architecture design, loopholes in network transmission are often unearthed for eavesdropping and hijacking of network information. How to analyze according to the characteristics of network traffic, reasonably plan and design the network structure to maximize the network transmission rate under reasonable cost conditions while ensuring the security of information transmission has become the research focus of network traffic analysis in recent years. This article reviews various methods and models proposed in this field, and puts forward the outlook for future development.

Key words: traffic analysis; network security; network transport; characteristics

1　引言

随着互联网的快速发展，当下互联网思维已在各领域得到了广泛的推广，应用日益广泛，网络用户数显著增加［1］。信息呈现爆发式增长，产生的大量数据所蕴含的价值也与日俱增，数据逐渐成为一种重要的战略性资源，在各行业的发展中起到了重要的推动作用。近年来移动互联网和宽带城域网迅速发展，网络带宽不断扩充，为我们带来方便的同时其业务的复杂性也随之带来了一些安全隐患和设计缺陷［2-4］。由于网络结构的发展存在滞后性，存在的漏洞也为攻击者提供了对网络中信息进行篡改、窃听、劫持的机会，甚至通过使用僵尸网络等技术对网络结构造成威胁，且随着云计算的发展，云环境下的攻击行为逐步表现出隐蔽性强、攻击路径复杂多步等特点［5］，对互联网的发展产生了极大的危害［6-7］。如何对于这些攻击进行特征分析与识别成为了近年来网络安全研究的重点。从第一代防火墙技术到入侵检测［8-9］，再到移动目标防御和网络主动防御，网络安全从强调外部安全逐渐转变为以通过对网络态势感知等技术进行提前精准识别恶意网络行为为重点，以提升计算机与网络的内生安全。因此网络流量分析在对于网络恶意行为进行

基金项目：无

准确检测方面具有重要研究意义。

随着移动互联网与宽带网络的技术革新，服务种类的多元化随之带来了各类复杂的网络场景，网络攻击与窃听行为也在复杂的环境中更加隐蔽和多样。在互联网协议层面，攻击者利用ICMP网际控制报文协议将数据隐藏在ICMP的有效负载部分，形成ICMP隐蔽通道来秘密进行数据的传输［10］；HTTP作为使用最多的传输协议，虽然具有加密处理，但在传输敏感信息时仍可能被重放攻击所窃听［11］；而在IP协议中，攻击者无节制地占用网络带宽，也将导致被攻击者通信效率显著降低，达到阻碍通信的目的［12］。在移动互联网中，随着移动智能终端设备的普及，其安全性正受到越来越多恶意攻击者和安全研究者的关注，Android系统由于流行度最高，针对该系统所开发的恶意软件也趋于多样，而该类软件的主要特点仍是通过访问网络完成攻击［13-14］。因此流量分析技术作为准确检测以上攻击行为的重要手段之一，为适应各类攻击手段的发展，近年来也更加侧重于分析流量的各类特征，以下针对其发展进行具体介绍。

2 相关背景知识

网络流量是单位时间内通过网络设备或传输介质的信息量（报文数、数据包数或字节数）［15］。在网络流量分析过程中，根据分析的需求，流量分析可以从不同方面进行展开。根据时间空间位置以及层级的不同，往往获得的流量分析结果也侧重不同。以下对于不同维度展开的网络流量分析进行简要介绍。

2.1 网络流量提取方式

根据网络测绘、网络性能测试以及网络架构设计等网络行为的数据需求，网络所采集数据在时效性、准确性以及采集成本等方面侧重点有所不同，因此根据多元化的数据需求，网络流量统计分析方法大致可以分为基于硬件提取方法以及基于软件提取方法。

(1) *基于硬件提取方法*

通过在网络物理层上安装流量监控设备以得到高准确性与时效性的网络流量，且在网络中如网卡、路由器等不同位置安装可以达到获得不同协议、粒度的数据流的目的。典型例子如Fluke公司的Optiview链路分析仪Fluke［16］，以及OCX-mon监控器通过使用分光器将光纤连接器中光信息进行分离以获取数据包头信息等［17］。基于硬件的提取方法可以有效地获取网络上的数据流且极大地避免了缓冲与延迟的发生，一个好的硬件测量系统可以获得任何缓冲区的丢包数量，并对此做出决策，可以满足对于数据具有高精度与时效性需求的领域［18］。但实现成本高，与网络设备耦合性过高等原因也导致了其部署相对困难。

(2) *基于软件提取方法*

与基于硬件提取方法不同，基于软件提取方法通过修改计算机操作系统网络接口模块，将主机作为网络中流量监控设备，以实现在不部署额外硬件设备基础上达到网络流量监控与提取的目的。较为典型的流量分析软件以sFlow以及Cisco公司提出并且形成RFC3954的Netflow流量采集技术为代表［19-20］，以及可以工作在大多数UNIX和Windows平台上的MRTG流量监控软件［21］.其主要思想是通过网络上主机或路由器采集数据并形成固定格式数据分组存入缓存，后经过特殊编码传输到后台服务器进行解析。具有部署简单，可扩展性强等优点，但由于其在相互通信过程中将产生一定的网络传输延迟，在网络流量监测的时效性以及准确性上低于基于硬件提取方法。

2.2 网络流量分析粒度

根据网络流量分析的目的还可以从数据的粒度层面将网络流量分析分为比特级，分组级以及流级的流量统计分析，其目的在于通过获取不同网络层面的数据信息，以表征网络物理特性与协议传输过程等特征。

比特级流量分析通过获取网络物理层信号以实时分析网络吞吐量变化以及带宽利用率等网络流量数据特征，以此对网络结构进行合理设计，将网络资源进行合理配置。

分组级流量分析通过分析网络网络层与传输层的IP分组及报文传输时延，丢包等特征，以发现网络上路由配置缺陷，对网络路由结构做出合理规划。

流级流量分析则主要基于应用层在时间空间维度上对网络流量进行分析。目的在于通过对于网络用户行为分析，获得用户的流量使用特征，进而发现网络的周期性、长相关性等特征，对于

未来网络流量的趋势进行预测。

2.3 网络流量特性

(1) 周期性

由于网络中大多数网络行为都是由用户请求所产生，而用户产生的网络流量在时间序列上并不是恒定不变的，往往根据现实生活中时间尺度为基准，呈现周期性变化。对于网络整体而言，网络流量也随用户请求数量变化而呈现周期性变化，这一现象称为“潮汐现象”。周期性变化规律是客观存在的，它可能是由于流量数据的周期采集引起的，也可能根据用户习惯与年龄构成等外部因素有所变化［22］。

(2) 自相似性

自相似性表示网络流量数据在较小时空维度上的结构与特性与较大时空维度上的整体结构与特性相一致，并不会随时间尺度的变化而改变。如果某个对象是自相似的，那么它的某一部分在放大时便会出现相似于整体的现象，传统的泊松模型可以很好的解释这一特性［15］。

(3) 长相关性

长相关性表示了网络流量特征并不是孤立存在的，而是以流量序列为表现形式，相互关联及影响。长相关性与自相似性类似，区别在于从不同角度对于同一网络特性进行描述，自相似性强调在时间维度上，整体与局部的一致性，而长相关性则强调导致这一现象的局部结构间的相互影响作用。该特性意为着各时刻网络流量统计信息不是孤立出现的，而是由过去时刻网络状态所决定，传统的马尔可夫模型可以作为这类长相关模型的典型代表。

(4) 混沌性

混沌是指确定的、宏观的非线性系统在一定条件下所呈现出的不确定的或者不可预测的随机现象［23］。由于网络是一个复杂系统，影响流量变化的因素十分复杂多变，且相互之间往往具有非线性规律，所以整体看来网络呈现复杂多变的状态。对于这类网络传统模型无法很好的对其进行预测，而采用神经网络及支持向量机等新技术可对于这类网络实现流量预测。

3 网络流量分析模型

网络环境中，异常网络流量来源主要包括异常网络操作、蠕虫病毒传播、闪存拥挤以及网络资源滥用［24］。大多数网络攻击行为主要是通过异常操作对网络进行攻击，对于蠕虫病毒而言，其主要特征是通过在网络中不断进行复制与传播而占用网络资源，其潜伏周期长，不易发现的特点对网络造成极大危害。对于Dos以及DDos而言，其通过控制大量正常主机在特定时刻对目标发出大量异常请求也会导致服务器无法响应正常的网络请求。这些通过占用网络流量导致网络资源耗尽的网络攻击方式无法使用传统的防火墙、入侵检测等网络安全技术进行有效检测。针对这一问题，流量分析技术却可以较好的解决，作为网络安全保障技术簇中极为重要的一个环节，如何准确识别网络异常流量的同时，不对正常网络流量产生误判，成为了目前网络流量分析技术的研究重点［25］。

3.1 基于阈值的网络流量检测

针对DDos这类通过采用集中访问耗尽网络带宽的攻击方法，基于阈值的网络流量检测方法通过提前设置网络流量阈值以达到对异常流量进行检测的目的。其核心思想是为网络检测系统提前设置一个单位时间内通过流量的正常阈值，当检测到一定时间内通过流量高于这一阈值，则将其识别为异常流量，为之后的异常流量分析提供依据。该方法操作简单且实现成本较低，但由于网络是复制多变的非线性系统，如何确定一个合适的阈值成为了该方法实现的难点，如果阈值设置过高，则可能会对异常流量出现漏报；而如果设置过低，则会大幅增加告警数量，对于网络异常流量检测效率也将显著下降［26］。

3.2 基于特征的网络流量检测

该方法基于特征识别思想，分析各类网络异常操作特征，建立异常网络行为特征库。当检测到网络行为发生，将其与检测系统中特征库进行匹配，该方法可以快速检测出常见的异常网络行为，例如DDos、蠕虫病毒等攻击。该方法可以分别通过为网络行为设置“白名单”或“黑名单”两种方式实现，但由于网络攻击行为往往隐蔽性较强，与正常网络行为特征差异化不明显，如何确定异常网络行为特征将显著影响该方法检测效率。且由于该方法基于已知网络攻击行为建立特征库，具有一定的滞后性的特点，对于无法匹配

特征的网络攻击行为检测效果不明显。

3.3 基于机器学习的网络流量检测

以上传统方法由于人为设置往往无法达到最优值，降低了检测系统的识别效率，导致漏报或误报现象频繁产生。机器学习作为模拟人类学习行为的研究方向，以现有数据与以往经验为基础，结合传统方法构造模型并学习，将异常流量分析问题转化为分类与聚类问题，可以较好地对异常流量进行识别分析。且根据分类与聚类问题描述不同，可以将机器学习方法分为有监督的学习与无监督的学习两类方法。

无监督学习方法不需要事先为网络流量设置标签，可以根据现有数据对异常流量进行识别，其核心思想是聚类，即通过流量数据之间的相似属性将网络聚集为不同的集群，以此区分网络中异常流量。较为经典的聚类算法有如K-Means以及PCA主成分分析方法。其中K-Means运用聚类的思想，从网络中最有可能成为中心点的节点开始设置K个簇，每一轮向外选择离簇中心最近节点加入形成新的簇，并根据质心或中心的标准设置簇新的中心，直到网络达到稳定，以此根据节点间相似属性将具有不同特征的流量区分开来，文献［27］运用这一思想，设计了基于K-Means的流量检测系统。但由于K-Means方法根据初始节点的不同最终结果会有所差异，稳定性不强。PCA方法从节点特征维度出发，通过找到节点主要特征并映射的方法，将节点从高维特征空间映射到低维特征空间，在简化了问题的同时，也保留了流量之间的主要差别。文献［28］基于将这一思想运用于大规模网络的异常流量识别中，在取得不错效果的同时，也显著降低了计算复杂度。

有监督学习方法与无监督学习方法不同，其基于分类思想，预先根据以往数据与经验为网络流量设置若干确定标签，在此基础上对网络流量进行分类，直到网络结构稳定位置，其较为经典的算法有如决策树、神经网络等。其中基于决策树方法根据预先设定标签构造一个分类树，其中根节点代表初始流量数据，分支节点代表数据根据相应标签进行的一次分类，叶子节点代表最终分类获得集群。通过构造决策树，数据根据确定标签被分为彼此互斥的集群，文献［29］基于这一算法提出了一种基于决策树的异常流量检测系统，该方法实现简单且结构清晰，但由于分类标签是根据以往数据获取的，识别效率对于标签的选取具有较大的依赖性，在没有自学习的条件下难以取得较好效果。具有自学习特点的模拟人类神经结构提出的神经网络模型在这一基础上，可以达到更好的分类效果，也成为了目前研究的热点，如BP神经网络、卷积神经网络（CNN）、深度神经网络（DNN）、循环神经网络（RNN）等各类模型的应用，对于异常流量分析具有显著的效果［30］。但这些由于梯度下降算法所构造的损失函数容易导致局部收敛，且算法迭代速度慢使模型的准确度没有达到理想的状态，无法胜任现在异常流量检测准确性和实时性的要求［31］。为优化这类模型提高分析效果，文献［32-33］使用量粒子群、遗传算法、人工蜂群等优化算法与神经网络模型相结合，在提高了迭代速度同时也避免了局部收敛导致的结果不最优问题。

3.4 基于拓扑结构的网络流量检测

在面对复杂海量的网络流量环境，如何在保证异常流量检测精度同时兼顾检测效率，是目前网络流量检测研究的重要问题之一。基于阈值检测方法虽然检测效率较高，但无法对于DDos攻击这类单条链路上流量正常，但同时出现时的集体异常进行有效检测。而基于机器学习的方法虽可以通过提取对应特征达到较高识别精度，但存在计算复杂度高，可迁移性低的缺点。因此近年来从网络拓扑结构角度对网络流量进行异常检测的方法研究受到了广泛关注［34］，该方法将通信网络抽象为复杂网络的形式，通过抽取节点局部结构，并计算局部结构在聚类系数、总边数、总权值等网络性质的分布，从而得到异常子图［35］。这一思想可以有效检测DDos攻击或网络扫描行为这类单条链路符合正常行为，但相互之间存在异常联系的集体异常行为［36］。利用较少的特征就可以达到可观的异常检测效果，对计算复杂度与检测效率进行了兼顾，可以应用于大规模网络中。

4 结束语

网络安全是一个全面的、复杂的工程，其难点在于攻防不对等，更新速度快。如何提前感知网络攻击行为，维护网络正常运行，成为了流量分析技术的重点。目前传统的基于阈值与特征进

行异常流量识别的技术由于对过去经验有较强依赖性而具有滞后的缺点，网络异常操作往往可以利用这一特点对流量检测分析进行规避。而基于机器学习的智能化、自学习的流量分析技术就成为了目前研究的重点，但在未来的研究过程中仍存在以下问题：如何合理设置网络流量获取位置以准确描述网络流量特征；如何运用优化算法对机器学习模型进一步优化以提高检测效率；在网络快速发展的今天，面对大规模网络流量时如何将网络流量检测技术与云计算等先进技术相结合以获取处理大规模流量的能力。

参考文献:

[1] 杨祎. 网络流量预测技术的研究[D]. 北京邮电大学,2014.

[2] 秦刘,智英,建贺磊,等. 802. 1x协议研究及其安全性分析[J]. 计算机工程,2007(07):153-154+157.

[3] 王涛,陈鸿昶. 全要素SDN指纹攻击及其模糊混淆防御机制研究[J]. 电子学报,2020,48(06):1213-1219.

[4] 袁庆军,张勋成,高杨,王永娟. 轻量级分组密码PUFFIN的差分故障攻击[J]. 电子与信息学报,2020,42(06):1519-1525.

[5] 李鹏,于洪涛,徐静波. 七号信令监测系统中基于Oracle的数据同步方案优化[J]. 电讯技术,2010,50(02):73-77.

[6] 王文娟,杜学绘,任志宇,等. 基于因果知识和时空关联的云平台攻击场景重构[J/OL]. 计算机科学,2020. 1-11

[7] 马陈城,杜学绘,曹利峰,吴蓓. 基于深度神经网络burst特征分析的网站指纹攻击方法[J]. 计算机研究与发展,2020,57(04):746-766.

[8] 张耀元,郭淑明,汪小雨. 基于入侵检测技术的MANET安全研究[J]. 电子科技,2016,29(11):157-160.

[9] 张桥,卜佑军,胡静萍,张稣荣. 入侵检测技术研究综述[J]. 网络安全技术与应用,2020(08):22-24.

[10] 李抒霞,周安民,郑荣锋,胡星高. 基于SVM的ICMP网络存储隐蔽信道检测[J]. 信息安全研究,2020,6(02):122-130.

[11] 何振宇. 基于流量分析的HTTP协议安全现状分析[J]. 科学技术创新,2020(03):82-84.

[12] 赵明. IP网络流量分析在网络管理中的应用[J]. 电子技术与软件工程,2016(21):17+47.

[13] 潘文强. 基于流量分析的安卓恶意软件检测[D]. 电子科技大学,2020.

[14] 李浩,马坤,陈贞翔,赵川. 基于网络流量分析的未知恶意软件检测[J]. 济南大学学报(自然科学版),2019,33(06):500-505.

[15] 苟娟迎,马力. 网络流量分析方法综述[J]. 西安邮电大学学报,2010,15(4):20-23.

[16] 刘铮. 网络测量方法及体系结构[D]. 天津大学,2002.

[17] 宋鸣. 基于流量分析的信息溯源关键技术研究[D]. 北京邮电大学,2014.

[18] ITU-T. Routing mechanisms in public packet telecommunication data networks[S]. 2010

[19] 王萌,王玲. 包分类算法在防火墙中的应用研究[J]. 通信技术,2011. 5(44):57-59.

[20] 张潇晓. 网络流量分析关键技术研究与系统实现[D]. 国防科学技术大学,2012.

[21] 杨家海,吴建平,安常青. 互联网络测量理论与应用[M]. 北京:人民邮电出版社,2009.

[22] 王西锋. 网络流量特性分析与预测研究[D]. 西北大学,2007.

[23] 周孝鹏. 基于网络安全的流量分析技术[J]. 信息与电脑,2019(12):201-202.

[24] 张宾. 互联网异常流量特征分析及其应用研究[D]. 清华大学,2012:74.

[25] 张楠,李洪敏,卢敏,柯明敏. 网络异常流量检测方法[J]. 兵工自动化,2016,35(9):66-69.

[26] Jianliang M, Haikun S, Ling B. The application on intrusion detection based on k-means cluster algorithm[C]//Information Technology and Applications，2009. IFITA'09.
International Forum on. IEEE，2009，1：150-152.

[27] 丁美美. 基于主成分分析的网络流量异常检测研究[D]. 北京交通大学,2017.

[28] Lee J H, Lee J H, Sohn S G, et al. Effective value of decision tree with KDD 99 intrusion detection datasets for intrusion detection system [C]//Advanced Communication Technology，2008. ICACT 2008.
10th International Conference on. IEEE，2008，2：1170-1175.

[29] 蒋品. 基于机器学习的蜂窝网络基站流量分析与预测研究[D]. 北京邮电大学,2019.

[30] 陈胜,朱国胜,祁小云,雷龙飞,镇佳,吴善超,吴梦宇. 基于机器学习的网络异常流量检测研究[J]. 信息通信,2017(12):39-42.

[31] 黄文明,徐双双,邓珍荣,雷茜茜. 改进人工蜂群算法优化RBF神经网络的短时交通流预测[J]. 计算机工程与科学,2016,38(4):713-719.

[32] 田中大,高宪文,李树江,王艳红. 遗传算法优化回声状态网络的网络流量预测[J]. 计算机研究与发展,2015,52(5):1137-1145.

[33] 邓瀚涔. 大规模粒子群算法及其在视频流量特征选择中的应用研究[D]. 济南大学,2019.

[34] Akoglu L , Tong H , Koutra D . Graph-based Anomaly Detection and Description: A Survey[J]. 2014.

[35] Akoglu L , Mcglohon M , Faloutsos C . oddball: Spotting Anomalies in Weighted Graphs[J]. 2010.

[36] Tong H , Lin C Y . Non-Negative Residual Matrix Factorization with Application to Graph Anomaly Detection. [C]// Proceedings of the Eleventh SIAM International Conference on Data Mining，SDM 2011, April 28-30，2011，Mesa，Arizona，USA. DBLP，2011.

[作者简介]

祁正伟（1980—），男，学士，工程师，主要研究方向：电子与通信工程。

石灏苒（1997—），男，在读硕士，主要研究方向：复杂网络链路隐匿，网络降噪。

卫红权（1971—），男，博士，研究员，主要研究方向：网

络信息安全，大数据处理技术。

李海涛（1982—），男，硕士，副研究员，主要研究方向：通信网安全、数据处理和嵌入式设计。

朱宇航（1982—），男，硕士，副教授，主要研究方向：网络科学。

基于统一描述网络结构模型的链路预测方法

吴翼腾[1]，顾泽宇[1，2]，于淑乔[3]

[1]信息工程大学 郑州 450002；

[2]中国电子工程设计研究院 北京 100089；

[3]墨尔本大学，澳大利亚 墨尔本 3010

摘　要：针对经典的链路预测模型无法利用实际网络中重叠结构等信息的问题，该文采用笛卡尔积、幂集、离散的度量空间等概念构建了一种对层次信息、重叠信息以及低阶环等微观信息进行统一描述的网络结构模型；并提出了基于该模型的链路预测算法。统一描述网络结构模型是层次结构模型和随机分块模型的一般化推广，可以对网络的层次结构、重叠结构等做出统一解释。真实网络数据集测试结果表明，相比主流算法，基于统一描述网络结构模型的链路预测算法可有效提高链路预测准确性。

关键词：复杂网络、链路预测、统一描述、网络结构模型

A Link Prediction Method Based on Uniform-Structure-Information Model

WU Yi-teng[1], GU Ze-yu[1,2], YU Xu-qiao[3]

1.National Digital Switching System Engineering and Technological R&D Center, Zhengzhou 450002, China;

2.China Electric Engineering Design Institute, Beijing 10089, China;

3.University of Melbourne, Melbourne, 3010, Australia

Abstract: Aiming at the problem that the classical link prediction model can not make use of the overlapping structure information in the actual network, the concepts of Cartesian product, power set and discrete metric space are adopted to construct a new network model, Uniform-Structure-Information model. It can give a uniform description of hierarchical information, overlapping structure information and microstructure information such as low order loop information. A link prediction method based on the proposed model is presented in this paper. Uniform-Structure-Information model is a generalization of hierarchical structure model and random block model, and hierarchical structure and overlapping structure of the network can be explained in a unified way. Experiment results on real datasets show that this method can effectively improve the accuracy compared with the existing mainstream link prediction methods.

Key words: Complex network; link prediction; uniform description; network model

1　引言

复杂网络中的链路预测问题是指运用网络结构信息，对一未知连接的节点对是否存在连接的可能性进行预测[1-2]，其具有很强的理论和实际应用价值。在理论意义上，链路预测可以帮助人们认识复杂网络的演化机制，刻画网络的结构信息等[3]。在实际应用价值方面，为指导生物蛋白质网络构建、电子商务商品推荐、资源贸易协调、电信用户通联关系挖掘等方面提供了重要方法[4-6]。

在无权无向静态网络的链路预测研究中，现有方法可分为：基于节点相似性的链路预测算法、节点相似性的指标融合算法、基于机器学习的链路预测算法以及基于似然分析的链路预测算法等4

基金项目：国家自然科学基金(61601513)

Foundation item: National Natural Science Foundation of China (61601513)

通讯作者：吴翼腾 wuyiteng1992@163.com

类。基于节点相似性的方法理论简洁，效率较高，主要从节点对的共同邻居数、路径数出发[7]，直接给出节点对的相似性评分。基于此方法，周涛等提出共同邻居加权的资源分配指标[8]；YAO Y等提出局部加权路径指标[9]；刘树新等提出基于局部拓扑信息加权的相似性指标[10-11]、资源传输匹配度指标[12]；文献［13-15］考虑到网络的社区信息，利用社区信息对经典相似性指标加权，或仅在节点所属社区内计算经典相似性指标，提升链路预测准确度。相似性指标是对微观网络结构的建模。

随着网络结构信息的研究深入，上百种相似性指标相继提出。多种维度的网络信息被充分挖掘；单一相似性指标在各类网络数据集上的鲁棒性一般。为进一步提高相似性指标的准确性和鲁棒性，人们纷纷提出组合规则法、OWA算子融合法[16]、AdaBoost融合法[17]等相似性指标的组合方法。于洪涛等提出的基于模糊积分的相似性指标融合算法[18]可大幅提升链路预测的准确性和鲁棒性。基于机器学习的链路预测方法将链路预测问题转化为有无连边的机器学习二分类问题[20]，其本质上也可看作多指标经分类器输出融合的组合方法。吴翼腾等详细研究了组合方法的理论极限问题[19]，提出并证明了采用组合方法进行链路预测的理论极限定理，阐释了组合方法相比单机制方法具有更高准确性和稳健性的原因。

基于似然分析的链路预测方法首先给出网络结构生成方式的假设，然后根据假设求出节点对产生链路的概率。典型的方法有层次结构模型[21]和随机分块模型[22-23]。与相似性指标相对应，该类方法是对宏观网络结构的建模。

网络结构的描述对于理解网络的生成演化机制、刻画网络各节点间的联系具有十分重要的作用[24]，从而为节点间的链路预测提供理论依据。基于似然分析的随机分块模型和层次结构模型从不同侧面提供了网络结构的描述方式，但是二者不能处理网络中各种规模（从宏观、中观的网络社区结构，到微观的低阶环或模体[25]结构等）的重叠结构问题，而实际网络中重叠结构无处不在。针对这一问题，本文对无向网络重新建模，提出统一描述网络结构模型（Uniform-Structure-Information模型，简称USI模型），该模型是层次结构模型和随机分块模型的一般推广，既可以包含节点的层次结构信息，又可以使节点从属于不同集合。避免了二者集合划分不可重叠的问题。在USI模型的基本假设下，可以将网络的社区结构、层次结构、网络的低阶环或模体等微观结构输入该模型。同时基于USI模型，本文提出一种基于该模型的链路预测算法，并通过数据实验验证算法的有效性。

本文的组织如下：第2节介绍链路预测的问题描述和评价指标；第3节构建统一描述的网络结构模型；第4节给出基于USI模型的链路预测算法；第5节对该链路预测算法在8个数据集上进行实验验证；第6节对本文做出总结。

2　问题描述和评价指标

2.1　问题描述

给定t时刻的网络$G(V,E)$，V和E分别表示节点集合和边集合。链路预测目的是预测未来的t'时刻将要出现的链路或消失的链路，或是预测当前t时刻网络未观测到的链路或错误链路[20]。即链路预测算法赋予节点对间链路预测的评分值，按照评分值的大小判决链路是否存在。

2.2　链路预测算法的评价指标

为了评估算法的准确性，要对网络进行训练集和测试集划分，链路预测算法只允许运用训练集信息进行预测。一般用AUC（Area Under the Receiver Operation Characteristic Curve）准确度衡量。AUC不受有无连边两类样本非平衡性的影响（无连边的节点对远大于有连边的节点对数量），AUC可以理解为在测试集中随机选择一条边的分数值比随机选择一条不存在的边的分数值高的概率[7]。即每次从测试集中随机选择一条边，再从不存在的边中随机选择一条边，若前者高则加1分，若相等则加0.5分，这样独立比较n次。若有n'次测试集得分高，有n''次二者相等，则AUC定义为：

$$\mathrm{AUC}=\frac{n'+0.5n''}{n} \tag{1}$$

3 统一描述网络结构模型

3.1 统一描述网络结构模型的定义

定义1

统一描述网络结构模型

$$A_1 = 2^{A_0}, A_2 = 2^{A_1}, A_3 = 2^{A_2}, \cdots, A_i = 2^{A_{i-1}}, i = 1,2,3,\cdots \tag{2}$$

对任意i, A_i中具有指定关系f的元素可以组成集合族D_k, $\mathrm{D}_k = \{D_1, D_2, \cdots D_n\}$

$$f:(A_i)^k \to \mathrm{D}_k, \mathrm{D}_k \subset (A_i)^k, k = 1,2,\cdots,\left|A_i\right| \tag{3}$$

其中

$$(A_i)^k = \left\{\left\{A_{i1}, A_{i2}, \cdots A_{ik}\right\} : A_{ij} \in A_i, 1 \le j \le k\right\} \tag{4}$$

$\left|A_i\right|$表示集合A_i的势，即集合A_i中元素的个数。

特别地，当i=0时，

$$f:(A_0)^k \to \mathrm{D}_k, \mathrm{D}_k \subset (A_0)^k \tag{5}$$

（2） 在集合族D_k的元素D_i上定义离散的度量空间：

$$d_i(x,y) = \begin{cases} p_i(p_i \leqslant 1), x \ne y \text{且} x,y \in D_i \\ 0, \qquad x = y \text{且} x,y \in D_i \end{cases} \tag{6}$$

USI模型可以解释无向网络中各种节点的连接和组织关系，为网络中各类节点的层次关系、重叠关系等建立了简明清晰的表示方法。定义中（1）可以简单地理解为有共同特性的元素可以组成集合。定义中（2），集合上建立的离散度量空间的度量值p可解释为该集合内部元素间发生联系的概率，且集合内部元素发生联系的概率相同（严格的讲，度量或距离衡量的是元素的疏离程度，可以用度量的倒数刻画紧密程度即元素发生联系的概率，因此在USI模型中不妨直接将度量理解为概率）。

根据模型定义，组成集合的元素可以是集合或非集合。当集合中元素为非集合时，集合与其上的度量直接表示节点对间存在链路的概率；当集合中的元素为集合时，度量p表示集合与集合建立联系的概率。例如，一个班级中的所有同学组成的集合，该集合中的元素为非集合，集合上的度量p表示该班级任意两个同学存在联系的概率，且均为p；又如，一个学校所有班级组成的集合，集合中的元素为班级，度量p则表示班级之间建立联系的概率，从而间接地表示班级之间的同学建立联系的概率。

（1） A_0是网络中所有节点组成的集合，定义幂集：

定义2

元素的阶

在USI模型中，A_i及其非空子集的元素为i阶元素，i = 0, 1, 2, 3, …。i阶元素记为$X^{(i)}$。2阶以上元素称为高阶元素。

定义3

集合的阶

在USI模型中，集合所含元素的阶数称为集合的阶。i阶集合记为$\mathrm{X}^{(i)}$。2阶以上集合称为高阶集合。

例如一个由3个节点组成的网络，$A_0 = \{1,2,3\}$，根据定义2，元素1，2，3都是0阶元素，根据定义3，A_0是0阶集合。由于元素$\{1,2\} \in 2^{A_0}$

$= \{\varnothing, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\} = A_1$

，因此元素$\{1,2\}$为1阶元素。设指定关系f为选取A_1中包含节点对$\{1,2\}$的元素，则$f\left[(A_1)^1\right] = \mathrm{D}_1 = \{\{\{1,2\}\}, \{\{1,2,3\}\}\}$，$f\left[(A_1)^2\right] = \mathrm{D}_2 = \{\{\{1,2\}, \{1,2,3\}\}\}$，$k \geqslant 3$时，$f\left[(A_1)^k\right] = \varnothing$。显然，$(A_i)^k \subseteq 2^{A_i} = A_{i+1}$为$i$ + 1阶集合。根据定义4，由于$\{1,2\}$为1阶元素，所以集合$\{\{1,2\}\}$为1阶集合。又如，$\Lambda_1 = \{\{1,2\}, \{1,3\}, \{1,2,3\}\}$是$A_1$的一个非空子集，即$\Lambda_1 \subseteq A_1$，所以$\Lambda_1$中的元素为1阶元素，$\Lambda_1$为1阶集合。同理，$\{\{1,3\}, \{1,2,3\}\} \in 2^{\Lambda_1} \subseteq 2^{A_1} = A_2$，因此元素$\{\{1,3\}, \{1,2,3\}\}$为2阶元素。

USI模型是对随机分块模型和层次结构模型的一般化推广。随机分块模型假设网络被分成若干个群，两个节点产生链路的概率只取决于节点所在的群。随机分块模型无法体现网络的层次结构和重叠结构信息。层次结构模型将网络用族谱树的形式表示，网络中$|A_0|$个节点作为叶子节点，族谱树通过$|A_0|-1$个非叶子节点将它们联系起来。将每个非叶子节点赋予一个概率值，每两个叶子节点连边的概率用它们最近共同非叶子节点处的概率表示。层次结构模型中若一个节点从属于某一叶子分支，其本身也属于上一级非叶子节点所属的叶子分支，即可以表示网络的层次结构特性。但是，节点不可从属于同级非叶子节点所属的其它叶子分支，即无法体现网络的重叠性。本质上，层次结构模型可以包含随机分块模型。USI模型假设有某种共同特性的元素可以组成集合，集合上定义离散的度量空间。从USI模型出发，若USI模型的最高阶集合为1阶集合，且所有1阶元素的交集为空集时，USI模型退化为随机分块模型。若每个集合有且仅有2个元素，且所有元素从属于唯一的对应阶集合时，USI模型退化为层次结构模型，层次结构模型可以含有高阶元素。基于该分析，层次结构模型是随机分块模型的推广，USI模型又是层次结构模型的推广。由此可以进一步分析，作为链路预测方法的随机分块模型和层次结构模型实际上是一种组合方法，是USI模型从不同角度退化后的加权组合，加权系数由该模型的“合理”程度决定，属于链路预测的前端融合算法。当给定指定关系f后，USI模型可以用于链路预测，具体算法将在后文给出。

USI模型本身可以看作加权网络模型，只要当i=0，k=2时，即两个元素构成0阶集合的指定关系f为任意两节点对组成集合时，度量p根据连边权重设定后，即转化为加权网络模型。

3.2 统一描述网络结构模型的性质

命题

$i(i\geqslant 1)$阶集合$\mathrm{X}^{(i)}=\{X_1^{(i)},X_2^{(i)},\cdots,X_n^{(i)}\}$可以通过集合中元素的并集运算$g$降阶为$i$-1阶集合。其中，

$$g:\mathrm{X}^{(i)}\rightarrow\bigcup_{j=1}^{n}X_j^{(i)} \tag{7}$$

显然，$\bigcup_{j=1}^{n}X_j^{(i)}$为$i$ - 1阶集合，记为$\mathrm{X}^{(i-1)}$。

推论1

$i(i\geqslant 2)$阶集合$\mathrm{X}^{(i)}=\{X_1^{(i)},X_2^{(i)},\cdots,X_n^{(i)}\}$可以通过集合中元素的并集运算$g$的$i$次迭代降阶为0阶集合。

证明

根据命题，i阶集合经1次元素的并集运算g可降为i-1阶集合；该i-1阶集合经第2次元素的并集运算g可降为i-2阶集合；以此类推，经过i次元素的并集运算g可降为0阶集合。 证毕

推论2

$i(i\geqslant 2)$阶集合$\mathrm{X}^{(i)}=\{X_1^{(i)},X_2^{(i)},\cdots,X_n^{(i)}\}$可以将其元素看作$i$-1阶集合，对每个$i$-1阶集合通过集合中元素的并集运算$g$的$i$-1次迭代，使原$i$阶集合降阶为1阶集合。

证明

将i阶集合的每个元素看作i-1阶集合，根据推论1，每个i-1阶集合经过i-1次元素的并集运算g的迭代，可降为0阶集合。则原i阶集合降为这些0阶集合作为元素构成的1阶集合。 证毕

设3阶集合$\mathrm{X}^{(3)}=\{X_1^{(3)},X_2^{(3)},X_3^{(3)}\}=\{\{\{\{1,2,3\},\{6\}\},\{\{4,5\}\}\},\{\{\{6\},\{7,8,9\}\}\},\{\{\{6,9\},\{9\}\}\}\}$，显然，$X_1^{(3)}=\{\{\{1,2,3\},\{6\}\},\{\{4,5\}\}\}$，$X_2^{(3)}=\{\{\{6\},\{7,8,9\}\}\}$，$X_3^{(3)}=\{\{\{6,9\},\{9\}\}\}$是3阶元素。根据推论1，该3阶集合可降阶为2阶集合：

$$\begin{aligned}\mathrm{X}^{(2)}=\bigcup_{j=1}^{3}X_j^{(3)}&=\{\{\{1,2,3\},\{6\}\},\{\{4,5\}\}\}\cup\{\{\{6\},\{7,8,9\}\}\}\cup\{\{\{6.9\},\{9\}\}\}\\&=\{\{\{1,2,3\},\{6\}\},\{\{4,5\}\},\{\{6\},\{7,8,9\}\},\{\{6,9\},\{9\}\}\}\end{aligned} \tag{8}$$

该2阶集合可以降为1阶集合：

$$\begin{aligned}\mathrm{X}^{(1)}=\bigcup_{i=1}^{4}X_j^{(2)}&=\\&\{\{1,2,3\},\{6\}\}\cup\{\{4,5\}\}\cup\{\{6\},\{7,8,9\}\}\cup\{\{6,9\},\{9\}\}\\&=\{\{1,2,3\},\{6\},\{4,5\},\{7,8,9\},\{6,9\},\{9\}\}\end{aligned} \tag{9}$$

该1阶集合可以降为0阶集合：

$$X^{(0)}=\bigcup_{j=1}^{3}X_j^{(1)}=$$

$$\{1,2,3\}\cup\{6\}\cup\{4,5\}\cup\{7,8,9\}\cup\{6,9\}\cup\{9\}=$$

$$\{1,2,3,4,5,6,7,8,9\} \quad (10)$$

根据推论2，$X^{(3)}=\{X_1^{(3)},X_2^{(3)},X_3^{(3)}\}$中的3阶元素$X_1^{(3)},X_2^{(3)},X_3^{(3)}$可降为2阶元素：

$$X_1^{(2)}=\{\{1,2,3\},\{6\}\}\cup\{\{4,5\}\}=\{\{1,2,3\},\{6\},\{4,5\}\}$$

$$X_2^{(2)}=\{\{6\},\{7,8,9\}\}$$

$$X_3^{(2)}=\{\{6,9\},\{9\}\}$$

进一步，2阶元素$X_1^{(2)},X_2^{(2)},X_3^{(2)}$可降为1阶元素：

$$X_1^{(1)}=\{1,2,3\}\cup\{6\}\cup\{4,5\}=\{1,2,3,4,5,6\}$$

$$X_2^{(1)}=\{6\}\cup\{7,8,9\}=\{6,7,8,9\}$$

$$X_3^{(1)}=\{6,9\}\cup\{9\}=\{6,9\}$$

所以最终得到由这3个1阶元素组成的1阶集合：

$$X^{(1)}=\{X_1^{(1)},X_2^{(1)},X_3^{(1)}\}=\{\{1,2,3,4,5,6\},\{6,7,8,9\},\{6,9\}\} \quad (11)$$

4 基于统一描述网络结构模型的链路预测方法

根据统一描述网络结构模型的定义和性质，用集合中的度量来衡量两集合关系紧密程度，进而刻画两节点的连接概率。基于USI模型的链路预测方法的基本假设是两个节点发生联系的概率主要依赖于其所在的群体（集合）。因此基于该模型的链路预测方法首先根据可利用的信息给出模型中集合的划分；其次利用最大似然估计给出离散的度量空间度量p的估计；最后假设各条路径产生联系概率相互独立的条件下，根据并联概率给出链路预测得分。链路预测方法的步骤如图1所示。

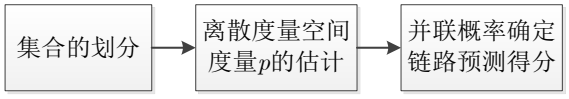


图1基于USI模型的链路预测方法步骤

4.1 集合的划分

对于含有属性信息和真实群组结构的网络数据，可以根据该信息给出指定关系f确定各阶集合的划分组成。对于只含有节点和连边拓扑信息的数据，只能通过算法识别和合理策略给出指定关系f。现对只含有拓扑信息数据进行分析。

对于0阶集合，复杂网络的社区发现算法给出了针对仅含拓扑信息数据的0阶集合划分方式。USI模型的链路预测算法中引入社区发现算法，按特定社区发现算法划分的结果，规定指定关系f划分0阶集合。网络中的环是另一种十分重要的网络结构，一个长度为h的环，是由h个节点$\{v_1,v_2,\cdots,v_h\}$和h条边

$$v_1,v_2\quad,\quad v_2,v_3\quad,\quad,\quad v_{h\ 1},v_h\quad,\quad v_h,v_1$$

组成的封闭回路（<i，j>表示边，且<i，j>=<j，i>）。环的存在尤其是低阶环的多少对网络功能有重要影响。USI模型的链路预测算法也考虑网络中的低阶环作为0阶集合的划分方式。

对于1阶集合，考虑社区发现算法划分0阶集合两两交互作用的情况，将任意两个0阶集合组成1阶集合。为减少计算量，在度量p的估计时可设置阈值限定1阶集合度量p的建立范围，且低阶环不划分1阶集合。由于仅含拓扑信息数据信息量有限，暂不考虑2阶以上集合的划分。

按照上述分析，给出指定关系f的如下策略：

（1）$i=0$，$k=1$，2，…，$|A_0|$时，按社区发现算法的划分结果作为指定关系f_1划分0阶集合。假设社区发现算法划分的社区结构全体为集合P，$P=\{V_1,V_2,\cdots V_n\}$，则指定关系f_1为：

$$f_1:(A_0)^k\rightarrow \mathrm{D}_{f_1k},\mathrm{D}_{f_1k}\subset(A_0)^k,k=1,2,\cdots,|A_0| \quad (12)$$

$$\mathrm{D}_{f_1k}=f_1[(A_0)^k]=\{V_i:|V_i|=k,V_i\in P\} \quad (13)$$

（2）$i=0$，$k=1$，2，…，$|A_0|$时，按指定关系f_2将只差1条边构成k阶环的元素组成0阶集合：

$$f_2:(A_0)^k\rightarrow \mathrm{D}_{f_2k},\mathrm{D}_{f_2k}\subset(A_0)^k,k=1,2,\cdots,|A_0| \quad (14)$$

$$\mathrm{D}_{f_2k}=f_2[(A_0)^k]=$$

$$\left\{\begin{array}{l}\{i_1,i_2,\cdots,i_k\}:\text{任意}i_1,i_2,\cdots,i_k\in V\text{且}i_1\neq i_2\neq\cdots\neq i_k,\\ \text{存在唯一}<i,j>,i=i_1,i_2,\cdots,i_k j=i_1,i_2,\cdots,i_k,i\neq j\\ \text{使得}<i,j>\notin E\end{array}\right\} \quad (15)$$

其中，V是无权无向网络$G(V,E)$节点的集合，E是网络边的集合。

（3）$i=1$，$k=2$时，按指定关系f_3将f_1划分的0阶集合两两组成1阶集合：

设：

$$D_i \in \bigcup_{k=1}^{|A_0|} \mathrm{D}_{f_1 k} = \mathrm{D}_{f_1} \tag{16}$$

$$f_3 : (A_1)^2 \to \mathfrak{D}, \mathfrak{D} \subset (A_1)^2 \tag{17}$$

$$\mathrm{D} = f_3\left[(A_1)^2 \right] = \left\{ \left\{ D_i, D_j \right\} : \text{任意} D_i, D_j \in \mathrm{D}_{f_1}, D_i \neq D_j \right\} \tag{18}$$

4.2 离散度量空间度量p的估计

根据集合阶数的不同，度量p的估计可以分为3种情况，分别为0阶集合上度量p的估计、1阶集合上度量p的估计以及高阶集合上度量p的估计。下面分别给出具体算法。

4.2.1 0阶集合$\mathrm{X}^{(0)}$上度量p的估计

设

$$\left| \mathrm{X}^{(0)} \right| = N_1 \tag{19}$$

在仅含0阶元素组成的集合中，元素与元素间只有连边与非连边之分，度量p即定义为元素连边的概率。集合中元素连边数为随机变量X，则X服从B（N，p）的二项分布。其中N为集合中元素的最大可能连边数$N = C_{N_1}^2$。对0阶集合上度量p的估计采用极大似然估计法。

似然函数为：

$$L(p,x) = p^x (1-p)^{N-x}, x = 0,1,2,\cdots,N \tag{20}$$

其中x是0阶集合观测到的实际连边数。

令：

$$\frac{dL(p,x)}{dp} = 0 \tag{21}$$

解得

$$\hat{p} = \frac{x}{N} \tag{22}$$

例如图2所示的0阶集合中共有10个节点，12条连边，则度量p的估计值为$\hat{p}$=12/C_{10}^2 = 4/15。

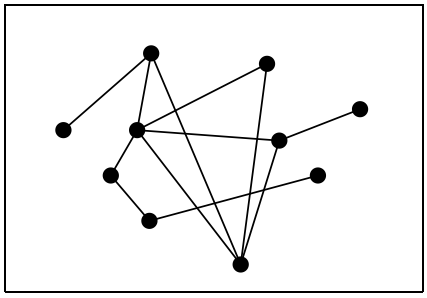


图2　0阶集合上度量p的估计示例

4.2.2 1阶集合$\mathrm{X}^{(1)}$上度量p的估计

$$\mathrm{X}^{(1)} = \left\{ X_j^{(1)} : X_j^{(1)} \text{是1阶元素}, j = 1,2,3,\cdots \right\} \tag{23}$$

$$\left| \mathrm{X}^{(1)} \right| = K \tag{24}$$

考虑到1阶元素可能存在交集，设：

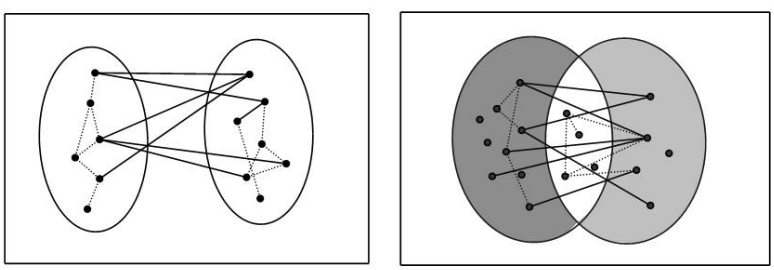
$$\left| X_j^{(1)} \setminus \bigcup_{j=1}^{K} X_j^{(1)} \right| = k_j \tag{25}$$

集合中1阶元素的最大可能连边数定义为：

$$N = \sum_{i \neq j} k_i k_j \tag{26}$$

由此，对于1阶集合$\mathrm{X}^{(1)}$，集合中1阶元素间仅存在0阶元素的连边与非连边，问题同样转化为二项分布$B(N,p)$（N表示1阶元素间0阶元素的最大可能连边数）的p值如何估计的问题，估计方法与4.2.1相同。

例如图3（a）所示的1阶集合不存在交集，只需考虑集合之间的实际连边数（为6）和可能最大的连边数（为6 × 7=42），则度量p的估计值为$\hat{p}$= 6/(6 × 7) = 1/7；图3（b）中集合存在交集，因此实际连边数仅考虑交集之外的实际连边（为7）和可能的最大连边（为9 × 5 = 45），则度量p的估计值为$\hat{p}$=7/(9 × 5) = 7/45。



(a) 1 阶集合间不存在交集　　(b) 1 阶集合间存在交集

图3　1阶集合上度量p的估计示例

4.2.2 高阶集合$X^{(i)}(i\geqslant 2)$上度量p的估计

根据推论2，将高阶集合通过元素的并集运算迭代降为1阶集合后，按照1阶集合上度量空间度量p的估计方法求解。

4.3 并联概率确定链路预测得分

由于USI模型中同一节点可以从属于不同阶的不同集合，因此存在两节点对产生联系的多条路径，与生活中人际交往十分类似，每增加一条两节点产生联系的路径，则两节点产生联系的概率随之增大。因此采用节点对各条路径产生联系的概率值的并联概率作为最终链路预测得分。假设产生联系的各条路径相互独立的条件下，最终链路预测得分可以表示为：

$$s_{xy}=1-\prod_{i=1}^{N_{xy}}\left(1-p_{xy}^{i}\right) \tag{27}$$

其中，s_{xy}为节点对xy的最终链路预测得分即连边概率，p_{xy}^{i}为节点对在第i个共同集合内连边的概率，N_{xy}为节点对xy所处的共同集合个数。

4.4 基于USI模型的链路预测方法与其它方法的对比分析

基于USI模型的链路预测方法属于链路预测的前端融合法，前端融合法主要包括基于拓扑信息[10-11]、社区信息[13-15]加权的方法、基于概率模型的似然分析法等[21-23]，前端融合法一般具有很好的解释性，方法的物理意义明确。后端融合法包括基于相似性的指标融合方法[16-18]、基于机器学习的分类方法[20]等，后端融合法提高预测准确度的机理是将多维度网络信息拟合成准确度的多元标量函数，并对目标函数进行优化，使准确率达到最大，但往往缺乏算法的解释性。相比于其它链路预测方法，USI模型的链路预测方法基于一条基本假设，即两个节点发生联系的概率主要依赖于其所在的群体。用USI模型的定义来表述，就是节点对从属于哪个度量空间，就用哪个度量空间上的度量p来衡量节点对的关系，若节点对从属于多个度量空间，就用这些度量空间共同作用的效果（即概率的并联）衡量节点对之间的关系。由于USI模型可以将多维度网络结构信息（包括已知的真实网络结构信息）输入进来，因此基于USI模型的链路预测可以综合利用网络的层次结构、重叠结构、微观结构等网络结构信息。基于以上信息和模型假设，用节点对从属的集合或度量空间解释链路的生成方式，进行链路预测。

5 实验分析

本文采用USI模型的链路预测算法，在8个数据集上进行链路预测实验。实验时，选用以下2个社区发现算法：

（1） Reichardt[26]：该算法将社区结构理解为自旋组态，使其最小化自旋玻璃态的能量而得到一种社区划分。

（2） SpectralClust[27]：对于图$G(V,E)$，利用基于谱分解的图划分算法定义代价函数，求解优化问题得到一种社区划分。

实验时取算法（1）Reichardt的参数为［3 2.5 2 1.5 1 0.5］，算法（2）SpectralClust的参数为6，不同尺度参数的社区结构同时作为输入，因此具有层次信息和重叠信息。网络中的环选取3阶环，即k=3。由Reichardt算法、3阶环作为输入的方法记为USI-1，由SpectralClust算法、3阶环作为输入的方法记为USI-2。

8个数据集分别为：（1） football（FB）[28]，（2） netscience（NS）[29]，（3） London transport1（LT）[30]，（4） CKM-3[31]，（5） A01[32]，（6） euroroad（ER）[33]，（7） Opsahl_powergrid（OP）[34]，（8） FWFB[35]。

8 个数据集的网络统计特性如表1所示

表1 8个网络数据集的统计特性

网络	|V|	|E|	<k>	<d>	C	r	H
FB	115	613	10.66	2.51	0.403	0.162	1.01
NS	1589	2742	3.45	1.33	0.889	0.462	2.01
LT	368	312	1.70	13.96	0.324	0.138	1.60
CKM-3	246	423	3.44	4.24	0.356	0.102	1.33
A01	311	640	4.12	4.15	0.374	0.115	1.93
ER	1174	1417	2.41	6.35	0.237	0.127	1.24
OP	4941	6594	2.67	18.99	0.416	0.0035	1.45
FWFB	128	2075	32.42	1.78	0.337	−0.112	1.24

其中|V|表示网络中的节点数，|E|表示网络中的边数，<k>表示网络中节点的平均度，<d>表示网络中节点对的平均距离，C表示网络的平均集聚系数，r表示网络的关联系数，H表示度的分布熵。

在每个数据集上计算节点对的链路预测得分，

每个数据集单独计算10次，每次独立地划分训练测试集，训练集比例占90%，测试集比例占10%，最终取10次计算的平均值作为最终链路预测结果。用AUC指标作为评价，得到以下实验结果：

表2　基于USI模型的链路预测算法与其它结构相似性链路预测算法结果比较

网络	USI-1	USI-2	CN	AA	RA	PA	LP	LRW4	LRW5	SRW3	SRW4	ACT
FB	**0.920**	**0.904**	0.866	0.865	0.863	0.246	0.877	0.899	0.896	0.888	0.895	0.569
NS	**0.999**	**0.999**	0.989	0.989	0.988	0.710	0.998	0.908	0.907	0.904	0.902	0.553
LT	**0.830**	**0.768**	0.590	0.591	0.590	0.686	0.697	0.513	0.539	0.503	0.509	0.752
CKM-3	**0.960**	**0.949**	0.848	0.847	0.849	0.638	0.922	0.870	0.866	0.836	0.870	0.648
A01	**0.866**	0.860	0.782	0.781	0.780	0.807	**0.862**	0.730	0.734	0.724	0.739	0.631
ER	**0.911**	**0.882**	0.548	0.547	0.547	0.426	0.594	0.654	0.690	0.594	0.654	0.768
OP	**0.967**	**0.932**	0.637	0.645	0.638	0.577	0.707	0.764	0.798	0.711	0.757	0.892
FWFB	**0.747**	0.688	0.627	0.632	0.638	**0.740**	0.642	0.699	0.802	0.734	0.726	0.734

实验选取了若干有代表性的基于节点相似性的链路预测算法，包括基于共同邻居的CN[36]算法，基于共同邻居和节点度加权的AA[37]、RA[8]算法，偏好连接相似性的PA[38]算法，基于局部路径的LP[8]算法，基于随机游走的LRW[39]、SRW[39]算法（其后数字表示步数，如LRW4表示随机游走的步数到4），以及全局相似性算法ACT[40]等10种。由表中实验数据可得知，仅使用网络的社区结构和3阶环信息的USI模型的链路预测算法，即可显著提升局部结构相似性和全局相似性算法的AUC指标。尤其在LT、ER、OP等数据集上，其它相似性指标的AUC在0.6左右，几乎不可用，而USI模型AUC可达0.9左右，预测准确性显著提升，从而体现了各种规模的网络结构信息对链路形成的影响。

从算法的效率上，设网络中节点数为N，则结构相似性指标CN、AA、RA、PA的复杂度为$O(N\cdot<k>^2)$；LP的复杂度为$O(N\cdot<k>^3)$；LRW、SRW的复杂度为$O(N\cdot<k>^n)$，其中n是随机游走的步数；ACT的复杂度为$O(N^3)$。USI模型的链路预测算法时间复杂度主要分为三部分，一是划分社区结构，二是集合上度量p的估计，三是计算并联概率。社区发现算法Reichardt的时间复杂度为$O(N^3)$，社区发现算法SpectralClust的时间复杂度为$O(N)$。设算法划分了$M(M\ll N)$个社区结构组成M个0阶集合和C_M^2个1阶集合，则根据式（22），阶集合和1阶集合上度量p的估计的复杂度分别为$O(M)$和$O(C_M^2)\approx\frac{1}{2}O(M^2)$。3阶环组成的0阶集合上度量$p$的估计可以等价转换为1次稀疏矩阵的乘法和归一化操作，复杂度为$O(N<k>^2)$。设共有$N_2(N_2<C_N^2)$个节点对同时从属于2个以上集合，则根据式（27）并联概率的复杂度为$O(<N_{xy}>\cdot N_2)\ll<N_{xy}>O(N_2)$，其中$<N_{xy}>$表示节点对$xy$所属共同集合个数的平均值。因此USI-1的复杂度约为$O(N^3+M^2+N_2)$，USI-2的复杂度约为$O(N+M^2+N_2)$。从实验结果上分析，USI-1算法的链路预测准确性普遍高于USI-2算法，这种预测准确性的提升是以算法的时间复杂度换取的，从而也间接说明网络的中观社区结构的质量对链路预测有重要影响，进而验证了USI模型链路预测算法假设的合理性。对于大规模网络，可选用USI-2算法，或在集合的划分中选用其它时间复杂度较低的社区发现算法。另外，1阶集合的划分具有灵活性，在大规模网络中可灵活调整1阶集合的划分数量，降低计算复杂度。

6　结束语

USI模型用笛卡儿积、幂集、离散度量空间等概念对多维度网络特征做出统一描述，相比经典的链路预测模型和结构相似性指标而言所能表达的信息更多，因此更好地反映了网络的本质特征。数据实验结果也验证了基于USI模型的链路预测方法可以更准确地进行预测。算法的本质是利用USI模型对输入信息进行前端融合，这种融合的好处是物理意义明确，因而相比其它对结构信息综合利用的算法，例如基于机器学习的分类算法，该算法更符合实际网络的生成演化机制，所以可以更好展现各种规模的网络结构对链路形成的影响；而有监督学习算法则是对网络结构信息进行后端

融合，其提高准确度的机理是对目标函数进行优化，将多维度信息拟合成多元标量函数，使拟合函数值达到最佳AUC，但往往缺乏算法的解释性。

本文提出的基于USI模型的链路预测方法只是该模型的一种实现，该方法的准确性还可通过融合结构相似性指标等方式进一步提升，可作为下一步的研究方向。

值得注意的是，当模型输入仅为社区发现算法划分的社区结构时，USI模型的链路预测AUC值可以作为社区发现算法的评价指标，从而可以对重叠、非重叠社区进行统一评价，且可以避免经典的模块度评价体系对于重叠社区评价普遍低于对非重叠社区评价的问题。

参考文献:

[1] GuillermoGarcía-Pérez, RoyaAliakbarisani, AbdorasoulGhasemi, et al. Precision as a measure of predictability of missing links in real networks. . 2020，101(5-1)：052318.

[2] Zhang J, Yu P S. Link Prediction[M]. Broad Learning Through Fusions, An Application on Social Networks. 2019. doi：10. 1007/978-3-030-12528-8_7

[3] 谭索怡,祁明泽,吴俊,吕欣. 复杂网络链路可预测性:基于特征谱视角[J]. 物理学报,2020,69(08)：188-197.
TAN Suoyi, QI Ming ze, WU Jun, et al. Link predictability of complex network from spectrum perspective[J]. Acta Phys. Sin. 2020，69(8) 188-197. DOI: 10. 7498/aps. 69. 20191817

[4] CANNISTRACI C V, ALANISLOBATO G, RAVASI T. From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks[J]. Scientific Reports，2015，3(4)：1613-1613.

[5] TUNINETTI M, TAMEA S, LAIO F, et al. To trade or not to trade: Link prediction in the virtual water network[J]. Advances in Water Resources，2016.

[6] 李冬,申德荣,寇月,林梦儿,聂铁铮,于戈. 基于层次化混合特征图的链路预测方法[J]. 中国科学:信息科学,2020,50(02)：221-238.

[7] 吕琳媛, 周涛. 链路预测[M]. 高等教育出版社，2013：58-66.
LinyuanLÜ, ZHOU Tao. Link Prediction[M]. High Education Press，2013：58-66.

[8] ZHOU T, LÜ L, ZHANG Y C. Predicting missing links via local information[J]. The European Physical Journal B，2009，71(4)：623-630.

[9] YAO Y, ZHANG R, YANG F, et al. Link prediction based on local weighted paths for complex networks[J]. International Journal of Modern Physics C，2017.

[10] LIU S, JI X, LIU C, et al. Similarity indices based on link weight assignment for link prediction of unweighted complex networks[J]. International Journal of Modern Physics B，2016.

[11] 王凯,李星,兰巨龙,卫红权,刘树新. 一种基于资源传输路径拓扑有效性的链路预测方法[J]. 电子与信息学报,2020,42(03)：653-660.

[12] 刘树新, 李星, 陈鸿昶,等. 基于资源传输匹配度的复杂网络链路预测方法[J]. 通信学报，2020，041(006)：70-79.

[13] BISWAS A, BISWAS B. Community-based link prediction[J]. Multimedia Tools and Applications，2017：1-21.

[14] DE BACCO C, POWER E A, LARREMORE D B, et al. Community detection, link prediction and layer interdependence in multilayer networks[J]. arXiv preprint arXiv:1701. 01369，2017.

[15] MALLEK S, BOUKHRIS I, ELOUEDI Z, et al. Evidential Link Prediction Based on Group Information[C] International Conference on Mining Intelligence and Knowledge Exploration, Mike. 2015.

[16] HE Y L, LIU J N K, HU Y X, et al. OWA operator based link prediction ensemble for social network[J]. Expert Systems with Applications，2015，42(1)：21－50.

[17] 吴祖峰, 梁棋, 刘峤,等. 基于AdaBoost的链路预测优化算法[J]. 通信学报，2014(3)：116-123.
WU Zufeng, LIANG Qi, LIU Qiao, et al. Modified link prediction algorithm based on AdaBoost[J]. Journal on Communications，2014(3)：116-123.

[18] YU H T, WANG S H, MA Q Q. Link prediction algorithm based on the Choquet fuzzy integral[J]. Intelligent Data Analysis，2016，20(4)：809-824.

[19] 吴翼腾, 于洪涛, 黄瑞阳,等. 采用组合方法进行链路预测的理论极限研究[J]. 通信学报，2020，041(006)：34-50.

[20] MARTÍNEZ V, BERZAL F, CUBERO J C. A Survey of Link Prediction in Complex Networks[J]. ACM Computing Surveys (CSUR)，2016，49(4)：69.

[21] CLAUSET A, MOORE C, NEWMAN M E. Hierarchical structure and the prediction of missing links in networks.[J]. Nature，2008，453(7191)：98-101.

[22] HOLLAND P W, LASKEY K B, LEINHARDT S. Stochastic blockmodels: First steps[J]. Social Networks，1983，5(2)：109-137.

[23] GUIMER R, SALES-PARDO M. Missing and spurious interactions and the reconstruction of complex networks[J]. Proceedings of the National Academy of Sciences of the United States of America，2009，106(52)：22073-8.

[24] 邱振宇,胡文斌,聂聪,唐传慧,刘中州,高旷,许平华. 基于三元组的社会网络演化分析方法[J]. 计算机学报,2019,：1-17.

[25] 王守辉, 于洪涛, 黄瑞阳,等. 基于模体演化的时序链路预测方法[J]. 自动化学报，2016(5)：735-745.
WANG Shouhui, YU Hongtao, HUANG Ruiyang, et al. A Temporal Link Prediction Method Based on Motif Evolution[J]. Acta Automatica Sinica，2016(5)：735-745.

[26] HESPANHA P. An Efficient MATLAB Algorithm for Graph Partitioning[J]. 2004.

[27] GIRVAN M, NEWMAN M E J. Community structure in social and biological networks[J]. Proceedings of the National Academy of Sciences of the United States of America，2002，99(12)：7821-6.

[28] NEWMAN M E J. Finding community structure in networks using the eigenvectors of matrices.[J]. 2006，74(3 Pt 2)：036104.

[29] DOMENICO M D, SOLÉRIBALTA A, GÓMEZ S, et al. Navigability of interconnected networks under random failures[J].

Proceedings of the National Academy of Sciences，2014，111(23)：8351-6.

[30] COLEMAN J, KATZ E, MENZEL H. The Diffusion of an Innovation among Physicians 1[J]. Sociometry，1977，20(20)：253-269.

[31] , datasetsPAJEK[OL]: http://vlado. fmf. unilj. si/pub/networks/data/mix/mixed. htm accessed May 2010. 2016. 1.

[32] ŠUBELJL. , BAJECM. . Robust network community detection using balanced propagation[J]. The European Physical Journal B，2011，81(3)：353-362.

[33] WATTS DJ, STROGATZ SH. Collective dynamics of 'small-world' networks[C]. Nature. 1998：440-442.

[34] ULANOWICZ R E, BONDAVALLI C, EGNOTOVICH M S. Network Analysis of Trophic Dynamics in South Florida Ecosystem，FY97: The Florida Bay Ecosystem [R/OL]. Technical report，CBL，1998：98-123.

[35] LORRAIN F, WHITE H C. Structural Equivalence of Individuals in Social Networks [J]. Social Networks，1971，1(1)：67-98.

[36] ADAMIC L A, ADAR E. Friends and neighbors on the Web[J]. Social Networks，2003，25(3)：211-230.

[37] BARABASI A L, ALBERT R. Emergence of scaling in random networks[J]. Science，1999，286(5439)：509-512.

[38] LIU W, LÜ L. Link Prediction Based on Local Random Walk[J]. EPL (Europhysics Letters)，2010，89(5)：58007-58012(6).

[39] KLEIN D J, RANDIĆ M. Resistance distance [J]. Journal of Mathematical Chemistry，1993，12(1)：81-95.

[作者简介]

吴翼腾（1992-）：男，博士研究生，研究方向为信息内容安全、对抗机器学习.

顾泽宇（1993-）：男，硕士，研究方向为网络大数据挖掘、网络安全.

于溆乔（1998-）：女，研究方向为网络大数据挖掘。

一种基于业务驱动的软件定义互连RapidIO控制器架构

李沛杰[1]，沈剑良[1]，吕平[1]，董春雷[1]，汪欣[2]，张传波[1]
[1]中国人民解放军战略支援部队信息工程大学，河南 郑州 450002；
[2]天津市滨海新区信息技术创新中心，天津 300457

摘　要：针对现有RapidIO控制器电路在业务处理时，逻辑资源冗余和可扩展性差的问题，提出一种基于业务驱动的软件定义互连（Traffic-Driven Software Defined Interconnection，TSDI）机制，通过设计统一的软件定义互连接口标准和归一化的互连拓扑，实现一种基于业务流处理特性的软件定义互连RapidIO控制器，在解决控制器灵活互连的同时，实现逻辑资源，延迟等关键指标的兼顾。实验结果表明，相比于传统的固定设计的RapidIO控制器电路，所设计的TSDI-Based RapidIO控制器最小收发延迟56.1ns，逻辑资源相比传统设计增加29.1%。
关键词：软件定义互连、业务驱动、控制器、接口标准、互连拓扑

A RapidIO Controller Based on Traffic-Driven Software Defined Interconnection Architecture

Li Peijie[1], Shen Jianliang[1], Lv Ping[1], Dong Chunlei[1], Wang Xin[2], Zhang Chuanbo[1]
1.Information Engineering University, Zhengzhou, Henan 450002;
2.Information Technology Innovation Center of Tianjin Binhai New Area, Tianjin 30045

Abstract: Aiming at the problems of redundant logic resources and poor scalability of existing RapidIO controller circuits in traffic processing, a Traffic-Driven Software Defined Interconnection (TSDI) mechanism is proposed, through the design of unified software defined interconnection interface standard and normalized interconnection topology, a kind of software defined interconnection RapidIO controller based on traffic flow processing characteristics is realized, which can not only solve the flexible interconnection of controllers, but also take into account the key indicators such as logical resources and delay. The experimental results show that compared with the traditional fixed RapidIO controller circuit, the TSDI based RapidIO controller designed in this paper has a transceiver delay of xxns, and the logic resource is reduced by XX%, but the area is increased by XX%.
Key words: SDI; Traffic-Driven; controller; Interface standards; interconnection topology

1　引言

以总线为中心的体系结构，已成为当前信息系统的主流[1]，尤其在以交换互连总线为中心的嵌入式系统中，负责业务处理的RapidIO控制器的性能和可扩展性直接决定系统的性能和可扩展性。RapidIO互连标准作为全球唯一的嵌入式系统互连国际标准，是面向高性能嵌入式系统互连通信的关键[2]。基于RapidIO控制器的嵌入式总线互连体系结构，是一种高性能、基于业务流处理的可靠互连体系结构，具有高带宽、低时延、高效率、高可靠性等优点。

RapidIO控制器在业务处理时呈现出流处理的特征，当前针对RapidIO控制器的诸多研究主要围绕成熟的IP核[3~5]展开，文献[2]在以交换总线中心加主控节点的架构上，提出一种软件可配置的RapidIO总线系统，用于动态分配路由、速率和带宽等配置。文献[6]在现有的Xilinx软IP的基础上，基于描述符的DMA传输模式，解决业务的高速处理问题。文献[7]针对传统控制器的结构缺陷，最

基金项目：国家科技重大专项核高基项目(No. 2016ZX01012101)

大限度利用资源，设计一种支持多种端口模式的RapidIO控制器。上述研究虽然试图解决RapidIO业务处理的性能和可扩展性问题，但在RapidIO控制器的设计上依然基于现有的成熟IP，采用固化的分层硬件处理架构，其在处理不同业务及协议升级时，需设计并行的多个硬件电路，从而造成资源的冗余和较低的可扩展性，这种刚性的互连结构使得基于RapidIO业务的流处理过程呈现出拓扑的固定和单一，新增业务的处理只能在协议栈自底向上分别增加新的旁路逻辑，这导致了RapidIO控制器架构随着业务的增加而呈现膨胀冗杂。

本文根据RapidIO协议的流处理特性，打破传统的分层控制器结构，构建业务驱动的软件定义化算粒资源池，并提出一种基于业务驱动的软件定义互连机制（Traffic-Driven Software Defined Interconnection，TSDI），通过定义一种基于流地址的软件定义互连接口协议，实现RapidIO控制器内部软件定义化算粒的独立地址输入和寻址输出，实现业务类型驱动的控制器处理。同时提出一种多端口共享信道的时分复用机制，结合软件定义化算粒接口的协议变动，实现接口带宽的最大优化，解决业务流量驱动的控制器处理。基于业务类型和流量驱动的控制器处理流程可抽象出归一化的互连算粒，这种归一化的算粒，一方面可以通过灵活简化业务流程实现目标性能的优化，另一方面基于互连算粒的组合可实现复杂互连拓扑的迭代，解决协议向上扩展的平滑性问题。

2 TSDI结构简介

TSDI主要由软件定义化处理算粒和软件定义互连网络构成，其中软件定义化处理算粒（Software-defined Grain，SDG）主要基于硬件可重构思想和软件定义协议控制器数学模型[8]，抽取不同业务处理下的私有算粒和公共算粒，并构成算粒资源池，其算粒类型可归一化为计算C，存储M和互连I三类；软件定义互连网络（Software-defined Interconnection，SDI）主要基于业务等效处理原则，根据软件配置的算粒上一跳地址输入AI和下一跳寻址输出AO，将软件定义化的算粒进行互连。如上所述，TSDI可抽象为如下模型：

$$TSDI_{out} = Flow_n(SDG_{n_m}, SDI_{n_m})$$

其中，$TSDI_{out}$表示业务的流输出，$Flow_n(SDG_{n_m}, SDI_{n_m})$表示业务处理的第n级流输出，$Flow_n(SDG_{n_m}, SDI_{n_m})$可表示为n-1级流输出经过第n处理后的输出，模型如下：

$$Flow_n(SDG_{n_m}, SDI_{n_m}) = Flow_n(Flow_n(SDG_{n-1_{m1}}, SDI_{n-1_{m1}}))$$

SDG_{n_m}表示第n级流处理的m个软件定义化算粒组合，其具体指从算粒资源池中选取的归一化算粒，可抽象为如下模型：

$$SDG_{n_m} = \{ F_{C,M,I}(0), F_{C,M,I}(1), \cdots, F_{C,M,I}(m) \}$$

SDI_{n_m}表示第n级流处理的m个软件定义化算粒的互连函数，其主要描述SDG中各个算粒的互连关系，其模型可抽象如下：

$$SDI_{n_m} = \{ F_{C,M,I}(0)(AI,AO), \cdots, F_{C,M,I}(m)(AI,AO) \}$$

TSDI在处理RapidIO业务时，会根据业务类型的不同（如IDLE1，IDLE2，IDLE3，控制符，DoorBell报文，NREAD报文，NWRITE报文等），将流处理相关算粒通过对应的拓扑进行互连。TSDI具有网络泛化的特性，可根据业务的不同，对各个软件定义算粒挂接的互连接口进行定义，以最大限度的利用互连接口的带宽，并实现时延的降低，TSDI也可扩展新的接口，并支持新增接口的独立软件定义，以最小代价实现RapidIO协议面向未来的升级、扩展及转变。

如图1所示为TSDI的一种典型业务处理示意图，其在处理IDLE序列解析时，从解码，解扰到错误检测，业务呈现一张串行流处理的特性，TSDI将拓扑优化为串行的处理形式，而在处理报文时，则存在并行业务处理的特性，TSDI将拓扑优化为混合的处理形式。由于TSDI可基于业务驱动优化拓扑结构，因此基于不同的处理业务，TSDI可实现固定的传输时延，这使得TSDI可以基于应用需灵活的修改网络拓扑（如图示为实现低时延，跳过CRC检测的算粒串行连接），实现传输时延等性能的优化，提升应用效率。

3 基于TSDI的RapidIO控制器架构

RapidIO协议的业务处理基本遵从协议栈的多层网络架构，其业务的处理呈现出流处理的特性，如图2所示为RapidIO协议业务处理的基本框架图。

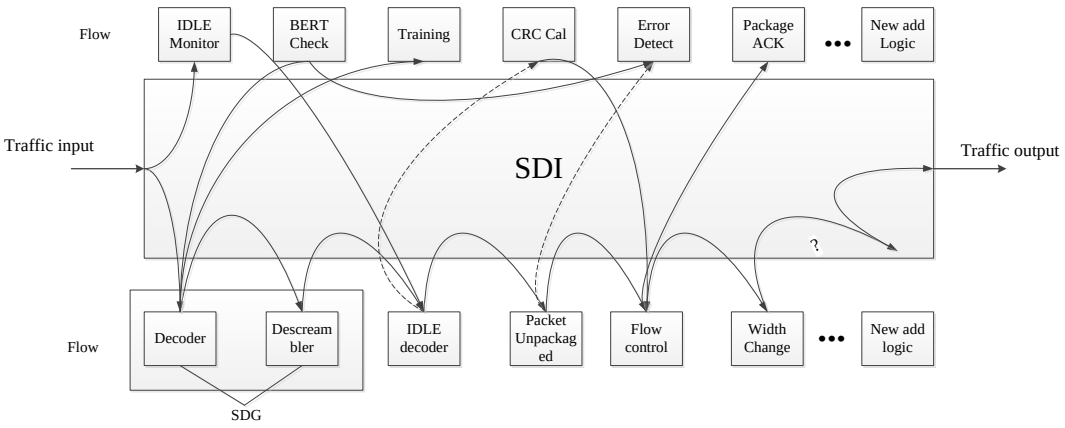


图 1　TSDI 典型业务处理示意图

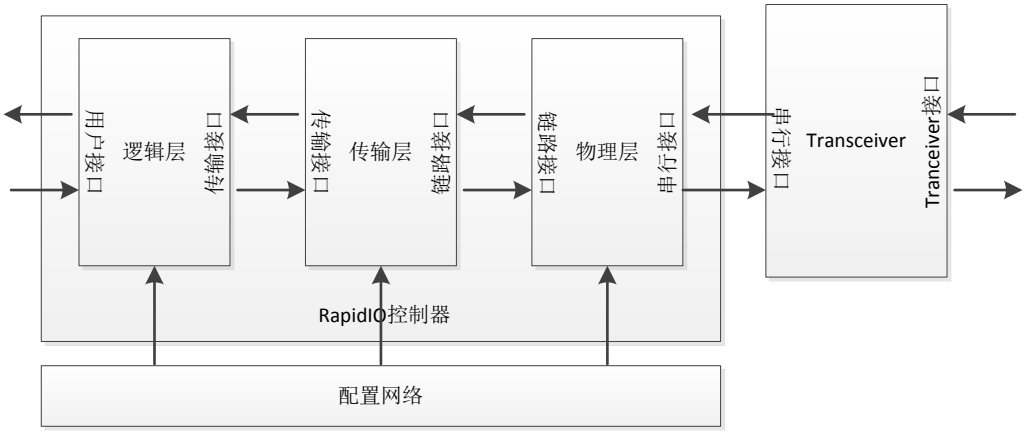


图 2　传统 RapidIO 协议业务处理基本框图

RapidIO 业务流从信道接收到的数据先经过 Transceiver 的串并转换，bit 同步，然后交由物理编码子层进行码元的解码（解扰，解码，纠错检错等），同时基于协议的不同，进行通道数据的处理和识别，从而支持可靠的或者非可靠的数据传输，并基于信息识别实现通信双方的信息交互。物理层和编码子层识别的数据，经过非报文数据的剥离，业务数据会在传输层进行报文解析，并基于业务流量的情况，进行数据的存储和读取，同时针对不同的业务对报文的头信息进行对应处理，处理完成的报文数据将在逻辑层进行报文负载内容的处理，同时传输层还将需响应的信息反馈到发送侧并通过物理层将响应的信息发送给对端。从而完成整个业务的处理。

从 RapidIO 业务的处理流程能够看出，业务在遵从流处理的同时，呈现出流分支，流可逆的特性及流存储和流计算的处理方式，且在业务处理的整个过程中，下游逻辑和上游逻辑均可抽象为一种前级输入和后级输出的简单模型。如图 3 所示，基于业务的流处理描述，业务的处理过程被抽象成为多个子 Function 的流处理，而在每个 Function 中，基于软件定义化的算粒，同样通过流处理的方式互连起来。

整体来看，协议在业务流处理时，无论是基于计算或存储的软件定义处理算粒还是基于 Function 的互连算粒，其基本结构均可归纳为输入和输出的形态，而为实现基于业务的算粒柔性互连，其接口协议必须统一满足某种特定的规范，因此必须研究确认一种软件定义互连的接口协议，这种接口协议不仅约束整个控制器的输入输出接口，也约束控制器各个协议层输入输出接口，同时更对算粒的输入输出接口进行约束，本文提出一种基于流地址的软件定义互连接口协议，其基本思路是基于流地址输入，算粒通过查找地址映射决定算粒输出的下一跳算粒，依次类推，实现整个业务中所有算粒的连接。

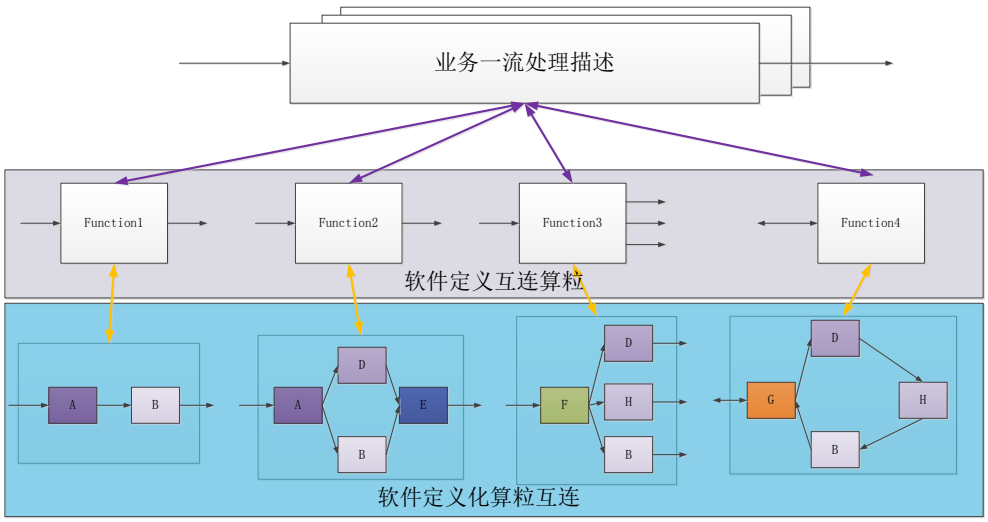


图 3　业务抽象为软件定义化算粒过程示意

定义一：业务的流处理为 $T(T_i, n, T_o)$。

其中 T_i 表示业务输入的 Function，我们称之为源业务输入互连算粒 STIG，T_o 表示业务输出的 Function，我们称之为目的业务输出互连算粒 DTIG，n 表示从 T_i 到 T_o 之间的跳数约束。

定义二：Function 的流处理为 $F(F_i, m, F_o)$。

其中 FI 表示互连 Function 中处理算粒输入，我们称之为源功能输入处理算粒 SFPG，FO 表示互连 Function 中处理算粒输出，我们称之为目的功能输出处理算粒 DFPG，m 表示从 FI 到 FO 之间的跳数约束。

定义三：处理算粒的处理函数为 $P(i, o)$。

其中 i 表示处理算粒 SDG 的输入，o 表示处理算粒 SDG 的输出。

因此，基于 TSDI 的 RapidIO 控制器业务的流处理可以表示为 $T\left(P_i(i, o), n, P_{m_{i+n-1}}\right)$。

3.1　基于流地址的软件定义互连接口协议

如上所述，业务的处理可以描述为 SDG 算粒输入和输出的关系。因此软件定义互连网络中所使用的处理和互连的算粒具备实现归一性设计的条件。考虑到算粒的互连均通过流地址的映射，因此在接口协议中必须具备流地址的输入和输出，同时接口协议需支持多输入的聚合和重分配，因此接口必须具备位宽可改变的输入输出数据接口，而为兼顾互连网络中各节点互连的灵活性和物理实现的可行性，输入输出接口应具备双向特性；此外，为实现接口带宽的可定义，接口协议需支持工作时钟的定义，因此在整个接口协议中必须具备时钟接口，整个接口协议可定义如下：

定义四：定义软件定义化算粒的互连接口为 $I\left(S_{clk}, S_{rst}, D_{clk}, D_{rst}, D_i, D_o, P_{dir}, S_a, D_a\right)$。

其中，S_{clk} 表示算粒输入的工作时钟，S_{rst} 为算粒输入的复位信号，D_{clk} 为算粒输出的工作时钟，D_{rst} 为算粒输出的复位信号，D_i 为输入的数据，D_o 为输出的数据，其接口位宽可通过软件定义确认其有效位宽及每 bit 的含义，D_i 和 D_o 方向均可以是双向的，P_{dir} 为数据接口的方向描述原语，在物理实现，可以描述 D_i 或 D_o 是水平方向上的输入输出，也可是垂直方向上的输入输出。S_a 表示本级算粒的地址输入，也是上一级算粒的寻址输出，D_a 表示本级算粒的寻址输出，也是下一级算粒的地址输入。

为最大限度的简化互连网络，降低物理实现的难度，整个互连网络将不再构建基于“米”字形的互连网络，而采用如图 4 所示的棋盘格互连网络形态。

其中 TIG 为业务流的互连算粒，其具有两输入两输出的形态，用于实现处理算粒的最快互连，TIG 内部的互连方式可灵活设计，具体的拓扑将在 3.2 节描述，TIG 在处理业务类型驱动时将相邻的处理算粒互连导通，其在互连的同时一般采用组合逻辑实现，此时时钟接口将采用同步设计，当处理业务流量驱动时，在业务流量切换的相邻算粒处，TIG 将通过 Gearbox 的设计实现算粒之间带宽匹配。

FPG 为业务流的处理算粒，其同样具有两输

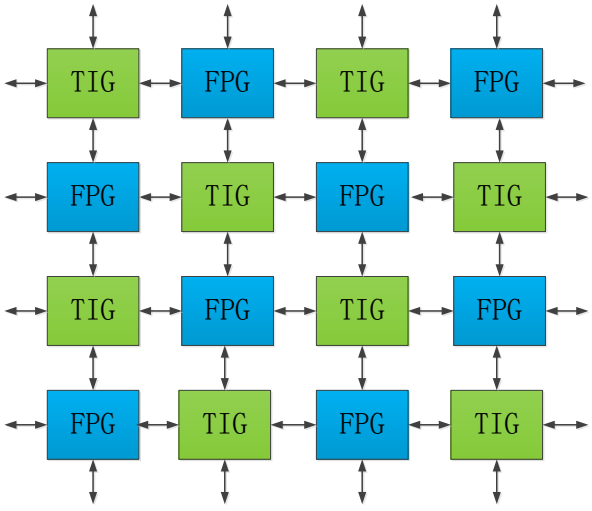


图 4　基于 TSDI 的棋盘格互连网络形态

入两输出的形态，用于实现不同 Function 下的算粒互连，其中可以基于更简化的互连算粒 SDI 及处理算粒 SDG 进行连接，一般来讲，FPG 内部的处理算粒和互连算粒基本采用同时钟域的处理，其内部的算粒在某一 Function 上具有一定的流处理特性，当然也支持业务的软件定义互连。

TSDI 中互连算粒及处理算粒的接口采用了一种基于流地址的软件定义接口协议，其创新性在于基于业务流处理特性抽取的基本算粒在接口带宽和方向上能够实现软件定义，算粒之间的互连则基于流地址的映射确定算粒的上级输入及下一跳的输出，从而在业务处理确定后，整个业务的互连方式将基本固定，这使得业务流处理下的关键特性将能够提前预知。

3.2　基于业务驱动的软件定义互连拓扑

在如图 4 所示的棋盘格互连网络中，每个 FPG 和 TIG 均采用两输入两输出的互连拓扑结构，而对于基本的处理算粒 SDG 和互连算粒 SDI，其接口拓扑必须更加简单，否则难以实现拓扑的归一。在如图 3 所示的软件定义化算粒互连结构中，算粒均存在输入和输出的接口，其基本的抽象模型可如图 5 所示，其中包括了单入单出，多入单出，单入多出及自入自出等四种形态。显然这对于接口形态来讲依然还很复杂，在软件定义互连接口的设计中，最简单的方式即为单入单出的形态，因此需要探索软件定义互连的接口协议，以实现互连拓扑形态的归一。

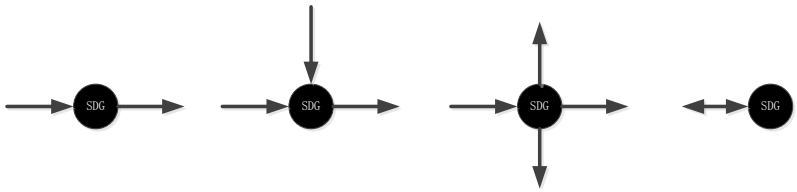


图 5　软件定义化算粒抽象模型

如上所述，要实现互连拓扑的形态归一，需对多入单出、单入多出及自入自出三种形态的拓扑进行等效处理：

对于多入单出的拓扑形态，在上级算粒输出时，其目的流地址均为相同的地址，其本质是多个相同目的流地址业务的汇聚，具体表现是 D_i 带宽的变动，而基于软件定义互连接口协议的约束，D_i 接口位宽可以软件定义，因此，对于多输入单输出的拓扑结构，其可以抽象为单输入单输出的拓扑形态，唯一需要注意的是，当接口位宽的扩展超过软件定义互连接口规范定义的最大位宽时，多输入单输出的算粒将扩展为多路并行的串行拓扑结构，直到处理带宽能够满足多路合入的算粒为止，或者将改变算粒的处理速率，使算粒处理带宽提高。其两种处理结构的拓扑如图 6 所示。

对于单输入多输出的拓扑形态，其实质是单输入的流地址被同时映射到多个不同的下一跳算粒，此种拓扑结构相对好理解，可以将该算粒复制多份，然后通过相同的输入，驱动多个不同的下一跳算粒。其处理结构可如图 7 所示。

对于自输入自输出的拓扑形态，其实质是源流地址与目的流地址相同的自环形态，其基本架构依然遵从单输入单输出的形态，配合 SDI 算粒可以很简单的实现该种形态，如下图所示。

综上，在算粒之间的互连拓扑中，其基本拓扑均可以归一为单输入输出的拓扑结构，结合基

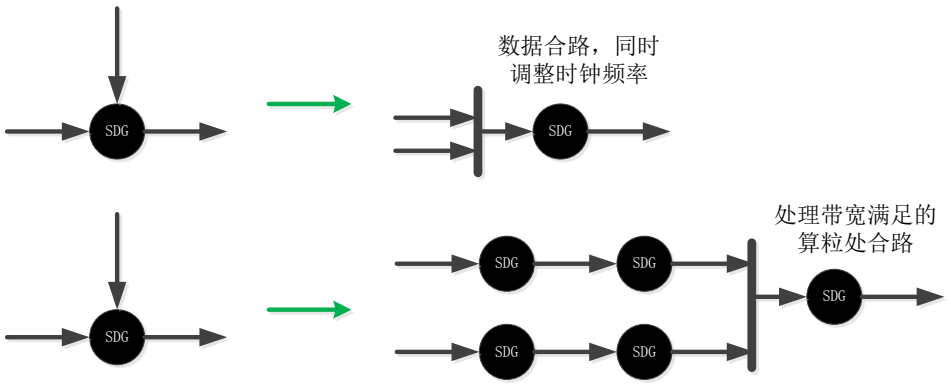


图 6　多输入单输出算粒的拓扑等效结构

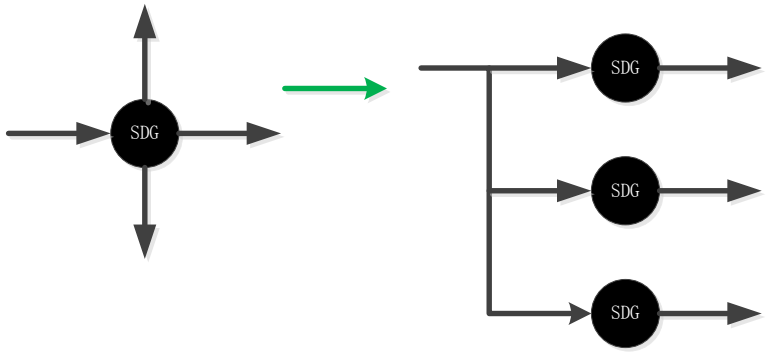


图 7　单输入多输出算粒的拓扑等效结构

图 8　自输入自输出算粒的拓扑等效结构

于流地址的软件定义互连接口协议，其接口的可定义特性及输入输出的双向特性，使得拓扑结构得以简化。

基于归一的算粒拓扑，在如图 3 所示的业务处理流程中，软件定义化算粒的互连拓扑主要表现为流拓扑，分流拓扑，环形拓扑，星型拓扑等形态，Function 内部的互连拓扑可软件定义，其中的算粒基于单输入单输出形态进行设计，其接口满足基于流地址的软件定义互连接口协议，与 FIG，TPG 不同的是，IG，PG 在设计过程中增加了接口输入输出方向（水平/垂直）及方向有效接口位宽的定义，四方向的接口仅有等效接口位宽的方向及数据有效，其结构如图 9 所示。

在上图所示的基本拓扑的结构中，SDI 和 SDG 间基本互连拓扑的有效组合与连接将实现 FIG 及 TPG 在 Function 内部的互连，而 Function 之间的互连将实现业务的互连及处理。反之基于业务驱动的流处理描述将业务流处理分为多个 Function 的连接，Function 内部的 FIG 及 TPG 的连接依托统一的接口协议及互连拓扑，将负责的互连关系分解为简单的拓扑连接，从而实现业务驱动的软件定义互连拓扑。

3.3　基于 TSDI 的 RapidIO 控制器典型业务的实现

如上所述，基于 TSDI 的 RapidIO 控制器可用基于 SDI 和 SDG 互连的棋盘格形式进行实现。在业务处理过程中，TSDI 的 RapidIO 控制器基于业务处理类型和处理流量的不同，首先在软件层面生成最优化的配置执行文件，然后通过配置处理核和重配置处理核共同定义软件定义硬件中混合粒度算粒的互连模式，也可通过动态感知实时数据变化追踪硬件电路的特性，实时更新硬件电路

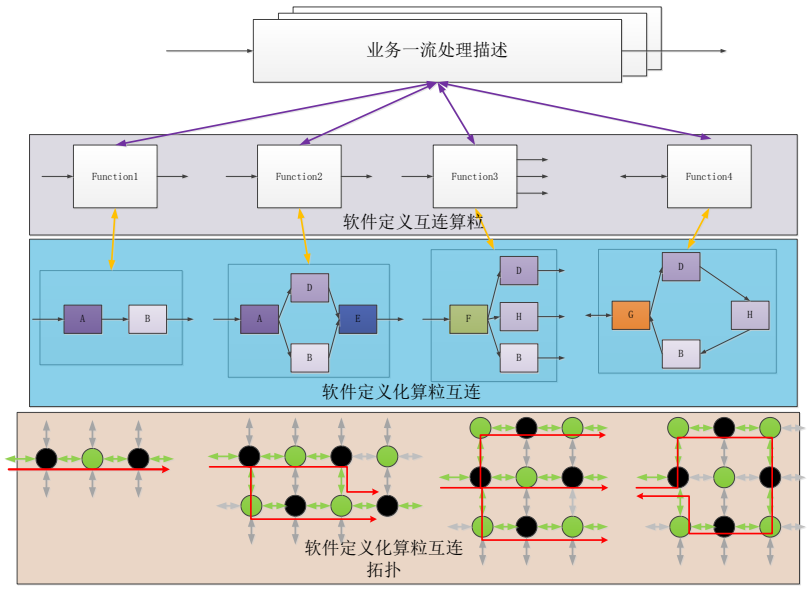


图 9　基于单输入单输出算粒形态的业务处理流程

的互连。配置执行文件按照软件定义互连接口协议的格式串行对 TSDI 中的 SDG 和 SDI 进行配置，配置完成后的网络将能够实现对应业务的处理。

如图 10 所示为 RapidIO 控制器中维护包业务处理的流程图，其输入的业务首先经过数据同步及解码，然后同步进行控制符，报文及 IDLE 序列的解析，对解析出来的维护包报文进行错误检测，并将正确进行存储，同时进行路由查表，当链路不拥塞和反压时，将存储的报文读出，并按照路由结果输出业务到下一级。

在维护包的业务处理流程中可以抽象出 5 个 Function，其中 Function1 主要实现数据同步及解码，其主要算粒包括 Bit 同步，解码和解扰；Function2 主要实现报文解析与错误检测，其主要算粒包括了报文解析和错误检测；Function3 主要实现报文转发，其主要算粒包括了错误检测，路由查表，报文存储及拥塞反压；Function4 主要实现路由查表与检测，其主要算粒包括错误检测和路由查表。维护包业务的处理流程可以抽象为 Function1，Function5，Function2，Function3，Function4 的流处理过程。其结构如图 11 所示。

基于归一化的软件定义互连拓扑结构，基于 TSDI 的 RapidIO 控制器在实现维护包处理时的结构如图 12 所示。

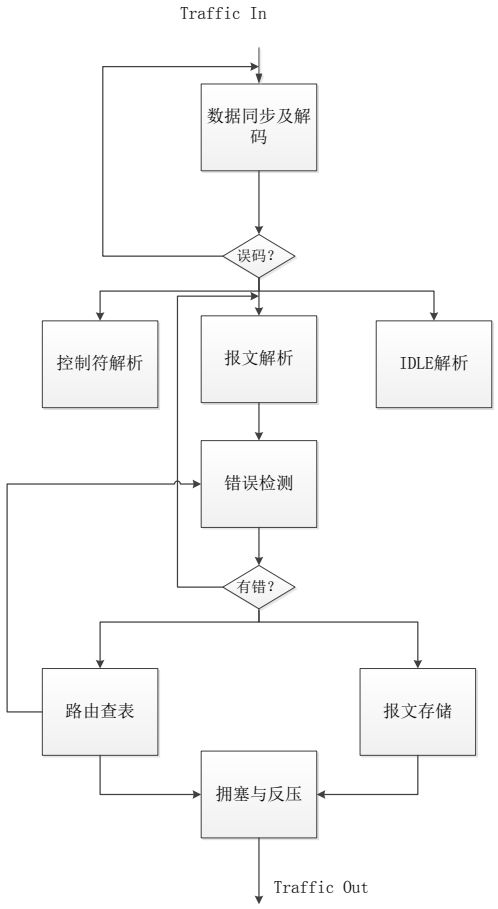


图 10　维护包业务处理的流程图

4　测试结果

本文基于 TSDI 实现了兼容 RapidIO Gen3/2/1 的控制器逻辑，如表 1 所示为本设计的主要参数与功能指标，与文献 [3~5] 所述的控制器进行了主要参数指标对比，可以看出，本文所设计的 RapidIO 控

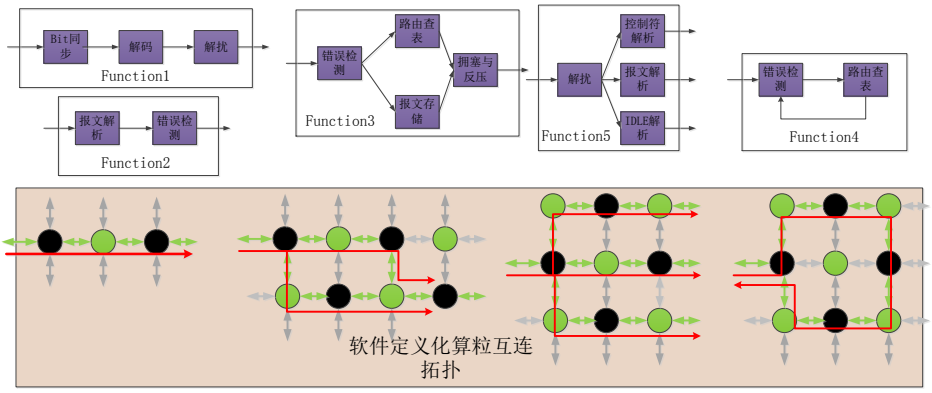


图 11　维护包业务流处理的抽象模型

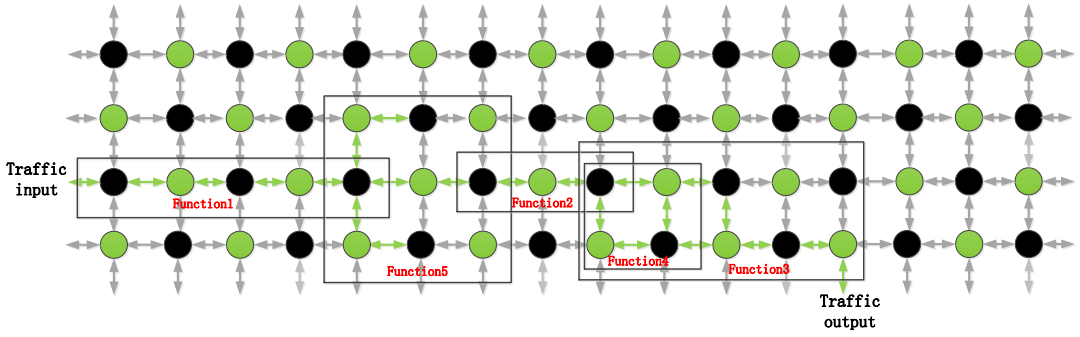


图 12　维护包业务驱动下的RapidIO控制器处理

制器在整体功能完备性上与Mobiveil及IDT设计的IP等同，但本设计在应用的灵活性和协议的可扩展性上有更优异的表现。

表 1　主要设计指标对比

主要参数指标	Xilinx 设计	Mobiveil设计	IDT设计	本文设计
支持协议	V2.2/V1.3	V3.1/2.2/1.3	V3.2/2.2/1.3	V3.2/2.2/1.3
端口模式	1x/2x/4x	1x/2x/4x	1x/2x/4x	1x/2x/4x
			2x+2x	2x+2x
			2x+1x	2x+1x+1x
			1x+1x	1x+1x+1x+1x
IDLE序列	IDLE2/1	IDLE3/2/1	IDLE3/2/1	IDLE3/2/1
高速侧接口位宽	20bit	10/20/40/64/67bit	10/16/20/32/40bit	10/16/20/32/40bit
用户侧接口位宽	128bit	256bit	256bit	64/128/256bit
DevID	16bit	8bit/16bit	8bit/16bit/32bit	8bit/16bit/32bit
寻址大小	34bit	34bit/50bit	34bit/50bit/66bit	34bit/50bit/66bit
链路训练	IDLE2	IDLE2/CW/DME	IDLE2/CW/DME	IDLE2/CW/DME
吞吐量	20Gbps	39.4Gbps	47.7Gbps	47.7Gbps
最大未确认报文	32	4096	128	1024可配置
V4.0可扩展性	不支持	不支持	不支持	支持

本文在性能的测试上，主要对比相似设计的时延和控制器的资源。其中对时延的测试主要从不同的包长度下的收发时延进行测试对比，采用与标准商用VIP进行对接的测试环境，通过VSC仿真波形进行收发时延的测试。对控制器资源的测试比较，主要通过Design Compiler工具综合后的数据进行对比。

5.1 延迟性能测试

对延迟的性能测试主要从两方面进行，一是不同包长度下的延迟情况，以验证不同控制器对不同业务所表现的不同延迟特性；二是不同业务负载下的延迟情况。本文主要对比IDT的RapidIO控制器设计，在测试时，分别将两种控制器IP的工作模式配置为1x模式12.5Gbps，并分别进行两组时延，其中第一组分别对比不同报文长度下单一报文的延迟特性，其中的延迟值为重复测量5次后的延迟平均值；第二组分别对比不同负载下小报文（64Byte）下的延迟特性，其中的延迟值为多个报文测试结果的平均值。如图13结果显示，最小报文长度下的延迟，本设计的要高于IDT的控制器延迟，但基本相当，这是因为TSDI的算粒实现软件定义互连接口协议的组合逻辑引入的，这种延迟的引入仅和业务所占用的算粒个数及拓扑结构相关，在不同报文长度下，本文所设计的RapidIO控制器在最大报文长度下的延迟变化不大，这主要是因为基于业务类型驱动的TSDI网络能够自动调整算粒的处理位宽，当传输报文长度变化时，可以最大限度的应用接口带宽，从而减小数据的处理周期；从不同报文负载下的延迟特性可以看出，基于TSDI的RapidIO控制器表现出来的特性更优，这主要是因为TSDI的RapidIO控制器支持最大1024个未确认报文，且基于软件定义互连接口，在系统软件的配合下，能够动态感知业务负载的大小，从而当业务负载增大时，动态的增大未确认报文的数量，但是当超过所能支持的最大范围后，TSDI的RapidIO控制器的延迟特性与IDT类似，呈现出线性增长的特性。

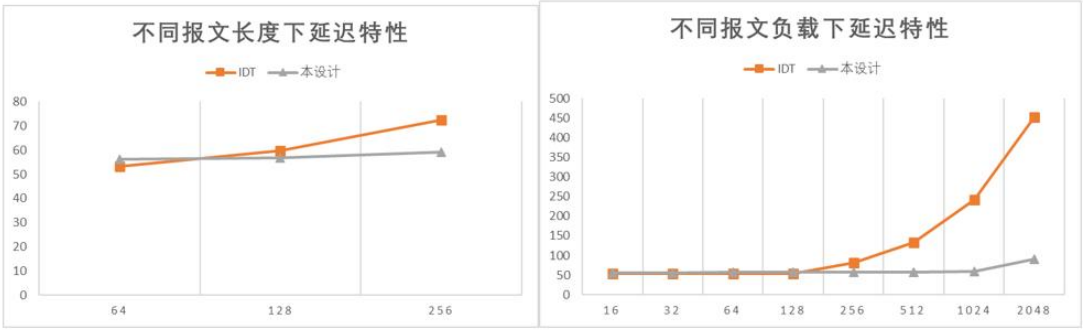


图13 传输延迟对比

5.2 控制器资源对比

本文通过在TSMC28 nm工艺上对所设计的RapidIO控制器进行综合，目标频率按照312.5MHz进行，同时对比IDT的10xN RapidIO控制器的资源，对比结果如表2所示，可以看出，Std Cell Instances的数量增长了8.33%，这主要取决于两方面的工作结果，一方面处理算粒软件定义化设计以及软件定义互连算粒的增加使得整体资源增大，但是由于公共算粒的提取，使得原有结构的冗余资源得到有效利用，整体资源减小。在存储大小上扩大了一倍，主要用于匹配算粒接口位宽最大256bit的情况，整体来看，相比于IDT的控制器面积增大了29.1%，基于TSDI的RapidIO控制器可以认为是用有限的资源提升换来了RapidIO业务处理的灵活可扩展。

表2 资源面积对比

对比项	IDT设计	本设计	增长百分比/%
Std Cell Instances	600K	650K	8.33%
Std Cell Area(mm^2)	0.56	0.61	8.92%
Memory Bits	214144	428288	100%
Memory Area(mm^2)	0.16	0.32	100%
Total Area(mm^2)	0.72	0.93	29.1%

6 结束语

本文基于业务驱动的软件定义互连机制设计了一款满足RapidIO 3.2协议规范的控制器电路，通过定义软件定义化算粒的接口规范及归一化互连拓扑结构，用有限的资源提升带来了RapidIO控制器的性能等价和灵活可扩展。TSDI互连机制具有网络的泛化性，对基于流处理特性的协议控制器均具有一定的参考价值和借鉴意义。

参考文献:

[1] 吕平,刘勤让,邬江兴,陈鸿昶,沈剑良．新一代软件定义体系结构[J]. 中国科学:信息科学，2018(03)，48:315-328.

[2] Chengfu D , Jiao L , Shiliang J , et al. Design of RapidIO Bus System Based on Software Configuration[J]. Microcontrollers & Embedded Systems，2020.

[3] Juncai S . Design of DMA High-speed Transmission Scheme Based on General RapidIO Controller[J]. Microcontrollers & Embedded Systems，2020.

[4] Xilinx. Serial RapidIO Endpoint LogiCORE IP Product Guide 7 [Z]. 2017.

[5] Mobiveil. Mobiveil RapidIO Controller (GRIO)[Z]. 2016.

[6] IDT. RapidIO-IP-10xN™ User Manual.[Z]. 2017.

[7] Xin-Tong G , Yuan-Wu L , Yang G , et al. Design and implementation of dual-channel serial RapidIO for multiple transmission modes[J]. Computer Engineering & ence，2019,41(2):233-239.

[8] Lv P , Liu Q , Chen H , et al. Domain-Oriented Software Defined Computing Architecture[J]. 中国通信:英文版，2019(6):162-172.

[作者简介]

李沛杰（1990-），男，硕士学位，助理研究员，主要研究方向为高速接口设计，软件定义互连技术，现代SoC设计技术。

复杂网络节点重要性排序算法及应用综述

郭程远[1]，潘世东[2]，陈鸿昶[1]，王庚润[1]

[1]中国人民解放军战略支援部队信息工程大学 河南省 郑州市 450000；

[2]江南计算技术研究所 江苏省 无锡市 214000

摘　要：复杂网络中节点的异质性导致各个节点的重要性是截然不同的，而关键节点往往对网络的结构和功能具有巨大的影响力，所以通过节点重要性排序算法对关键节点进行挖掘一直受到各方面的关注。本文将复杂网络中关于节点重要性的排序算法进行了研究，比较了各种方法的优势和缺陷，并对重要节点挖掘算法的应用方向进行介绍，最后对现有研究进行了总结并对未来的研究方向进行展望。

关键词：复杂网络、关键节点、节点重要性排序

Research on node importance ranking based on complex network

Guo Chengyuan[1], Pan Shidong[2], Chen Hongchang[1], Wang Genrun[1]

1.People's Liberation Army Strategic Support Force Information Engineering University, Zhengzhou 450000, China;

2.Jiangnan Institute of Computing Technology, wuxi,214000, China

Abstract: The heterogeneity of nodes in a complex network leads to the distinct importance of each node, while key nodes often have great influence on the structure and function of the network. Therefore, the mining of key nodes through node importance ranking algorithm has been paid attention to by all aspects. This paper studies the ranking algorithm of node importance in complex networks, compares the advantages and disadvantages of various methods, and introduces the application direction of important node mining algorithm. Finally, it summarizes the existing research and looks into the future research direction.

Key words: complex network; Key nodes; Node importance ranking

1　引言

从生态系统到社会关系，从社交网络到互联网，在人类社会及自然界中存在着大量的复杂系统，而复杂系统恰恰可以通过复杂网络来表示[1]。随着互联网技术的发展，人们已经生活在一个充满着各种各样复杂网络的世界中[2]。生活中任何具有部分或者全部特性如：小世界、无标度、鲁棒性、脆弱性、自组织以及自相似的网络都可称为复杂网络。如生物网络、水电网络、证券交易网络、社交网络等。

复杂网络中对于关键节点的挖掘研究作为复杂网络中主要研究方向之一，目的是为了能够发现在网络结构和信息传递过程中起着重要作用的节点。例如：2003年一条瑞士与意大利电网间的互联线跳闸却引发了意大利全国范围内的大规模

停电；在城市的交通网中往往存在着几个重要的交通枢纽，当这些少量的重要枢纽发生拥堵时则会间接的引发全城的交通拥堵现象；在短视频和网络直播发展迅猛的当前，往往是少数自媒体或者主播这些关键节点掌握着大量平台资源和流量。因此，对于网络中关键节点的挖掘在复杂网络理论研究和生活实践中都具有重要意义。

虽然前人已经提出了有效的度量指标和算法，但是由于网络科学的发展，近两年出现了一些新的重要节点挖掘方法，这些方法对于前人的研究进行了改进和扩展而且在时间复杂度和计算精度上有所提升为节点挖掘的研究提供了一些新的研究思路，所以有必要对近两年的研究进行一个总结。

本文的结构如下：在第二章中对复杂网络重要节点挖掘的研究现状和应用进行介绍。第三章，从不同的角度分类介绍节点重要性排序算法，包括对一些经典评估指标、经典算法的介绍和近两年的发展。 在第四章对近些年来复杂网络技术的一些热门应用做些介绍。 在第五章中，对当前节点发现算法进行总结并对未来的研究方向以及发展趋势进行展望。

2　研究现状

复杂网络的研究最早来源于瑞典数学家欧拉在1736年提出的“欧拉图问题”， 也称为基于哥尼斯堡七桥问题。在二十世纪五十年代末，数学家们又提出了一种更加贴合实际的网络称为随机网络，但是随着研究的深入，生活中存在的各类网络并不能用随机网络完美的解释。现实世界的网络大多并不具有规则网络和随机网络的特征，反而是与前两者都不相同的网络，这样的网络被称为复杂网络（Complex Networks），自1998年Watts和Strogatz［3］提出了网络的小世界特性和1999年Barabasi和Albert ［4］提出网络的无标度特性引爆了人们对复杂网络的研究热情。

由于复杂网络的无标度特性，网络中大多数的节点度都很小，而一小部分节点度却很大导致网络的度分布呈幂律分布，节点的度具有异质性，不同节点在网络传播中的作用一般不同。而关键节点是指在网络中相比其他节点更能影响网络的结构和信息传递。网络结构主要指节点的度、节点间距离、网络的连通性、节点的聚类系数等。网络功能包含网络的抗毁性、传播、控制等［5］。关键节点在复杂网络中的巨大影响力在理论研究和实际应用中都有着巨大价值。目前，对于复杂网络中节点重要性排序的研究，主要是从节点的邻居个数、节点的位置、网络鲁棒性、网络全局信息等方面进行研究。近几年基于图熵理论、机器学习算法和多属性混合评估的方法对网络中的关键节点挖掘也逐渐成为了研究热点。

同样通过对重要节点的挖掘也解决了许多现实问题如：对城市交通网络中的关键节点进行挖掘，制定相应的拥堵疏导策略减少大规模交通拥堵情况的发生［6］、在军事供应链管理中找到关键节点就能提高物资保障的可靠性和效率［7］、近年频繁发生的勒索病毒、电信诈骗、网络暴力等问题通过网络科学的手段提前发现和预警［8］。

3　节点重要性排序的典型算法

目前，对于复杂网络中节点重要性排序算法的理论研究可以从5个部分去讨论：

1）基于邻居节点信息的算法主要通过节点的局部信息对节点的重要性进行评估；

2）基于网络信息传播路径的算法考虑了节点在信息传播路径上的位置；

3）基于节点位置的排序算法考虑了节点在网络全局结构中的位置；

4）基于特征向量的算法在度的基础上，不仅考虑邻居节点的数量还考虑了邻居节点的重要程度；

5）最后一部分将介绍一些新思路的算法，如：基于图熵理论的、基于机器学习技术、基于多属性融合和基于网络拓扑变化的算法；

接下来将对不同类型的算法进行介绍。

3.1　基于邻居节点的排序算法

节点的重要性往往遭到其本身信息和邻居节点影响，基于邻居节点的重要性排序算法主要通过对邻居节点的数量进行评估，判断出节点的重要程度。

该类经典算法主要有度中心性、半局部中心性算法，总结如表1所示。

YANG等人［11］考虑到节点的度对于信息传播起到积极作用，而节点的聚类系数越大则会导

表1　基于邻居节点的经典排序算法

算法	特点	优点	缺点
度中心性	节点周围邻居节点越多,信息的传播范围就越广,则该节点的影响力就越大[9]	计算复杂度较低,是最为经典的重要性评价指标且应用范围广泛	在大规模网络中应用有限
半局部中心性	节点邻居节点以及间接邻居节点都能够影响节点的重要性[10]	计算精度相对于度中心性高且计算量比其他考虑全局结构的算法小	在一些特殊结构的网络中不适用

致信息的传播范围越小，因此在半局部中心性的基础上综合考虑了节点的度、邻居节点的度、聚类系数和邻居的聚类系数四个方面，提出了一种新的半局部中心性算法DCC。Tulu等［12］提出了一种将节点的度和社区结构结合的半局部中心性算法，经过计算节点的CbM来判别节点的重要性，该算法考虑了节点的度和社区结构对节点重要性的影响，不但能发现重要节点还能发现在不同社区中担任枢纽的重要节点。但是使用CbM算法必须先使用合适的社区检测算法将网络进行划分，因此提出与CbM兼容的社区发现算法将是未来研究的方向。Opsahl等［13］将度中心性应用到了加权网络中，综合了边的权重和相邻节点个数，提出加权网络中节点强度的算法。赵构恒等［14］在Opsahl的基础上进一步考虑有向网络中节点间连边的方向性并引入了节点的入强度和出强度的概念，提出了JP-Degree中心性。

3.2　基于路径的排序算法

基于邻居节点的排序算法把周围的邻居节点看作同等重要，而现实中不同邻居节点的重要性一般是不同的。比如：位于两个社区之间起到连接不同社区作用的节点往往比其他节点更加重要。因此，基于路径的排序算法考虑了网络的全局结构，从信息的传播路径对节点的重要性排序。

该类经典算法主要有离心中心性、接近中心性、介数中心性、Katz中心性和随机游走介数中心性等算法，总结如表2所示。

表2　基于路径的经典排序算法

算法	特点	优点	缺点
离心中心性	离心中心性(EC)通过计算节点与其他节点间距的极值对节点的重要性进行评估[15]	计算时间复杂度低且能够直观反应节点与网络中其他节点的距离	特别容易受到极值影响
接近中心性	接近中心性(CC)通过计算节点与其他节点距离平均值的倒数对节点的重要性进行评估[16]	计算节点与网络中其他节点间距均值的倒数能够避免受到个别孤立节点产生的距离极值的影响	对于节点数量较多的网络来说计算时间复杂度较高
介数中心性	介数中心性(BC)认为穿过节点的最短路径越多,则节点越重要[17]。	能够准确的发现网络中那些能够提高信息传输效率的枢纽节点	对于大型网络中那些不在最短路径上的节点重要性无法准确区分
Katz中心性	Katz中心性认为短路径比长路径越发重要,因此通过对不同长度的路径设计一定的权值来计算节点的重要性[18]	不仅考虑节点对之间的最短路径,还考虑其他非最短路径,相对于只考虑最短路径的算法更加客观	时间复杂度较高,适用范围受到网络规模和拓扑结构限制
随机游走介数中心性	随机游走介数中心性认为节点的重要性与随机游走过程中穿过节点的次数相关[19]	能够直观的反映出节点在网络中与其他节点的连接数量	时间复杂度较高

Fei等人［20］认为节点的强度也是基于该节点与网络中其他节点之间的吸引力之和来表示的。提出了通过计算节点V_i、V_j的度k_i与k_j之和并与最短距离d_{ij}平方的反比得出两个节点的相互作用力，即：

$$F_{ij}=\frac{k_i \times k_j}{d_{ij}^2} \tag{1}$$

通过计算出网络中每个节点与待测节点的相互作用力之和得出节点的强度（重要性），该算法结合了节点的局部信息和最短路径，弥补了离心中心性算法容易受到极值影响的问题，且该算法的节点重要性排序精度要高于传统的排序算法。但是该算法无法区分强度相同的节点，因此增加一些参数去区分这些节点将是下一步研究的方向。

Salavati［21］等提出了一种新的排序方法，利用社区检测算法提取网络中的社区结构。然后，将各个社区经过排序算法选举出关键节点作为社区的网关节点；最后，计算各个社区中关键节点到其余社区网关节点的最短路径和，并进行排序。该算法运用了节点的局部结构，提高了节点的接近度中心性，降低了计算复杂度，而且可以快速检测出具有高扩散能力的节点，适用于大规模网络，但是无法对每个节点例如：社区结构中的非关键节点的重要性进行排序，另外该算法的准确性也在一定程度受到社区检测算法的影响。卢鹏丽［22］等基于图熵的理论提出网络的信息熵定义，将信息熵与介度中心性结合提出了介度熵的节点重要性识别方法，并根据一定的网络攻击策略对网络中识别出的重要节点进行攻击，实验证明新提出的介度熵可以更好的识别网络中节点的重要性。

3.3 基于节点位置的排序算法

相对于局部属性和路径信息，基于节点位置的排序算法从网络的整体结构进行考虑，依据节点在网络中的位置对节点的重要性进行评判。

基于节点位置的经典排序算法主要是由Kitsak［23］等人在2010年首次提出的k-shell算法，该类算法如表3所示。

表3 基于节点位置的经典排序算法

算法	特点	优点	缺点
K-shell	K-shell算法认为在邻居节点数量相同的情况下，节点的影响力受到该节点在网络中位置的影响	原理简单、计算耗时少，能够对网络中的节点准确的划分层级	同层节点重要度无法区分

Zeng等［24］认为k-shell方法的局限性是由于在分解网络时完全忽略了被删除节点的链接数。考虑结合分解后节点的残余度与其被移除的邻居节点的信息来改进K-shell的方法，提出了MDD算法。该算法的计算公式如下：

$$k^m = k^r + \lambda * k_i^e, \tag{2}$$

其中k^m为混合度值，k^r为残余度值（连接到其余节点的链接数），k_i^e为消耗度值（连接到删除节点的链接数），λ是0到1之间的可调参数。该算法利用节点残余度和消耗度值混合计算，将K-shell分解得到的层级进一步细致的划分。Liu［25］等发现网络中存在k-shell指数较大的类核群，但扩展效率较低，因此存在通过k-shell分解的核并不是真正的核这一现象。为了解决这个问题提出了使用连接熵来衡量网络中各个层之间差异的方法。樊燕妮等［26］提出的MSC算法，将节点在网络中的位置与其邻居之间的拓扑结构和连边的重要性考虑在内，将节点的信息多方面结合，相对于K-shell算法更加精确。

3.4 基于特征向量的排序算法

前面三种排序算法是把所有节点看作同样重要的节点，只考虑邻居节点的数量和其在网络中的位置去权衡节点的重要程度，然而不同节点的重要程度是不相同的。为此，基于特征向量的排序方法将克服这些问题。

基于特征向量的经典排序算法主要有特征向量中心性、累计提名、PageRank和HITS，该类算法如表4所示。

Dai等人［32］在特征向量的基础上提出一种新的算法LNC，该算法定义了节点自身影响力、邻居和下一个邻居节点贡献的计算方法，通过将这两种指标结合计算出节点的重要性。该算法由于扩大了节点邻域信息的度量范围以及加入了贡献概率这个指标，使得LNC算法相较于特征向量中心性（EC）、PageRank算法的排序准确性更高且时间复杂度更低。Lu［33］等人提出了PageRank的改进算法LeaderRank，通过在网络中增加一个节点g（Ground node），将其与网络中的所有节点双向链接，从而得到一个强连接的新网络进而替代了PageRank中的跳转概率q。 由于加入了节点g，使得原网络变为强连接网络后，收敛速度变快，其无参数特性解决了PageRank需要对参数q频繁校准的缺点。通过实验发现LeaderRank在节点重要性排序和抗干扰能力的表现更加优秀。Li［34］等人进一步改进了LeaderRank算法，允许拥有连接的节点从g节点中得到更多的分数，即用偏随机漫步代替标准随机漫步。由于节点的入度直接反映了节点的重要程度，因此Li根据节点的影响力来确定节点的权重。加权LeaderRank比LeaderRank在重要节点挖掘、容错性和鲁棒性上表现

表4　基于路径的经典排序算法

算法	特点	优点	缺点
特征向量中心性	节点的重要性由邻居节点的数量和重要程度共同决定[27]	能够发掘出影响力最强的节点	收敛速度较慢，有时会陷入无休止的循环中
累计提名	累计提名算法认为网络中处于中心的节点将会被更多的节点提及[28]	相对于特征向量中心性的算法，累计提名的算法能够快速收敛	除社交网络外应用受限
PageRank	基于网页之间的关系上构建网络并通过随机游走来区分不同网站的重要性[29]	应用范围广，计算精度高	随机跳转机制并不符合现实情况，参数值q需要进行大量实验计算
HITS	HITS算法[30]相对于PageRank算法它考虑了网络中每个节点的两个度量值（权威值、枢纽值）	综合节点的两个不同属性进行重要性排序，能够有效的发现网络中的枢纽节点	计算过程需要多次迭代，运行效率低且容易产生主题偏离现象
SALSA	SALSA算法在HITS算法的基础上用随机游走的方法确定网页的权威值和枢纽值[31]	能够避免主题漂移现象，由于对权威值和枢纽值的计算是独立进行的，不会出现相互增强的现象	时间复杂度高

得更好。Xu等［35］考虑到现实中的复杂网络往往是动态的，因此将H-index指数的偏加权机制与LeaderRank算法结合提出了能够适应网络结构变化的算法ALR。与之前的算法相比，ALR能够较好地应用于网络结构动态变化的场景之中。

3.5　其他排序算法

前几章介绍的都是一些比较常见的算法，本章将对近几年新出现的和一些不常用的算法类型进行介绍，算法总结如表4所示。

表4　其他排序算法

算法	特点	优点	缺点
基于熵理论的排序算法	该类算法认为，节点发出的信息到达每个节点的概率越平均则越利于信息的传播，节点越重要[36]	适用于社交网络、交通网络，能够挖掘出极具影响力的节点	对于大型网络计算时间复杂度高且精度有限

Tong等［37］提出了将图分解为子图和计算邻居节点熵的新型熵中心性模型。胡钢等［38］提出了一种考虑节点邻居节点信息熵的节点重要性排序算法，该算法能够很好的适应各种类型的网络。林鸿基等［39］考虑到当前的节点排序算法往往只是考虑了一个或者两个中心性指标，而这样的评估方法往往不能全面的对节点的重要性合理评价，于是提出了将信息熵权法和层次分析法结合进行关键节点的识别。Xu［40］等人受到熵理论的影响通过比较节点的邻接信息熵对节点的重要性进行排序，在加权网络用节点的强度来计算信息熵而在有向网络则综合了节点的入度和出度值。

基于机器学习的排序算法	将重要节点的特征作为样本，通过训练机器学习模型挖掘重要节点	面对大型网络时间复杂度低具有扩展性	准确性低于传统的中心性度量方法

Yu等［41］受图卷积网络概念的启发，将复杂网络中的关键节点识别问题转化为一个回归问题。Wandelt等［42］提出将深度学习的方法应用于重要节点识别中，相对于其他度量方法该方法大大加速了节点挖掘的过程。汪黎明［43］通过训练神经网络得到能给出两节点对之间最短路径的模型，定义DQNRank值为经过待测节点的最短路径和总路径的比值，通过该值反应节点对于网络的重要程度从而对节点的重要性进行评价。

多属性结合的排序算法	该类算法认为，每种排序算法都有其局限性因此综合考虑多种属性求出最优结果	准确性相对于单一中心性度量算法精度更高	对节点需要计算多种中心性度量方法，时间复杂度较高

Du［44］等人认为每种中心性度量算法都有其局限性，因此提出用TOPSIS方法将多种中心性度量算法融合，综合评价节点的重要性。Hu［45］等人认为TOPSIS算法中将每一种属性都认为相同重要的机制是不合理的，因此提出了基于理想解相似度的排序加权算法，当多属性融合评估时赋予每种属性一个权重使得TOPSIS算法更加准确合理。KuoTing［46］将TOPSIS方法进行了简化并定义了新的排序指标。Yang［47］等人考虑到有些网络的结构是动态变化的，因此提出基于灰色关联分析方法和易感感染恢复（SIR）模型结合的TOPSIS动态加权排序方法，动态地为每个属性分配合适的权重。

基于网络拓扑变化的排序算法	该类算法认为，由于节点的异质性导致删除重要节点相对于其他节点对网络的损害更大	能够有效增强电力传输网、物流网、交通网等现实中复杂网络的鲁棒性	对于大型网络来说计算时间复杂度较高；一定程度上依赖网络攻击策略

Wang等［48］在最短距离法的基础上将网络上所有节点间最短距离倒数的平均值定义为网络的信息传播效率（EFFC），通过移除网络中的单个节点，并计算删除节点前后的网络的信息传播效率对节点重要性进行评估。Dangalchev［49］提出了通过节点移除计算网络中残余节点的接近中心性来衡量节点重要性的方法。ZHONG［50］等同时通过特定的攻击程序对网络节点进行攻击，评估网络中剩余节点的连通性和通过顺序删除节点来评估网络中孤立网络节点个数对节点的重要性进行评估。

4 典型应用场景

前面的章节对复杂网络节点重要性排序的算法进行了介绍，由于复杂网络的相关技术在生活中具备极大的应用价值，所以本章将对复杂网络中关键节点发现技术的主要应用进行介绍。

4.1 在社交网络中的应用

社交网络由于其用户数量庞大，因此关键节点的影响力也就更大。近些年来由于暴恐行为频发不断，对于恐怖组织的打击和及时发现，是维护社会安全和公共秩序的重要保障。Kanokwan［51］等人通过对不同关键节点发现方法进行比较实验，选择了一种最合适的算法用来挖掘恐怖主义骨干网络。蒋昌礼［52］通过使用PageRank算法和接近中心性算法开发出识别微博中关键用户和关键链路的软件，为社交网络的监督、管理以及商业兴趣导向的分析提供了参考依据。周飞［53］等通过改进PageRank算法对用户自身影响力、用户动态行为以及动态内容带来的真实影响这3个方面挖掘出知乎网络中的意见领袖。Lu［54］等通过将改进的h指数中心性和累积中心性结合提出了扩展的混合中心性用于社交网络中有影响力节点的挖掘中，提高了关键节点识别的准确度。

4.2 在金融网络中的应用

在金融行业运用复杂网络技术进行分析也是当前一个热门研究。庄新田［55］等认为可以用复杂网络来抽象和描述证券市场。以股票为节点集，通过测量不同股票间的相关系数，将相关系数值大于一定阈值的节点对作为连边构建股票网络，而通过关键节点发现算法得出股票市场上一部分影响力大的股票股价能对其他股票价格产生影响。冷炳荣［56］等则从宏观的角度，从中国各个城市之间的资本流、经济流、人口迁移等因素综合考虑构建了城市经济网络，通过对网络中节点的聚类分析将城市划分为三大城市区并且分析出区域核心城市。

近些年来金融诈骗、p2p暴雷、非法集资等高危金融事件频发不断，然而由于缺乏监管，在这些高危金融事件的预警方面一直缺少有效的技术支撑。而结合复杂网络节点发现技术，将用户的转帐、消费信息以及其他金融行为用复杂网络表示，构建用户资金流向网络，在此基础上使用重要节点发现算法及时发现上述非法用户和行为，为公安等部门预防金融违法提供技术支撑。另外，传统的推荐算法往往是基于机器学习的方法，将用户的消费信息构建复杂网络从而开发出推荐算法也是一个研究热点。

4.3 在交通网络中的应用

在交通方面，地面交通供给和交通需求的不匹配往往导致交通堵塞。谈进辉［57］从复杂网

络理论得到启发，将道路交叉口作为网络节点、路段作为节点连边，通过对交通网络关键连边移除测试其鲁棒性得到道路的承载能力，根据不同道路的承载能力制定出相应的出行策略，减小了汽车的拥堵可能。随着国内航班的增多导致航空飞行的冲突可能性日益增加，为了改善这个问题，刘飞等［58］和吴明功［59］认为在飞行冲突态势网络中，少数的航空器扮演着“关键节点”的作用，它们能够对冲突态势的演变产生重要影响，因此使用关键节点发现算法识别飞行网络中关键飞行器，有效的降低飞行状态网络的复杂性，降低了管制人员的调配难度。Meng［60］等人结合变分系数法、TOPSIS 和加权 TOPSIS 算法提出了一种多属性决策方法，通过对深圳地铁系统仿真实验揭示了重要节点在城市轨道交通网络发展中的影响。

4.4 在生物网络中的应用

在生物学方面，由于脑网络具有天然的复杂性，因此使用复杂网络技术研究脑网络也是十分常见的。梁曼［61］考虑了脑网络的拓扑特性，通过TOPSIS法将多种中心性算法结合对脑网络关键节点进行判别，相对于单一属性的排序算法，该算法对于重要节点的发掘更加精准。同样在防止疾病传播的方面，刘洋［62］受到复杂网络节点异质性的启发，认为在疾病传播过程中hub节点被感染相比于普通节点被感染传播能力更强，通过对hub节点制定免疫策略减少了疾病爆发的可能性，解决了需要大范围注射疫苗来预防疾病传播的难题。由于细胞的功能和蛋白质的合成中某些蛋白质发挥着重要作用，因此Dai［63］等人提出了一种基于衰减系数的改进指数算法将其应用在蛋白质网络中，从而发现关键蛋白质。

4.5 在电力网络中的应用

在电力网络方面，由于电力网络大多拓扑结构复杂，并且在过去经常出现因为电网中关键节点或者传输线路导致电网级联故障，因此很适合使用复杂网络理论进行研究。何铭［64］基于TOPSIS法结合紧密中心性、阶数中心性、凝聚度等指标通过多属性结合的方法准确挖掘出电力网络中的关键节点。傅杰［65］从节点收缩法获得启发通过去掉连边而不去除节点的方法发现电力网络中的关键输电线路，通过提高这些线路的抗毁性从而提高网络整体的鲁棒性。肖宝强［66］等人将对于电网重要节点的研究扩展到对于重要群组节点的研究，由于关键群组节点相对于单个的关键节点影响力更大，因此通过挖掘关键群组节点有效的提高了电力网络的抗毁性。

5 总结和展望

关键节点挖掘是复杂网络中的一个重要研究方向，本文研究了复杂网络节点重要性排序方面的常用算法和最新进展。目前，复杂网络中重要节点的挖掘经过十几年的发展已经相对成熟，然而还是存在着一些问题有待研究。

度中心性和半局部中心性的算法虽然计算简单，但是由于其考虑的信息有限，所以在大型网络上并不适用；基于路径和特征向量等全局信息的算法虽然准确性相对较高，但是由于其计算复杂导致时间复杂度较高。目前，虽然有一些结合两者优点的算法，但是相对于传统算法优势不是特别明显。因此对于高精度和低时间复杂度的重要节点挖掘算法的研究仍会是未来研究的热门方向。同时，随着数据量的激增和机器学习技术的进步，将机器学习的方法应用于重要节点挖掘的研究受到越来越多人的关注。从根本上来说，复杂网络和机器学习的目标都是为了发掘数据内在的规律，因此融合复杂网络理论和机器学习技术将会是未来研究的一个方向。另一方面，对于排序算法的评价手段大多数都是基于经典的传染病传播模型，即从传染病传播模拟信息传播的模式。然而，由于在现代互联网的帮助下，信息传输所需的成本要低得多，而且比物理传染的速度要快得多。所以提出针对现代信息传播特点的新评价手段或者模型将是未来的一个研究方向。

参考文献:

[1] 方锦清,汪小帆,郑志刚,毕桥,狄增如,李翔．一门崭新的交叉科学:网络科学(上)[J]，物理学进展 2007年03期

[2] 汪小帆, 李翔, 陈关荣．网络科学导论[M]．北京：高等教育出版社，2012

[3] Watts D J, Strogatz S H. Collective dynamics of 'small-world'networks[J]. Nature，1998，393：440－442

[4] Barabási A L, Albert R. Emergence of scaling in random networks［J］. Science，1999，286：509－512

[5] 任晓龙, 吕琳媛．网络重要节点排序方法综述[J]．科学通报，

2014，59：1175－1197

[6] 修志博. 城市交通复杂网络节点重要度评估与级联失效研究[D]. 吉林大学. 2020

[7] 王梓行,姜大立. 基于可靠性的军事供应链网络节点重要度评估研究[J]. 军事运筹与系统工程. 2020,02

[8] 赫南, 李德毅, 淦文燕, 等. 复杂网络中重要性节点发掘综述[J]. 计算机科学，2007，34：1－5

[9] Burt R S, Minor M J, Alba R D. Applied network analysis: A methodological introduction. Sage Publications Beverly Hills，1983

[10] Chen D B, Lü L, Shang M S, et al. Identifying influential nodes in complex networks. Physica A，2012，391：1777－1787

[11] Yang Y Z, Wang X, Chen Y, et al. A Novel Centrality of Influential Nodes Identification in Complex Networks [J]. IEEE ACCESS，2020,8:58742-58751

[12] Tulu M M,Hou R H,Younas T. Identifying Influential Nodes Based on Community Structure to Speed up the Dissemination of Information in Complex Network[J]. IEEE ACCESS,2018,6:7390-7401

[13] OPSAHL T, AGNEESSENS F, SKVORETZ J. Node centrality in weighted networks:generalizing degree and shortest paths[J]. Social Networks,2010,32(3):245-251.

[14] 赵构恒,贾鹏,周安民. 有向加权网络中的改进度中心性[J]. 计算机应用,2020,40(S1):141-145.

[15] Hage P, Harary F. Eccentricity and centrality in networks[J]. Soc Netw，1995，17：57－63

[16] Freeman Linton C.. Centrality in social networks conceptual clarification[J]. Freeman Linton C. ,1978,1(3).

[17] Freeman Linton C.. A set of measures of centrality based on betweenness, Sociometry 40 (1) (1977) 35－41.

[18] Katz L. A new status index derived from sociometric analysis. Psychometrika，1953，18：39－43

[19] Newman M E J. A measure of betweenness centrality based on random walks. Soc Netw，2005，27：39－54

[20] Fei L G, Zhang Q, Deng Y. Identifying influential nodes in complex networks based on the inverse-square law[J]. Physica A，2018,512:1044-1059

[21] Salavati C, Abdollahpouri A, Manbari Z. Ranking nodes in complex networks based on local structure and improving closeness centrality [J]. NEUROCOMPUTING,2019,336:36-45

[22] 卢鹏丽,郭旭东,董瑞,曹乐. 基于介度熵的复杂网络节点重要度识别方法[J]. 兰州理工大学学报,2020,46(02):111-115.

[23] Kitsak M, Gallos L K, Havlin S, et al. Identification of influential spreaders in complex networks. Nat. Phys. ，2010，6：888-893.

[24] Zeng An, Zhang C J. Ranking spreaders by decomposing complex networks[J]. PHYS LETT A,2013,377:1031-1035

[25] Liu Y, Tang M, Zhou T, et al. Improving the accuracy of the k-shell method by removing redundant links: From a perspective of spreading dynamics[J]. SCI REP-UK. ,2015,5

[26] 樊燕妮, 刘三阳,白艺光. 基于多尺度中心性算法的复杂网络节点影响力研究[J],数学的实践与认识. 2020年10期 第159-167页

[27] Bonacich P. Factoring and weighting approaches to status scores and clique identification[J]. J Math Sociol，1972，2：113－120

[28] Poulin R, Boily M C, Mâsse B. Dynamical systems to define centrality in social networks[J]. Soc Netw，2000，22：187－220

[29] Brin S, Page L. The anatomy of a large-scale hypertextual web search engine[J]. Comput Netw ISDN Sys，1998，30：107－117

[30] Kleinberg J M. Authoritative sources in a hyperlinked environment. JACM，1999，46：604－632

[31] Lempel R, Moran S. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. Comput Netw，2000，33：387－401

[32] Dai J Y, Wang B, Sheng J F, et al. Identifying Inflfluential Nodes in Complex Networks based on Local Neighbor Contribution[J]. IEEE ACCESS,2019,7:131719-131731

[33] Lu L Y, Zhang Y C, Yeung C H, et al. Leaders in Social Networks, the Delicious Case[J]. PLOS ONE,2011,6

[34] Li Q, Zhou T, Lü L, et al. Identifying Influential Spreaders by Weighted LeaderRank. 2013，Physica A，2014，404：47－55

[35] Xu S, Wang P. Identifying important nodes by adaptive LeaderRank [J]. PHYSICA A,2017,469:654-664

[36] TutzauerFrank. Entropy as a measure of centrality in networks characterized by path-transfer flow [J]. Social Networks，2006,29(2).

[37] Tong Qiao, Wei Shan, Chang Zhou. How to Identify the Most Powerful Node in Complex Networks? A Novel Entropy Centrality Approach[J]. Tong Qiao;Wei Shan;Chang Zhou,2017,19(11).

[38] 胡钢,徐翔,高浩,过秀成. 基于邻接信息熵的网络节点重要性识别算法[J]. 系统工程理论与实践,2020,40(03):714-725.

[39] 林鸿基,林振智,林冠强,莫天文. 基于信息熵权和层次分析法的电网关键节点识别[J]. 广东电力,2016,29(12):50-56.

[40] Xu Xiang, Zhu Cheng, Wang Qingyong, Zhu Xianqiang, Zhou Yun. Identifying vital nodes in complex networks by adjacency information entropy. [J]. Scientific reports,2020,10(1).

[41] Yu E Y, Wang Y P, Fu Y, et al. Identifying critical nodes in complex networks via graph convolutional networks [J]. KNOWL-BASED SYST,2020,198

[42] WandeltS. et al. : Complex Network Metrics: Can Deep Learning Keep up With Tailor-Made Reference Algorithms? [J]. IEEE ACCESS,2020,8:68114-68123

[43] 汪黎明. 基于深度强化学习的复杂网络关键节点识别[D]. 安徽财经大学,2020.

[44] Yuxian Du, Cai Gao, Yong Hu, MahadevanSankaran, Yong Deng. A new method of identifying influential nodes in complex networks based on TOPSIS [J]. Physica A: Statistical Mechanics and its Applications,2014,399.

[45] Jiantao Hu, Yuxian Du, Hongming Mo, Daijun Wei, Yong Deng. A modified weighted TOPSIS to identify influential nodes in complex networks[J]. Physica A: Statistical Mechanics and its Applications，2016,444.

[46] TingKuo. A modified TOPSIS with a different ranking index [J]. European Journal of Operational Research,2017,260(1).

[47] Pingle Yang, Xin Liu, Guiqiong Xu. A dynamic weighted TOPSIS method for identifying influential nodes in complex networks [J]. Modern Physics Letters B,2018,32(19).

[48] Wang S S,Du Y X,Deng Y. A new measure of identifying influential

nodes: Efficiency centrality[J]. COMMUN NONLINEAR SCI，2017，7

[49] Dangalchev C. Residual closeness in networks. Physica A，2006，365：556－564

[50] Zhong J L, Zhang F M, Li Z X. Identification of Vital Nodes in Complex Network via Belief Propagation and Node Reinsertion[J]. IEEE ACCESS，2018，6，29200-29210

[51] MalangKanokwan, Shuliang Wang, PhaphuangwittayakulAniwat, Yuanyuan Lv, Hanning Yuan, Xiuzhen Zhang. Identifying influential nodes of global terrorism network: A comparison for skeleton network extraction[J]. Physica A: Statistical Mechanics and its Applications，2020，545.

[52] 蒋昌礼. 微博网络关键节点和关键链路识别方法研究与软件研制[D]. 电子科技大学，2013.

[53] 周飞,高茂庭. 基于PageRank的网络社区意见领袖发现算法[J]. 计算机工程，2018，44(02)：203-209+219.

[54] Pengli Lu,Chen Dong. EMH: Extended Mixing H-index centrality for identification important users in social networks based on neighborhood diversity[J]. Modern Physics Letters B，2020，34(26).

[55] 庄新田,闵志锋,陈师阳. 上海证券市场的复杂网络特性分析[J]. 东北大学学报(自然科学版)，2007(07)：1053-1056.

[56] 冷炳荣,杨永春,李英杰,赵四东. 中国城市经济网络结构空间特征及其复杂性分析[J]. 地理学报，2011，66(02)：199-211.

[57] 谈进辉. 东莞市路网复杂网络特性及鲁棒性研究[J]. 现代交通技术，2019，16(04)：59-62.

[58] 刘飞,余敏建,李佳威,温祥西,李双峰. 基于复杂网络理论的飞行冲突关键点识别[J]. 空军工程大学学报(自然科学版)，2019，20(04)：19-25.

[59] 吴明功,王泽坤,甘旭升,杨国洲,温祥西. 基于复杂网络理论的关键飞行冲突点识别[J]. 西北工业大学学报，2020，38(02)：279-287.

[60] Yangyang Meng,Xiangliang Tian,Zhongwen Li,Wei Zhou,Zhijie Zhou, Maohua Zhong. Exploring node importance evolution of weighted complex networks in urban rail transit[J]. Physica A: Statistical Mechanics and its Applications，2020，558.

[61] 梁曼. 基于复杂脑电网络的关键节点识别分析[D]. 河北工业大学，2015.

[62] 刘洋. 基于复杂网络的免疫策略研究[D]. 西南大学，2016.

[63] Dai Caiyan, He Ju, Hu Kongfa, Ding Youwei. Identifying essential proteins in dynamic protein networks based on an improved h-index algorithm. [J]. BMC medical informatics and decision making，2020，20(1).

[64] 何铭,邹艳丽,梁明月,李志慧,高正. 基于多属性决策的电力网络关键节点识别[J]. 复杂系统与复杂性科学，2020，17(03)：27-37.

[65] 傅杰,邹艳丽,谢蓉. 基于复杂网络理论的电力网络关键线路识别[J]. 复杂系统与复杂性科学，2017，14(03)：91-96.

[66] 肖宝强,赵旭东,陈顺,席俊鹏,肖元昊. 电力生命线网络多节点关键群组评估[J/OL]. 电测与仪表：1-8[2020-10-17]. https://kns-cnki-net. e2. buaa. edu. cn/kcms/detail/23. 1202. th. 20200918. 1735. 010. html.

[作者简介]

郭程远（1997—），男，硕士研究生，主要研究方向为复杂网络关键节点发现。

潘世东（1987—），男，研究生学历，工程师，主要研究方向为信息处理。

陈鸿昶（1964-），男，教授，博士生导师，主要研究领域为网络分析。

王庚润（1987-），男，助理研究员，博士，主要研究方向为电信网安全、数据处理。

信号处理与深度学习硬件加速的一致性计算方法

高彦钊[1]，陶常勇[2]

[1]战略支援部队信息工程大学，郑州 450001；

[2]天津市滨海新区信息技术创新中心 天津 300450

摘　要：针对边缘设备与移动终端等对信号处理与深度学习任务同时支撑的应用需求，本文在计算需求层面深入分析多种典型信号处理算法与深度学习算法并模块化分解的基础上，提取了两者共有其适合并行硬件加速的计算模块，提出了一种信号处理与深度学习的一致性计算方法，设计了两类应用一体化硬件加速的软件定义层次化处理单元（PE）以及阵列化计算结构，最后基于Zynq计算平台从重构效率与计算性能等两个方面对一致性计算方法与计算结构进行了验证，结果表明基于一致性计算方法的软件定义可重构计算结构具有较高的计算性能与重构效率。

关键词：深度学习、信号处理、硬件加速、计算结构

Hardware-accelerated Consistent Calculation Method for Signal Processing and Deep Learning

Gao Yanzhao[1], Tao Changyong[2]

1.Information Engineering University, ZhengZhou, 450001;

2.Information technology Innovation Center of Tianjin Binhai New Area, Tianjin, 300450

Abstract: The signal processing and deep learning are both required for the mobile terminals or other devices. In this paper, on the basis of modular decomposition of typical signal processing algorithms and deep learning algorithms, the two common computing modules suitable for parallel hardware acceleration are extracted, and a consistent calculation method for signal processing and deep learning is proposed, and two types of application-integrated hardware-accelerated software-defined hierarchical processing units (PE) and arrayed computing structures are designed. Finally, based on the Zynq computing platform, the consistent computing method and computing structure are verified from the aspects of reconstruction efficiency and computing performance. The results show that the software-defined reconfigurable computing structure based on the consistent computing method has high computing performance and weight. Architectural efficiency.

Key words: Deep Learning; Signal Processing; Hardware Acceleration Computing Architecture

1　引言

近年来，人工智能技术的飞速发展与广泛应用，对现代战争[1][2]、工业范式[3][4]以及日常生活[5][6]产生了深刻的影响。随着边缘终端与移动终端广泛使用，对未来计算系统提出了更高的要求。

从应用需求角度看，人工智能计算任务虽然

在计算系统中占据的比重越来越大，但是在当前以及未来很长的一段时间内，信号/信息处理等科学计算仍然是计算系统任务的最重要组成部分。因此，未来计算系统不仅需要支撑诸如信号处理等科学计算，而且必须能够承担深度学习等人工智能处理任务。如对目标检测任务，复杂天气会导致图像模糊，需要通过科学计算对模糊图像进行去雾和图像增强等预处理，然后采用人工智能算法进行目标检测；对语音识别任务，为了消除人类发声器官本身和由于采集语音信号的设备所带来的混叠和高次谐波失真等因素的影响，必须通过科学计算对其进行预加重、分帧、加窗等预处理操作，以保证人工智能语音识别阶段的信号更均匀、平滑。但是因为传统信号处理与深度学习在算法与成熟度等层面存在较大差异，两者的研究与应用一直呈相对割裂的状态。然而从深度学习研究热点如卷积神经网络（CNN）、循环神经网络（RNN）等算法的计算中包含大量的、可并行化处理的数值计算；而科学计算不论应用场景为何，处理对象为何，计算方法为何，计算器件为何，其计算算法优化、计算过程管理以及计算资源分配等，都在逐步向智能化方向发展。因此，两者之间不仅具有明显的相通之处，并且具有强烈的相互支撑，融合发展的必要性。

从计算需求角度看，在数据量爆炸的信息时代，不论是科学计算还是人工智能，均面临着海量数据的实时处理、计算方法灵活调整、计算功耗有效降低、计算过程智能管控以及计算系统稳健可靠等严峻的挑战。在摩尔定律与Dennard缩放定律逐步放缓的历史背景下，单纯依靠工艺水平的提高或者在冯诺依曼计算架构下从单核到众核的扩展已经很难应对上述问题。因此，不论是科学计算还是人工智能，基于粗粒度可重构计算等新型计算方式实现对计算任务的硬件加速受到了越来越多的关注。

对此，本文针对科学计算与人工智能一体化硬件加速需求，在深入分析多种典型信号处理算法与深度学习算法的基础上，针对两者在同一硬件平台加速的计算需求，提出了两者一致性硬件加速的计算方法，并基于软件定义硬件以及可重构计算技术，设计并分析了可行的硬件加速计算架构，为科学计算与人工智能的一致性硬件加速提供可行的技术思路。

2 典型信号算法分析

2.1 STAP处理方法

空时自适应处理（STAP）是基于一维空域滤波技术发展而来的，目前已成为信号处理领域的重要研究方向。从相控阵雷达各子阵下行信号开始到CFAR检测报告，以mDT算法[7]为基础的STAP信号处理流程及其主要计算模块如图1所示。

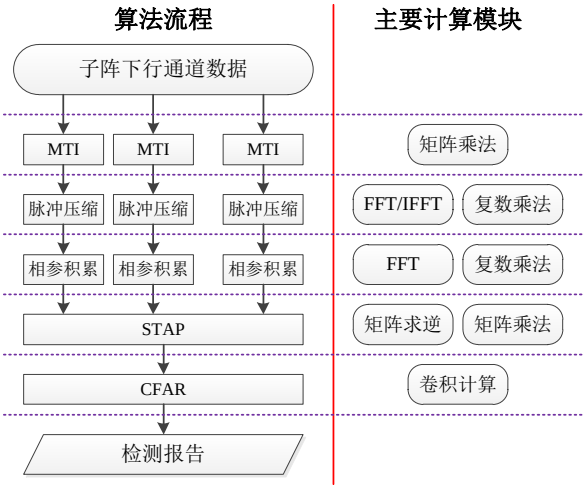


图1 STAP算法流程及其主要计算模块

2.2 PD处理方法

脉冲多普勒（PD）雷达是基于多普勒原理的雷达体制，在距离分辨力、速度分辨力以及杂波抑制等方面具有非常突出的能力，能在强杂波背景中分辨出运动目标[8][9]。PD处理流程及其主要计算模块如图2所示。

2.3 大斜视SAR成像

合成孔径雷达（SAR）采用脉冲压缩技术和合成孔径原理实现地面场景全天候、全天时以及远距离成像。与正侧视SAR成像相比，大斜视SAR成像具有更好的机动性，可通过调整天线指向对感兴趣区域多次重复观测[10][11]。大斜视SAR成像处理流程及其主要计算模块如图3所示。

2.4 遥感光学图像目标识别

对于遥感光学卫星影像中的舰船目标识别问题，为了解决云杂波、海杂波以及舰船浪迹等造成的干扰，克服不同目标尺寸大小对检测带来的困难，文献［12］［13］提出了无监督的基于视觉

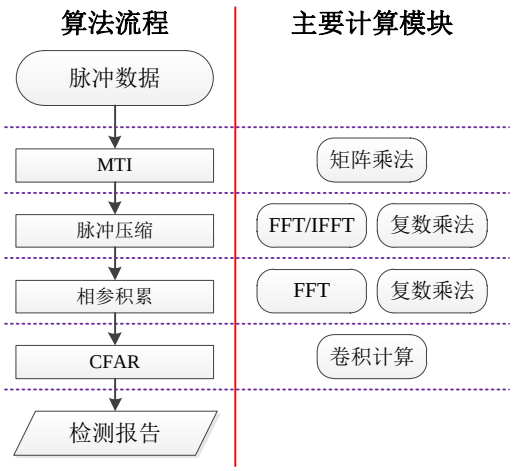


图 2　PD 算法流程及其主要计算模块

图 3　大斜视 SAR 成像算法流程及其主要计算模块

显著性与 S-HOG 描述子的遥感光学图像目标识别算法，处理流程及主要计算模块如图 4 所示。

3　典型深度学习算法分析

3.1　卷积神经网络

卷积神经网络（CNN）属于前馈型神经网络，是目前深度学习领域非常具有代表性的神经网络之一，在大型图像处理方面表现出色，目前已广泛应用于图像分类、定位等领域中。以 LeNet-5 [14] 为例，CNN 的处理流程及主要计算模块如图 5 所示。

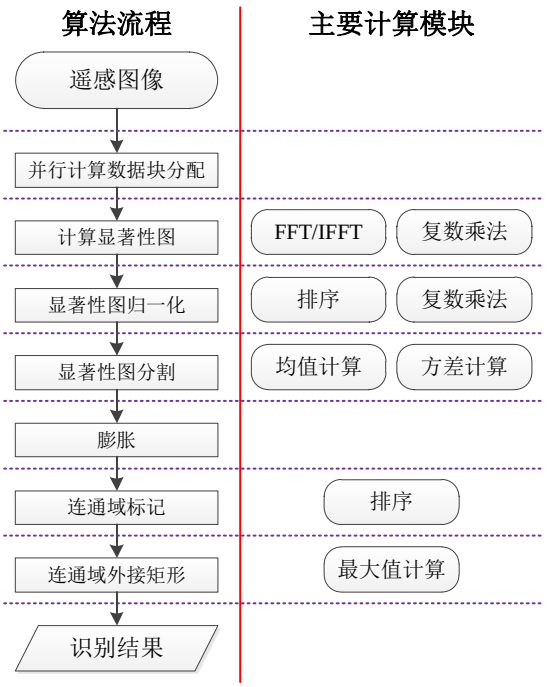


图 4　遥感光学图像目标识别算法流程及其主要计算模块

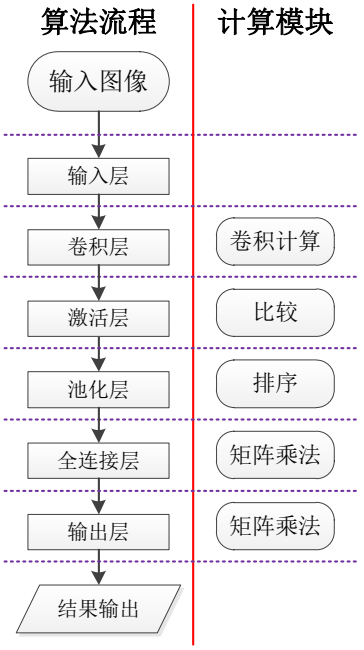


图 5　CNN 算法流程及其主要计算模块

3.2　循环神经网络

循环神经网络（RNN）与卷积神经网络不同，以序列数据作为输入，通过对时序数据进行学习实现上下文信息的存储与表达，具有记忆性与参数共享性，是一种全连接神经网络，已经在自然语言处理领域广泛应用，如语音识别、文本分类和情景分析等。其处理流程与主要计算模块如图 6

所示。

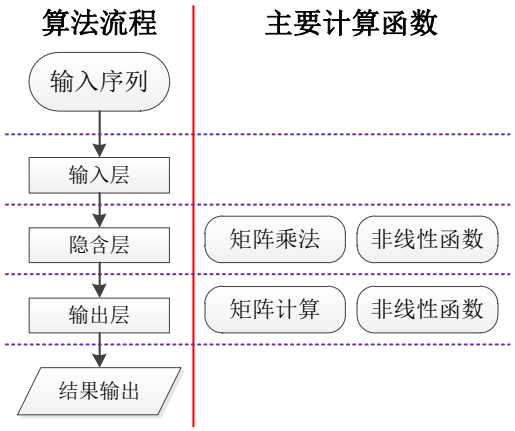


图 6　RNN算法流程及其主要计算模块

通过对上述一维脉冲处理、二维脉冲处理、二维SAR成像、SAR图像解译等、CNN以及RNN等多个典型算法分析，虽然应用场景不同，计算算法不同，但是其主要计算模块主要包括FFT/IFFT计算、矩阵乘法、矩阵求逆、卷积运算、比较、排序、复数乘法以及实数乘法等，其中适应于硬件并行加速的模块则包括FFT/IFFT计算、矩阵乘法、矩阵求逆以及卷积计算等四类。而事实上，这些计算模块也是科学计算与人工智能硬件加速的主要研究对象[15]-[20]。

4　一致性计算方法

4.1　计算模型

1)　FFT/IFFT

根据FFT计算方法，按频率抽取（DIF）的基-2蝶形计算表达式为：

$$\begin{cases} Y_1 = \omega_1 X_1 + \omega_2 X_2 \\ Y_2 = \omega_1 X_1 - \omega_2 X_2 \end{cases} \tag{1}$$

同样，按频率抽取的基-4蝶形计算表达式为：

$$\begin{cases} Y_1 = \omega_1 X_1 + \omega_2 X_2 + \omega_3 X_3 + \omega_4 X_4 \\ Y_2 = \omega_1 X_1 - \omega_2 X_2 - j\omega_3 X_3 + j\omega_4 X_4 \\ Y_3 = \omega_1 X_1 + \omega_2 X_2 - \omega_3 X_3 - \omega_4 X_4 \\ Y_4 = \omega_1 X_1 - \omega_2 X_2 + j\omega_3 X_3 - j\omega_4 X_4 \end{cases} \tag{2}$$

其中，$Y_i(i=1,2,3,4)$表示蝶形运算计算结果；$\omega_i(i=1,2,3,4)$表示蝶形运算的旋转因子；$X_i(i=1,2,3,4)$表示蝶形运算输入。

2)　矩阵乘法

假设矩阵$\mathbf{Y}=\mathbf{A}\cdot\mathbf{B}$，其中$\mathbf{A}=\{a_{ij}|i=1,2,\cdots,M;\ j=1,2,\cdots,K\}$，$\mathbf{B}=\{b_{ij}|i=1,2,\cdots,K;\ j=1,2,\cdots,N\}$，则矩阵$\mathbf{Y}$的任一元素$y_{ij}(i=1,2,\cdots,M,j=1,2,\cdots,N)$表示为：

$$y_{ij}=\sum_{k=1}^{K} a_{ik}b_{kj} \tag{3}$$

3)　矩阵求逆

采用基于LU分解的矩阵求逆方法计算矩阵$\mathbf{A}=\{a_{ij}|i=1,2,\cdots,N;\ j=1,2,\cdots,N\}$的逆矩阵$\mathbf{Y}=\{y_{ij}|i=1,2,\cdots,N;\ j=1,2,\cdots,N\}$，包括三个步骤：□ LU分解，将矩阵$\mathbf{A}$分解为上三角矩阵$\mathbf{U}=\{u_{ij}|i=1,2,\cdots,N;\ j=1,2,\cdots,N\}$与下三角矩阵$\mathbf{L}=\{l_{ij}|i=1,2,\cdots,N;\ j=1,2,\cdots,N\}$，其计算表达式为：

$$u_{ij}=\begin{cases} a_{ij} & (i=1;j=1,\cdots,N) \\ a_{rj}-\sum_{k=1}^{r-1} l_{rk}u_{kj} & (r=1,\cdots,N;j=r,\cdots N) \end{cases} \tag{4}$$

$$l_{ij}=\begin{cases} a_{ij}/u_{11} & (i=1;j=1,\cdots,N) \\ \dfrac{a_{ij}-\sum_{k=1}^{j-1} l_{ik}u_{kj}}{u_{jj}} & (i=j+1,\cdots N;j=1,\cdots,N) \end{cases} \tag{5}$$

□ L/U求逆，假设矩阵$\mathbf{L}$的逆矩阵表示为$\mathbf{V}=\{v_{ij}|i=1,2,\cdots,N;\ j=1,2,\cdots,N\}$，矩阵U的逆矩阵表示为$\mathbf{R}=\{r_{ij}|i=1,2,\cdots,N;\ j=1,2,\cdots,N\}$其计算表达式分别为

$$v_{ji}=\begin{cases} l_{ii}^{-1} & (i=j) \\ -v_{ii}\left(\sum_{k=i+1}^{j} v_{jk}l_{ki}\right) & (i<j) \\ 0 & (i>j) \end{cases} \tag{6}$$

$$r_{ij}=\begin{cases} u_{ii}^{-1} & (i=j) \\ -v_{ii}\left(\sum_{k=i+1}^{j} u_{ik}r_{kj}\right) & (i<j) \\ 0 & (i>j) \end{cases} \tag{7}$$

□ L与U乘法，其计算表达式为：

$$y_{ij}=\sum_{k=1}^{N} r_{ik}v_{kj}\ \ (i=1,\cdots,N;j=1,\cdots,N) \tag{8}$$

4)　卷积计算

假设3×3维卷积核为$\mathbf{W}=\{w_{ij}|i=1,2,3;\ j=1,2,3\}$，输入图像为$\mathbf{A}=\{a_{ij}|i=1,2,\cdots,N;\ j=1,2,\cdots,N\}$，卷积结果为$\mathbf{Y}=\{y_{ij}|i=1,2,\cdots,N-2;\ j=1,2,\cdots,N-2\}$。则卷积计算结果的任意元

素y_{ij}表示为：

$$y_{ij} = w_{11}a_{i-1,j-1} + w_{12}a_{i-1,j} + w_{13}a_{i-1,j+1} + w_{21}a_{i,j-1} + w_{22}a_{ij} + w_{23}a_{i,j+1} + w_{31}a_{i+1,j-1} + w_{32}a_{i+1,j} + w_{33}a_{i+1,j+1}$$

不论FFT/IFFT、矩阵乘法、矩阵求逆还是卷积计算，如果将其计算输入视为矩阵（其维数可变），综合式（1）-（），上述计算的数学模型可一致性表示为：

$$y_{ij} = (\sum a_{ij} \cdot b_{ij} + c_{ij}) \cdot d_{ij} \tag{10}$$

其中，a_{ij}、b_{ij}、c_{ij}、d_{ij}分别是四个计算输入矩阵**A**、**B**、**C**、**D**中的元素，y_{ij}为结果矩阵**Y**中的元素。基于式（10），实现FFT/IFFT、矩阵乘法、矩阵求逆以及卷积计算等不同任务的计算，主要包括四个步骤：1）根据计算过程设计确定计算结果y_{ij}角标i与j的变化规律；2）确定实现元素y_{ij}计算所需要的输入a_{ij}、b_{ij}、c_{ij}以及d_{ij}的集合及其地址变化规律；3）将所需输入元素集合从存储器中读取出来并组成算式；4）通过乘累加模块组成的算粒完成计算过程，并回传y_{ij}的计算结果。

4.2 计算结构

虽然FFT/IFFT、矩阵乘法、矩阵求逆以及卷积计算等不同的计算模块具有不同的计算方法，但是能够一致性表示如式（10）所示。在式（10）的统一描述下，不同计算模块在具体执行过程中，有所不同的是计算过程的顺序组织以及各个元素对应的计算输入序列组成。对此，将完成各模块计算的PE统一划分为3层：第一层控制计算过程的顺序组织及各个元素对应的计算输入序列组成，实现对计算过程的解析；第二层基于第一输出，实现计算输入序列的读取与组合，实现计算算式的生成；第三层采用特别设计的乘加计算结构，实现具体的计算过程。

4.2.1 PE结构

基于式（10）设计的配置流与数据流共同驱动的可重构处理单元结构设计如图7所示，在逻辑上共分为三层，自上而下依次为算法控制层、数据调度层以及计算执行层。其中算法控制层由多个算式规则控制模块（表示为五角星）组成，每个规则控制模块以软件定义的方式实现对不同计算功能的过程控制，解决“怎么算”的问题，即通过计算结果**Y**的跳变顺序实现计算进程的控制；数据调度层由多块RAM组成的分布式数据存储空间（以田字格表示）与算式生成模块（由小圆圈表示）组成，计算数据按不同的存储方式分散存储在多个RAM中，并可在层内进行灵活调度，而算式生成模块响应上一层的控制流信息，完成数据的读写访问，解决“算什么”的问题，即根据**Y**的跳变顺序实现计算输入**A**、**B**、**C**、**D**对应元素的选择，并完成计算数据读取；计算执行层由多组乘法器、加法器、累加器、比较器组成，接收待计算数据进行计算并返回计算结果，解决“具体算”的问题，即根据计算输入**A**、**B**、**C**、**D**元素选择执行具体的计算。

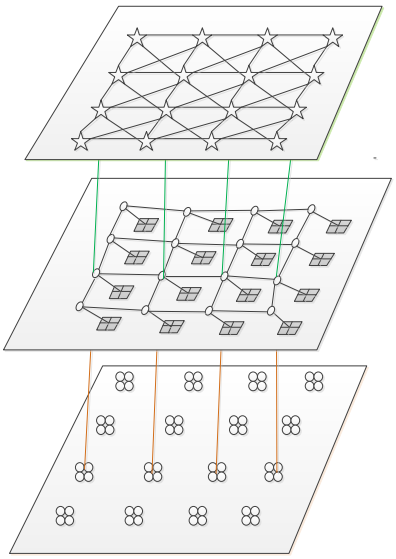


图7 计算结构示意图

同层内各模块之间可以进行信号或数据交互，如算法控制层各算式规则控制模块之间可以进行控制信号交互、数据调度层各算式生成模块可以读取各个RAM的数据、计算执行层相同的计算模块可以共同完成同一个算式的计算任务等；层间不同模块之间也可实现灵活连接，如算式规则控制模块可与数据调度层相应位置及其周围的算式生成模块相连接、算式生成模块可与计算执行层相应位置及其周围的计算模块相连接。在计算过程中，配置流先于数据流下发，完成对计算结构的配置，包括模块功能、数据存取以及模块互连等，适应不同的计算任务。

1) 算法控制层

不同计算任务可采用不同计算跳转顺序与数据组织形式完成。在计算跳转顺序方面，将算法控制过程分为两个层次：算式间循环控制与算式

内循环控制，如图 8 所示。算式间循环控制是第一层循环，指示计算结果 **Y** 元素的角标跳转顺序，即计算过程的推进顺序，可以有多种安排方式，根据计算需求而设定；算式内循环控制是第二层循环，指示与当前 **Y** 元素计算对应的计算输入 **A**、**B**、**C**、**D** 的元素的角标解析。在两层循环控制下，不仅可实现不同计算任务的计算顺序控制，而且可快速实现不同计算阶段待计算数据的解析、输入以及存取等，保证计算效率的提升。

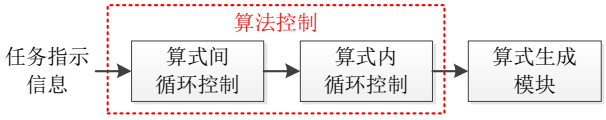


图 8　算法控制模块框图

在数据组织形式方面，根据不同算式间的数据是否可复用将四种应用的计算算式分为两类：组合算式与非组合算式。其中组合算式包括 FFT/IFFT、矩阵乘法与卷积计算，其特点是相邻算式间的计算输入数据可复用，一次数据读取可用于多个算式的计算；非组合算式包括矩阵求逆，其特点是相邻算式间的计算输入数据不可复用，一次数据读取仅用于当前的计算。在组合算式中，充分利用数据复用特性可有效减少数据存取。

2)　数据调度层

数据调度接收并解析上层指令，完成数据读取、组合与下发，其功能包括：1）将待计算数据按计算需求的方式进行分布式存储，包括按矩阵行/列存储、按上/下三角矩阵分别存储、按矩阵元素奇偶分别存储等方式；2）算式生成模块按照算式规则控制模块指示实现从 RAM 阵列中任意 RAM 中读取相关待计算数据；3）算式生成模块将所读取数据组成算式并下发至计算执行层；4）根据算式规则控制模块指示将计算执行层返回的计算结果按一定的方式存入相应的 RAM 阵列中。其结构示意图如图 9 所示，包括 RAM 阵列与算式生成模块，数字用来标识各自的位置。

对 RAM 阵列与算式生成模块的索引格式为二元组 (i,j)，分别表示其行列号，则算式生成模块与 RAM 阵列的位置号是统一的，便于通过配置信息指定相应的路由选择策略，并能够实现任意一个算式生成模块从任意 RAM 中进行数据存取。

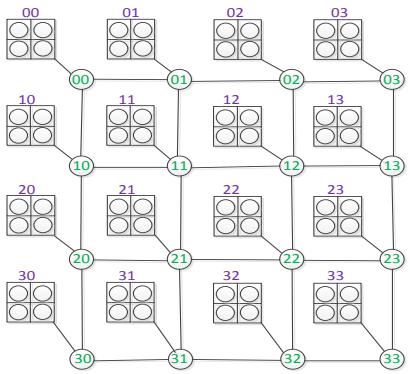


图 9　数据调度层组成示意图

3)　计算执行层

根据式（10）所示的一致性计算公式，计算执行层中计算模块必须包含多个复数乘法器、复数加法器、复数累加器等基本单元，其中复数乘法器包含四个实数乘法器与两个实数加法器，复数加法器包含两个实数加法器。根据计算场景不同，复数乘法器与复数加法器既可以实现复数乘加运算，也可根据配置信息拆分进行多个实数乘加运算。另外，多个复数乘加运算模块既可单个依次完成一个算式的计算，也可多个并行共同完成一个算式的计算。计算结果通过互连结构返回上层算式生成模块，并根据算法控制模块的指令要求存入相应的 RAM 阵列中。

4.2.2　阵列结构

从 PE 的角度来看整个计算架构，以 3×3 个 PE 组成计算阵列为例，内部由 PE 阵列及数据通路与配置通路组成，并通过 SRIO 接口、DDR 接口以及本地管理接口与外部连接，其具体组成与互连结构如图 10 所示。各模块功能为：

1）PE：用于完成不同的计算任务，主要包括算法控制层模块、数据调度层模块以及计算执行层模块三大部分；

2）PE 状态控制模块：用于对阵列中的各个 PE 状态进行控制实现多 PE 之间的工作协同，主要包括配置接口模块、PE 空闲、启动、工作及结束等各种状态的控制模块、PE 间数据流向的控制模块等。

3）Localbus 转 AXI_lite 模块：完成外部软件定义配置或控制命令格式向本地总线格式的转化。

4）AXI_crossbar 模块：实现软件定义配置向各个 PE、DMA0、DMA1 等模块的路由。

5）NIU 模块：实现数据在 PE 之间的路由

传输。

6）DMA0 模块：实现 PE 阵列与外部 SRIO 接口之间的数据传输。

7）DMA1 模块：实现 PE 阵列与外部 DDR 接口之间的数据传输。

8）封解包模块：实现 DMA0 与 SRIO 接口之间的数据组帧与切帧。

9）SRIO 接口：实现 PE 阵列数据与片外或板间的数据交互。

10）DDR 接口：实现 DDR 集中式大数据存储与 PE 阵列之间的数据交互。

11）本地管理接口：实现本地软件定义配置或控制指令下发。

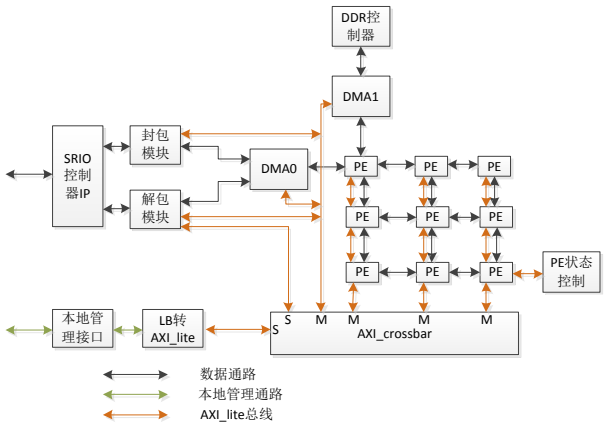


图 10　由 PE 组成的计算阵列框图

通过控制与计算分离的层次化 PE 设计、分布式存储结构设计以及柔性可定义互连结构设计，可实现数据位宽可定义（64bit 或 32bit）、PE 功能可定义（FFT、矩阵乘法、矩阵求逆、卷积计算）、数据通道可定义（PE 之间全互连，数据流程可规划）以及计算模式可定义（阵列分割支持时空域计算）等多尺度灵活可重构，具有灵活性与高效性兼顾的优势。

5　实验验证

5.1　实验环境

基于可重构计算架构的信号处理与深度学习硬件加速一致性计算方法的实验验证基于 Xilinx 的 Zynq 开发板（型号为 ZC706）开展，验证环境结构如图 11 所示，其中设计计算阵列 PE 数量为 2×3 个。PC 机通过 JTAG 加载 Zynq 逻辑文件，并通过 RART 和以太网接口与 Zynq 上的 ARM 核进行通信。Zynq 的 PS 外挂 DDR、Flash 和以太网 PHY，PS 的 ARM 内核工作频率为 667MHz，DDR 接口工作频率为 533MHz，计算阵列工作频率为 100MHz，计算精度为单精度浮点，复数数据宽度为 64bit，实数数据宽度为 32bit。

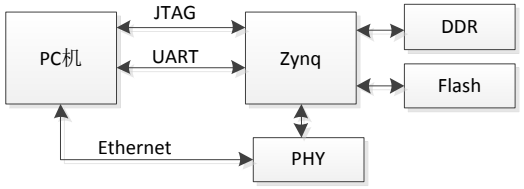


图 11　验证环境结构示意图

5.2　实验结果

实验一：计算阵列重构效率

单个 PE 配置文件为 25.6Kb，全阵列 6 个 PE 配置文件为 153.6Kb，在配置数据通路位宽为 32bit，时钟频率 50MHz 的条件下，实现单个 PE 的配置耗时 16μs，实现全阵列 6 个 PE 的配置耗时 96μs，与 FPGA 秒级的 bit 文件加载时间相比，具有巨大的重构效率优势，具体对比如表 1 所示。

表 1　重构时间对比

	文件大小	时钟频率	重构时间(6 个 PE)
PE 重构	153.6Kb	50MHz	96μs
FPGA 加载	12.7Mb	6MHz	9.38s

实验二：FFT 计算性能

将本文计算结构实现 1K 点 FFT 的计算性能与其他处理器进行对比，包括 RASP [21] 可重构处理器 NoC [22]、MorphoSys [23] 以及 TI 公司 C6678 等。因各类处理器的工作时钟频率不同，本文可重构计算架构工作频率仅为 100MHz，为方便比较，将本文方法的计算时间按时钟频率为 1GHz 进行等比例折算，则各类处理器的 FFT 计算性能对比如表 1 所示。

从表中可以看出，基于本文一致性计算方法及可重构计算结构实现 1K 点单精度浮点 FFT 计算仅需 1.20μs，计算性能是 RASP 的 2.14 倍，是

表 2　FFT 计算时间对比

处理器	RASP	NoC	MorphoSys	C6678	本文方法
计算时间/μs	2.57	76.30	7.40	12.50	1.20

63.58倍。

实验三：矩阵乘法计算性能

将本文计算结构实现单精度浮点实数的两个128×128维矩阵相乘的计算性能与其它基于FPGA的矩阵乘法器进行对比，计算时间如表2所示。从表中可以看出，基于本文一致性计算方法及可重构计算结构实现矩阵乘法计算在同工作时钟频率下优于基于FPGA的矩阵乘法计算性能。

实验四：矩阵求逆计算性能

将本文计算结构实现单精度浮点实数的32×32维矩阵求逆的计算性能与其它基于FPGA的矩阵求逆计算器进行对比，计算耗时如表3所示。在相同工作时钟频率下，针对相同维数的矩阵求逆计算本文方法优于其它基于FPGA的矩阵求逆计算性能。

表3　矩阵乘法计算时间对比

计算方法	方法一[24]	方法二[25]	本文方法
矩阵维数	100×100	128×128	128×128
时钟频率/MHz	60	250	100
计算时间/μs	1351	986.30	908.76

表4　矩阵求逆计算时间对比

计算方法	方法一[26]	方法二[27]	方法三[28]	本文方法
矩阵维数	32×32	32×32	32×32	32×32
时钟频率/MHz	100	100	100	100
计算时间/μs	70.81	53.82	87.34	48.36

基于上述四个实验可以看出，从不同计算结构重构的效率来看，本文方法极大地由于FPGA加载效率，能够实现对不同应用计算结构的实时重构；而对FFT计算、矩阵乘法以及矩阵求逆的计算性能来看，本文方法同样具有较大的优势。

6　结论

本文在对信号处理与深度学习典型算法分析的基础上，提取了两类应用共有且适合并行加速的计算模块，提出了一体化硬件加速的一致性计算方法，并根据计算模块的特点，基于可重构计算技术设计了软件定义计算结构，通过PE内算法控制、数据调度以及计算执行等层次化设计、分布式存储结构设计以及PE间软件定义互连设计，该计算结构可实现PE内与PE间多尺度灵活重构，满足科学计算与人工智能典型计算算法的一体化硬件加速需求，具有较高的灵活性与计算性能。

参考文献:

[1] 孙强. 人工智能对现代战争的影响[J]. 数码世界，2018，5：446.

[2] 陆震. 人工智能在军用机器人的应用[J]. 兵器装备工程学报，2019，40(5)：1-5.

[3] 胡冰洋. 推动我国第四次工业革命及颠覆性技术创新的分析和建议[J]. 中国经贸导刊，2019，8：30-33.

[4] 薛加玉. 人工智能赋能制造业转型升级[J]. 现代工业经济和信息化，2019，3(9)：9-10.

[5] 曾伟良, 吴淼淼, 孙为军, 谢胜利. 自动驾驶出租车调度系统研究综述[J]. 计算机科学，2020，47(5)：181-189.

[6] 谢林利. 智慧城市中基于异构物联网的智慧家居[J]. 计算机科学与应用，2020，10(1)：29-34.

[7] 向聪, 冯大政, 和洁. 机载雷达三维空时两级降维自适应处理[J]. 电子与信息学报，2010，32(8)：1869-1873.

[8] 袁兴生, 段红, 姚新宇, 冯晓梅. 脉冲多普勒雷达信号处理仿真系统研究. 计算机应用，2009，29：294-297.

[9] 姚旺, 金红新, 赵鹏飞, 丛彦超, 王雪. 基于多DSP的PD脉冲压缩雷达信号处理机的设计. 嵌入式技术，2017，43(7)：51-54.

[10] 顾福飞, 张群, 杨秋, 霍文俊, 王敏. 基于NCS算子的大斜视SAR压缩感知成像方法. 雷达学报，2016，5(1)：16-24.

[11] 李震宇, 陈溅来, 梁毅, 邢孟道, 保铮. 带有多普勒中心空变校正的大斜视SAR成像方法. 西安电子科技大学学报(自然科学版)，2016，43(3)：19-24.

[12] 漆昇翔. 视觉显著性及其在自动目标识别系统中的应用[D]. 华中科技大学，2015.

[13] 白婷. 基于视觉显著性的红外小目标检测算法研究[D]. 华中科技大学，2016.

[14] Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recongnition. Proceedings of the IEEE，1998，86(11)：2278-2324.

[15] 龚彤艳,张广婷,贾海鹏,袁良. 一种偶数基Cooley-Tukey FFT高性能实现方法[J],计算机科学，2020，47(1)：31-39.

[16] 张多利, 张玲佳, 宋宇鲲. 变维度FFT硬件加速器结构设计及FPGA实现[J]. 微电子学与计算机，2017，34(12)：34-39.

[17] 杨飞, 马昱春, 侯金, 徐宁. 基于MPSoC并行调度的矩阵乘法加速算法研究[J]. 计算机科学，2017，44(8)：35-40.

[18] 于敬巨, 张多利, 宋宇鲲. 高性能矩阵求逆硬件加速器的设计与实现[J]. 合肥工业大学学报(自然科学版)，2018，41(12)：1652-1658.

[19] 杨博文, 杨海涛, 高浩浩. CNN加速器中卷积计算单元的硬件设计[J]. 数学技术与应用，2019，37(10)：136-137.

[20] 秦华标, 曹钦平. 基于FPGA的卷积神经网络硬件加速设计. 电子信息学报，2019，41(11)：2599-2605.
[21] 何国强, 李丽, 李世平. 面向雷达信号处理应用的可重构处理器设计[J]. 现代雷达，2016，38(8)：46-51.
[22] BAHN J H, Yang J S, et al. Parallel FFT algorithms on network-on-chips[J]. Journal of Circuits System & Computers，2011，18(2)：255-269.
[23] KAMALIZAD A H, PAN C, et al. Fast parallel FFT on a reconfigurable computation platform[C]. Proceedings of the 15th Symposium on Computer Architecture and High Performance Computing：IEEE Press 2003：254-259.
[24] 田翔, 周凡. 基于FPGA的实时双精度浮点矩阵乘法器设计[J]. 浙江大学学报: 工学版，2008，42(9)：1611-1615.
[25] 刘沛华, 鲁华祥等. 基于FPGA的全流水双精度浮点矩阵乘法器设计[J]. 智能系统学报，2012，7(4)：302-306.
[26] 陈晓东, 李世平等. 基于FPGA的Cholesky分解矩阵求逆[J]. 现代雷达，2019，41(10)：58-62.
[27] 张多利, 蒋雯等. 一种用于矩阵求逆的原位替换算法及硬件实现[J]. 合肥工业大学学报(自然科学版)，2020，42(1)：75-80.
[28] 张多利, 叶紫燕等. 任意阶矩阵求逆的算法优化和硬件实现[J]. 合肥工业大学学报(自然科学版)，2019，42(9)：1227-1233.

[作者简介]
高彦钊（1984-），男，河北平山，博士，信息工程大学助理研究员，主要研究方向为高效能计算、雷达信号处理。

陶常勇（1982-），男，山东莱芜人，硕士，天津市滨海新新区信息技术创新中心高级工程师，主要研究方向为拟态计算、高效能计算。

一种基于MISO系统的星型网络组密钥生成方法

宋宣妍，金梁，黄开枝，肖帅芳
战略支援部队信息工程大学，河南 郑州 450002

摘　要：为了保证组内多个用户通信的安全，提出了一种基于MISO系统的星型网络组密钥生成方案。传统的组密钥方案大多为单天线用户，本文分析了中心节点为多天线，其他子节点为单天线时的组密钥生成方案。在MISO系统中，中心节点进行信道探测获得信道增益后，利用获得的信道增益进行波束成形来传输下行的探测信号以提高子节点的接收功率。中心节点和子节点均获得信道增益后，中心节点与子节点利用信道增益生成相同的组密钥并进行一致性验证。最后利用仿真实验验证了这种方法的有效性。实验结果表明，与现有组密钥生成方法相比，在信噪比为15dB的情况下，本文方法的组密钥生成速率最多可以提高0.625倍。

关键词：MISO系统、组密钥、密钥生成速率、多天线

A group key generation method of star network based on MISO system

Song Xuanyan, Jin Liang, Huang Kaizhi, Xiao Shuaifang
PLA Strategic Support Force Information Engineering University, Zhengzhou 450002, China

Abstract: In order to ensure the communication security of multiple users in a group, a group key generation scheme of star network based on MISO system is proposed. Most of the traditional group key schemes are single antenna users. This paper analyzes the group key generation method when the central node is multi antenna and the other sub nodes are single antenna. In the MISO system, after the center node detects the channel gain, it uses the channel gain to carry out beamforming to transmit the downlink detection signal to improve the receiving power of the sub node. After both the central node and the child node obtain the channel gain, the central node and the child node generate the same group key by using the channel gain and verify the consistency. Finally, the effectiveness of this method is verified by simulation experiments. The experimental results show that compared with the existing group key generation methods, the group key generation rate of the proposed method can be increased by 0. 625 times when the signal-to-noise ratio is 15 dB.

Key words: MISO system; group key; key generation rate; multi antenna

1　引言

随着无线通信技术的高速发展，保证无线通信的安全至关重要，传统的无线安全通信方法难以满足无线通信的安全需求。近几年无线物理层安全备受关注，物理层安全是保护无线通信安全的“第一层屏障”。无线物理层密钥生成技术是利用信道的时变性、互易性和空间去相关性的特点生成密钥[1]。大部分密钥生成方案是基于两个合法用户进行设计的，然而存在同一个网络中的多个合法用户需要共享同一个密钥进行保密通信的情况，因此提出了组密钥生成方案来保证组内信息的安全传输。

现有的组密钥分发方案将网络拓扑结构分为星型、链型、网状结构，并针对了不同的网络拓扑结构对组密钥的密钥生成和分发进行设计。Liu H[2]等人利用接收信号强度生成组密钥，具体介绍了密钥生成的过程和可达的组密钥速率。Shuaifang xiao[3]，提出了一种安全私密共享方法，在组内节点完成信道探测后，选取一个中心节点向其他节点发送辅助信息，其他节点利用自身得到的信道估计值获得其他节点的信道估计值并生成相

基金项目：国防科技创新特区项目；重点院校和重点学科专业建设项目；国家自然科学基金资助项目（No. 61871404）。

同的组密钥。Guyue Li [4] 等人针对星型网络、链型网络提出了一种轻量级组密钥生成方案，避免了密钥协商时的信息泄露问题。

以上均考虑的是组内节点为单天线的情况，文献［5］针对网状拓扑结构提出一种多天线用户的密钥生成方案，通过每个节点进行信道探测并广播估计的信道信息后，每个节点将信道信息量化编码生成密钥，此方法不会向窃听者泄露信息。文献［6］提出了在MISO场景下，发送方采用波束成形技术，选择质量最佳的方向进行下行的数据传输以提高通信质量。发送方为了设计该波束成形滤波器，在上行导频训练阶段，需要获取正确的信道信息。

针对物联网中多个用户在彼此的通信范围内需要进行加密通信的场景，在文献［4］-［6］的基础上，本文提出了一种基于多输入单输出（multiple-input and single-output， MISO）系统的组密钥生成方案，考虑中心用户为多天线，其他用户为单天线的星型网络场景。为了提高通信质量，本文采用波束成形技术，选择通信质量最佳的方向进行下行数据的传输，提高了接收信号的功率。本文首先分析了组密钥生成的系统模型与组密钥生成方法，然后分析了组密钥生成方法的安全性。最后通过仿真实验对这种组密钥方法的有效性进行了验证。

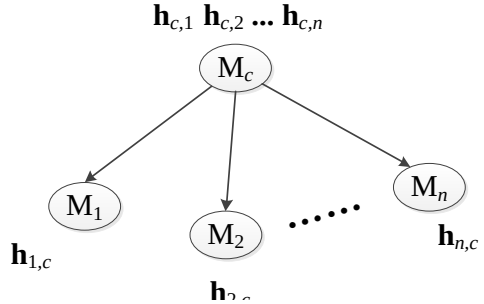
2　系统模型与组密钥生成方法

2.1　系统模型

本文考虑包含1个中心节点M_c与n个子节点M_1，M_2，…，M_n的星型网络拓扑结构如图2-所示。

图2-系统模型

组内任意两个节点均存在直达链路，可以通过无线信道通信，同时在所有节点的通信范围内



存在一个窃听者Eve。其中中心节点M_c为多天线，天线数为m，n个子节点均为单天线，且所有节点均采用TDD通信模式，保证任意两个节点间的信道在相干时间内是互易的。为了保证通信安全，中心节点与子节点共同生成相同的组密钥K_g用于加密通信。当组内的节点需要通信时，利用组密钥K_g对信息进行加密得到密文，并通过无线信道将密文广播出去，当组内节点收到加密的信息后，利用K_g对密文解密恢复信息。从M_c的第m根天线到M_n的信道表示为$h_{c,n}^m$，两个节点之间的信道在相干时间内是互易的，即$h_{c,i}^m = h_{i,c}^m, \forall i = \{1, 2, ..., n\}$。节点之间的无线信道建模为准静态无线信道。便于分析假设所有无线衰落信道增益均为服从独立同分布的高斯随机变量，均值为0，方差为σ_h^2，所有节点接收信号时的噪声信号均为加性高斯白噪声。存在一个被动窃听者Eve，他可以通过无线信道接收到所有组内节点发送的信号。将节点M_n到Eve的信道增益记为g_n。Eve的位置是未知的，并且距离合法节点的距离大于半波长，即g_n与$h_{c,1}, h_{c,2}, ..., h_{c,i}$均是不相关的。

2　2组密钥生成方法

本节提出了星型网络的组密钥生成方法，中心节点与每个子节点进行点对点的通信并提取出信道增益，利用每个信道增益分别对生成的随机序列R加密并广播。子节点利用已知的信道增益对随机序列R进行解密并获得其他的信道增益，生成一致的组密钥。该方法包括信道探测、密钥分发、密钥生成与一致性验证四个步骤。

2.2　1信道探测

为了获得信道增益，首先要进行信道探测，组内节点轮流发送探测信号并对二者之间的信道进行信道估计。假设每个子节点发送信道探测信号的功率均为P_0，中心节点发送信道探测信号的功率为P_c，发送信道探测信号的时间为T_c，在相干时间内，一共进行n次信道估计。

每次的信道探测过程包含三个步骤，以第i次信道探测为例：

（1）首先，子节点M_i以功率P_0发送探测信号$x(i)$，中心节点在第i个时隙接收到的信号为

$$\mathbf{y}_{c,i}(i) = \mathbf{h}_{c,i}x(i) + \mathbf{n}(i), i = 1,...,n \qquad (1)$$

其中$\mathbf{n}(s)$是均值为0，方差为σ_n^2的加性高斯白噪

声（AWGN）。$\mathbf{h}_{c,i}$ 为信道增益，$\mathbf{h}_{c,i}=[h_{c,i}^1,h_{c,i}^2,...,h_{c,i}^m]^T$，且其中每个元素均为独立同分布的高斯随机变量，其均值为0方差为σ_h^2。重写式（1）为矩阵形式

$$\mathbf{Y}=\mathbf{h}_{c,i}x+\mathbf{n} \tag{2}$$

其中 $\mathbf{Y}=[\mathbf{y}_{c,i}^1,...,\mathbf{y}_{c,i}^m]^T$，并且 $\mathbf{n}=[\mathrm{n}_{c,i}^1,\mathrm{n}_{c,i}^2,...,\mathrm{n}_{c,i}^m]^T$，其方差为$m^2\sigma_n^2$。由于导频信号$x$已知，所以，利用迫零算法估计出信道$\hat{\mathbf{h}}_{c,i}$

$$\begin{aligned}\hat{\mathbf{h}}_{c,i}&=\frac{x^T}{\|x\|^2}\mathbf{Y}_{c,i}\\&=\mathbf{h}_{c,i}+\frac{x^T}{\|x\|^2}\mathbf{n}\end{aligned} \tag{3}$$

中心节点M_c计算$\hat{h}_{c,i}=\sum_{k=1}^{m}h_{c,i}^k,i=1,...,n$并对其量化生成量化比特序列$\hat{h}_{c,1}^{\Delta},\hat{h}_{c,2}^{\Delta},...,\hat{h}_{c,n}^{\Delta}$，由于$h_{c,i}^1,h_{c,i}^2,...,h_{c,i}^m$独立同分布，则$\hat{h}_{c,i}\sim\mathcal{N}(0,m\sigma_h^2+\frac{m\sigma_n^2}{\sqrt{P_0T_0}})$。

（2）然后中心节点使用该信道估计结果$\hat{\mathbf{h}}_{c,i}$来设计向子节点发送导频时的波束成形滤波器$\mathbf{w}$以使子节点能够获得最大的接收信噪比。这个波束成形滤波器[6]的设计方法为：

$$\mathbf{w}_i=\frac{\hat{\mathbf{h}}_{c,i}}{\left\|\hat{\mathbf{h}}_{c,i}\right\|}=[\frac{\hat{h}_{c,i}^1}{\left\|\hat{\mathbf{h}}_{c,i}\right\|},\frac{\hat{h}_{c,i}^2}{\left\|\hat{\mathbf{h}}_{c,i}\right\|},...,\frac{\hat{h}_{c,i}^m}{\left\|\hat{\mathbf{h}}_{c,i}\right\|}] \tag{4}$$

（3）中心节点M_c利用波束成形滤波器发送探测信号x_c，$\mathrm{E}\left\{\left|x_c(s)\right|^2\right\}=1$，子节点$\mathrm{M}_i$（$i=1,2,...,n$）在第$i$个时隙接收到的信号为

$$y_{i,c}(s)=\mathbf{h}_{c,i}^T\mathbf{w}_ix_c(s)+n_{i,c}(s),i=1,...,n \tag{5}$$

其中$\sqrt{P_c}$是中心节点的发射功率，$n_{i,c}(s)$表示节点M_i接收探测信号时的噪声。$\mathbf{h}_{c,i}$表示信道增益，$\mathbf{h}_{c,i}=[h_{c,i}^1,h_{c,i}^2,...,h_{c,i}^m]^T$。节点$\mathrm{M}_i$采用迫零算法对信道进行估计[7]，得到

$$\begin{aligned}\hat{h}_{i,c}&=\frac{\left\|\hat{\mathbf{h}}_{c,i}\right\|x_c{}^T}{\left\|x_c\right\|^2}y_{i,c}=[(h_{c,i}^1)^2+(h_{c,i}^2)^2+...+(h_{c,i}^m)^2]+\\&\frac{\left\|\hat{\mathbf{h}}_{c,i}\right\|x_c{}^T}{\left\|x_c\right\|^2}n_{i,c}\end{aligned} \tag{6}$$

由于h_i和$n_{i,c}$均为均值为0的高斯随机变量，所以$\hat{h}_{i,c}$也为均值为0的高斯随机变量，其方差为$\sigma_h^2+\sigma_n^2/\left\|x_c\right\|^2$，$x_c$的发送功率为$P_c$，发送时间为$T_c$个符号周期，因此$\left\|x_c\right\|^2=P_cT_c$，可得$\hat{h}_{i,c}\sim\mathcal{N}(0,m\sqrt{(h_{c,i}^1)^2+(h_{c,i}^2)^2+...+(h_{c,i}^m)^2}\,\sigma_h^2+m\sigma_n^2/P_cT_c)$。

中心节点利用波束成形滤波器分别向子节点发送探测信号后，子节点M_1，M_2，…，M_n分别计算$\hat{h}_{i,c}$并对其量化得到量化比特序列$\hat{h}_{1,c}^{\Delta},\hat{h}_{2,c}^{\Delta},...,\hat{h}_{n,c}^{\Delta}$。

2.2 2密钥分发

经过信道探测阶段，中心节点和子节点均获得信道特征序列后，组内的节点未获得共享的随机信息，需要通过密钥分发使得所有节点获得共享的随机信息。本文采用文献［4］的轻量级密钥生成方法进行安全的密钥分发。中心节点生成随机序列R后，对其进行纠错编码，利用信道探测阶段获取的信道特征序列对编码后的随机序列R进行加密并发送给各个子节点。子节点收到信号后，通过信道探测阶段获取的信道特征序列对信号进行解密，获得随机序列R。子节点利用随机数获得中心节点与其他节点的信道特征序列并组合随机序列与各个信道的信道特征序列得到组密钥。具体分为以下5个步骤进行介绍：

1 中心节点M_c生成随机序列R，并对其进行LDPC纠错编码[8]，编码后生成序列R_c

$$R_c=C(R) \tag{7}$$

2 中心节点M_c将序列R_c分别通过$\hat{h}_{c,1}^{\Delta},\hat{h}_{c,2}^{\Delta},...,\hat{h}_{c,n}^{\Delta}$对其进行异或加密，得到辅助信息$\mathbf{Q}=[Q_{1,c},Q_{2,c},...,Q_{n,c}]$，以复用的方式将辅助信息广播至各个子节点；

$$Q_{i,c}=R_c\oplus\hat{h}_{c,i}^{\Delta},i=1,...,n \tag{8}$$

3. 子节点M_i收到辅助信息$\mathbf{Q}=[Q_{1,c},Q_{2,c},...,Q_{n,c}]$后，由于信道互易性，$\hat{h}_{i,c}^{\Delta}$与$\hat{h}_{c,i}^{\Delta}$是高度相近的量化比特序列，所以利用信道探测获得的$\hat{h}_{i,c}^{\Delta}$对$Q_{i,c}$进行异或解密获得随机序列$R_i'$信道，如式（9）所示，其中$D(\cdot)$表示纠错解码函数

$$\begin{aligned}R_i'&=D(R_c)\\&=D(Q_{i,c}\oplus\hat{h}_{i,c}^{\Delta})\\&=D(R_c\oplus\hat{h}_{c,i}^{\Delta}\oplus\hat{h}_{i,c}^{\Delta})\\&=D(L(R)\oplus\hat{h}_{c,i}^{\Delta}\oplus\hat{h}_{i,c}^{\Delta})\end{aligned} \tag{9}$$

子节点M_i获得R_i'后，通过式（9）推导出$R_i'=R$，是因为$\hat{h}_{c,i}^{\Delta}$和$\hat{h}_{i,c}^{\Delta}$是中心节点与子节点分别经过

信道探测得到的具有互易性的信道特征值，但是存在一定的不一致比特位。而LDPC纠错码可以在量化比特序列中某些不一致的比特位纠正过来，即子节点得到R_i'并且$R_i' = R$。

4. 子节点M_i获得R_i'后，利用R_i进行异或解密获得其他信道的信道特征序列。子节点M_i收到的辅助信息为$\mathbf{Q} = [Q_{1,c}, Q_{2,c}, ..., Q_{n,c}]$，由步骤3可知，其中$Q_{i,c}$用于计算出$R_i'$，步骤4对剩下$n$-1个值分别异或解密得到其他信道的信道特征序列。以子节点M_i获取第j个子节点的信道特征序列为例，由式（8）可知$\mathbf{Q} = [Q_{1,c}, Q_{2,c}, ..., Q_{n,c}]$中

$$Q_{j,c} = R_c \oplus \hat{h}_{c,j}^{\Delta} \tag{10}$$

R_c为随机序列R进行LDPC纠错编码后的值，所以由式（9）计算出的随机序列R的估计值R_i'的值也需代入式（7）中对其进行LDPC纠错编码得到R_i

$$R_i = C(R_i') \tag{11}$$

利用R_i与$Q_{j,c}$进行异或并解码得到第j个节点的信道特征序列$\hat{h}_{c,j}^{\Delta}{}'$

$$\begin{aligned} \hat{h}_{c,j}^{\Delta}{}' &= D(R_i \oplus Q_{j,c}) \\ &= D(R_i \oplus (R_c \oplus \hat{h}_{c,j}^{\Delta})) \end{aligned} \tag{12}$$

同理可得子节点M_i利用$\mathbf{Q}$中的其他序列可以获得其他节点的信道特征值，最终可以获得信道特征序列$[\hat{h}_{c,1}^{\Delta}{}', \hat{h}_{c,2}^{\Delta}{}', ..., \hat{h}_{c,n}^{\Delta}{}']$。

最后，所有节点均能获得随机序列R_i与所有信道的信道特征序列$[\hat{h}_{c,1}^{\Delta}{}', \hat{h}_{c,2}^{\Delta}{}', ..., \hat{h}_{c,n}^{\Delta}{}']$。

2.2 3组密钥生成与一致性验证

中心节点利用R_c与所有信道的信道特征序列$[\hat{h}_{c,1}^{\Delta}, \hat{h}_{c,2}^{\Delta}, ..., \hat{h}_{c,n}^{\Delta}]$组合得到组密钥$R_i \| \hat{h}_{c,1}^{\Delta}{}' \| \hat{h}_{c,2}^{\Delta}{}' \| ... \| \hat{h}_{c,i}^{\Delta}{}'$。子节点$M_i$利用随机序列$R_i'$的LDPC编码值$R_i$与所有信道的信道特征序列$[\hat{h}_{c,1}^{\Delta}{}', \hat{h}_{c,2}^{\Delta}{}', ..., \hat{h}_{c,n}^{\Delta}{}']$组合得到组密钥$K_g = R_i \| \hat{h}_{c,1}^{\Delta}{}' \| \hat{h}_{c,2}^{\Delta}{}' \| ... \| \hat{h}_{c,i}^{\Delta}{}'$。

在各个子节点生成组密钥之后，需要与中心节点进行组密钥一致性的验证。中心节点利用哈希函数将生成的散列值广播至各个子节点，各个子节点利用相同的哈希函数验证自己获得的组密钥的散列值与中心节点广播的散列值是否相同。只有所有节点组密钥一致性验证均通过才完成了密钥分发的步骤。

3 安全性分析

通过上节的组密钥生成过程，组内n个子节点得到了私密随机源。同时Eve通过监听无线信道和公共协商信道，得到了窃听信道信息$\hat{\mathbf{g}} = (\hat{g}_1, \hat{g}_2, ..., \hat{g}_n)$，辅助信息$\mathbf{Q} = [Q_{1,c}, Q_{2,c}, ..., Q_{n,c}]$，可达密钥速率是由密钥生成双方对共享随机源的观测值的互信息决定的[3]，本文的组密钥生成过程中，是由中心节点与其他n个子节点进行信道探测完成的，因此组密钥生成速率由中心节点和估计误差最大的组内节点的观测值的互信息决定的。

根据文献［9］和文献［10］的结论，组密钥生成速率可表示为

$$R_{g,Q} = \lim_{\Delta \to 0} \frac{1}{T} I(R_i, \hat{h}_{c,i}^{\Delta}{}' ; R_c, \hat{h}_{c,i}^{\Delta} \mid \hat{\mathbf{g}}, \mathbf{Q}) \tag{13}$$

根据互信息的性质，可以对式（9）进行简化，首先得到

$$\begin{aligned} I(R_i, \hat{h}_{c,i}^{\Delta}{}' ; R_c, \hat{h}_{c,i}^{\Delta} \mid \hat{\mathbf{g}}, \mathbf{Q}) &= I(R_i, \hat{h}_{c,i}^{\Delta}{}' ; R_c, \hat{h}_{c,i}^{\Delta}, \hat{\mathbf{g}}, \mathbf{Q}) - \\ & I(R_i, \hat{h}_{c,i}^{\Delta}{}' ; \hat{\mathbf{g}}, \mathbf{Q}) \end{aligned} \tag{14}$$

由于窃听信道与合法信道是不相关的，所以可以得到

$$I(R_i, \hat{h}_{c,i}^{\Delta}{}' ; R_c, \hat{h}_{c,i}^{\Delta}, \hat{\mathbf{g}}, \mathbf{Q}) = I(R_i, \hat{h}_{c,i}^{\Delta}{}' ; R_c, \hat{h}_{c,i}^{\Delta}, \mathbf{Q}) \tag{15}$$

$$I(R_i, \hat{h}_{c,i}^{\Delta}{}' ; \hat{\mathbf{g}}, \mathbf{Q}) = I(R_i, \hat{h}_{c,i}^{\Delta} ; \mathbf{Q}) \tag{16}$$

将式（15）和式（16）代入式（14）可得

$$\begin{aligned} I(R_i, \hat{h}_{c,i}^{\Delta}{}' ; R_c, \hat{h}_{c,i}^{\Delta} \mid \hat{\mathbf{g}}, \mathbf{Q}) &= I(R_i, \hat{h}_{c,i}^{\Delta}{}' ; R_c, \hat{h}_{c,i}^{\Delta}, \mathbf{Q}) - \\ & I(R_c, \hat{h}_{c,i}^{\Delta} ; \mathbf{Q}) \end{aligned} \tag{17}$$

因此式（13）可以简化为

$$\begin{aligned} R_{g,Q} &= \lim_{\Delta \to 0} \frac{1}{T} [I(R_i, \hat{h}_{c,i}^{\Delta}{}' ; R_c, \hat{h}_{c,i}^{\Delta}, \mathbf{Q}) - I(R_c, \hat{h}_{c,i}^{\Delta} ; \mathbf{Q})] \\ &= \lim_{\Delta \to 0} \frac{1}{T} [I(R_i, \hat{h}_{c,i}^{\Delta}{}' ; R_c, \hat{h}_{c,i}^{\Delta}, \mathbf{Q}_{i,c}) - I(R_c, \hat{h}_{c,i}^{\Delta} ; \mathbf{Q}_{i,c})] \end{aligned} \tag{18}$$

由于$Q_{i,c} = R_c \oplus \hat{h}_{c,i}^{\Delta}$，且随机序列$R_c$与$\hat{h}_{c,i}^{\Delta}$不相关，即$I(R_c ; \hat{h}_{c,i}) = 0$。根据异或不相关特性可得$I(R_c, \hat{h}_{c,i}^{\Delta} ; Q_{i,c}) = 0$。利用互信息的性质，可以对式（18）进一步化简得到

$$\begin{aligned} R_{g,Q} &= \lim_{\Delta \to 0} \frac{1}{T} [I(R_i, \hat{h}_{c,i}^{\Delta}{}' ; R_c, \hat{h}_{c,i}^{\Delta}, Q_{i,c}) - I(R_c, \hat{h}_{c,i}^{\Delta} ; Q_{i,c})] \\ &= \lim_{\Delta \to 0} \frac{1}{T} [I(R_i, \hat{h}_{c,i}^{\Delta}{}' ; R_c, \hat{h}_{c,i}^{\Delta}, Q_{i,c})] \end{aligned} \tag{19}$$

由于$Q_{i,c}$与R_c、$\hat{h}_{c,i}^{\Delta}$不相关，式（19）可化

简为

$$R_{g,Q}=\lim_{\Delta\to 0}\frac{1}{T}\left[I\left(R_i,\hat{h}_{c,i}^{\Delta}{}';R_c,\hat{h}_{c,i}^{\Delta}\right)\right] \tag{20}$$

从式（20）可以看出，经过密钥分发阶段发送的辅助信息**Q**不会影响密钥速率，不会给窃听者Eve泄露任何信息。

4 结果分析

为了验证本方案的有效性，对密钥生成速率（Key Generation Rate， KGR）和比特不一致率（Bit Error Rate， BER）进行了仿真分析。假设收发双方均采用数字信号处理，每个节点的信道增益均采用了16比特量化。可达组密钥速率与最短相干时间$T_{\min}$成正比，本章所提方法的密钥速率可以表示为

$$R_{g,Q}=\lim_{\Delta\to 0}\frac{1}{T_{\min}}\left[I\left(R_i,\hat{h}_{c,i}^{\Delta}{}';R_c,\hat{h}_{c,i}^{\Delta}\right)\right] \tag{21}$$

采用蒙特卡洛方法进行10000次实验，每次随机产生一组信道增益值和噪声值，采用Matlab中的ITE工具包估计互信息。节点之间的无线信道建模为准静态块衰落信道，且相互独立，所有无线衰落信道的增益均为服从独立同分布的高斯随机变量，均值为0，方差为σ_h^2，所有节点接收信号时收到的噪声信号均为加性高斯白噪声，方差为σ_n^2。

表1 仿真参数表

参数	值
P_0	1
P_c	1
σ_h^2	1
$T_{\min}$	$2n$

如图分析了子节点n=2，3，4，5的情况下的密钥生成速率随信噪比（Signal-to-Noise Ratio，SNR）变化的对比，组密钥速率均随SNR的提高而增大。从图中可以看出密钥生成速率与子节点数有关。虽然子节点数增加后，生成的组密钥长度变长，但是由于子节点数增加后所需的最短相干时间也增加了，所以子节点数越小KGR越大。

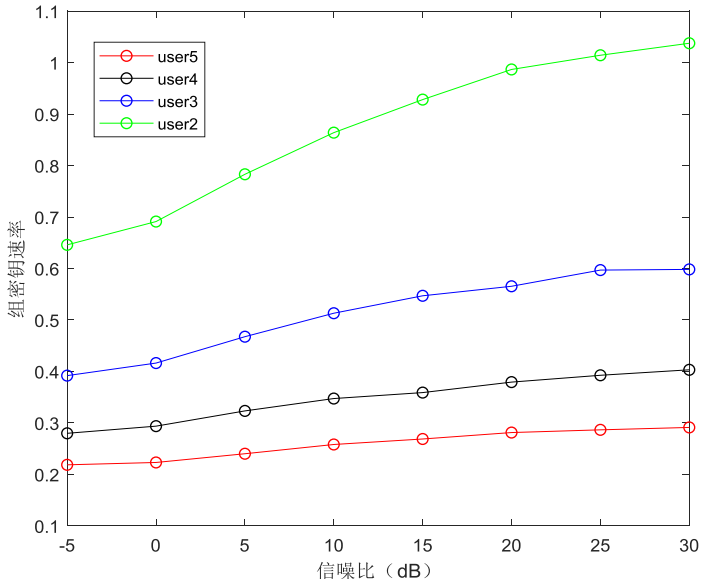


图3-1 密钥生成速率随SNR变化曲线对比

为了验证本文方法，选取文献［3］和文献［11］的组密钥生成方法与本文的方法进行对比。文献［3］提出了一种安全私密共享方法，在组内节点完成信道探测后，选取一个中心节点向其他节点发送辅助信息，其他节点利用自身得到的信道估计值获得其他节点的信道估计值并生成相同的组密钥。文献［11］利用接收信号强度生成组密钥，中心节点向其他用户发送接收信号强度差值，子节点收到后可计算获得其他节点的接收信号强度的估计值。

如图3-2为组内共有4个节点时，三种方法的组密钥生成速率随SNR变化的对比图，从图中可以看到本文的方法明显优于参考方法。由于中心用户为多天线，利用收到的信道状态信息进行波束成形，增加了接收信号的功率，提高了传输的可靠性，从而提高了密钥生成速率。

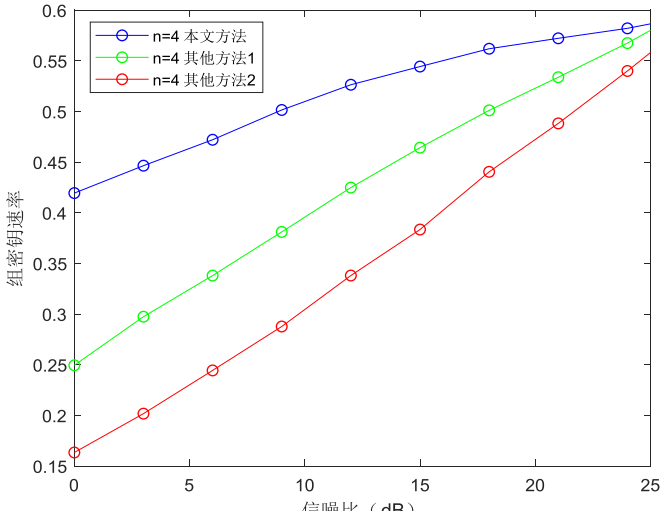


图3-2　密钥生成速率随SNR变化曲线对比

图3-3记录了子节点数n为2，3，4，5时BER随SNR的变化曲线对比。一方面，从图中可以看出随着SNR提高增加BER明显降低；另一方面，从图中可以看出随着子节点数增加，BER也越高。这是由于子节点数增加后，生成组密钥长度也随之增加，噪声对生成密钥的影响也增加。所以SNR相同时，子节点数越多，BER越高。

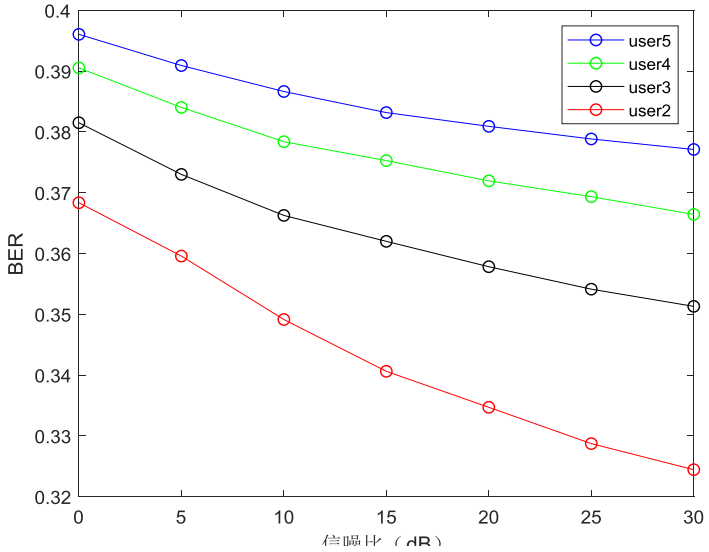


图3-3　比特不一致率随SNR变化曲线

5　总结与展望

本文分析了中心节点为多天线时的组密钥生成方法。下行节点通过波束成形向子节点发射导频，与子节点进行信道探测，可以提高传输可靠性，进而提高了密钥生成速率。从实验结果可以

看到信道探测和传输的时间影响密钥生成速率，后续研究可以根据系统特点设计传输方案，以降低最短相干时间，进而提高密钥生成速率。本文研究了中心节点为多天线和子节点为单天线情况下的组密钥生成方法，对于中心节点和子节点均为多天线的情况还需要进一步进行研究。

参考文献:

[1] 金梁,蔡奥林,黄开枝,钟州,楼洋明. 基于多随机信号流的密钥生成方案[J]. 电子与信息学报，2019，41(06)：1405-1412.

[2] Liu H, Yang J, Wang Y, et al. Group secret key generation via received signal strength: Protocols, achievable rates, and implementation [J]. IEEE Transactions on Mobile Computing，2014，13(12)：2820-2835.

[3] 肖帅芳. 面向物联网的物理层密钥生成技术研究[D]. 战略支援部队信息工程大学，2018.

[4] Guyue Li, Liangjun Hu and Aiqun Hu. Lightweight Group Secret Key Generation Leveraging Non-Reconciled Received Signal Strength in Mobile Wireless Networks [C]. IEEE International Conference on Communications Workshops，Shanghai，China，20-24 May. 2019，pp. 1-6

[5] Thai C, Lee J, and Quek T. Secret group key generation in physical layer for mesh topology [C]. IEEE Global Communications Conference，San Diego，USA，2015：1-6.

[6] 田筱雯. 多天线无线通信系统中的物理层安全研究[D]. 大连理工大学，2019.

[7] MIMO-OFDM无线通信技术及MATLAB实现[M]. 电子工业出版社，2013.

[8] ChungS-Y, ForneyJ G D, RichardsonT, and UrbankeR. On the design of low-density parity-check codes within 0. 0045 dB of the Shannon limit[J]. IEEE Commun Lett，2001，5(2)：58-60.

[9] Maurer U. Secret key agreement by public discussion from common information[J]. IEEE Transactions on Information Theory，1993，39(3)：733-742.

[10] Ye C and Reznik A. Group secret key generation algorithms [C]. IEEE International Symposium on Information Theory，Nice，France，2007：2596-2600.

[11] Liu H, Yang J, Wang Y, et al. Group secret key generation via received signal strength: Protocols, achievable rates, and implementation [J]. IEEE Transactions on Mobile Computing，2014，13(12)：2820-2835.

[作者简介]

宋宣妍（1997-），女，信息工程大学硕士生，主要研究方向为无线物理层安全、无线物理层密钥生成技术等。E-mail：songxuanyan1@163. com

金 梁（1969-），男，信息工程大学教授、博士生导师，主要研究方向为移动通信技术、阵列信号处理、物理层安全等。

黄开枝（1973-），女，信息工程大学教授、博士生导师，主要研究方向为通信信号处理及无线通信安全。

肖帅芳（1989-），男，信息工程大学助理研究员，主要研究方向为无线移动通信。

无线内生安全:问题、属性、构造与功能

胡晓言[1]，金梁[1,2]，楼洋明[1,2]，钟州[1,2]，邬江兴[1,2]

[1]战略支援部队信息工程大学，河南 郑州 450002；

[2]网络通信与安全紫金山实验室，江苏 南京 211111

摘　要：无线通信系统的信息安全性和功能可靠性问题一直是关注的焦点，基于动态异构冗余（Dynamical Heterogeneous Redundancy， DHR）的内生安全原理为无线通信安全技术的发展提供了新方向。文章首先梳理了当前无线通信面临的内生安全问题，然后分析了无线信道天然具有的内生安全属性；在此基础上，定义了无线通信的内生安全构造；最后列举了无线通信的内生安全功能及应用。

关键词：无线通信、内生安全、信息安全、功能安全、DHR

Wireless Endogenous Security and Safety: Issues, Attributes, Structures and Functions

HU Xiao-yan[1], JIN Liang[1,2], LOU Yang-ming[1,2], ZHONG Zhou[1,2], WU Jiang-xing[1,2]

1.PLA Strategic Support Force Information Engineering University, Zhengzhou 450002, China;

2.Purple Mountain Laboratories: Networking, Communications and Security, Nanjing 211111, China

Abstract: The information security and functional reliability of wireless communication systems have become the focus of current research. The endogenous security principle based on Dynamic Heterogeneous Redundancy (DHR) provides a direction for the development of wireless communication security and safety technology. The article first sorts out the endogenous security problems faced by the current wireless communication system, and then analyzes the endogenous security and safety attributes of the wireless channel. After that, the endogenous security and safety structure model of the wireless communication system is given, and finally the applications of the existing wireless communication endogenous security and safety functions are listed.

Key words: wireless communication; endogenous security; information security; functional safety; DHR

1　引言

无线信号以电磁波为载体，在空间中以光速自由、开放地传输信息，极大地方便了人们的生活，推动了社会进步与发展。与此同时，无线通信面临的信息安全性和功能可靠性问题也受到广泛关注。无线信号自由传播地同时也暴露了电磁波的内源性“基因”缺陷，即在信号覆盖范围内任何人都可以在物理层实现窃听或攻击。而现有安全机制都在高层设计，主要沿用了传统有线通信中的加密机制，无法对因无线信道开放性带来的安全问题做到精准施策。一方面，随着计算能力的发展，加密算法在未来难免存在被攻破的风险。例如，2G 中的 A51、A52 加密算法早已被攻破。3G 中的 KASUMI 加密算法由于存在漏洞，也在 2013 年被有条件破解[1]。另一方面，若继续在无线网络中沿用“资源规模堆叠”、“技术捆绑集成”等传统“拼盘”式的安全机制设计思路，势必导致安全大量占用通信资源，降低网络整体效能[2]。此外，随着未来的无线接入设备日趋多样，

基金项目：国防科技创新特区项目；重点院校和重点学科专业建设项目；国家自然科学基金资助项目（No. 61871404）

Foundation item: The National Natural Science Foundation of China(61871404); National Defense Science and Technology Innovation Special Zone of China.

无线环境愈发复杂，通信链路面临着更加严峻的安全形势，空中接口的安全短板带来的水桶效应也会更加明显[3]。因此，亟需从无线安全问题的本性属性着手探寻新质安全元素，弥补安全短板。

近几年，安全与通信的同步研究正在成为新的发展趋势[4]。例如，2017年美国提出将安全作为关键要素，创建“独一无二、从一开始就受到保护的5G通信网络”。我国国家战略也指出网络安全与信息化需统一谋划、统一部署、统一推进、统一实施，并在5G研发布局之初就启动对5G安全的研究，部署了“未来无线接入物理层与系统安全通信技术研究”等系列重大安全项目，意图实现5G网络服务与安全功能融合式发展。在这一背景下，集安全与通信于一体的内生安全技术不断涌现。例如，华为针对IP网络固有安全威胁设计全新的IP协议，提出安全网络架构；奇安信基于“数据驱动安全”的理念，提出以数据为核心重构安全生态。这些技术的诞生标志着网络安全技术由外挂式不断地向内生性转变[5][6]，并且，在这一过程中逐步形成了诸多新的安全理念。其中，邬江兴院士提出的内生安全理论最具代表性[4]-[10]。内生安全理论是利用具有动态、异构、冗余属性的内生安全构造，实现内生安全功能，具有先验知识依赖度低、能够抵御已知和未知安全威胁、架构开放融合等内生安全效应。近些年来，内生安全理论在理论探索、技术突破、系统研制等方面均取得了重要成果，因此有望成为未来无线网络安全研究的革新方向，提供具有代际效应的特色安全增量[7]。

本文无线通信的内生安全问题、属性、构造和功能这四个方面展开，首先介绍了电磁波及网络空间内源性缺陷产生的内生安全问题，然后概述了无线通信系统自身构造或运行机理中的内生安全属性，进而探讨了基于这些属性的内生安全构造，最后介绍了几种现有的无线内生安全功能及关键技术。

2 内生安全问题

2.1 网络空间内生安全问题

推动无线通信安全向内生的“先天免疫”转变，首先应理清什么是内生安全问题。本质上，内生安全问题与网络空间是相互联系、相互依存的，这是因为任何功能往往都存在伴生或衍生的显式的副作用或隐式的暗功能，这些副作用可能是良性的，也可能是不良的，但暗功能则完全是未知的。因此，在网络空间的各类软硬件发挥功能之时，无法根除的漏洞及后门也会产生副作用或暗功能[8]。如果这些副作用或暗功能影响了系统的安全性，则称它们为“内生安全”（Endogenous Safety and Security，ESS）问题。一旦内生安全问题被人为或自然引起，就会引发的非期望的安全事件，称为广义不确定扰动[9]。大量的网络安全事件表明，网络空间内生安全问题往往是由自然或人为导致的广义不确定扰动这个外因，通过目标对象自身存在的显式的副作用或隐式的暗功能之内因相互作用而形成的[10]。

2.2 无线通信内生安全问题

无线网络也存在着上述普遍的内生安全问题。由于不同体制、架构地无线网络不断融合和异构共存，以及eMBB、mMTC和uRLLC场景中海量的异构通信节点的接入，任何一个芯片、软件、接口，都有可能是威胁的来源。攻击者在开放的无线网络架构下，可以实施“单向透明、里应外合”协同攻击，而无线网络中的防御者在面对未知协议漏洞、未知设备后门和未知攻击时，势必无法有效应对信息保密性、完整性、可用性、可信性等方面产生的不确定威胁。

除此之外，无线通信网络还面临着电磁传播内源性缺陷带来的特有的内生安全问题。由于电磁环境的不确定性和不可操控性，无线通信在具有广播特性（良性副作用）同时，也不可避免的引入了环境噪声和干扰（非良性副作用）。同时，这还会导致不可预测的暗功能，即任何地方都可能截获无线信号（窃听、定位），任何地方都可能发起无线攻击（接入、干扰）。上述副作用和暗功能使得无线信号即面临着自然和人为因素导致的通信可靠性问题（即**功能安全Safety**），又面临着被截获、假冒、篡改等各种人为扰动带来的保密性和可信性问题（即**信息安全Security**），这些都属于电磁空间传播机理的内源性缺陷引发的广义不确定扰动（即自然因素扰动和人为因素扰动的总和）而产生的内生安全问题。

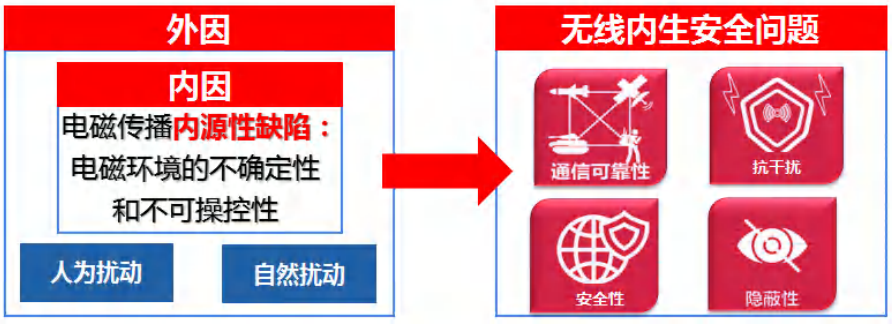


图1　无线内生安全问题

3　内生安全属性

在面对内生安全问题时，传统网络安全手段大多采用基于威胁感知的精确防御，如打系统补丁、查毒杀马乃至设蜜罐、布沙箱等亡羊补牢式的防护措施。这些方式只有在获得攻击来源、特征、途径、行为、机制等先验知识的前提下才能实施有效防御，而对于难以根除的副作用或暗功能带来的不确定威胁，既不能彻底规避，也无法预知或化解针对未知威胁的攻击。为了应对这一网络安全难题，文献［4］认为既然内生安全问题出自结构本身，就应遵循“结构决定功能”的内源性安全机理，设计具有内 生安全属性的信息系统架构。在这一理念的启发下，文献［9］［10］提出并完善了基于拟态防御的内生安全理论，该理论以动态、异构、冗余（Dynamic Heterogeneous Redundancy， DHR）属性的安全架构为核心，利用“有毒带菌”、不可信、不可控的软硬构件，将安全、可信的“基因”赋予网络信息系统和设备诞生之初，建立内生的网络安全机制，“条件规避或化解”基于内生安全问题的未知安全缺陷的不利影响。

如上所述，解决内生安全问题的关键在于利用内生安全属性，实现内生安全功能。不同于有线网络系统中人为构造的DHR属性，无线通信系统本身就具有天然的异构、动态、冗余的内生安全属性。其中，异构、动态特性主要源于无线信号在传输过程的多径效应和噪声的影响，造成信道的特征复杂多变、难以预测[11]。不同位置、不同波长的信道特征不相关，使得合法用户的信道特征相对于攻击方具有天然的“测不准”特性。在此基础上，利用冗余的通信资源可以进一步提升无线通信系统的鲁棒性，抑制干扰和噪声因素的影响。因此，基于DHR属性的内生安全原理对于无线通信安全机制的研究具有启示意义，是无线内生安全问题新的解决之道。本节从电磁波传播原理的角度分析了无线通信的内生安全属性，为研究无线通信内生安全构造，开发内生安全功能提供了思路。

3.1　异构性

无线通信的内生安全属性主要源于复杂的电磁传播环境。其中，无线信道的异构性是实现内生安全功能的关键。本小节用时变的冲激响应，分析多径效应影响下无线信道的异构性。假设发送信号$x(t)$经过N个信道传播到达接收机，$x(t)$的载波频率为f_c，则$x(t)$的实部可表示如下：

$$x(t)=\Re\left\{x(t)e^{j2\pi f_c t}\right\} \tag{1}$$

考虑每条路径的多普勒频移，则第n条路径到达接收机的信号为：

$$R_n(t)=\mu_n(t)x\left[\tau-\tau_n(t)\right]e^{j\left(2\pi f_c\left(t-\tau_n(t)\right)-\phi_{D_n}(t)\right)} \tag{2}$$

其中$\mu_n(t)$、$\tau_n(t)$和$\phi_{D_n}(t)$为第n条路径的接收信号振幅、时延和多普勒相移，一般认为三者相互独立。在无噪声的条件下，经过N条路径叠加后的接收信号为：

$$R(t)=\sum_{n=1}^{N}\mu_n(t)e^{-j\phi_n(t)}x\left[\tau-\tau_n(t)\right] \tag{3}$$

其中$\phi_n(t)=2\pi f_c\tau_n(t)-\phi_{D_n}(t)$。由于接收信号为发送信号和信道冲激响应的卷积，故信道冲激响应可表示为：

$$h(t)=\sum_{n=1}^{N}\mu_n(t)e^{-j\phi_n(t)}\delta\left[\tau-\tau_n(t)\right] \tag{4}$$

由上式易知，无线信道由若干个相互独立的、具有不同时延和多普勒频移的电磁波叠加而成，

这说明在无线信号的传输过程中，会经历直射、反射和散射等不同的传播方式，以不同的路径、不同的时间到达接收端，导致各个路径的信号按各自相位相互叠加后产生随机的多径衰落[12]。当多径传播过程中没有直达径，且n足够大时，信道环境可视作均匀散射环境，根据中心极限定理，$h(t)$是具有复高斯分布的广义平稳随机过程，对于均匀散射环境中接收同一信号源的两个不同位置的接收机，信道相关函数可以表示为：

$$\rho = \frac{E\left[h_1(t)^{\dagger}h_2(t)\right]}{D\left[h_1(t)\right]D\left[h_2(t)\right]} = J_0(2\pi\frac{d}{\lambda}) \qquad (5)$$

其中$J_0(\cdot)$为零阶贝塞尔函数[13]，λ为载波波长，d两个接收机的距离。由上式可见，两个不同位置接收机接收同源信号时，无线信道之间的相关性会随着距离的拉大而降低。实际上，即使信道环境中的散射体不够丰富，无线信道的传播环境也具有一定程度的多样性，不同位置的信道的多径衰落也不相同［14］。由此可见，无线信道受环境**多样性**的影响，具有**异构性**，即该位置处的信道特征信息对于不同位置的攻击者具有“测不准”属性。

此外，当信号载波频率变化时，同一位置的无线信道也会发生改变。这是由于电磁波沿不同路径传播后，即使传播的路程差不变，电磁波到达接收端的波程差也会不同，使得叠加后的多径衰落发生变化，因此同一位置不同频率的无线信道也具有异构性。

3.2 冗余性

无线信道的冗余性来源于信道的开放性。与传统有线通信相比，无线通信以电磁波取代电缆作为信息传输的载体，摆脱了对通信终端位置的桎梏，具有在空间中以光速进行自由、开放传播的物理特性，在一定程度上模糊了通信边界的限制，使得接收范围内各个位置的接收机都可以获取信号并还原信息内容，因此无线信道具有**空域冗余性**[14]。

除了无线信道在空间上的冗余性，对于空间上收发双方位置固定的情形，信号还可以通过不同的载波频率发射。当采用两个或两个以上具有一定频率间隔的电磁波频率同时发射时，接收端通过将多个射频信号按一定权重合并，可以提高信息传输的鲁棒性。这是由于载波频率不同时信道的衰落特性不相关，采用多个频率发射可以有效对抗频率选择性衰落，因此无线信道也具有**频域冗余性**。在无线通信系统中，利用这两类冗余资源均可以换取传输信息可靠性的提升，使系统具有容忍噪声、干扰或随机故障的能力。

3.3 动态性

信道衰落包括快衰落和慢衰落。其中快衰落主要源于多径效应，是指由发射机、接收机和传播路径周围的各类散射、反射和折射体造成的多径传播，使多径信号在接收机相互叠加，引起接收信号幅度快速起伏变化的现象。慢衰落是指由发射机或接收机的运动，或者电磁波传播环境变化，造成衰落的局部中值不断变化的现象。从时间维度上看，慢衰落比多径效应引起的快衰落要慢得多。具体而言，传播环境包括传播媒介、气象条件、传播距离、地形等因素的变化，例如大气湍流、大气层结变化而引起的空气密度变化，或是信号穿过楼宇、雾气等不同媒质，均会导致慢衰落的产生。由于以上两类信道衰落都是随着时间变化的，因此也被称为时间选择性衰落，无线信道的**动态性**正是来源于信道衰落的**时变性**。

尽管信道衰落在多数通信系统中被视为一种有害于传输的可靠性的因素，但在网络空间内生安全理念的启发下，可以将无线信道作为一种具有动态、冗余特性的异构执行体，构建拟态的防御机制，使防御场景和行为获得“测不准”属性，从而显著增加攻击难度。并且无线信道所具有的DHR特性是天然存在的，不依赖人为构建，相比于传统架构减少了工程上的开销。

4 内生安全构造

典型的网络空间DHR模型如图3所示，主要包括输入代理、异构执行体和输出代理组成，三者所包含的区域称为拟态界。其中输入代理根据负反馈控制器的指令，动态地从m个异构执行体中选出相应的（多个）异构体作为一个执行体集，并将输入送至执行体集中；然后，多个异构执行体集受到输入激励，独立地产生输出矢量并多模裁决；最后，多模裁决器研判输出内容的一致性情况。当输出内容不一致时，负反馈控制器被激活，并根据控制参数生成的控制算法决定是否要向输入代理发送替换（迁移）异常执行体的指令，

或者指示异常执行体实施在线/离线清洗恢复或等价重组重构重配等操作，直至输出矢量一致或差异低于给定阈值。不难看出，功能等价、结构变化的执行体可以使攻击者对目标系统的结构、运行机制产生认知困境，从而干扰或瓦解攻击链的稳定性。因此，DHR架构本身具有高可靠、高可用和高可信的“三位一体”的内生安全属性，可以有效应对基于未知漏洞、后门、病毒或木马等未知副作用和暗功能引发的内生安全问题，为破解当前网络空间安全难题的提供了具有创新意义的方法[10]。

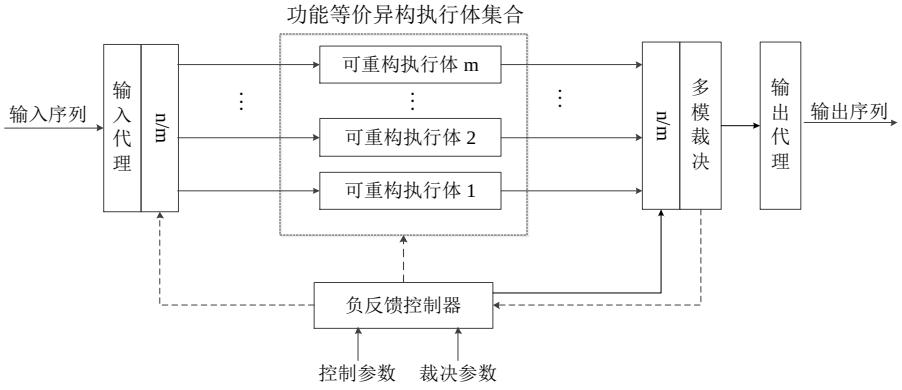


图2　典型DHR模型

利用无线信道天然具有的内生安全属性，可以构造内生安全模型，实现内生安全功能。无线通信系统的DHR架构可归纳为如图3所示，与典型拟态系统对应的异构执行体是无线信道，输入代理为发信机，输出裁决器为接收机，以上三者所包含的区域称为拟态界。无线通信内生安全的研究范围包含拟态界内的无线空口环境及射频链路，在信号层面保障信息安全和功能安全。其中，信息安全主要是保护信息的机密性、可信性和完整性，功能安全主要是指保障信号的可靠传输。此外，输出裁决器进行的分集技术（如最大比合并等），可以减小随机噪声带来的影响，类似于拟态系统输出代理中的拟态裁决机制；发信机利用接收机发送的导频估计信道后，进行的波束形成技术可类比于拟态系统的输入分发、策略调度和反馈控制。

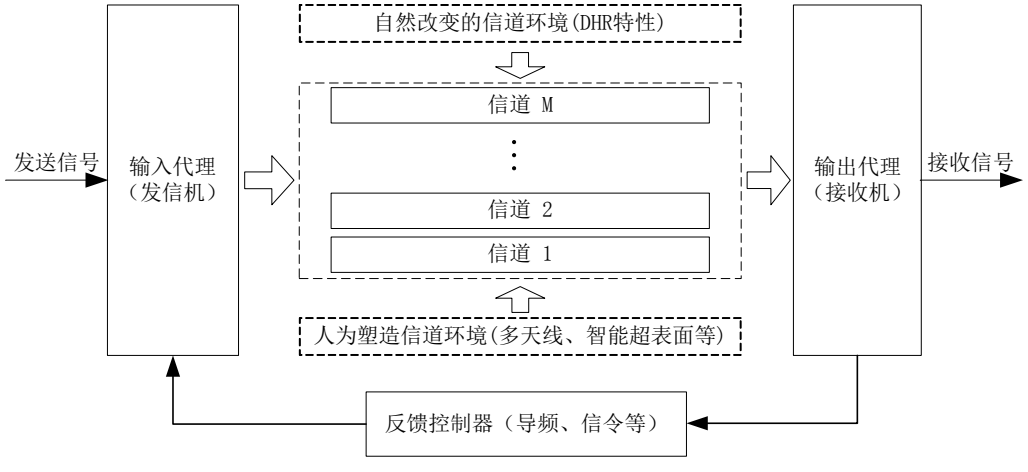


图3　无线通信系统的内生安全构造模型

如上所述，无线内生安全的实质是在利用信道的内生安全属性（DHR属性）的基础之上，解决自然和人为的广义不确定扰动问题，而经典无线通信的实质是在对抗自然信道不确定性带来的扰动。两者的差别在于设计理念，无线内生安全是建立在对电磁环境的精细认知、优化定制和精

细操控基础之上，而经典无线通信则是被动适应电磁环境。两者的共同点在于，都是利用了信道的内生安全属性。这说明，不同于传统的捆绑式、拼接式的安全机制，无线内生安全与经典无线通信之间存在着共生关系，两者既是内生的融合，又能共生的发展。例如，信道状态信息（Channel State Information，CSI）估计技术的进步不仅能更好的对抗信道随机衰落，也能更好的利用CSI对信号加扰，增强内生安全算法。类似的还有信道编码技术，一种好码既可以提升通信系统的鲁棒性，也有利于物理层安全传输技术得到实现。

5　内生安全功能及应用

依靠无线通信内生安全架构可以实现内生安全功能，解决内生安全问题。本节举例说明了无线通信内生安全功能的两种主要应用，包括无线内生信息安全技术和无线信号功能安全与鲁棒性控制技术。

5.1　无线内生信息安全技术

无线内生信息安全技术的关键利用电磁波的传播机理，提取内生安全元素——信道指纹。信道指纹是自然界中一种天然的随机源，源于电磁波传播过程中直射、反射、衍射、散射、折射等各种效应组合，其产生机理决定了信道指纹具有信道的异构性、空时频资源的冗余性和无线传播环境变化的动态性的特点，可用于实现内生安全功能。如图4所示，信道指纹可用于搭建安全通信的“专属信道”，既可以实现对信号随机加扰的安全传输技术[15][16][17]，也可以实现逼近“一次一密”的物理层密钥生成技术[14][18][19]。

物理层安全传输技术的实质是根据信道指纹

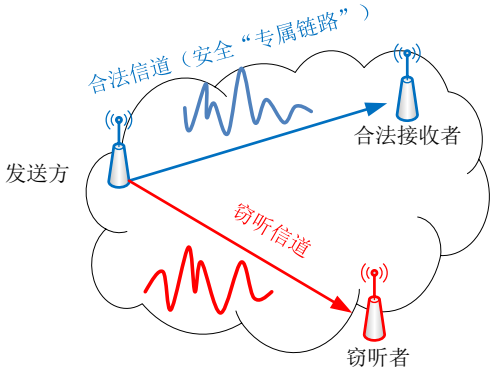


图4　基于信道指纹的无线内生信息安全技术

的差异设计与位置强关联的信号传输和处理机制，使得只有在期望位置上的用户才能正确解调信号，而在其它位置上的信号是置乱加扰、污损残缺、不可恢复的。物理层密钥生成技术的实质是利用通信双方私有的信道指纹，提供实时生成、无需分发的快速密钥更新手段，使得生成的密钥可以阻断任何企图通过试错方法达成的协同破解。并且，这种逼近“一次一密”的安全机制即使面临着攻击者暴力破解，成功破解的成果也无法继承。以上两种无线内生安全功能的实现都不依赖攻击者任何先验知识和行为特征，因此对独立的攻击资源、攻击技术和方法形成的“差模攻击效应”具有天然的抑制功效。与基于DHR架构的拟态防御技术类似，对于无线内生信息安全技术，突破其防御只有通过时空一致性的精准协同攻击才有逃逸的可能。然而，攻击者首先要克服时空非一致性的“测不准效应”，要逾越“异构环境形成的物理隔离距离”才可能形成“共模逃逸”态势。理论上，“差模逃逸”不可能发生，而“共模逃逸”也是极小概率事件。因此，在无线内生安全架构内的攻击行动或成果都不具有稳定鲁棒性和品质鲁棒性。

“一次一密”的存在性条件已经在文献[20]中得到了初步证明。如图5所示，该文献指出客观环境的信息量是足够大的，传递信源的通信过程中天然蕴含了与之等量的环境信息量，可提供足够的密钥。因此，只要能够实现对环境精细认知和塑造，那么不论可达通信速率多高，都可以从环境信息这个内生“富矿”中提取与之匹配的密钥速率，并且提取过程中不耗散额外的能量、不占用额外的信道资源。该文献也说明，我们可以通过一些使能技术增大信道指纹的随机性与动态性。例如，以超材料智能表面为代表利用材料科学与信息科学交叉融合产生的非线性增益还可强化赋能内生安全，为内生安全功能的实现提供关键的技术支撑。

超材料智能表面具有对电磁波传播方向、相位、强度、极化等内生属性进行高效、快速、灵活调控的特性，可以丰富、放大、加速电磁环境的异构复杂度和随机时变性，并通过对电磁波传播的精细操控达到按需实时重构无线环境的效果，提高对电磁波传输性态的操控自由度，从而“人

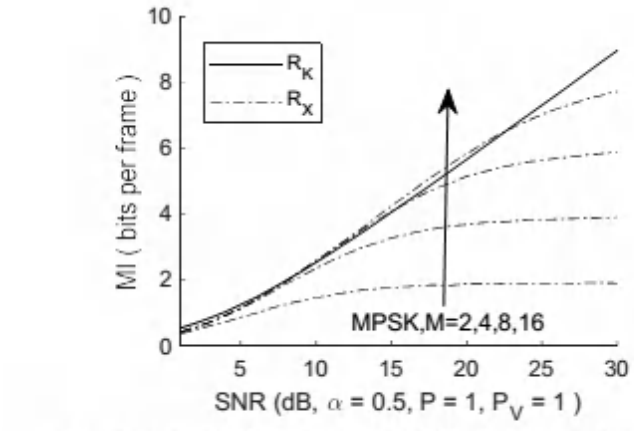


图5　通信过程中实现一次一密的存在性仿真

工编辑”无线信道的内生安全基因，加速密钥生成。文献［21］也给出了一种通过充分探索电磁环境和信号熵的方法，可以在静态环境中提高密钥生成速率。这些使能技术都有助于实现基于电磁环境内生安全属性的“一次一密”内生安全功能。

此外，射频指纹源于发射机中的电路容差和元器件容差（制造容差和漂移容差）、工艺条件等造成的系统不一致性，其产生机理决定了射频指纹具有唯一性和第三方不可仿冒性，这一内生于无线设备的物理属性天然适合作为认证的可信根，对增强认证机制、实现海量设备的快速认证提供了技术可行性，也可用于保护信息安全[22]。

5.1　无线内生功能安全技术

无线内生安全不仅能提供高可信的信息安全，也能提供高可靠、高可用的功能安全，实现通信、安全和抗干扰的一体化内生安全功能。许多传统无线通信技术的目的都属于保障通信的功能安全，例如扩频通信［23］，多天线分集技术等，它们主要是对抗包含自然信道的不确定干扰和人为扰动在内的广义不确定扰动。而无线内生功能安全需要在辨识无线信道天然安全基因基础上，进行人为可控的基因优化与改造，减少和消除环境的不确定性和不可操控性，实现对电磁环境恶性因素的非特异性免疫。

例如，无线通信抗干扰技术就是一种典型的内生功能安全技术。抗干扰技术按域一般可分为时域[24]、空域[25]和频域[26]抗干扰，其本质均是利用无线信道的DHR特性，以经典的跳时、跳频和跳空通信为例，它们分别利用时域、频域或空域冗余性，根据跳变图像随机选择不同的时间、频点或天线进行通信，其根本都在于冗余构建异构执行体的跳变，使得攻击方无法有效实施窃听和干扰攻击。抗干扰的策略集包括时域、频域和空域以及混合域抗干扰技术，由信宿的策略表决反馈至策略调度进行抗干扰策略选择，再由信源选择机制创建异构执行体进行通信。例如，信源首先选择跳频策略进行传输，经过异构执行体到达信宿后进行频域滤波、解调。如果无法满足性能需求，则信宿将其反馈至策略调度，重新选择空时混合域的抗干扰策略，之后再经过新的异构执行体后到达信宿进行空域滤波，直到能够准确解调，则实现了抗干扰策略的收敛。因此，与基于DHR架构的内生信息安全技术类似，无线通信抗干扰技术引入了动态化和随机化要素，使得防御机制具有测不准效应，能够有效增加攻击方的干扰难度。

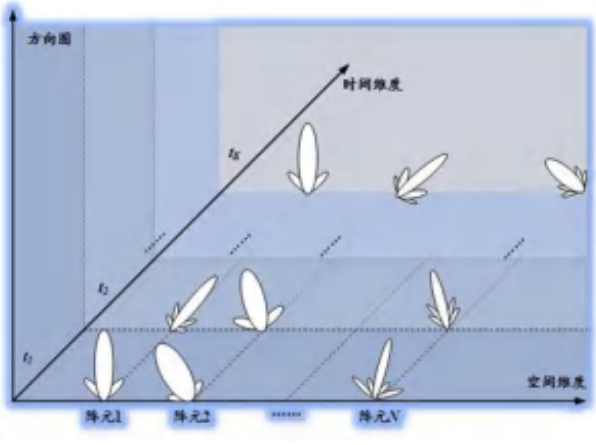


图6　空时异构阵元模型图

此外，从内生理念出发，借助超材料等使能技术，还可以在精细控制无线信道的基础上，进一步挖掘阵元多样性潜力。例如超材料表面可为收发双方提供对电磁环境的精细感知能力，丰富、放大、加速电磁环境的异构复杂度和随机时变性，使无线信道的开放性向个性化定制演进。基于超材料打造的DHR天线阵列，通过空域异构构造，使得不同位置的阵元具有异构、不相关的方向图，使得空域资源摆脱空谱墙的限制，能够在天线孔径受限条件下提升效能。如图6所示，理论上N个阵元可实现NK的实际自由度，接收容量随（时域异构度×空域异构度）线性增长[27]。因此，开发有规模决定论向构造决定论转型发展，拓展内生安全的新维度

5 结束语

本文从内生安全原理的角度，梳理了无线通信普遍和特有的内生安全问题，分析了无线内生安全属性——DHR特性，给出了无线内生安全的构造模型，最后介绍了无线内生信息安全和功能安全的应用。可以预见的是，在无线通信技术的未来发展中，网络空间内生安全理论将发挥如文中所述的重要引导作用。并且，无线通信系统特有的DHR架构也将会成为网络空间内生安全理论的重要补充。以未来广泛存在的海量机器类通信场景为例，通信节点具有密集分布、高并发通信、低通信时延、动态迁移的特点，使移动互联网络面临着更为复杂的安全管理难题，人为引入DHR特性势必增加体积功耗、维护使用等方面的开销。而采用无线信道作为异构执行体可以在有效降低攻击者的协同性的同时，放宽DHR架构对软硬件实现结构或算法上的相异性要求，从而减少引入异构冗余的实现成本，具有重要的工程意义。由此可见，内生安全理论与传统无线通信的信息安全与功能安全技术的相互融合、相互渗透，不仅能够为开发新的无线通信内生安全技术提供指导，也能够为基于拟态原理的网络空间内生安全理论提供进一步发展的思路。

参考文献:

[1] Dunkelman O, Keller N, Shamir A. A practical-time related-key attack on the KASUMI cryptosystem used in GSM and 3G telephony [J]. Journal of cryptology，2014，27(4)：824-849.

[2] 季新生, 黄开枝, 金梁,等．5G安全技术研究综述[J]. 移动通信，2019.

[3] 杨红梅, 林美玉．5G网络及安全能力开放技术研究[J]. 移动通信，2020，44(4)：65-68.

[4] 邬江兴．拟态防御技术构建国家信息网络空间内生安全[J]. 信息通信技术，2019(6).

[5] 张铮, 马博林, 邬江兴．web服务器拟态防御原理验证系统测试与分析[J]. 信息安全学报，2017，2(1)：13-28.

[6] 刘彩霞, 季新生, 邬江兴．一种基于MSISDN虚拟化的移动通信用户数据拟态防御机制[J]. 计算机学报，2018，41(2)：275-287.

[7] 邬江兴．"网络安全再平衡战略"之抓手: 拟态防御[J]. 中国信息安全，2018(06)：46-50.

[8] 宋克, 刘勤让, 魏帅,等．基于拟态防御的以太网交换机内生安全体系结构[J]. 通信学报，2020，041(05)：18-26.

[9] 邬江兴．网络空间拟态防御导论[M]. 北京：科学出版社，2018.

[10] 邬江兴．网络空间拟态防御原理——广义鲁棒控制与内生安全[M]. 北京：科学出版社，2020.

[11] 王旭, 金梁, 刘璐,等．均匀散射环境中物理层安全密钥容量分析[J]. 通信学报，2016，37(09)：75-81.

[12] Jakes W C, Cox D C. Microwave mobile communications[M]. New Jersey，USA: Wiley-IEEE Press，1994.

[13] Chen C, Jensen M A. Secret Key Establishment Using Temporally and Spatially Correlated Wireless Channel Coefficients [J]. IEEE Transactions on Mobile Computing，2010，10(2)：205-215.

[14] 楼洋明, 金梁, 钟州,等．基于MIMO接收信号空间的密钥生成方案[J]. 中国科学:信息科学，2017，047(03)：362-373.

[15] Goel S, Negi R. Guaranteeing Secrecy using Artificial Noise [J]. IEEE Transactions on Wireless Communications，2008，7(6)：2180-2189.

[16] 钟智豪, 罗文宇, 彭建华,等．多层异构蜂窝网协作传输和协作干扰机制的安全性能分析[J]. 中国科学:信息科学，2016(1)：33-48.

[17] 张波, 黄开枝．异构携能通信网络中基于人工噪声辅助的鲁棒安全传输方案[J]. 电子与信息学报，2019，041(001)：1-8.

[18] Maurer U M. Secret key agreement by public discussion from common information[J]. IEEE Transactions on Information Theory，1993，39(3)：733-742.

[19] 胡晓言, 金梁, 黄开枝,等．基于信号传播特性的物理层密钥生成方案[J]. 电子学报，2019，47(02)：229-234.

[20] Liang Jin, Xu Wang, Yangming Lou, et al. Achieving one-time pad via endogenous secret key in wireless communication[C]. IEEE International Conference on Communications in China. Chongqing, China. 2020. (accepted)

[21] Liang Jin, Shengjun Zhang, Yangming Lou, et al. Secret Key Generation with Cross Multiplication of Two-way Random Signals [J]. IEEE Access，2019：113065- 113080.

[22] 袁红林, 胡爱群．射频指纹的产生机理与惟一性[J]. 东南大学学报(自然科学版)，2009(02)：230-233.

[23] Madhow U, Honig M L. MMSE interference suppression for direct-sequence spread-spectrum CDMA [J]. IEEE Transactions on Communication，1994，42(12)：3178-3188.

[24] AdemN, HamdaouiB, YavuzA. Pseudorandom Time-Hopping Anti-

Jamming Technique for Mobile Cognitive Users［C］. IEEE Globecom Workshops，2015：1-6.

[25] 郭素霞, 李翔宇, 金梁,等．基于空域信道跳变抗干扰DOA估计方法[J]. 中国科学:信息科学，2016，46(7)：899.

[26] HanawalMK, Abdel-RahmanMJ, KrunzM. Joint Adaptation of Frequency Hopping and Transmission Rate for Anti-Jamming Wireless Systems［J］. IEEE Transactions on Mobile Computing，2016，15(9)：2247-2259.

[27] Liang Jin, Yangming Lou, Xiaoming Xu, et al. Separating Multi-Stream Signals Based on Space-Time Isomerism［C］. 2020 International Conference on Wireless Communications and Signal Processing (WCSP). Nanjing，China. 2020

[作者简介]

金 梁（1969-），男，国家数字交换系统工程技术研究中心教授、博士生导师，主要研究方向为移动通信技术、阵列信号处理、物理层安全等。E-mail：liangjin@263. net

胡晓言*（1992-），男，国家数字交换系统工程技术研究中心在读博士研究生，主要研究方向为物理层安全及信息安全。E-mail：ndscpls@163. com

楼洋明（1991-），男，国家数字交换系统工程技术研究中心助理研究员，主要研究方向为移动通信、通信信号处理及信息安全。

钟州（1982-），男，国家数字交换系统工程技术研究中心讲师，主要研究方向为移动通信、通信信号处理及信息安全。

邬江兴（1953-），男，国家数字交换系统工程技术研究中心主任，教授，博士生导师，主要研究方向为信息通信网络、网络安全。 Email： 17034203@qq. com

基于OFDM信号的无线内生安全密钥性能研究

——邓哲元，金梁，黄开枝，陈亚军（战略支援部队信息工程大学，河南 郑州 450002）

摘　要：随着无线通信的不断发展，现有的通信安全防护手段已经越来越难以防范已有的或即将到来的安全威胁。近年来，内生安全概念得到迅速发展，因此将内生安全概念引入无线通信是顺应时代发展。无线内生安全密钥技术是移动通信安全技术中一个热门研究领域。然而目前针对其研究大都停留在理论证明和方案设计上，实验验证较少，尤其是缺少对基于CSI的密钥生成中基本问题的实验研究。本文采用软件定义无线电（Software Defined Radio， SDR）设备实现了基于正交频分复用（Orthogonal Frequency Division Multiplex， OFDM）信号的内生安全密钥提取系统，通过该系统我们可以获取CSI信息。首先研究了静态和非静态时的CSI变化情况，并从子载波间隔和子载波个数两个方面探究了CSI对密钥性能的影响。

关键词：无线通信、内生安全、密钥生成

Research on performance of wireless endogenous security key based on OFDM signal

DENG Zheyuan, JIN Liang, HUANG Kaizhi, CHEN Yajun

PLA Strategic Support Force Information Engineering University, Zhengzhou 450002, China

Abstract: Wireless endogenous security key technology is a hot research field in mobile communication security technology. However, at present, most of the researches focus on the theoretical proof and scheme design, and there is little experimental verification, especially the lack of experimental research on the basic problems of key generation based on CSI. In this paper, We implemented an endogenous secure key extraction system based on OFDM signal by Software Defined Radio (SDR). The channel changes under static and non-static conditions are studied, and the influence of sub carrier spacing and number of subcarriers on key performance is explored.

Key words: wireless communication; endogenous security; key generation

1　引言

邬江兴院士在［1］中说“任何自然的功能或人造的功能都存在伴生或衍生的显式的副作用或隐式的暗功能。副作用可能是良性的，也可能是不良的，但暗功能的性状及影响则完全是未知的”。简言之，任何东西有利必有弊，这种由内源性缺陷带来的安全问题被称为内生安全问题。

无线通信利用电磁波承载信息，其借助电磁波的广播特性可与传播区域内的任意节点通信，使人们摆脱了有线通信的束缚。然而，正是这种广播特性，使得攻击者在电磁波传播区域内的任何地方都可能截获无线信号（如窃听）或发起无线攻击（如干扰），给无线通信系统带来安全问题，我们称之为无线内生安全问题。

本文主要针对无线信号在通信过程中被窃听的问题展开讨论。1949年，香农［2］首次提出了“完美加密”模型，奠定了信息理论安全和密码学基础。目前，针对通信中窃听问题，普遍的解决思路是用传统密码学算法对通信内容进行加密及认证，只有合法通信双方可以利用提前分发的密钥对通信内容进行加解密，得到实际通信信息。

基金项目：国家自然科学基金资助项目（No. 61871404）；国防科技创新特区项目；重点院校和重点学科专业建设项目

但是，随着量子计算领域的不断进步，基于计算复杂度的传统密码学算法安全性无法保证；其次，物联网时代已经到来，大量物联网终端对体积、功耗和计算资源方面有所限制，难以部署复杂的安全算法，急需轻量级加密算法；最后，传统密码学算法本质上都独立于无线通信，是一种被动式的安全方案，无法从根本上解决问题。因此，我们需要以电磁波传播特性为中心，从无线通信本质出发去研究问题，从根本解决无线内生安全问题。

电磁波在传播过程中，受环境中的散射体影响发生直射、反射、折射、衍射，从而产生多径信号使得无线信道具有时变性、互易性以及唯一性等属性［3］。其中，时变性指无线信道是时刻变化的，且该变化具有随机性，可确保无线信道特征不被预测；互易性指无线信道在相干时间内基本不变，可确保收发双方获得相同的无线信道特征；唯一性指无线信道与用户位置绑定，不同位置获得的信道特征不同，可确保窃听者无法获知合法用户的信道特征。基于以上三种特性，人们提出了基于信道特征的无线物理层密钥生成技术（简称“信道密钥”）。信道密钥由无线通信环境内生，又因无线信道特性可免去密钥分发环节，是一种真正意义上的环境内生技术，其可从根本解决无线内生安全问题，因此我们将信道密钥也称为无线内生安全密钥（简称“内生密钥”）。

内生密钥研究源于Maurer提出利用共享随机源生成密钥的方法［4］，而无线信道的三种特性满足了其对共享随机源的要求，因此针对内生安全密钥理论研究快速发展，［5］-［14］针对密钥生成中的问题进行了理论分析，并设计了密钥生成方案。由于商业无线收发器易于获取接收信号强度（Received Signal Strength， RSS）信息，因此目前大部分实验都基于RSS提取密钥。但RSS是信道的粗粒度信息，可获知的信道情况较少，导致普遍密钥生成速率不高。而CSI是信道的细粒度信息，可获知的信道情况更多，但仅有部分实验利用软件定义无线电（Software-defined Radio，SDR）［15］，［16］等设备提取CSI开展研究。［15］从相干时间、信道带宽、硬件指纹三个影响密钥生成的基本方面去对信道问题进行了研究探讨；［16］从该文选择了3个比较典型的信道场景（室内、走廊和室外）进行实验，并使用了一个新指标——图像熵来评价信道随机性。虽然上述实验研究对CSI中的许多问题进行了探讨，但对OFDM信号的特性并未做充足考虑。

本文采用GNU Radio结合SDR设备进行实验。通过GNURadio结合SDR的方式，我们可以用软件思维开发硬件，大大降低开发难度及周期。实现了基于IEEE 802.11帧结构［18］的OFDM信号收发机，首先研究了静态和非静态时的CSI变化情况，并从子载波间隔和子载波个数两个方面探究了CSI对密钥性能的影响。

本文剩下部分为：第2节阐述本文研究的系统模型及密钥生成流程；第3节介绍了本文搭建的基带信号收发链路；第4节通过采集的实际场景数据分析了不同情况下的密钥性能；第5节为本文结论。

2 系统模型及方案

2.1 系统模型

本文所应用的模型可抽象为图1。其中Alice和Bob是合法通信的双方，二者在初始阶段互相交换导频并测量CSI信息，CSI信息用于后续密钥生成。Eve为被动窃听者，其与Alice和Bob的距离大于半波长，以本文实验所在2.4G频段为例，半波长为6.25CM。三者均为单天线。

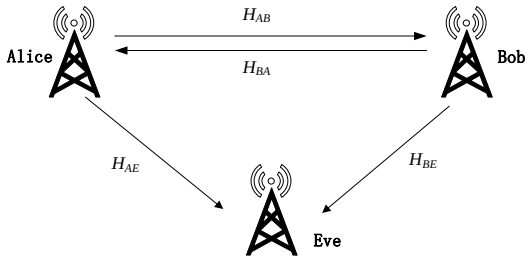


图1 系统模型

Alice和Bob通信后获得的接收信号分别为

$$Y_A = H_{BA}X_B + N_A \tag{1}$$

$$Y_B = H_{AB}X_A + N_B \tag{2}$$

Eve收到来自Alice和Bob的信号分别为

$$Y_{AE} = H_{AE}X_A + N_{AE} \tag{3}$$

$$Y_{BE} = H_{BE}X_B + N_{BE} \tag{4}$$

其中 X_A, X_B 表示Alice和Bob发送的信号；$H_{AB}, H_{BA}, H_{AE}, H_{BE}$ 表示3个节点间的无线信道；

Y_A, Y_B, Y_{AE}, Y_{BE} 表示3个节点的接收信号；N_A, N_B, N_{AE}, N_{BE} 表示接收信号的噪声。

2.2 无线内生安全密钥生成方案

收发双方分别为Alice和Bob，如图2，方案共分4个阶段：信道探测、量化、协商以及隐私放大。

（1）信道探测：目的是使Alice和Bob可以获得一致的信道特征值，本文利用的信道特征信息是CFR。

（2）量化：通过信道探测会得到信道特征的模拟值，而用于加密的密钥是二进制的0、1序列，因此该步骤完成的是模拟值到离散值的映射。

采用单门限量化［12］，通过式（5）将模拟值转换为离散比特。

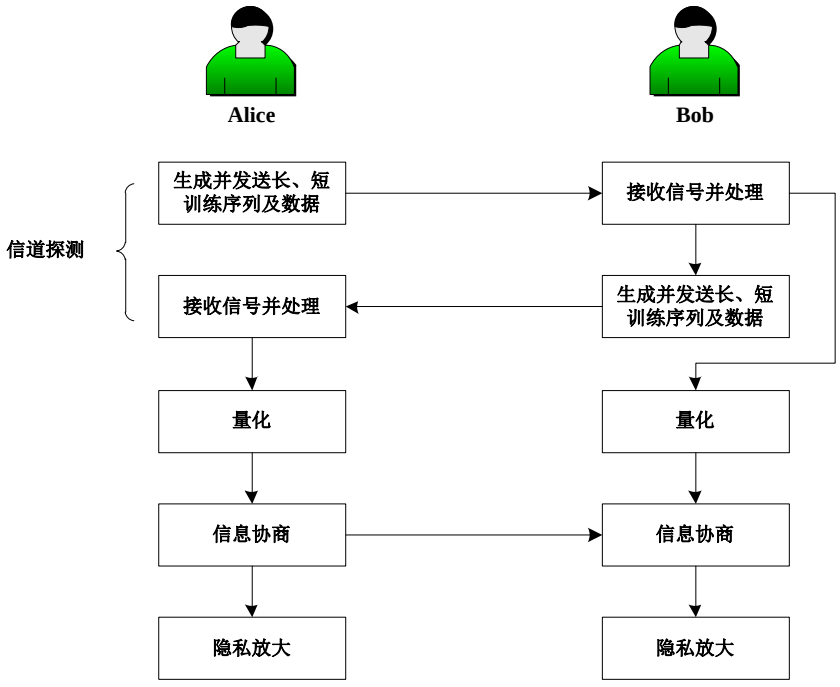


图2 方案流程图

$$Q_n = \begin{cases} 0, & H_n < T \\ 1, & H_n > T \end{cases} \tag{5}$$

其中，H_n 表示信道估计值 $\hat{H}_{AB}, \hat{H}_{BA}$ 的模，Q_n 表示量化值。

（3）信息协商：受信道噪声、接收机射频指纹等的影响，Alice和Bob获得的信道特征值并不能完全互易，因此二者需要进行交互，来纠正不一致的信息。

采用基于BCH码的信息协商方法［3］，其具体步骤如下：

1）Alice首先产生与量化序列 Q_A 等长的随机数 n，并用BCH码将其编码生成BCH(N,K)，再将BCH(N,K)与 Q_A 作异或生成协商序列S，将S发给Bob。

$$S = BCH(n) \oplus Q_A \tag{6}$$

2）Bob收到S后，与Bob的量化序列 Q_B 作异或后，对序列进行译码得到随机数 $\hat{n}$。再将 $\hat{n}$ 进行BCH编码并与S进行异或即可纠正Bob的错误比特。

$$\begin{aligned} \hat{n} &= BCH_{decode}\left(S \oplus Q_B\right) \\ &= BCH_{decode}\left(BCH(n) \oplus Q_A \oplus Q_B\right) \\ &= BCH_{decode}\left(BCH(n) \oplus e\right) \\ \hat{Q}_B &= BCH(\hat{n}) \oplus S \end{aligned} \tag{7}$$

经过上述过程后，Alice和Bob可以获得完全一致的密钥序列。

（4）隐私放大：由于信息协商中协商序列是在信道中明文传输的，因此会泄露一部分信息给窃听者。同时，协商后的密钥序列可能随机性不够。因此采用Hash函数对协商后序列进行隐私放大。

3 收发系统架构

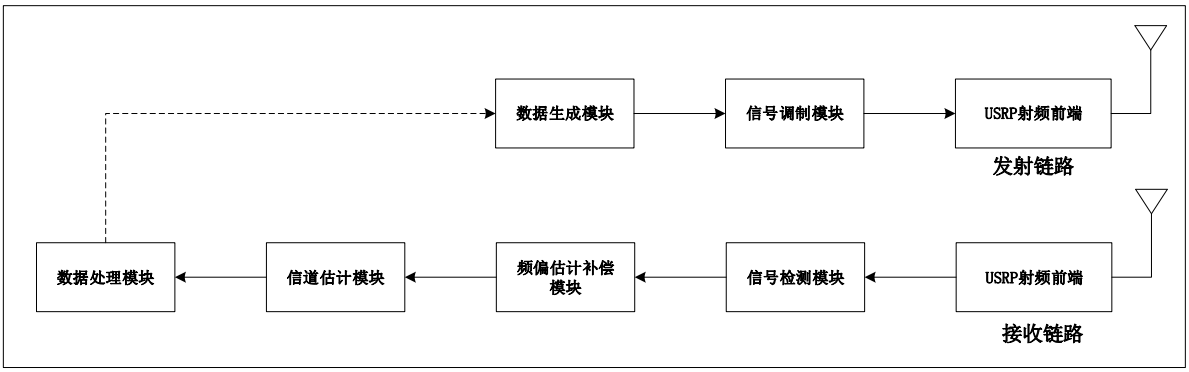


图3　无线收发机结构

发射链路主要包含3部分：

1） USRP射频前端：负责将基带信号上变频到射频信号。

2） 调制模块：该模块需要完成OFDM调制，生成基带信号。

3） 数据生成模块：该模块负责生成长、短训练序列及数据部分。

采用基于IEEE 802.11的非高吞吐模式帧结构[18]（如图3）。其中，短训练序列用于帧同步和频偏估计，该序列包含10个周期为16的短前导；长训练序列用于信道估计，该序列包含2个周期为64的块状导频；Signal部分用于指示数据部分的调制格式和长度；数据部分承载传输的数据，由于Signal部分和数据部分与本文无关，故也将Signal部分看作数据部分。总数据部分越短，信号的传播时延越低，越有利于信道的互易性。

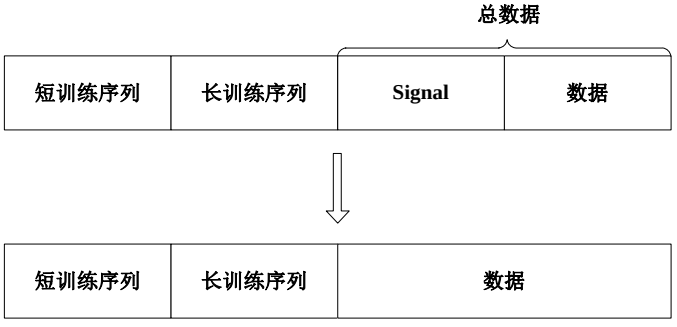


图4 数据帧结构

采用该帧结构具体有两点原因：1）该帧结构采用OFDM调制，非常适合基于导频的信道估计方法，从而提取出CSI；2）该帧结构中长训练序列采用块状导频，利于提取信道完整的CSI。

接收链路主要包含5部分：

1）USRP射频前端：负责将射频信号下变频到基带信号。

2）信号检测模块：该模块利用短训练序列进行接收信号检测以及帧同步。由于数据包是突发传输，因此需要进行信号检测同时进行帧同步。数据帧中短训练序列包含10个相同的连续短前导，因此可以采用类似双滑动窗口的信号检测方法。具体方法［19］如下：

首先设置两个窗C_n, P_n，如公式（8）（9）。其中r_n表示接收信号；N_{win}表示窗大小。在C_n窗中对接收信号与接收信号的延迟作互相关，在P_n窗中对接收信号及其共轭作互相关。

$$C_n = \sum_{k=0}^{N_{win}-1} r_{n+k} r_{n+k+16}^{*} \tag{8}$$

$$P_n = \sum_{k=0}^{N_{win}-1} r_{n+k} r_{n+k}^{*} = \sum_{k=0}^{N_{win}-1} |r_{n+k}|^2 \tag{9}$$

最后求C_n与P_n的比值m_n，如公式（10）。当且仅当m_n大于门限值时表示收到的信号为有用信号，且m_n发生阶跃处为信号起点，即帧的同步。

$$m_n = \frac{|C_n|^2}{(P_n)^2} \tag{10}$$

3）频偏估计补偿模块：该模块利用短训练序

列对接收信号粗频偏估计及补偿。利用［20］中频偏估计方法，具体如下：

理想情况下不存在频偏，短前导周期为16，因此 $r_n = r_{n+16}$。若有频偏，可由式（11）求得频偏 Δf。

$$\Delta f = \frac{1}{16}\arg\left(\sum_{n=0}^{15} r_n^* r_{n+16}\right) \tag{11}$$

随后由算出的频偏，对各子载波进行频偏补偿，如式（12）。

$$r_{n\_补偿} = r_n \cdot e^{-j\frac{2\pi n\Delta f}{16}} \tag{12}$$

4）信道估计模块：该模块利用长训练序列进行信道估计。由于采用块状导频，可估计出所有用于传输信息的子载波的CSI。

Alice和Bob在信道探测阶段互相发送信号，二者接收到的信号由式（1）、（2）表示

采用最小二乘估计（Least Squares， LS）对信道 $H_{AB,}\ H_{BA}$ 进行估计。接收端利用已知导频 X_A, X_B，可获得如下信道估计值 $\hat{H}_{AB}, \hat{H}_{BA}$，如式（13）。

$$\begin{aligned} \hat{H}_{BA} &= \frac{Y_A}{X_B} = H_{BA} + \frac{N_A}{X_B} \\ \hat{H}_{AB} &= \frac{Y_B}{X_A} = H_{AB} + \frac{N_B}{X_A} \end{aligned} \tag{13}$$

5）数据处理模块：用于完成后续的量化、信息协商、隐私放大等工作。

4 实验结果

为研究OFDM信号中的信道变化情况，在USRP X310和ADALM PLUTO SDR硬件平台上搭建了密钥生成系统，如图5。硬件各项指标如表1所示。软件环境采用GNU Radio，借助其开源库gr-ieee80211完成了第3节中的收发机设计。

表1 硬件指标

指标	USRP X310	ADALM PLUTO SDR
采样率	200 MS/s	61.44 MS/s
ADC精度	16 bit	12 bit
频率范围	10MHZ–6GHZ	325MHZ–3.8GHZ
发射功率	>10 dBm	7 dBm

图5 实验硬件平台

实验在室内场景中进行，将Alice放置于桌子上，Bob放置于椅子上，二者以100ms时间间隔互发导频信号，如图6。实验时，首先测试了无人行走（静态场景）时不同系统带宽对信道的影响。OFDM调制的FFT点数固定为64点，系统带宽分别为5MHZ、10MHZ、20MHZ，则子载波间隔频率 Δf 分别为78.125KHZ、156.25KHZ、312.5KHZ。

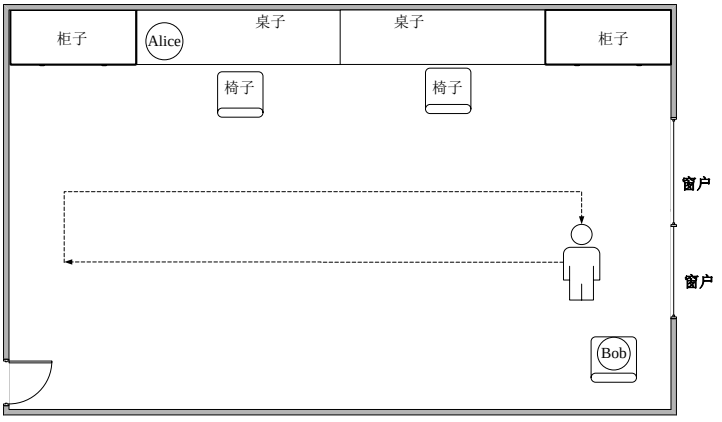


图6 实验环境

通过测量得出不同带宽下子载波变化数据，随后对不同频率间隔的子载波相关性进行分析。虽然该系统设计时FFT点数为64点，但抛弃保护频带及直流占用的12个点，实际使用中只有52个点用于发送数据。实验时选择第一个子载波的幅值作为基准，分别与之后51个子载波做式（14）的计算，得出结果如图7。

$$\rho_{XY} = \frac{E\{XY\} - E\{X\}E\{Y\}}{\sqrt{E\{X^2\} - E\{X\}^2}\sqrt{E\{Y^2\} - E\{Y\}^2}} \quad (14)$$

观察图7可发现，随着频率间隔的增大，子载波之间的相关性有明显的下降。因此在密钥生成过程中，应尽量增加子载波之间的间隔，以降低子载波之间的相关性，从而提高密钥的随机性。

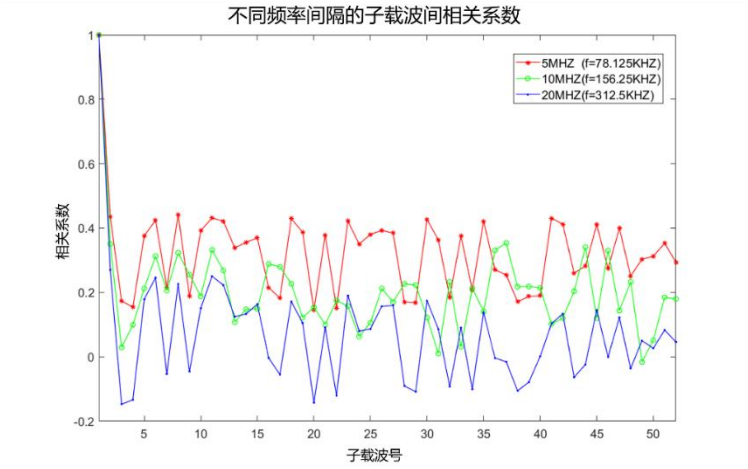


图7　不同频率间隔的子载波相关性

随后，又在20MHZ带宽系统中，分别测试了有人行走（非静态）和无人行走（静态）时的信道变化情况，如图8-9。其中x轴表示测量次数，即从x轴观察，可观察信道在时间上的变化；y轴表示子载波序号，即从y轴观察，可观察信道随频率变化而发生的频率选择性衰落。

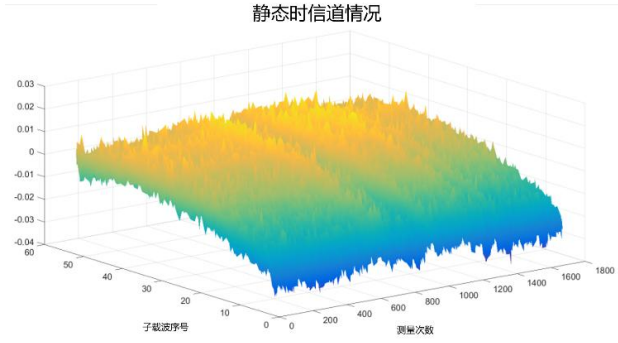


图8　静态时的信道情况

两次测量时，收发机均固定放置，环境中除了图6中标记的物体外，无其他物体，唯一区别是两侧收发机中间是否有人走过。观察结果可知，静态时测量结果在时间及频域上变化不大，幅值在-0.03~+0.03之间波动，其主要变化来自于环境

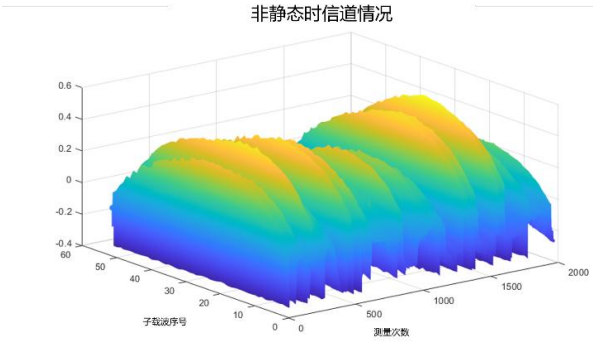


图9　非静态时的信道情况

中的噪声。但当收发机中间有人走过时，信道在时间上出现上下波动，并在某些子载波处发生频率选择性衰落，幅值在-0.4~+0.4之间波动，信道变化情况较静态场景提高了一个数量级，这非常有利于密钥生成。

最后，分别计算静态和非静态时的密钥生成速率。通过以上分析及观察非静态和静态时的信道变化情况可知，不同子载波在时间和频率上都有相关性，这对最终的密钥生成速率会有影响，因此分别实验了利用不同个数子载波进行密钥生成时对速率的影响，目的是找到最适合用于密钥

生成的子载波个数。为了防止频率间隔过近以及边带滤波带来的影响，用于密钥生成的不同个数子载波对应的子载波号如表2所示。

表2　选择的子载波号

子载波个数	子载波号	子载波个数	子载波号
1	(26)	6	(8,15,22,30,37,44)
2	(18,35)	7	(7,13,20,26,32,39,45)
3	(13,26,39)	8	(6,12,18,23,29,35,40,46)
4	(11,21,31,41)	9	(6,11,16,21,26,31,36,41,46)
5	(9,18,26,35,43)	10	(5,10,14,19,24,28,33,38,42,47,52)

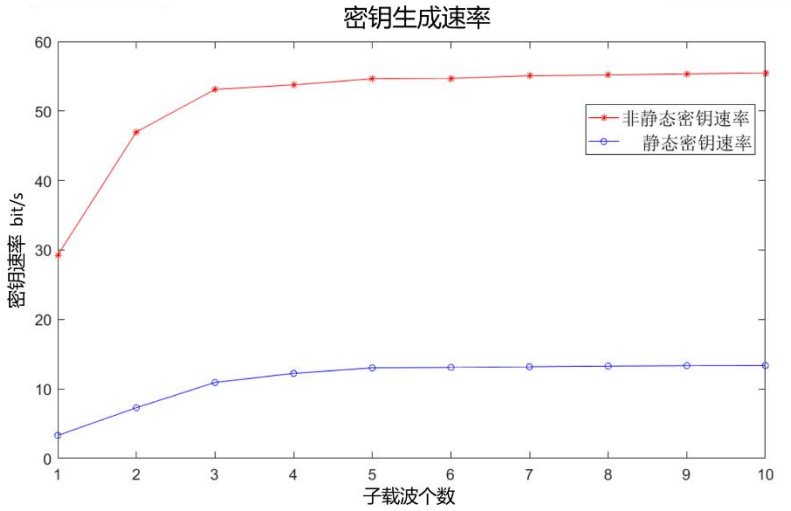


图10　不同子载波个数的密钥生成速率

结果如图10，通过分析，首先可以发现非静态时的密钥生成速率远大于静态时的密钥生成速率，非静态时的密钥生成速率约为静态时的5倍。这意味着非静态环境非常有利于密钥生成。其次，可以发现随着使用的子载波数增加，密钥速率并不是线性增长的，而是趋于某一值稳定的。观察图10可知，当用于密钥生成的子载波数大于4时，对密钥生成速率几乎无增长，这意味着我们在设计密钥系统时，仅仅需要4个子载波的数据，密钥速率为53.77 bit/s，即可达到该系统的上限，可以避免计算大量数据，从而节省硬件资源及时间。

5　总结

本文设计并通过结合GNURadio和SDR设备实现了基于IEEE 802.11帧结构的内生安全密钥提取实验。分别在室内静态与非静态场景中进行实验，证明了频率间隔越大，子载波间相关性越低，频率选择性衰落越明显。证明了非静态信道密钥生成速率高于静态密钥生成速率，前者约为后者的5倍。实验最终发现当进行密钥生成时，仅需利用其中4个子载波即可达到系统上限，密钥生成速率为53.77 bit/s。

本文研究了在OFDM信号中的不同频率间隔及不同信道场景对内生安全密钥生成技术的影响，通过实验对基于OFDM信号的密钥生成问题有了进一步认识。虽然当前密钥生成速率仍然较低，但随着超材料天线等新兴技术的应用，密钥生成速率必将成倍的增长。而提升密钥生成速率也将是本文今后的研究方向。

参考文献:

[1] WU J. Cyberspace Mimic Defense Generalized Robust Control and Endogenous Security[M]. Springer，Cham，2020.

[2] SHANNON C E. Communication Theory of Secrecy System[J]. Bell System Technical Journal，1949，28(4)：656-715.

[3] 张胜军. 基于互易信道和随机信号的物理层密钥生成[D]. 战略支援部队信息工程大学，2019.

[4] MAURER U M. Secret key agreement by public discussion from common information[J]. IEEE transactions on information theory，

1993，39(3)：733-742.
[5] 李晶琪．多天线信道物理层特征密钥提取方法[D]．东南大学，2018.
[6] 胡晓言．窃听信道相关条件下的物理层密钥生成技术研究[D]．战略支援部队信息工程大学，2018.
[7] JIN H, HUANG K, XIAO S, et al. A Two-Layer Secure Quantization Algorithm for Secret Key Generation With Correlated Eavesdropping Channel[J]. IEEE access，2019，7：26480-26487.
[8] JI Z, ZHANG Y, HE Z, et al. Vulnerabilities of Physical Layer Secret Key Generation Against Environment Reconstruction Based Attacks[J]. IEEE Wireless Communications Letters，2020，9(5)：693－697.
[9] XU W, JHA S, HU W. LoRa-Key: Secure Key Generation System for LoRa-Based Network[J]. IEEE internet of things journal，2019，6(4)：6404-6416.
[10] HAN B, PENG S, WU C, et al. LoRa-Based Physical Layer Key Generation for Secure V2V/V2I Communications[J]. Sensors (Basel，Switzerland)，2020，20(3)：682.
[11] LI K, NI W, EMAMI Y, et al. Design and Implementation of Secret Key Agreement for Platoon-based Vehicular Cyber-physical Systems[J]. ACM Transactions on Cyber-Physical Systems，2019，4(2)：1-20.
[12] AONO T, HIGUCHI K, OHIRA T, et al. Wireless Secret Key Generation Exploiting Reactance-domain Scalar Response of Multipath Fading Channels[J]. IEEE Transactions on Antennas and Propagation，2005，53(11)：3776-3784.
[13] MEHMOOD R, WALLACE J W, JENSEN M A. Key Establishment Employing Reconfigurable Antennas: Impact of Antenna Complexity[J]. IEEE Transactions on Wireless Communications，2014，13(11)：6300-6310.
[14] RUOTSALAINEN H, ZHANG J, GREBENIUK S. Experimental Investigation on Wireless Key Generation for Low-Power Wide-Area Networks[J]. IEEE Internet of Things Journal，2020，7(3)：1745-1755.
[15] Li J, Hu A, Li G. Analysis of non-reciprocity factors in extracting secret key from wireless channels for practical indoor scenarios[C]// 2016 2nd IEEE International Conference on Computer and Communications (ICCC). IEEE，2016.
[16] 袁瑞, 彭林宁, 李古月, 等．不同环境下无线信道密钥生成性能研究[J]．密码学报，2020，7(2)：261-273.
[17] 彭灏昌．基于软件无线电的OFDM研究平台的开发[D]．东南大学软件工程，2015.
[18] COMMITTEE L M S. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications[S]. IEEE Std. 802. 11-1999, 2003.
[19] LIU C H. On The Design of OFDM Signal Detection Algorithms for Hardware Implementation: IEEE Global Telecommunications Conference[C]，2003.
[20] SOUROUR E, EL-GHOROURY H, MCNEILL D. Frequency Offset Estimation and Correction in the IEEE 802. 11 a WLAN: VTC2004-Fall. 2004 IEEE 60th[C]，2004.
[21] BLOESSL B, SEGATA M, SOMMER C, et al. An IEEE 802. 11a/g/p OFDM Receiver for GNU Radio: Proceedings of the Second Workshop on Software Radio Implementation Forum[C]，2013.

[作者简介]

邓哲元（1996-），男，信息工程大学在读硕士研究生，主要研究方向：物理层安全。

金 梁（1969-），男，信息工程大学教授、博士生导师，主要研究方向为移动通信技术、阵列信号处理、物理层安全等。E-mail：liangjin@263. net

黄开枝（1973-），女，信息工程大学教授、博士生导师，主要研究方向为通信信号处理及无线通信安全。

陈亚军（1989-），男，信息工程大学助理研究员，主要研究方向为无线通信、物理层安全。

Study on Software-Defined Protocol Controller

LV Ping[1], LIU Qin Rang[1], Shen Jian Liang[1], Wang Xin[2], Chen Ting[1]

1.National Digital Switching System Engineering and Technological Research Center,Zhengzhou China;

2.Information Technology Innovation Center of TianJin Binhai New Area, China lp@ndsc.com.cn, lqr@ ndsc.com.cn,sjl@ ndsc.com.cn, wx@ tj. ndsc.com.cn, ct@ ndsc.com.cn

Key words: Protocol controller; Software-defined; Software-defined protocol controller; Computational kernel

Abstract. Software-defined chips are becoming a popular research topic for extending Moore's law. A protocol controller is an external communication interface of a data-intensive chip. Existing protocol controllers are implemented with solidification logic， which prevents the chip from defining the external communication interface with software； this has become a key technological bottleneck that the software-defined chip must break through. Based on the idea of mimic computing， in this paper we proposed a software-defined protocol controller technology， provided a mathematical model and a performance eval- uation model for the software-defined protocol controller， and designed a soft- ware-defined protocol controller with a unified hardware architecture and hybrid granularity. Under 40-nm processing conditions， we achieved a soft- ware-defined protocol controller that supports RapidIO， FC， and Ethernet pro- tocols. Compared to current solidification logic with equivalent function and performance， it consumes 36.32% less resources， 48.4% less area， and 42.41% less power. We also verified the versatility and effectiveness of the soft- ware-defined protocol controller.

1 Introduction

With the continuous development of Moore's Law and Dennard scaling， the performance power area （PPA） of integrated circuits （ICs） has become the most im- portant index in the IC industry. In the pursuit of higher performance， lower power

consumption， and smaller areas （cost）， the process node of the chip continues to ad- vance and has now entered the 7-nm level. However， the more advanced the process node is， the higher the costs of production and design. To maintain the validity of Moore's Law， software-defined hardware is becoming a popular research topic in academia and industry. The goal is that， by expanding the range of applications and the life cycles of the chips with software-defined attributes， the validity of Moore's Law can be maintained. In 2017， Defense Advanced Research Projects Agency（DARPA） of the United States announced the launch of the Electronics Resurgence Initiative ［1］ and deployed a software-defined hardware （SDH） project ［2］ oriented toward new system architectures. The goal was to create a new generation of high speed， pro- grammable， high-performance software-defined chips.

Serving as data storage and exchange centers， the data center is an important in- frastructure of the Internet of Things， cloud computing， and artificial intelligence； it is an important carrier for cloud computing and future business development and is a typi-

cal information system infrastructure. Early data centers were "chimney-like" vertical architectures incapable of sharing resources and having low utilization rates. Data exchange between computing and storage must be achieved through specific bridging techniques and protocol overbearing (Over) technologies, such as IPoIB [3] (Internet Protocol Over InfiniBand), FCoE [4] (Fiber Channel over Ethernet), and NVMe over Fabric [5]. As data centers evolved in the cloud era, the chimney-like vertical architecture of conventional data centers was broken down by software-defined networks (SDN) [6], software-defined memory (SDM) [7], software-defined data centers (SDC)[8], hyper-converged infrastructures [9, 10], and in-memory computing [11], [12]. However, virtualization has also prevented system efficiency and perfor- mance from fully meeting the application requirements in certain scenarios, and has greatly reduced the efficiency of computing, storage, and network resources utilization. This is a prominent problem of current data centers [13].

With the advent of the protocol-independent switch architectures (PISAs) and the emergence of the Network Data Plane Programmable Language P4 [14], the technol- ogy of programmable network switch chips has matured. In 2016, Barefoot released the Tofino programmable switch chip based on the PISA architecture. The block diagram of the programmable switch chip [15] shows that the Tofino is a programmable Ethernet switching chip above the network layer and that the underlying hardware structure (physical and link layers) is fixed and incapable of building a heterogeneous protocol converged system with integrated storage, computing, and transmission.

Software-defined interconnection (SDI) [16] is a software-defined hardware technol- ogy for the network domain, and the software-defined protocol controller (SDPC) technology is key to the interconnection and linking of heterogeneous protocols.

The protocol controller is a high-speed data-intensive external communication interface; it forms a different protocol environment around computing, storage, and network communications. The CPU, GPU, and other computing chips are usually integrated to form PCIe and Ethernet protocol controllers. Embedded processing and control chips, such as DSPs and MPUs, are usually integrated to form RapidIO and Ethernet protocol controllers. Memory controllers are usually integrated to form pro- tocol controllers, such as SATA, SCSI, SAS, and FC controllers. The NPU and network switch chips are usually integrated to form protocol controllers, such as Ethernet, IB, and RapidIO. As the technologies of computing, processing, memory, and communi- cation continue to fuse, the need for data communication between computing, pro- cessing, control, and storage systems has become increasingly acute. Objectively, there is a need for a software-defined, application-environment-based protocol controller that can not only greatly expand the realm of applications of computing, processing, storage, and switching chips, and avoid the various inflexible connections introduced by different protocols, but also greatly extend the lifetimes of computing, processing, storage, and switching chips.

In Section 2 of this paper, we propose a mathematical model and an effective evaluation method for a software-defined protocol controller, and we present a soft- ware-defined protocol controller based on a unified hardware architecture that is ca- pable of supporting mixed-grained reconfigurations. In Section 3, we implement the design of a 40-nm-level software-defined protocol controller that is capable of sup- porting RapidIO, Ethernet, and FC protocols. We then compare its logic resources, area, and power consumption with those of equivalent protocol controllers based on current technology. Finally, we provide a conclusion and future outlook in Section 4.

2 Software-Defined Protocol Controller

The software-defined protocol controller is an

important component for building a large end-to-end software-defined interconnected system. It is characterized by appli- cation drivers， which determine the characteristics of the protocol controller based on the network environment， extraction and reconfiguration through the analysis of typical

interconnection protocol characteristics in multiple network environments， particularly in the fundamental kernel set ［23］， ［24］ that contains coarse-grained reconfigurable units ［17-20］， fine-grained programmable units ［21］，［22］ and solidification logic. Protocol controllers capable of supporting different frequencies and binding modes， and protocols are obtained by configuration management based on the application needs and by reconfiguration and interconnection within and between fundamental kernel sets. Due to the differences in speed， code， and protocol function hierarchy， the hardware implementation of the protocol controller is also very different； this has become the most challenging task in software-defined interconnected technology. The software-defined protocol controller is based on the idea of mimic computing ［25］. On a unified hardware architecture， mixed-grained reconfigurable kernels ［26］ for differ- ent functions in the software-defined protocol controller are called through configura- tion management to construct transmission protocols for satisfying different interconnection needs and to complete the functions in the physical layer， link layer， and parts of the network layer of the Open System Interconnection （OSI） reference model ［27］.

2.1 Model Definition of SDPC

Definition 1 (transmission protocol):

Transmission protocols refer to all proto- cols that can run on the external communication interface of data-intensive chips for interconnection purposes. We designate a set of transmission protocols as

$$\mathrm{PRTL} = \{a_1, a_2 \ldots a_n\}$$

λ_i

l ai， A single transmission protoco is often described by the application scenario ， the transmission rate ， the port mode ， and the transmis- sion quality ； that is， a task is designated as .

v_iApplication function requirements can be divided into communication, storage, and calculation based on the field of the application and object. The transmission rate refers to the transmission rate of the protocol in the transmission medium. The port mode $b_i$is the port binding mode (1X, 2X, 4X, 8X, …) of the transport protocol, and the port flow control mode (Rx flow control and Tx flow control). The transmission quality q_i refers to the target requirements or constraints placed by the user on the protocol bus to complete the transmission task. Usually, q_i includes the protocol overhead, latency, and symbol error rate, designated as $q_i = (o_i, l_i, r_i)$.

v_i b_i
$q_i a_i = (\lambda_i, v_i, b_i, q_i)$

Definition 2 (variable kernel element):

A variable kernel element refers to the core module that is extracted from the characteristics of the transmission protocol to be solved and possesses a clear boundary， an independent function， a complete algo- rithm， a moderate granularity， and good commonality. Its functional algorithm is im- plemented through hardware and becomes a processing component to be stored in a hardware function library for calling and reuse. All the kernel elements are designated as $\mathrm{PE} = \{pe_1, pe_2, \ldots . pe_m\}$， where PE is the hardware function library.

Definition 3 (variable component):

The variable components dynamically con-

figure a protocol controller based on the application needs and the system state of the transmission protocol and hardware resources. They reconfigure the processing com- ponents， readjust the storage hi-

erarchy， recombine the components in the node， and rebuild the connection between networks. The set of all variable components is des- ignated as AR.

Definition 4 (SDPC model): The SDPC model consists of two parts, a variable object and decision function, and can be expressed as a sextu- ple: $SDPC = (PRTL, PE, IF, AR, RF, OF)$ where PRTL, PE, and AR are variable objects and IF, RF, and OF are decision functions. Specifically, $PRTL = \{a_1, a_2 \ldots a_n\}$ $a_i = (\lambda_i, v_i, b_i, c_i)$ $a_i = (a_i, l_i, r_i)$ $TQ = \{q_1, q_2, \ldots q_n\}$ $PE = \{P_1, P_2, P_3, P_4\}$ represents a collection of transport protocols with , , and .

represents a computational kernel library extracted and constructed from the transport protocol set PRTL， where $P_1 = (S_1, S_2, \ldots S_t)$ and $P_2 = (F_1, F_2, \ldots F_u)$and represents a set of computational kernel elements constructed and implemented by functional modules specific to different transport protocols， such as line order inversion of the RapidIO protocol， code-word insertion， channel bonding， forward error correction of the Ethernet protocol， and functional modules， such as the EEE Controller. They belong to the private module kernel set of each protocol and are configured as solidification logic and fine-grained programmable logic as necessary. P1 is the kernel set of solidification logic resources， and P2 is the fine-grained pro- grammable logic kernel set. $P_3 = (A_1, A_2, \ldots A_v)$ represents a set of computing kernel elements constructed from and implemented by functional modules shared by differ- ent transmission protocols， such as Rx/TX Polarity Control， Gearbox， PRBS Gen， and others. Solidification logic is used inside the chip and belongs to the common module kernel set. $P_4 = (R_1, R_2, \ldots R_w)$ represents a set of computing kernel elements con- structed from and implemented by the configurable modules of different transport protocols， such as encoding functions， scrambling functions， and others that belong to a coarse-grained reconfigurable kernel set.

$IF = \{IF_1, IF_2, IF_3\}$ represents the decision function extracted from the transmis- sion protocols and constructed from the computational kernel element. $IF_1: PR \rightarrow P_1/P_2$ represents the implementation function of the computing kernel element extracted from the core module of the transmission protocol and configured through specific function modules. $IF_2: PR \rightarrow P_3$represents the implementation func- tion of the computing kernel element extracted from the core module of the transmis- sion protocol and configured through the common function modules. $IF_3: PR \rightarrow P_4$， represents the implementation function of the computing kernel element extracted from the core module of the transmission protocol and constructed from the configu- rable function modules.

$AR = (P, M, C)$ represents a collection of all variable components. The software-defined protocol controller is implemented by the processing unit P, the memory hierarchy M, and the connection topology C. Hence, AR is characterized by (P, M, C), where $P = \{pe_1, pe_2, \ldots pe_m\}$represents a processing component set of the current system, and the function of each processing component pe_i is variable. $M = (m_1, m_2, \ldots m_n)$ represents an intelligent memory component set of the current system, and the capacity and hierarchy of each intelligent memory component mi is

variable. $C = (c_1, c_2, \ldots c_l)$ represents the topological architecture set interconnection relationship within the kernel of the protocol controllers and between the memories. The interconnection topology is configured based on the characteristics of the com- munication protocols.

The functional performance of different protocol controllers can be implemented by adjusting the relationship between P-M-C. This may include packet generation, control symbol processing, error management and recovery, link maintenance, generation and deletion of an Idle sequence, Cyclic Redundancy Check (CRC) calculations, and clock compensation. These functions may be implemented by calling different combinations of processing components in PE, while routing lookup and flow control may be accomplished by choosing a different memory component M based on the variation of protocol types. In specific selection of P-M-C plans, the configuration management element is determined according to the system status and target require- ments.

OF represents the target function that is achievable when the system automati- cally adopts a software-defined protocol controller for reconfiguration; that is, .

$OF: PRTL \times PE \times AR$ The above definitions show that the SDPC can implement the mapping of the transmission protocol to the implementation scheme, call the computing kernel ele- ments, and perform automatic dynamic reconfiguration of the SDPC using the func- tions IF_1, IF_2 and IF_3 based on the characteristics of the transmission protocol, the requirements of the transmission quality, and the state of the interconnection system. The final goal is to optimize the efficiency function OF.

2.2 SDPC Implementation Plan

In data-intensive chips, the protocol controller mainly performs high-speed serial transmission and reception, physical coding, protocol analysis, error recovery, event monitoring, event reporting, flow control processing, packet buffering, and routing lookup of related communication protocols in the system. Different protocol control- lers have the same implementation architecture. Due to the similarity of communica- tion protocols, different communication protocols have a certain uniqueness and in- dependence in their processing. Therefore, while maintaining the differences in com- munication processing characteristics, different communication protocols can extract the same or similar functional modules, thereby achieving a complete set of soft- ware-defined programmable and reconfigurable functional blocks. A model of the software defined protocol controller is shown in Figure 1.

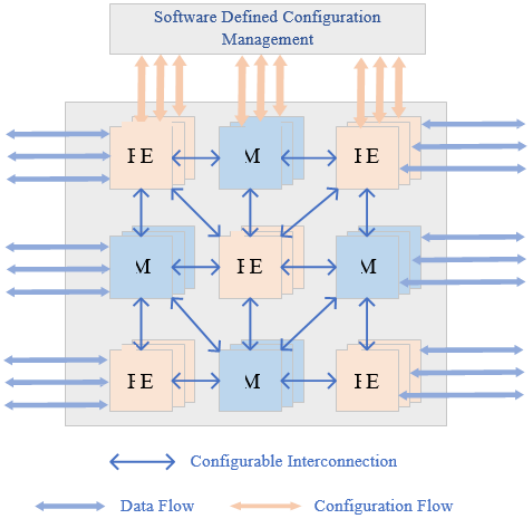


Fig. 1 Software Defined protocol controller hardware model

The building process of the software-defined protocol controller is as follows:

1. nThe characteristics of the types of communication protocols in the system are analyzed and the private functions, common functions, and configurable functions of each protocol and the storage unit capacity required by each protocol are determined;

2. Integrated consideration of the implementation difficulty, energy consumption ratio, and cost performance ratio is performed. The decision function IF = $\{IF_1, IF_2, IF_3\}$ is constructed;

there is a total of kernels. The number of internal programmable memory units of the SDPC is , and there are K configurable interconnection struc- ture

classes;

1. A computing kernel element library $PE = \{P_1, P_2, P_3, P_4\}$ is constructed from the decision function IF based on the characteristics and storage capacity of each protocol, as $L = t + u + v + w$ analyzed in step 1. Here, $P_1 = \{S_t\} \cap P_2 = \{F_u\} \cap P_3 = \{A_v\} \cap P_4 = \{R_w\}$ and J

1. The variable component in the SDPC corresponding to the implementation protocol a_n is $AR_n = (P_n, M_n, C_n)$, $P_n = \{pe_{ni}\} \cap M_n = \{m_{nj}\} \cap C_n = \{c_{nk}\}$; that is, the pro- tocol controller a_n adopts nk types of interconnection topologies to connect ni ( ) computing kernels and nj ($nj \le K$) memory units. Here, the ni kernels in- clude m_1 fine-grained computing kernels, u_1+v_1 solidified computing kernels, and w_1 reconfigurable computing kernels. In addition, the programmable and configurable computing kernels are reconfigured to implement a protocol controller that meets the transmission requirements. $ni \le L$

To summarize the software-defined protocol controller construction process de- scribed above, the SDPC is divided into four modules based on the protocol pro-cessing flow: a software-defined configuration management module; a software de- fined ingress engine (SDIE); a software defined egress engine (SDEE); and unified storage, unified routing, clock reset, and other modules. Its structure is shown in the block diagram of Figure 2:

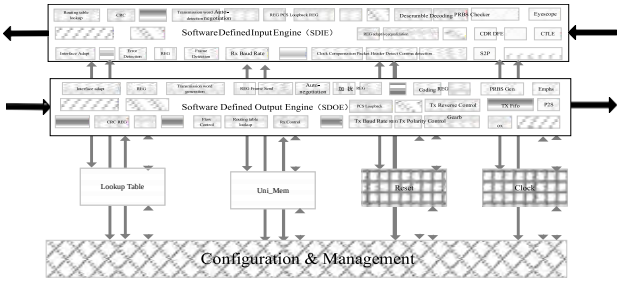


Fig. 2 Software - defined protocol controller structure block diagram.

Configurable RAM Immutable Private Kernel Programmable Unit Immutable Common Kernel Reconfigurable Kernel

1. The software-defined configuration management module belongs to the internal computing kernel of the SDPC; it is used as a channel for receiving and transmitting software-defined configuration flow;

2. The clock control and reset control module is a coarse-grained configurable com- mon computing kernel; it provides clock signals with different frequencies for the internal modules of the SDPC and performs different types of reset controls for each module based on the requirements;

3. The unified storage and routing module is a programmable storage unit; it is a common module for implementing each protocol controller and for allocating recon- figurable storage units as needed based on the characteristics of each protocol controller;

4. The software-definition ingress (egress) engines contain four types of kernels: a protocol private module programmable kernel, a protocol private module solidifica- tion kernel, a public solidification kernel (high-speed serial transceiver part), and a public configurable kernel. Here, the common solidification kernel is mainly the serial transceiver for decision feedback equalization (DFE), clock data recovery (CDR), continuous time linear equalization (CTLE), generation and detection of pseu- do-random sequences, serial-to-parallel conversion (S2P), and parallel-to-serial con- version (P2S). The public configurable computing kernel includes the codec module that supports the configuration of 8B/10B, 64B/66B, 64B/67B, the scrambling/descrambling module, frame generation,

frame transmission， CRC check， and interface adaptation. The private kernels will be implemented in the form of solidifi- cation logic and programmable logic within the SDPC based on the characteristics of the protocol in the system while taking full consideration of the performance， area， and power consumption. To achieve high data throughput， a multistage pipeline pro- cessing structure is used inside the SDPC， for which flow registers （REG） are inserted at the appropriate locations.

In the chip implementation process， the solidification logic resource kernel set P1

of each protocol private module and the common kernel set P3 shared by the protocols are implemented in the chip in the form of ASIC. The fine-grained programmable logic kernel set P2 of each protocol private module is implemented by an embedded programmable logic circuit. The kernel P4 of the configurable function module is in the form of a coarse-grained reconfigurable unit. Given the same hardware resources， the more communication protocols that are implemented in the SDPC， the more ap- plication scenarios there will be for the SDPC， and the higher the flexibility of the software-defined interconnection chip will be.

2.3 SDPC Performance Analysis

There is a general and special relationship between the SDPC and a single pro- tocol controller. A single protocol controller is a subset of the SDPC， and the SDPC is a collection of all the protocol controllers in the system. The SDPC is obtained by connecting a variety of different kernels and storage based on the application re- quirements using different interconnection methods. It is a programmable， reconfigu- rable hardware entity， which embodies the idea that the underlying hardware of the software-defined interconnect system can be flexibly configured.

In different application scenarios， software-defined protocol controllers can achieve different performances. The SDPC is an important part of software-defined interconnect chips. The performance of the SDPC directly affects the performance of software-defined interconnect chips and their application systems. Power consump- tion and area are important indicators for evaluating chip performance. In this paper， we analyze the performance of the SDPC in terms of power consumption and area.

2.3.1 Area analysis

There are n types of transmission networks in the whole system, so the software-defined protocol controller must support the transmission of the n types of communication protocols: $\mathrm{PRTL} = \{a_1, a_2 \ldots a_n\}$. Here, there are a total of L kernels, including $m+v$ fixed kernels, u programmable kernels, w reconfigurable kernels, and x storage units, with a total area of S. Because different kernels and storage units are called to implement different protocol controllers, the power consumption is also different. To implement a certain protocol controller AR_i in the SDPC, it is possible that fine-grained programmable private kernels of relevant protocols may be called. As a result, the area SA_i is greater than or equal to the protocol controller area

$SE_i(1 \le i \le n)$ [28] implemented in the ASIC mode but less than the total area of the SDPC； that is， $S > SA_i > SE_i$.

The average rate of area utilization in the interior of the SDPC is：

$$1 - \frac{\sum_{i=1}^{n} S - SA_i}{n \times S} \quad (1)$$

When the protocols in the interior of the SDPC have few or no private logic ker- nels， the area of protocol controllers implemented through software-defined modes will be close to the area of the SD-

PC， that is， $SA_i \approx S$， hence：

Average utilization rate in the interior of the SDPC is：

(2)

Meanwhile， since there are only a few or no private computing kernels inside the SDPC， we have：

(3)

When the SDPC satisfies the requirements of Eq.（2） and Eq.（3） simultaneously， the power consumption of the single ASIC-type protocol control is basically the same as that of the protocol implemented by the SDPC. At this time， the efficiency of the SDPC is the highest and the state is optimal for the SDPC performance. However， when the SDPC is used in an actual system， private kernels exist in the various pro- tocol controllers in the SDPC kernel set due to differences in the implementation of the communication protocols. As a result， the resource areas required to implement the protocol controllers are also different. For this reason， the internal area utilization rate of the SDPC is also different.

2.3.2 Power Analysis

Based on the above resource occupancy analysis of the software-defined proto- col controller， the SDPC contains some redundancy logic in addition to the logic oc- cupied by the protocol controller. The power consumption PAi of the protocol con- troller ARi implemented by the SDPC is as follows：

(4)

where PA_{si} represents the static power consumed by the redundant logic inside the SDPC， and PA_{di} indicates the power consumed by the implementation of the protocol controller logic within the SDPC.

Power consumption is related to the area and rate. Therefore， the power for im- plementing each protocol controller in the SDPC is also different. Using more redun- dant logic results in a greater PA_{si} and a higher fraction of the total SDPC power consumption and a poorer performance. There are mainly two ways to reduce the redundant logic power consumption. First， the computational kernel set is constructed from the design perspective of lowering the power consumption， enabling low-power design for redundant logic， and reducing redundant static power consumption. Second， the reduction of redundant controller logic is treated as an important index for evalu- ating the kernel set to improve the resource utilization rate of the SDPC. Although the power consumption for implementing the SDPC increases compared to that of a fixed protocol controller， the power consumption of the entire system in a complex applica- tion scenario will be less than the sum of the power consumed by multiple fixed pro- tocol controllers.

From the above analysis， having more protocols supported by the soft- ware-defined protocol controller results in a nimbler system. However， part of the system's performance will be sacrificed in meeting the system flexibility require- ments.

3 Test Comparison and Analysis

The functional performance of the SDPC was verified based on a soft- ware-defined interconnect chip. The chip was based on a 40-nm processing design， and the software-defined protocol controller supported three protocol types： Ethernet， FC， and RapidIO. Each port could be software-defined as one of the three protocols for connecting with processing endpoint devices and switching devices of different protocol types.

To verify the performance of the SDPC， we compared the resources， area， and power consumption of a single protocol controller under the same application envi- ronment. The specific evaluation conditions were as follows：

1. Single protocol controller implementation description： Four 8G FC protocol con- trollers， four 10G Ethernet protocol controllers， and four 1× RapidIO 3.0 protoco l controllers；

2. The software-defined protocol controller implementation description included four internal high-speed serial transceiver channels that could implement any combination of the three protocol control-

lers above；

3. The logic， area， and power consumption data were the statistical values after the completion of the back-end design. Since the power consumption of the SDPC was slightly different when implementing different protocols， the SDPC power consump- tion was the average value in various application scenarios.

Table 1 **Comparison** of logic resources

	Reg	Comb	Memory	Cell
FC	106992	273068	65	550779
Ethernet	216176	649984	90	977842
SRIO	125900	912847	75	1078819
Total: Res	449068	1835962	230	2607740
SDPC	275657	1211733	160	1660743
SDPC/Res	61.38%	65.99%	69.56%	63.68%

Number of cells is the total of register， composition logic，， and Memory

Table 2 shows that the implementation of the three different transmission proto- cols could be achieved in a software-defined manner by using the SDPC unified

hardware architecture. Although the resource utilization （logical and area） of each protocol did not reach 1， the resources occupied by the SDPC were respectively 56.5% and 51.6% of the total of the three-protocol-hardened hardware.

The power consumption estimate listed in Table 3 was evaluated based on the worst-case operation scenario. As shown in Table 3， the power consumed by redun- dancy logic in the SDPC was controlled to within 15% of the total power consumed by the SDPC. Compared to the power consumed by the whole chip， the power con- sumed by the redundant logic will be further reduced.

In summary， the software-defined protocol controller not only greatly improved the system flexibility， but also simplified the system composition. Although the extra power consumption and area of the chip increased， in terms of the overall effect， the reduction in performance introduced by the software-defined protocol controller was controllable and limited； thus， it created no significant effect on the system.

4 Summary and Outlook

In this paper， we proposed a software-defined protocol controller based on the software definition of the external communication interface in the software-defined

chip， and provided a design model and engineering implementation. We verified the feasibility， flexibility， and efficiency of the software-defined protocol controller. Along with the development of hyper-converged computing and integrated infor- mation processing， the demand for a software-defined interconnected environment will become increasingly acute. In this paper， we proposed that software-defined pro- tocol controllers may replace the existing solidified protocol controllers and expand the application scenarios of DSPs， CPUs， GPUs， NPUs， and network switching chips. Furthermore， self-defined private communication protocols with even higher efficien- cies may be used to implement a real software-defined information ecosystem.

Table 2 Comparison **of** area resources mm2

	Area		Resource utilization rate
	Fixed protocol	SDPC	
Ethernet	5.7611		67.84%
FC	4.0924	8.4923	48.19%
RapidIO	6.5939		77.65%
Average resource utilization rate		64.56%	
Total area of single protocol SE= $\sum_{i=1}^{3} SE_i$		SDPC area S	S/SE
16.4474		8.4923	51.6%

Table 3 Power consumption estimates watt

Protocol name	Power consumption Pi	PAdi	PAsi	PAsi/PAi
FC	0.8724	0.7807	0.0917	11.75%
Ethernet	1.308	1.1648	0.1432	12.29%
RapidIO	1.5784	1.4336	0.1448	10.10%
Total Power1	3.7588			
SDPC (Ps)	2.165			
Ps/Power1	57.59%			

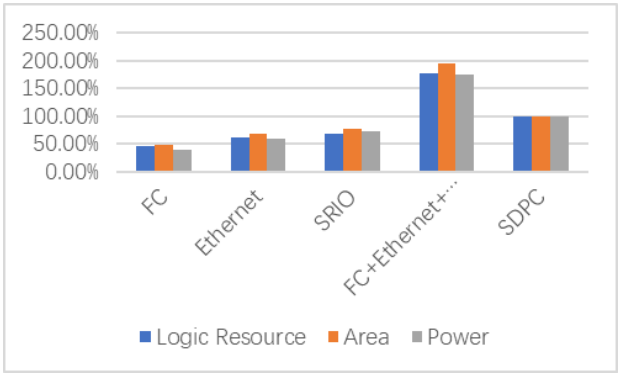


Fig.3 Overall comparison of logic resources, area, and power consumption.

References:

[1] DARPA Electronics Resurgence Initiative [online] Available: https://www. darpa. millwork-with-us/electronics-resurgence-initiative.

[2] Shen W. DARPA's Data Driven Discovery of Models (D3M) and Software Defined Hardware (SDH) Programs[C]//Proceedings of the 2018 on Great Lakes Symposium on VLSI. ACM, 2018: 1-1.

[3] Chu J, Kashyap V. Transmission of IP over InfiniBand (IPoIB)[J]. 2006.

[4] Wu Y Wang F, Hua Y, et al. I/O Stack Optimization for Efficient and Scalable Access in FCoE-based SAN Storage[J]. IEEE Transactions on Parallel and Distributed Systems, 2017:1-1.

[5] Huber N, et al. Evaluating and Modeling Virtualization Performance Overhead for Cloud Environments[C], Closer 2011 - Proceedings of the International Conference on Cloud Computing and Services Science, Noordwijkerhout, Netherlands, 7-9 May. DBLP, 2011: 563-573.

[6] AzodolmolkySiamak. Software Defined Networking with OpenFlow [M]. China Ma- chine Press, 2014.

[7] Ye Yurui. Software Defined Storage. Beijing: China Machine Press, 2016.

[8] Chen Xi, RickySun. Softwares Defined Data Center. Beijing: China

[9] Qian Chaoyang, Lu Mingsheng, Introduction to Hyper-Converged Infrastructure [J], Mathematical Technique and Application, 2016 (9): 216-217.

[10] Azeem S A, Sharma S K. Study of Converged Infrastructure & Hyper Converge Infrastructre As Future of Data Centre [J]. International Journal of Advanced Re- search in Computer Science, 2017, 8(5).

[11] Luo Le, Liu Yi, Qian Depei. Survey on In-Memory Computing

[12] Sebastian A, Tuma T, Papandreou N, et al. Temporal correlation detection using computational phase-change memory [J]. Nature Communications, 2017, 8(1): 1115.

[13] Chen Yanling, Wu An, Zhang Bin, alet, Distributed Network Infrastructure and Technology Oriented toward Cloud Server [J], Telecommunication Network Tech- nology, 2017(8): 8-11.

[14] Bosshart P, Dan D, Gibb G, et al. P4: programming protocol-independent packet pro- cessors [J]. Acm Sigcomm Computer Communication Review, 2014, 44(3):87-95.

[15] Gibb G. Reconfigurable Hardware for software-defined networks [D]. Stanford Uni- versity, 2013.

[16] Ping LV, QinRang LIU, JiangXing WU,, et al. New Generation Software Defined Archi- tecture [J]. SCIENTIA SINICA Informationis, 2018(3).

[17] Shen Jianliang, Li Sikun, Liu Leiet al. Hierarchical Configuration Memory Design for Coarse-Grained Reconfigurable SoC [J]. Journal of Computer Research and Develop- ment, 2017, 54(5):1121-1129.

[18] Tehre V, Kshirsagar R. Survey on coarse grained reconfigurable architectures [J]. International Journal of Computer Applications, 2012, 48(16): 1-7.

[19] AnsaloniG. et al., "EGRA: A Coarse Grained Reconfigurable Architectural Template", IEEE TVLSI, vol. 19, no. 6, pp. 1062-1074, 2011.

[20] Akbari O, Kamal M, Afzali-Kusha A, et al. PX-CGRA: Polymorphic approximate coarse-grained reconfigurable architecture [C]//2018 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2018: 413-418.

[21] Lin T J, Zhang W, Jha N K. Fine-grain dynamically reconfigurable FPGA architecture: U. S. Patent 9,735,783[P]. 2017-8-15.

[22] Yu H, Xu Q, Leong P H W. Fine-grained characterization of process variation in FPGAs [C]//2010 International Conference on Field-Programmable Technology. IEEE, 2010: 138-145.

[23] Shen Laixin, Wang Wei. Reconfigurable Architecture Based on Operator Grain Per- ception [J]. Computer Engineering, 2013, 39 (9):114-118.

[24] Chen Ang, Zhang Zhihong. Extraction and authentication of reconfigurable algorithm set for password domain [J]. Computer Applications and Software, 2014(11):292-294.

[25] Wu Jiangxing. Meaning and Vision of Mimic Computing and Mimic Security Defense [J]. Telecommunications Science, 2014, 30(7).

[26] Li Z, Huang Z, Chen S, et al. A modeling and mapping method for coarse/fine mixed-grained reconfigurable architecture[C]//2012 IEEE 11th International Con- ference on Solid-State and Integrated Circuit Technology. IEEE, 2012: 1-4.

[27] Day J D, Zimmermann H. The OSI reference model[J]. Proceedings of the IEEE, 1983, 71(12): 1334-1340.

[28] KuonI., RoseJ. "Measuring the gap between FPGAs and ASICs", IEEE Trans. Com- put. -Aided Design Integr. Circuits Syst., vol. 26, no. 2, pp. 203-215, Feb. 2007.

About the authors

LV Ping received her Ph. D. degree at National Digital Switching System Engineering and Technological Research Center（NDSC） in 2019. Currently. Her research interests is new gen- eration network information system architecture， and she is engaged in Large scale integrated circuit design.（Email： lp@ndsc. com. cn）

LIU Qin Rang received his Ph. D. degree at National Digital Switching System Engineering and Technological Research Center （NDSC） in 2004， Currently， he is a professor and doctoral. supervisor at NDSC. His research interests include New generation computer architecture and security of cyberspace.（Email： lqr@ndsc. com. cn）

Shen Jian Liang received his Ph. D. degree at National University of Defense Technology in 2013， Currently， he is a Vice professor and Master supervisor at NDSC. His research interests are reconfigurable computing and embedded SoC design method， and he is engaged in Large scale integrated circuit design.

Wang Xin received his master's degree at xidian university in china in 2013 ， Currently， he is a senior engineer at Information Technology Innovation Center of TianJin Binhai New Area. His research interests are reconfigurable computing and software defined network， and he is engaged in Large scale integrated circuit design.

Chen Ting received his Ph. D. degree at National University of Defense Technology in 2014， Currently， he is a an assistant researcher at NDSC. His research interests are digital signal pro- cessing， high performance microprocessor architecture， and high speed switching circuit. ， and he is engaged in Large scale integrated circuit design.

基于多层级日志分析的拟态Web表决与清洗方法

唐源，张铮，赵玉风，季新生
信息工程大学 郑州 450001

摘　要：表决与清洗是拟态防御技术中的重要机制，通过表决与清洗可以实现攻击检测与异常处理。已有的表决与清洗依赖于固定算法，缺乏对系统内部异构执行组件反馈信息的参考，给系统带来了误报和性能损耗。因此，在拟态Web服务系统上，提出一种基于多层级日志分析的拟态Web表决与清洗方法。首先通过比较不同执行体的运行结果发现异常请求，接着结合异常请求在拟态Web服务系统中不同执行体不同层级组件产生的日志进行威胁分类与分级，并将分析结果应用于表决与清洗。实验与实际业务数据分析结果表明，该方法能降低系统误报，并能根据执行体的运行情况进行清洗。同时，威胁分类与分级的结果有利于安全维护人员进行取证分析。
关键词：拟态Web服务系统、多层级日志分析、拟态Web表决、拟态Web清洗

Mimic Web Voting and Cleaning Method Based on Multi-Level Log Analysis

唐 源, 张 铮, 趙 玉风, 季 新生
Information Engineering University, Zhengzhou 450002, China

Abstract: Voting and cleaning, both of important mechanisms in mimic defense technology, could realize attack detection and abnormal handling through voting and cleaning. Unfortunately, the existing voting and cleaning methods rely on fixed algorithms and lack the reference to the feedback information of the heterogeneous execution components in the system, which brings false alarms and performance losses to the system. Therefore, in the mimic Web service system, a mimic Web voting and cleaning method based on multi-level log analysis is proposed. Firstly, abnormal requests are found by comparing the running results of different executor. Secondly, the threat classification and threat level of the abnormal request are obtained by analyzing the logs who generated by the different level components of the different executor of the abnormal request in the system, and apply it to voting and cleaning. Experimental and actual business data analysis results show that this method can reduce system false alarms and clean the executor according to its operating conditions. At the same time, the results of threat classification and threat grading are conducive to forensic analysis by security maintenance personnel.
Key words: mimic Web service system; multi-level log analysis; mimic Web voting; mimic Web cleaning

1　引言

根据国家计算机网络应急技术处理协调中心的最新数据表明，2019年CNVD收录的网络安全漏洞创历史新高，共计16193个，Web应用漏洞与应用程序漏洞占79.5%，Web安全与应用程序安全的问题仍不容忽视[1]。网络系统存在未知的设计缺陷使得漏洞层出不穷，现有的理论与工程水平无法彻底解决这种缺陷，使得网络安全态势总体呈现无法根除所有漏洞的不均衡局面[2]。为了应对未知的漏洞后门威胁，研究人员提出了拟态防御技术，期望在系统存在设计缺陷的情况下，通过系统的动态异构冗余机制，达到防御未知漏洞后门的目的[3]。Web服务作为重要的服务载体平

基金项目：基金项目：本课题得到国家自然科学基金项目（61521003），国家重点研发计划（Grant No.2018YF0804003, and Grant No. 2017YFB0803204）以及网络通信与安全紫金山实验室资助。
通讯作者：通信作者：张铮（ponyzhang@163.com）

台，保障其安全性十分必要，将拟态防御技术应用到Web服务中，通过构造拟态Web服务系统来防范已知与未知的Web攻击[4]。2016年拟态构造的Web服务器通过科技部组织的长达6个月的大型测试，验证了该系统的有效性。2018年-2020年连续三届“强网”拟态防御国际精英挑战赛，攻击者无法突破拟态Web服务系统，尤其在2020年6月的第三届比赛上，将真实的业务网站部署在拟态Web服务器上，在长达48小时的比赛中，多国“白帽黑客”精英战队对该网站进行攻击无一得手，充分证明了拟态Web服务系统的防御能力与实际业务实用性。

不同于基于规则检测的防御技术，拟态Web服务系统通过对多个执行体的运行结果进行表决检测威胁攻击，并通过调度算法对异常执行体进行调度清洗[4]。然而，由于系统只对执行体的执行结果进行差异性表决，无法识别因执行体自身问题带来的差异，如执行体输出结果表达语法不一致或性能瓶颈造成的差异也有可能导致不一致结果出现。同时，并非所有的异常告警都会对系统造成安全威胁。因此，解决非恶意请求在不同执行体上执行结果不同产生的误报，以及减少误报或低安全威胁引起的非必要调度清洗问题，是目前拟态Web服务器迈向工程化应用过程中必须要解决的问题。此外，由于拟态Web服务系统异构执行体数量多，执行体内部异构层级多，造成了一次异常请求的攻击取证工作量大、难度高。因此，需要将多层级日志分析与拟态Web表决与清洗技术相结合，实现降低拟态Web服务系统误报率、合理调度清洗执行体与异常请求取证分析的目的。

本文提出了一种多层级日志分析的拟态Web表决与清洗方法。首先将拟态Web服务系统分为多个执行层级，并给出了多层级日志分析模型，对用户请求进行分级与分类；其次将多个层级日志分析结果引入到拟态Web表决与拟态Web清洗机制中，对异常请求进行拦截，对异常执行体进行清洗；最后，通过实际业务数据分析与实验验证了该方法的有效性。

2 研究现状

2.1 表决机制

拟态防御技术的核心实现机制是动态异构冗余技术，主要包括请求分发机制、调度清洗机制、多执行体机制、表决输出机制[2]。用户的请求经请求分发机制等价复制发送至多执行体执行，表决输出机制比对多执行体的执行结果，通过表决机制反馈给用户响应[2]。调度清洗机制控制着当前用户请求由哪些执行体执行，哪些执行体需要被下线清洗恢复。表决机制，也称为多模裁决机制，通过对比多执行体执行结果的异同度，感知威胁。具体的异同度划分，依据不同的表决方法。

现有的表决方法主要分为两类，一类是一致性表决算法，即所有执行体的执行结果一致才将该执行体结果输出，否则认为是异常请求[5]，然而这种方法过于严格，忽略了执行体之间相同输出结果表达语法不同形成的文本差异会带来误报。为了消除文本差异，现有的文本相似方法包括基于字符串的相似度，基于语料库的相似度，基于知识的相似度[6-8]。这些方法都依赖于作者的先验知识，且文本相似比较的仅为执行体输出结果，忽略了执行体是由多个组件集成的复杂集合，每一个组件的输出都会影响执行体的输出特征。

另一类是多数表决方法，即将超过半数相同的执行体结果或相同数量最多的执行体结果输出[9-10]，这种输出容易将异常结果输出。为了改进这种方法，文献［11］通过执行体之间的异构程度与执行体的历史表决通过率，考虑对表决器的反馈作用。文献［12］将博弈论应用于拟态表决机制中，将表决器的输出正确率作为攻击者与防御者的收益函数，求出最优解。文献［13］针对执行体的防御效果、执行效率与恢复能力进行综合分析，并将其反馈至表决中。上述方法都是通过表决器的输出日志，对执行体进行评估分析，将分析后的结果用于拟态系统的表决与调度。上述方法仅关注表决器的执行结果而忽略了拟态系统各层级异构组件的执行情况，尤其是在执行体出现性能问题或是异构组件之间的输出信息存在语法差异时，会给拟态系统的表决带来误报。

2.2 清洗机制

拟态Web清洗机制将系统异常的执行体进行

下线清洗恢复操作，消除用户的攻击对执行体造成的影响。目前清洗的触发方法主要包括[2]：

1） 主动清洗，固定时间内进行清洗。这种方法无法及时消除恶意请求对执行体造成的影响；

2） 事件激活，固定的事件激活清洗。这种方法需要考虑好哪种事件激活清洗，同时需要综合考虑系统的运行状况；

3） 依据历史信息进行下线，减少易被攻击执行体的上线机率。这种方法容易使上线的执行体固定，不利于系统的高可用；

4） 依靠系统的安全系数，在系统处于容易被攻击的情况下，增加系统的清洗频率，减少执行体在线的时长。这种方法难以定义系统的威胁情况，并且频繁的清洗也不利于系统提供服务。

3 多层级日志分析模型

3.1 拟态Web分层架构

拟态Web服务系统是拟态防御技术在Web服务上的具体应用，其分层架构如图1所示。

图1 拟态Web服务系统分层架构图

拟态Web服务器主要在操作系统层、服务器软件层、软件运行环境层、数据存储层进行了异构，每一层级都有数种可被选择的异构元素，不同层级的异构元素通过可兼容组合构成一个执行体[14]。如某个执行体的结构可以为：

{Ubuntu16、Apache2.4、PHP_1、$Mysql_1$}

异构元素之间的异构方式主要包含两类：第一类是利用不同厂商的不同版本的软硬件产品构成，如Linux系统与Windows系统，Linux系统还包括CentOS、Ubuntu等；第二类是为了防御某特定或某一类型漏洞而人为制造的异构元素，如图1软件运行环境层中，可以将PHP函数与关键字用不同的加密方式来构成不同的异构元素PHP_1、PHP_2等。

拟态Web服务器通过表决器的3模表决机制识别恶意请求，即比较3个执行体的执行结果[4]，当执行结果之间出现差异则输出异常，并标记出现差异的执行体信息，下面通过案例1说明如何防御WebShell。

案例1 假设某Web网站被用户成功上传了PHP后门文件，且该用户掌握了PHP_1的密钥，然而上传的PHP文件无法在不包含PHP_1的执行体上运行，这样不同执行体的执行结果进行3模表决的时候，就会被表决器发现该异常；表决器发现该异常后将结果反馈给调度器，调度器便下线包含PHP_1的执行体并重新选择新的执行体，下线后的执行体会被调度器进行清洗恢复操作。

3.2 多层级日志分析模型

由图1可知，拟态Web服务系统是一个多层级运行结构。将表决分发器所在层级定义为代理层，并将其设置为第1层，操作系统层设置为第2层，服务器软件层设置为第3层，软件运行环境设置为第4层，数据存储层设置为第5层。用户请求在每一层的运行情况，都会影响执行体的运行结果，

进而影响多模表决结果。传统的拟态表决方法比较的信息单一，未考虑执行体内部各组件对执行结果的影响。为了提高拟态Web表决的准确率，准确分析异常请求的行为，需要对拟态Web服务系统多层级组件的日志进行综合分析，以便增加分析的广度，让表决结果更可靠[15]。图2为拟态Web多层级日志综合分析模型图。

图2　拟态Web多层级日志综合分析模型图

拟态Web多层级日志综合分析模型的符号介绍如下：

表1　多层级日志分析模型符号

符号	定义
log_i	单次请求在层级i的日志
e_i	单次请求在层级i的异常量
$\boldsymbol{f_i}$	单次请求在层级i的错误系数矩阵
Num	系统提供服务的执行体数量
m_i	单次请求在层级i异常执行体的数量
Brk	拦截系数
C_j	执行体j的清洗系数,$1 \leqslant j \leqslant N$
INF	当前系统可设置的最大整数

其中，$\forall i, 1 \leqslant i \leqslant 5$。

定义1

单次请求在层级i是否出现异常情况用e_i表示，e_i的取值为：

$$e_i = \begin{cases} 0, log_i\text{未出现异常}; \\ 1, log_i\text{出现异常}; \end{cases}$$

其中异常是指该请求在不同执行体的log_i出现错误信息、响应不同状态码或者操作出现不一致的情况。如在log_i中发现运行错误信息、文件操作行为不一致等。

表决器对执行体的输出结果进行3模裁决发现异常后，设表决器标记错误执行体的数量为M，并记录错误执行体IP。由于拟态Web表决是3模表决，$M = 2$时说明有2个执行体的执行结果与其它执行体不一致，而其它执行体的数量为1，若这2个执行体的执行结果相同则$M = 1$，若这2个执行体的执行结果不相同则$M = 3$，所以$M \neq 2$。

定义2

m_i表示在层级i出现异常执行体的数量，形式化描述为：

$$m_i = \begin{cases} M, i = 1, M \in N, 0 \leqslant M \leqslant 3\text{且}M \neq 2; \\ t,\ i > 1, t \in N\ \text{且}\ 0 \leqslant t \leqslant 3; \end{cases}$$

定义3

单次请求下执行体在层级i的错误系数为：

$$\{ f_{ij} \in N \mid 0 \leqslant f_{ij} \leqslant 4, j \in N, 1 \leqslant j \leqslant Num \mid,$$

f_{ij}的取值情况如下：

（1） 取值为0，即0级异常。表示执行体在层级i未处理用户请求；

（2） 取值为1，即0级异常。表示执行体在该层处理了用户请求，未发现异常；

（3） 取值为2，即1级异常。表示执行体线程池不够、CPU满负载等系统资源问题带来的请求超时等异常；

（4） 取值为3，即2级异常。表示用户请求在执行体的层级i上非法运行错误，如在数据存储层，某执行体数据库查询语句的关键字与函数进行了加密处理，“AND”加密为“EAND”，攻击者输入“XXX AND 1=1 -- ”的注入方式将会在该执行体的数据存储层报“语法错误”异常；

（5） 取值为4，即3级异常。代表用户输入在

不同执行体上数据的修改不一致、文件操作不一致等资源篡改型的异常。

则处理单次请求的所有执行体在层级i的错误系数组合矩阵$\boldsymbol{f}_i$为：

$$\boldsymbol{f}_i = \{ f_{i1}, f_{i2}, f_{i3} \}^T$$

其中f_{i1}, f_{i2}, f_{i3}在分析模型中为了分析方便无排列顺序之分，不指代特定执行体，例如$\{0,1,1\}^T$与$\{1,1,0\}^T$在分析模型中是等价的，而在实际运行过程中由执行体的IP标识。则$\boldsymbol{f}_i$的可能组合：

（1） 当$m_i = 0$，可能的组合为$\{1,1,1\}^T$、$\{0,0,0\}^T$，分别代表正常访问或未访问该层级；

（2） 当$m_i = 1$，可能的组合为$\{0,1,1\}^T$、$\{2,1,1\}^T$，$\{3,1,1\}^T$，$\{4,1,1\}^T$此时仅$\{0,1,1\}^T$、$\{2,1,1\}^T$为正常访问；

（3） 当$m_i > 1$，此时全为异常组合。其中，$\{0,0,1\}^T$、$\{2,2,1\}^T$由于有执行体正常访问该层，其它执行体响应了两个异常状态码，所以$m_i = 2$，并且此时系统已弱化为单执行体运行的结构，系统处于高风险状态；

则单次请求在所有层级的错误系数矩阵$\boldsymbol{F}$为：

$$\boldsymbol{F} = \{ \boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3, \boldsymbol{f}_4, \boldsymbol{f}_5 \}^T$$

则单次请求的可能访问系数矩阵为：

表2　访问系数矩阵

访问系数矩阵	说明	是否异常
$\{\boldsymbol{f}_1, \mathbf{O}, \mathbf{O}, \mathbf{O}, \mathbf{O}\}^T$	用户访问代理层或分发器分发信息异常	是
$\{\boldsymbol{f}_1 \boldsymbol{f}_2, \mathbf{O}, \mathbf{O}, \mathbf{O}\}^T$	用户访问在操作系统层被处理完成	是
$\{\boldsymbol{f}_1 \boldsymbol{f}_2 \boldsymbol{f}_3, \mathbf{O}, \mathbf{O}\}^T$	用户请求在服务器软件层被处理完成	否
$\{\boldsymbol{f}_1 \boldsymbol{f}_2, \mathbf{O} \boldsymbol{f}_4, \mathbf{O}\}^T$	用户绕过服务器软件访问执行体的操作系统	是
$\{\boldsymbol{f}_1 \boldsymbol{f}_2, \mathbf{O}, \mathbf{O} \boldsymbol{f}_5\}^T$	用户绕过服务器软件访问数据存储系统	是
$\{\boldsymbol{f}_1 \boldsymbol{f}_2 \boldsymbol{f}_3 \boldsymbol{f}_4, \mathbf{O}\}^T$	用户访问服务器业务代码	否
$\{\boldsymbol{f}_1 \boldsymbol{f}_2 \boldsymbol{f}_3 \boldsymbol{f}_4 \boldsymbol{f}_5\}^T$	用户访问服务器业务，业务代码调用数据存储层数据	否

其中$\mathbf{O}$为零矩阵，代表对应层级未处理用户请求。表2中是否异常的判断依据为用户请求是否符合Web服务系统处理逻辑，即用户无法绕开代理层、服务器软件层来访问执行体内部的其它层级，所以无f_3的访问集合都是异常访问。

定义4

是否拦截用户当前请求用拦截系数Brk表示，Brk的取值为：

$$Brk = \begin{cases} 0, \text{不拦截用户当前请求}; \\ 1, \text{拦截用户当前请求}; \end{cases}$$

定义5

执行体是否被下线清洗由执行体的清洗系数C决定，在系统启动时，设置：

$$\forall C_j, C_j = 0, 1 \leqslant j \leqslant Num$$

C的值是一个全局计算值，下文会给出计算方式。

在用威胁值进行攻击分析之前，有以下说明：

说明1 Web服务系统后一层级如果反馈错误信息，那么前一层级也会报错误信息，即不会出现前一层级标记错误的执行体与后面层级不匹配的情况。

说明2 假设当前处理用户请求的3个执行体为：$\{A_1, A_2, A_3\}$，这3个执行体的当前清洗系数分别为：$\{C_1, C_2, C_3\}$，设置出现异常的执行体标号为k $(1 \leqslant k \leqslant 3, k$在一次请求中可取多个值$)$。

在前述说明的情况下，则一次请求的多层级日志分析方法如下：

（1） 表决器比较3个执行体的执行结果，相同则不拦截本次请求，设置$Brk = 0$；不相同则进入下一步；

（2） 从第5层到第1层的顺序，判断3个执行体在层级i是否出现异常，无异常则继续比较下一层，直至比较完所有层级，比较方法如下：

1） 若$\exists m_i = 1$，且$\forall m_i, 0 \leqslant m_i < 1$。当错误系数组合矩阵$\boldsymbol{f}_i$为$\{0,1,1\}^T$、$\{2,1,1\}^T$，设置$Brk = 0$，该次威胁记录为层级$i$记录的相关异常，出现异常的执行体清洗系数$C_k = C_k + 1$；

2） 若$\exists m_i > 1$。

• 若仅仅出现1级异常。若访问集合为$\{\boldsymbol{f}_1, \mathbf{O}, \mathbf{O}, \mathbf{O}, \mathbf{O}\}^T$，说明代理层与执行体连接异

常，该次威胁记录为执行体分发异常“Send_error”，设置$Brk = 0, C_k = INF$。否则设置$Brk = 1$，出现异常的执行体$C_k = C_k + 1$，该次威胁记录为层级i记录的相关异常；

• 出现3级异常，则设置$Brk = 1, C_k = INF$。若3级异常出现在操作系统层或访问集合为$\{ \boldsymbol{f}_1, \boldsymbol{f}_2, \mathbf{O}, \mathbf{O}, \mathbf{O} \}^T$、$\{ \boldsymbol{f}_1, \boldsymbol{f}_2, \mathbf{O}, \boldsymbol{f}_4, \mathbf{O} \}^T$，则威胁记录为“操作系统层威胁攻击”。若3级异常出现在数据存储层或访问集合为$\{ \boldsymbol{f}_1, \boldsymbol{f}_2, \mathbf{O}, \mathbf{O}, \boldsymbol{f}_5 \}^T$，数据库组件中的异常记录为“sql_injection”，文件创建类的记录如果与软件运行环境组合出错，则记录为“webshell”，否则记录为“file_upload”，文件修改类的记录为“file_update”；

• 其它情况则设置$Brk = 1, C_k = C_k + 2$，出现2级异常的层级记录威胁为对应层级对应组件出现的威胁；若多个层级出现2级异常，按照操作系统层、数据存储层、软件运行环境层、服务器软件层的优先顺序进行威胁记录。

4 基于多层级日志分析的表决与清洗方法

拟态防御技术通过比较执行体之间的输出结果发现异常，此时用户的请求已在部分执行体内部产生影响，因此，需要建立一种机制及时消除异常请求对执行体的影响，防止出现异常的执行体影响系统功能。

消除异常执行体的异常状态，异常状态包括异常数据消除、内部系统宕机、执行体易掉线或易受攻击等影响执行体正常运行的状态。异常执行体的清洗流程为事件触发清洗、异常执行体下线、异常执行体重构与根据现场快照重启、执行体上线四个流程。执行体的重构为重新为执行体选择内部异构元素，通过可兼容且与当前线上执行体结构不同的组合，组合成一个新的执行体。拟态Web防御系统一般采取根据现场快照重启与数据同步的方法进行执行体异常状态的消除或清洗。

目前对于拟态Web防御系统来说，执行体处于何种状态或者触发了怎样的事件进行清洗，目前还处于研究阶段[2]。而且，由于目前的表决方法只关注执行体的输出结果，未关注执行体内部反馈信息对执行结果的影响，造成了目前的表决方法存在误报率过高的现象，不利于拟态Web防御技术的推广。因此，本章将展示如何将拟态Web服务系统多层级日志分析结果反馈到拟态Web表决与拟态Web清洗当中，并通过线上部署业务的数据分析验证该方法的有效性。

4.1 基于多层级日志分析的表决方法

在第3节中，一次请求经过多层级日志分析之后，会得到一个拦截系数Brk。在表决器对线上执行体的运行结果进行比对之后，如果发现比对结果不一致，则结合裁决日志与多层级日志分析结果对该次请求进行分析。即相当于在整体一致性表决的基础上，结合多层级日志分析进行表决，具体的一次请求的表决方法步骤如下：

（1）表决器对执行体的执行结果进行整体一致性表决，当表决一致时，响应用户正常执行结果；否则，触发多层级日志分析，进入步骤2）；

（2）对当前请求进行多层级日志分析。当$Brk = 0$，说明本次请求无安全威胁，给用户响应正常执行体的执行结果；当$Brk = 1$，说明本次请求存在安全威胁，拦截本次请求，给用户响应自定义的异常页面。

4.2 基于多层级日志分析的清洗方法

传统的拟态Web清洗触发是在拟态Web表决触发异常后，调度器直接对出现异常的执行体进行下线清洗。本文提出一种多层级日志分析的清洗方法，在异常请求经多层级日志分析后，更新执行体的清洗系数，根据执行体的清洗系数决定是否清洗执行体。

在第3节的多层级日志分析方法中，一次请求经过多层级分析之后，会得到当前运行执行体的清洗系数C，且从分析方法中可以发现，用户的请求仅在并发量过高或者对执行体造成3级异常时才会对执行体的运行造成影响。因此，设置执行体的清洗阈值为C_{MAX}，在上一章中介绍了INF为当前系统可设置的最大整数值，且仅仅在3级异常时，设置执行体的清洗系数值为INF。为了使出现3级异常或已无法提供正常服务的执行体能被快速清洗，设置$C_{MAX} \ll INF$。C_{MAX}的设置是为了让系统即能立即处理高危执行体，同时又能让系统在未出现3级异常时，且执行体数量充足的情况下，系统能处理多次出现异常情况的执行体。所以执行体的清洗系数值有以下几种情况：

（1） $C = 0$，说明当前执行体暂未被调度器调

度上线或未出现异常现象，执行体处于正常状态；

（2） $C < C_{MAX}$，说明当前执行体资源不足或者已被攻击但未对执行体造成影响，执行体存在异常情况。

（3） $C \geqslant C_{MAX}$，说明执行体内部信息被注入修改、代理层无法访问执行体或执行体多次出现异常情况，执行体处于高危状态，急需被清洗。

假设当前拟态Web系统平均1小时发生α次1级异常，β次2级异常，γ次3级异常，假设未发生3级异常的执行体运行到第t小时进行下线清洗，则这些执行体的清洗系数：

$$C = t \times \frac{\alpha + \beta}{C_N^3}$$

当前由于3级异常已被或要被清洗的执行体数量为$\frac{\gamma}{C_N^3} \times t$，所以当前拥有最大清洗阈值的执行体数量为$Num - \frac{\gamma}{C_N^3} \times t$，所以当前不被清洗的执行体数量为：

$$N_{nc} = \frac{\gamma}{C_N^3} \times t$$

为了保证系统能正常提供服务，则满足$N_{nc} \geqslant 3$，所以有：

$$t \geqslant \frac{3C_N^3}{r} + 2$$

所以此时的C为C_{MAX}。当$r = 0$或$\alpha + \beta = 0$时，为了使系统在某时间段能处理大量的异常请求与稳定提供服务考虑，建议C_{MAX}选取前一天的清洗阈值或某一段时间内清洗阈值的均值，同时建议在$r \neq 0$的情况下t选取最小值。

系统初始时，设置所有执行体的清洗系数值为0，C_{MAX}可根据整个防御系统上线前，系统的测试情况来设置。用户的某一次请求时，假设当前访问的执行体为执行体1、执行体2、执行体3，则该次请求的反馈清洗流程如下：

□ 用户请求经过多层级日志分析后，更新当前线上执行体的清洗系数C_1, C_2, C_3。

□ 多层级日志分析模块将C_1, C_2, C_3的更新情况反馈给调度器，若

$$\exists C_k \geqslant C_{MAX}, k \in \{1,2,3\}$$

调度器将对清洗系数值大于或等于C_{MAX}的执行体进行异常执行体插入算法，即将异常执行体插入处理队列；

□ 调度器对处理队列中的执行体，进行异常执行体清洗算法，并控制输入代理，不再分发用户请求至下线执行体；

□ 当异常执行体清洗完成，调度器发送信号至数据同步监控程序，后者将对清洗后的执行体进行数据同步；

□ 数据同步完成后，调度器发送信号至输入代理，允许输入代理分发用户请求至清洗后的执行体，执行体清洗完成。

其中，使用优先队列来实现对执行体的清洗，通过执行体的清洗系数C与达到当前清洗系数值最早的时间T来划分优先队列。设置优先队列*Priority_Queue*记录即将被清洗的执行体，将执行体的清洗系数C作为*Priority_Queue*的第一优先值，在清洗系数相等的情况下将达到当前清洗系数值最早的时间T作为*Priority_Queue*的第二优先值。则异常或高危执行体的插入*Priority_Queue*的算法伪代码如下：

算法1　异常执行体插入算法

Insert algorithm
Input:执行体A_i**Output**:插入结果
1.**if** A_i **in** Priority_Queue2.　**if** $A_i < INF$3.　Insert_ Priority_Queue (A_i)4.　**end if**5.**else**6.　Insert_ Priority_Queue (A_i)7.**end if**

当*Priority_Queue*中有值大于或等于清洗阈值C_{MAX}，拟态Web服务系统将对该执行体进行下线清洗，具体算法伪代码如下：

5　实验与分析

本节进行多层级日志分析实验，介绍多层级日志分析模型如何进行多层级日志分析。然后将基于多层级日志分析的表决方法（以下将称为本文提出方法）与一致性表决方法、多数表决方法的进行表决对比，通过实际应用数据分析比较本文提出方法的有效性，并根据应用场景数据得到执行体的清洗系数图，验证本文提出的清洗方法的有效性。最后，将一致性表决方法与本文提出方法进行性能对比。

算法2　异常执行体清洗算法

Cleaning algorithm

Input:Priority_Queue**Output**:清洗结果

1.**while**(Priority_Queue.**size**())　//当队列不为空2.　node=Priority_Queue.**top**()　//取出队头元素3.　**if** $C_{node} \geqslant C_{MAX}$

4.　shutdown(node)　//下线执行体node5.　clean(node)　//清洗执行体node6.　C_{node}=0　//重置执行体node的清洗系数

7.　Priority_Queue.**pop**()　//删除队头元素

8　**end if**9.**end while**

5.1　多层级日志分析实验

本小节通过案例来展示拟态Web多层级日志分析，分别选取了软件运行环境层WebShell、数据存储层SQL注入、组件运行出错三个不同层面或者不同类型的典型案例。

实验5.1 软件运行环境层WebShell

据W3Techs在2020年10月8日公布的网站服务器端编程语言使用情况统计，PHP使用率占78.9%，所有本案例采用PHP文件做WebShell实验演示。假设攻击者已成功上传后门脚本“webshell.php”文件至网站文件目录，脚本内容为：

＜？ php $a= $_GET ［'shell'］； echoPHP13 shell_exec PHP13 （$a）;？＞

其中“PHP13”为IP为“192.168.102.13”执行体（以下将以执行体IP的第四网段指代执行体，如本执行体标号为“13”）PHP程序的加密方法。单独访问“13”执行体的结果为：

图3　“13”执行体WebShell注入实验结果图

可以发现单个执行体可被攻击者植入的后门程序利用，输入指令“pwd”，返回了注入文件所在目录。

接下来，访问拟态Web服务系统前端，访问的裁决日志为：

图4　WebShell注入实验裁决日志截图

裁决日志中，“13”执行体的运行结果与“11”、“12”执行体运行结果不一样，而3个执行体如图所示的顺序为“11”、“12”、“13”，分别为1号、2号、3号执行体，“13”执行体位置为3，所以在时间之后跟的标号为3，表示排第3的执行体运行结果与其它执行体结果不一样，所以$m_1 = 1$。所以攻击利用后门程序被表决器发现异常，触发多层级日志分析，记录3个执行体执行信息，具体信息如下，从上至下分别为1号、2号、3号执行体；

可以发现1号执行体与2号执行体因为案例1所述防御方法，产生了PHP程序解析错误，错误

其中“error_type”代表多层级日志分析的异

状态码“Run_Code”取值为3，即该层级1号执行体与2号执行体的错误系数：

$$f_{41} = f_{42} = 3$$

即该层级出现2级异常。3号执行体正常运行，所以无信息记录，但由于有访问记录，所以

$$f_{43} = 1$$

因此共有两个执行体出现异常信息，即$m_4 = 2$。又因为在资源中该资源被标记为用户上传资源，即用户上传资源出现了PHP解析异常，所以该资源为后门程序，所以设置：

$$f_{51} = f_{52} = 3$$

所以多层级日志分析结果如下：

常类型，本案例为“WebShell”。可以发现，本次

图5（c） WebShell注入实验3号执行体日志截图

实验识别攻击的关键点在于：裁决日志发现疑似异常请求、内部运行组件的关键日志对比；

实验5.2 数据存储层SQL注入

SQL注入是数据存储面临的一个经典威胁，本实验演示SQL注入的多层级日志分析。访问实验网页的查询点，访问信息如下：

图6 实验4.2常规执行体注入实验结果图

如图6所示，向“http：//192.168.85.101/db_demo/doFindStudentById. php”页面POST了“id=1 or 1=1 -- ”，该店返回了注入后的结果，所以该点存在SQL注入点；接下来，访问拟态Web服务系统该页面，访问“http：//192.168.85.21/db_demo_label/doFindStudentById.php”，则裁决日志为：

图7 实验4.2裁决日志截图

与实验4.1同理，图7说明2号执行体（192.168.102.12）与其它执行体的执行结果不相同，1号执行体（192.168.102.11）与3号执行体（192.168.102.13）结果相同。查看多层级日志记录：

通过图8中的（a）、（b）、（c），可以发现1、2、3号执行体由于数据库语句加密策略，识别了非法字符，所以状态码标记为“3”，即在数据存储层中1号、2号、3号执行体的错误系数为：

$$f_{51} = f_{52} = f_{53} = 3$$

图8（a）1号执行体多层级日志记录

图8（b）2号执行体多层级日志记录

图8（c）3号执行体多层级日志记录

图8（d）多层级日志分析结果

然而PHP运行的错误信息却不一样，1号与3号由于访问数据库，没有信息返回，所以报空地址异常；2号执行体由于数据库防护策略拦截了语句，造成了超时信息，程序无法读取超时结果，所以日志中即报超时错误，又报读取结果错误；但3个执行体出现错误的原因却是一样的，且都出现了错误，所以$m_4 = m_5 = 3$。这也是为什么在多层级日志分析过程中，如果多层出现错误信息，优先考虑最终访问资源异常的原因。由于数据存储层数据库语句出现问题，所以异常结果“error_type”为数据库注入“sql_injection”。

通过本实验除了说明多层级分析如何发现数据存储层SQL注入外，也可以说明，裁决结果中即使某两个执行体的执行结果一样，但并不说明这两个执行结果就是正常的，且在本例中，这三个执行体的执行结果都是异常的，如果仅仅只根据裁决日志得出结论，显然是不合理的。

实验5.3 组件运行出错

组件运行出错造成执行体的输出结果异常，是拟态Web服务系统面临的主要误报之一，本实验将说明多层级日志分析如何发现这种误报。

使用Apache Jmeter对拟态Web服务系统进行

压力测试，模拟高并发情况下网页的访问请求，选取其中一条信息的多层级日志分析结果进行展示。其中，裁决日志截图为：

图9　实验5.3裁决日志截图

与实验5.1同理，裁决日志说明3号执行体（192.168.85.103）与1号（192.168.85.101）、2号执行体（192.168.85.102）输出结果不一致。多层级日志记录为：

图10（a）1号执行体多层级日志记录

图10（b）2号执行体多层级日志记录

图10（c）3号执行体多层级日志记录

图10（d）代理层日志记录

图10（a）、（b）中，由于用户请求正常运行，所以无错误信息记录，请求的状态码为“1”，即用户请求了该层；图（c）中，由于网页运行PHP文件需要请求PHP解析，PHP-FPM（PHP进程管理器）开放了“9000”端口，网页请求PHP-FPM超时，所以报了图（c）所示的超时异常与原因；所以在服务器软件层中，各执行体的错误系数分别为$f_{21}=f_{22}=1$，$f_{23}=2$，所以$m_2=1$。由于3号执行体运行超时，所以代理日志也记录了该执行体的超时异常，记录情况如图（d）所示。所以多层级日志分析结果记录如下：

多层级日志记录结果为运行超时（run_time-out）。

5.2　表决对比

表决对比实验采用紫金山实验室门户网站的各层级日志数据进行分析，其中，紫金山实验室门户网站已进行拟态化改造。日志数据包含一致性表决日志与后端各执行体组件的运行日志。在一致性表决日志的基础上，分别用多数表决的方法与本文提出方法对日志进行了综合分析。以2020年6月19日-2020年6月25日共7天的数据为分析样本，一致性表决方法、基于一致性表决的多数表决方法、本文提出方法的异常请求分析结

果图11所示。

图11　三种方法识别的异常请求数对比图

由于多数表决方法与本文提出方法都是基于一致性表决方法的，设计的目的是为了降低一致性表决方法带来的误报，所以这两种方法检测的异常请求数会低于一致性表决方法。从上表中发现多数表决方法降低了系统的误报，然而从多数表决方法认定为正常的数据中发现了6542次异常数据，也称逃逸数据，部分逃逸数据如下：

表3　多数表决方法部分逃逸数据

攻击者请求信息	响应状态码
/uploads/allimg/200507/000H46396-0.png/test%E3%80%81.php	200,200,403
/uploads/allimg/200507/000H46396-0.png/b.php%EF%BC%9Fusername=admin	200,200,404
/templets/cn/scripts/html5media/1.1.8/html5media.min.js/?--%3E%3C/script%3E%3Cscript%3Ealert(2412)%3C/script%3E	200,200,404
/templets/cn/scripts/jquery-1.9.1.min.js/"><script>alert(2508)</NonExistentFile.js/"><script>alert(2508)</script>	200,200,404
/templets/cn/scripts/jquery.SuperSlide.2.1.1.js/%22%3e%3cscript%3ealert%282535%29%3c%2fscript%3e	200,200,404

从表3中数据可以发现，攻击者向拟态Web系统注入PHP脚本（WebShell）或进行跨站脚本攻击（Cross Site Scripting，XSS），由于其中两台响应了攻击者预期结果，而剩余一台未响应攻击者预期结果，但由于多数表决会认为多数一致的结果是正常结果，所以系统会向攻击者响应攻击者预期结果，造成攻击逃逸。

本文方法对一致性表决认定为威胁的数据的分析结果如下：

图12本文提出方法威胁数据分析结果

（1）“system”为“操作系统层攻击威胁”，共计3042条，其中主要包括文件上传攻击，如请求信息为“/FCKeditor/editor/filemanager/browser/default/browser. html？ Type=monyerConnector=connectors/asp/connector.php”；

（2）“server”为“服务器软件层攻击威胁”，共计189037条，其中主要包括：

□ 目录遍历尝试，如“/dede”、“/pmadmin”、“/ManageAdmin”、“/absadmin”；

□ 文件遍历尝试，如“/Web.config”、“/access.log”、“/admin.db”；

□ 漏洞扫描、爬虫，如“/ADClick.aspx？Sit-

eID=206&ADID=1&URL=http：//jy1. zhubobagua.com/20200618ksfukxv14156379*85197398*71691.html”、“/_vti_bin/..%255c/..%255c/..%255c/winnt/system32/cmd.exe？/c+dir”；

（3）“ok”为结合执行体各组件日志与一致性表决日志发现的正常请求，共计20063条，未出现逃逸数据，

（4）“environment”为“软件运行环境层攻击威胁”，共计96154条，其中主要包括：

 HTTP消息头注入：“/plus/diy.php%2fFoo-bar% 0d% 0aAppScanHeader：% 20AppScanValue%2f1%2e2%2d3%0d% 0aSecondAppScanHeader：%20whatever”；

 命令注入：“/；print（md5（acunetix_wvs_security_test）） ”、“/plus/% 24 （nslookup%20WUdtCLZ7）/default/images”；

 WebShell：“/mouangliang. jpg/. php”、“/shell.php”；

 XSS：“/servlet/%0ARefresh： 0； URL=javascript：prompt（1）%0A1”。

（5）“data”为“数据存储层攻击威胁”，共计908条，其中主要包括“SQL注入”，如“/templets/（select （0） from （select （sleep （9） ） ） v）/*'% 2b （select （0） from （select （sleep（9） ） ） v）%2b'%22%2b（select（0）from（select（sleep（9）））v）%2b%22*/”。

通过3种表决方法的分析对比，我们可以得出以下结论：

 相较于一致性表决方法，本文提出方法能降低一致性表决方法带来的误报，一致性表决方法总计误报率为6.63%，其中包含了影响门户网站使用的误报；本文方法误报率为0.14%，能降低97.92%的误报。本文提出方法误报的来源为6月23日与6月25日用户请求数分别到达160多万与270多万，此时出现部分动态网页中某模块加载过慢带来的告警，不影响系统的实际使用；

 相较于多数表决方法，本文提出方法不会造成攻击逃逸。多数表决方法带来了6542次攻击逃逸；

 相较于一致性表决方法、多数表决方法，本文提出方法能准确分析出攻击威胁出现的层级，便于安全维护人员及时定位出现威胁的执行体及组件。

5.3 清洗分析

通过线上业务的日志来模拟7天内某执行体的清洗系数变化，展示本文提出清洗方法的效果。随机选取一个执行体，记为执行体A，图（a）-（c）为执行体A在6月19日、6月23日、6月25日，清洗系数波动较大的几天清洗系数变化图，纵轴为清洗系数值，横轴为时间。为了不暴露实际业务的执行体数量，故图中隐去了清洗系数值。其中，α、β、γ、t由系统前一天的数据计算出。

如图（a），6月19日执行体A的清洗系数前半段上升，后半段处于水平状态，说明在前一段时间系统遭受了大量的异常请求或者误报。通过日志信息发现波动的时间段为8：31-9：00，波动原因为漏洞扫描、爬虫与误报，无4级威胁；

如图（b），6月23日出现大幅度波动，这一天系统收到大量异常请求，多次出现4级威胁或达到清洗阈值C_{MAX}，执行体被清洗次数较多，15：15：50后执行体恢复平稳；

如图（c），6月25日前14个小时无变化， 15点有大量请求，出现了一些误报，也有一部分异常请求，16点后处于稳定状态。

通过图（a）-（c）的清洗系数变化图可发现，针对某执行体的3级威胁在实际业务中是少数的，结合5.2的分析，大量的恶意请求是扫描、爬取与

试探，即如果不对恶意请求进行分析，仅仅以系统受到攻击来清洗执行体是不合理的。通过图（a）-（c），本文提出的清洗方法能实时反应执行体当前受到的威胁情况，并且能在执行体清洗系数过高或受到3级威胁时及时清洗执行体。

5.4 性能对比

性能对比实验环境配置CPU型号为Intel（R） Core （TM） i5-7200U CPU @ 2.50GHz，CPU核心数为4，内存为16GB，测试工具采用Apache JMeter。由于多数表决方法在前文实验中漏报率过高，故性能对比测试仅将本文提出方法与一致性表决方法进行比对，测试在不同并发请求下两种方法的系统吞吐量（Throughput）、平均响应时间（Average）、系统误报率（Errors）。吞吐量与平均响应时间测试情况如图7。

图7中一致性表决方法1500的并发数到2500的并发数呈现上升的趋势，主要原因是2500并发的丢包数是1500并发丢包数的10倍，系统在一定条件下处理的请求少了，所以吞吐量上升了，但平均响应时间增加了，属于正常情况，本文提出方法2500并发到5000并发也是这种情况。

通过图14可以发现一致性表决方法的系统吞吐量高于本文提出方法，且平均响应时间也要低于本文提出方法，即一致性表决方法的系统性能高于本文提出方法。本文提出方法造成的性能损耗来自于各执行体组件上的日志推送模块、日志收集模块与表决器请求日志收集模块。

在压测清洗下两种方法系统误报数对比表如表4所示。

表4 系统误报数对比表

测试方法	1000并发数	1500并发数	2500并发数	5000并发数
正常系统压测	0	0	0	0
一致性表决方法	0	24	43	85
本文提出方法	0	0	0	1

通过表4可以发现，虽然本文提出方法带来了性能损耗，但是本文提出方法能降低系统误报，表中本文提出方法的误报原因为分发、表决器所在的基座，访问量过大带来的误报，在实际应用中可以通过负载均衡来缓解分发、表决器的压力；而一致性表决方法的由于不同执行体响应的超时信息不一样，无法通过负载均衡提升性能的方法来解决。

6 总结

表决与清洗是拟态Web服务系统的重要机制。本文针对已有表决方法存在误报、难以指导执行体清洗的情况，引入Web服务分层的概念，提出了一种基于多层级日志分析的表决与清洗方法。该方法首先比对用户请求在不同执行体的执行结果，在出现不相同结果的情况下，分析用户请求在系统不同层级组件的产生的错误日志、操作日志，通过分析结果对用户请求进行威胁分类与分级，并根据分类与分级的结果进行表决与指导执行体清洗。实验结果表明，该方法能降低误报率，并能指导执行体进行清洗。威胁分类与分级的结果也有利于安全维护人员进行取证分析。

由于本文方法无法准确给出异常请求的攻击分类，下一步将结合用户请求特征、用户请求软件执行栈、系统相关组件日志信息等威胁分析技术做进一步的分析，并根据分析结果优化表决方法，降低系统误报。

参考文献:

[[1]] 国家计算机网络应急技术处理协调中心. 2019年我国互联网网络安全态势综述[R]. 北京:2020.

National Internet Emergency Center. Review of China's Internet Network security situation in 2019[R]. Beijing:2020

[[2]] Wu J X. Cyberspace mimic defense[M]. Springer International Publishing, 2020.

[[3]] 邬江兴. 网络空间拟态防御研究[J]. 信息安全学报,2016,1(04):1-10.

WU Jiangxing. Research on Cyber Mimic Defense[J]. Journal of Cyber Security, 2016, 1(04):1-10.

[[4]] 仝青,张铮,张为华,邬江兴. 拟态防御 Web 服务器设计与实现[J]. 软件学报,2017,28(4):883-897.

TONG Q, ZHANG Z, ZHANG WH, et al. Design and implementation of mimic defense Web server[J]. Ruan Jian Xue Bao/Journal of Software, 2017, 28(4):883-897 (in Chinese)

[[5]] Parhami B. Voting algorithms[J]. IEEE transactions on reliability, 1994, 43(4): 617-629.

[[6]] Gomaa W H, Fahmy A A. A survey of text similarity approaches[J]. International Journal of Computer Applications, 2013, 68(13): 13-18.

[[7]] Islam A, Inkpen D. Semantic text similarity using corpus-based word similarity and string similarity[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2008, 2(2): 1-25.

[[8]] Mihalcea R, Corley C, Strapparava C. Corpus-based and knowledge-based measures of text semantic similarity[C]//Aaai. 2006, 6(2006): 775-780.

[[9]] Alotaibi B, Elleithy K. A majority voting technique for wireless intrusion detection systems[C]//2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT). IEEE, 2016: 1-6.

[[10]] Barbara D, Garcia-Molina H. The reliability of voting mechanisms[J]. IEEE Transactions on Computers, 1987(10): 1197-1208.

[[11]] 武兆琪,张帆,郭威,卫今,谢光伟. 一种基于执行体异构度的拟态裁决优化方法[J]. 计算机工程,2020,46(05):12-18.

WU Zhaoqi, ZHANG Fan, GUO Wei, WEI Jin, XIE Guangwei. A Mimic Arbitration Optimization Method Based on Heterogeneous Degree of Executors[J]. Computer Engineering, 2020,46(05):12-18.

[[12]] 王禛鹏. 拟态网络操作系统调度与裁决机制研究及实现[D]. 战略支援部队信息工程大学,2017.

Zhenpeng Wang. Research on the Scheduling and Decision-making Mechanism of Mimic Network Operating System[D]. Information Engineering University, 2017.

[[13]] 李卫超,张铮,王立群,邬江兴. 基于拟态防御架构的多余度裁决建模与风险分析[J]. 信息安全学报,2018,3(05):64-74.

LI Weichao, ZHANG Zheng, WANG Liqun, WU Jiangxing. The Modeling and Risk Assessment on Redundancy Adjudication of Mimic Defense. Journal of Cyber Security, 2018,3(05):64-74.

[[14]] 张杰鑫,庞建民,张铮. 拟态构造的Web服务器异构性量化方法[J]. 软件学报,2020,31(02):564-577.

ZHANG Jie-Xin, PANG Jian-Min, ZHANG Zheng. The Research of the Quantification Method for Heterogeneity on Web Server with Mimic Construction[J]. Ruan Jian Xue Bao/Journal of Software, 2016 (in Chinese), 2020,31(02):564-577.

[[15]] Chen C L P, Liu Z. Broad learning system: An effective and efficient incremental learning system without the need for deep architecture[J]. IEEE transactions on neural networks and learning systems, 2017, 29(1): 10-24.

[作者简介]

唐源（1995—），男，信息工程大学硕士生，主要研究方向为主动防御和网络安全。

张铮（1976—），男，博士，副教授，硕士生导师，主要研究方向主动防御技术和高性能计算。

趙玉风（1996—），女，信息工程大学硕士生，主要研究方向为主动防御和网络安全。

季新生（1968—），男，教授，博士生导师，主要研究方向为网络安全，新一代移动通信技术。

异构云计算中虚拟机调度方法综述

周梦丽[1]，刘文彦[2]，陈福才[2]

[1]郑州大学，中原网络安全研究院，郑州 450000；

[2]国家数字交换系统工程技术研究中心，郑州 450002

摘　要：云计算中，为用户提供服务交互模式。通过这种方式，用户只需支付一定费用就可以使用相关设施。由于，云计算需要为多用户的每个任务做出合适决策。因此，在本文中，详细地回顾了为解决云计算中的任务调度问题而提出的各种算法。本文对不同算法的研究进行了详细回顾，目的是为解决不同调度任务中常见的问题。并基于资源和工作负载来平衡云中的负载，负载平衡器使用各种调度算法来确定向虚拟机发送请求的后端服务器，此外，提供商还负责跨物理机器动态地重新分配或迁移虚拟机，以整合工作负载，避免资源的过度使用或未充分利用。

关键词：云计算、任务调度、虚拟机

Review Of Scheduling Methodologies Of Virtual MachinesIn Heterogeneous Cloud Computing

Zhou Meng-li[1], Liu Wen-yan[2], Chen Fu-cai[2]

1.Zhongyuan Network Security Research Institute, Zhengzhou University, Zhengzhou 45000, China;

2.China National Digital Switching System Engineering & Technological R & D Center, Zhengzhou 450002,China

Abstract: In cloud computing, service interaction mode is provided for users. In this way, users can use the relevant facilities for only a certain fee. Because cloud computing needs to make appropriate decisions for each task of multi-user. Therefore, in this paper, we review in detail the various algorithms proposed to solve the task scheduling problem in cloud computing. The purpose of this paper is to solve the common problems in different scheduling tasks. The load balancer uses various scheduling algorithms to determine the back-end servers that send requests to the virtual machine. In addition, the provider is also responsible for dynamically reallocating or migrating virtual machines across physical machines to integrate the workload and avoid resource overuse or underutilization.

Key words: cloud computing; task scheduling; virtual machine

1　引言

云计算这个词出现在21世纪末，它描述的是一种结合了其他几种技术的操作模式，而不是一种新技术本身。定义多种多样，而且相当全面。美国国家标准与技术研究所（National Institute of Standards and Technology， NIST）的定义是将云计算视为“一种模型，用于实现对可配置计算资源的共享池（例如，网络、服务器、存储、应用程序、，和服务），它可以以最小的管理工作或服务提供商交互来快速配置和发布”[1]。此外，还定义了“按需自助服务”、“广域网络接入”、“资源池”、“快速弹性”和“计量”五个基本云特征。

云计算包含了广泛的应用：社交网络、税务和健康应用、存储解决方案、虚拟机租赁等等。通常，这些服务分为三种服务模型软件即服务、

基金项目：国家自然科学基金创新研究群体项目（NO.65121003）

Foundation item: The Foundation for Innovative Research Groups of the National Natural Science Foundation of China (NO.61521003)

平台即服务和基础设施即服务［1］。

在软件即服务（Software as a Service， SaaS）中，客户通过Web浏览器或特定客户端使用“在云基础设施上运行的提供商应用程序”［1］。例如Twitter和Google云消息。

在平台即服务（Platform as a Service， PaaS）中，云提供了一个平台，包括编程语言、库等，以运行“消费者创建或获得的应用程序”［1］。谷歌应用引擎就是一个例子。

在基础设施即服务（Infrastructure as a Service， IaaS）中，云为客户提供资源。后者能够在这些资源上“运行任意软件，包括操作系统和应用程序”［1］。例如Amazon Elastic Compute Cloud（EC2）和谷歌计算引擎。

云提供了大量资源，包括用于计算、数据中心、存储、网络、防火墙和服务形式的软件的平台。同时，它还提供了管理这些资源的方式，以便云用户可以访问它们而不会遇到任何与性能相关的问题。关于云计算中的调度机制，文献［2］中提出了几种算法。本文的工作安排如下：在这一部分，简要介绍了本文的研究重点。第二节讨论了各种研究方法。在第三节中，对整个研究工作的结果进行了简要总结。

2 文献综述

虚拟机（Virtual machine， VM）是执行单元，充当云计算技术的基础。虚拟化包括为各种应用程序和资源创建、执行和管理托管环境。云计算环境中的虚拟机共享资源，例如处理核心，系统总线等。每个虚拟机可用的计算资源受到总处理能力的限制。在这种环境模型中，任务到达模式是不可预测的，而且每个虚拟机的功能也各不相同。因此，迫切需要开发一种算法，该算法可以通过调度任务来处理虚拟机之间的需求，从而提高系统性能。

本文概述了三种主要类别的各种调度算法的详细信息：一般调度算法，混合调度算法和机器学习调度算法。在第2.1节中讨论了常用的优化调度方法的细节，第2.2节讨论了用于云计算调度问题的混合优化方法的细节，最后第2.3节讨论了用于云计算调度问题的带延迟优化方法的细节计算。在讨论了所有这些算法之后，以表格的形式讨论了阅读中发现的推论。

2.1 一般调度算法

文献［3］Atiewi等人研究了几种调度算法的综述。云任务调度的目的是获得较高的系统吞吐量，并将多个计算资源分配给应用程序。调度问题的复杂性随着任务的规模的增大而增大，成为一个亟待解决的问题。Min-Min算法用于在任务长度确定的情况下以减少任务的完成时间。维护这一点，云提供商必须获得用户满意度。文献［4］Agarwal等人提出了一种新的通用优先级算法，并与“先到先得”（First come first served， FCFS）和“循环调度”（Round robin， RR）进行比较。该算法在cloudSim工具箱中进行测试，结果证明，与其他现有调度算法相比，该算法提供了更好的性能。文献［5］Ebadifard等人在云计算的资源调度策略中引入了改进的粒子群优化算法（Particle swarm optimization， PSO）算法。实验结果表明本文提出的方法比基本的PSO算法收敛到接近最优解的速度快，并且在执行更多任务时更有效。文献［6］Basker等人提出了一种针对云基础设施服务的增强的加权循环调度（Weighted round robin， WRR），其考虑了任务长度和资源能力，有效地预测未充分利用的虚拟机并避免任何虚拟机的过载，通过静态和动态调度来优化利用参与的虚拟机，从而最小化任务的响应时间。虽考虑了相互依赖的任务，但尚未考虑在任务迁移的重载场景中实现负载平衡。文献［7］Zuo等人提出了一种资源成本模型，目的是更详细地描述任务对资源的需求。该模型返回用户的资源成本和预算成本之间的关联。基于该资源成本模型，提出了一种多目标优化调度方法。这种多目标优化算法将制造周期和用户的预算成本作为优化问题的指标，从而获得了性能和成本的多目标优化。

2.2 混合调度算法

文献［8］Elmougy等人提出了一种新的最短任务优先和具有动态可变量子时间（Short job first and round Robin with Dynamic variable Quantum time， SRDQ）任务调度算法的混合算法，该算法结合了使用动态变量任务量子的最短任务优先（Shortest job first， SJF）和RR算法的调度程序。该算法主要取决于两个基本密钥：第一个密钥具有动态任务量，以平衡短任务和长任务之间的等

待时间；第二个密钥则考虑将就绪队列分为两个子队列，Q1为短任务队列和另一个为长任务的。实验结果表明，该算法在减少等待时间、响应时间和部分长任务饥饿感等方面优于现有的控制方法。文献［9］ Velde等人描述了将任务划分为不同组，并将其复制到系统的本地中间件上。它使系统具有容错性和负载均衡性，提高了响应时间和资源利用率。文中采用Lexi搜索方法，在降低成本的同时，将任务调度到不同资源上。基于一种概率度量分配任务，该度量基于资源的可用性和任务的执行时间计算。负载均衡减少了每种资源在调度程序上产生的开销。文献［10］Qader介绍了一种以最小-最小调度算法为基础的负载平衡（Load balancing algorithm based on Min-Min ， LBMM）方法。该算法改善了Min-Min的负载不平衡性，使每个节点的执行时间最小化，但不指定如何为需要大规模计算的复杂任务选择节点。文献［11］Guo等人描述了云计算系统中的延迟最优虚拟机调度问题，云计算系统中的CPU、内存和存储等基础设施资源都是固定的。首先针对异构和动态工作负载采用排队模型。然后，将排队云计算系统中将虚拟机调度问题表述为一个决策过程，其中决策变量是虚拟机配置的向量，优化目标为平均任务完成时间而言的延迟性能。提出了一种结合了SJF缓冲和最小-最小最佳拟合（Min-Min best fit ， MMBF）调度算法的低复杂度在线方案，即SJF-MMBF，来确定解决方案。进一步提出了另一种结合基于SJF缓冲和强化学习（Reinforcement learning ， RL）的调度算法的方案，即SJF-RL，以避免SJF-MMBF中潜在的任务饥饿问题。文献［12］Al-Arasi等人描述并实现了一种独立的任务调度算法，旨在将用户的任务分配给大量的计算资源。提出了一种基于遗传算法（Genetic algorithm ， GA）和PSO的云计算任务调度混合算法。该算法使用CloudSim仿真实现。结果表明，该混合算法在缩短制造时间和提高资源利用率方面优于其他遗传算法和粒子群算法。

2.3 分类调度

文献［13］中Dong-Ju等人在云工作流系统中提出了一种分层存储调度策略。为了减少复制调度次数，首先利用数据节点的CPU负载、内存负载、网络负载、存储负载和网络距离等信息来评估节点的可用性。其次，选择最优方案。利用访问频率和硬件配置实现复制调度。在高处理性能和高配置节点上进行高频数据存储，提高了集群的响应率。实验结果表明，该策略能够提高异构集群中副本的访问效率和数据存储的本地均衡。文献［14］ Pranav等人提出了一种基于QoS感知的虚拟机分配策略在云使用基于信用的调度算法。当越来越多的作业开始到来时，它们很难获得虚拟机，因此它们被放在一个队列中。通过使用不同的调度算法，从队列中调度这些作业。现在，保证客户对虚拟机的等待时间不太长，即保证对客户的QoS是非常必要的。在提出的模型中，通过回收为特定类型的请求创建的vm，并将它们分配给需要这些请求的其他作业，可以确保QoS。通过这种方式，可以合理地节省创建和销毁虚拟机所浪费的时间。此外，我们使用了一种基于信用的调度算法，它不让队列中的作业等待更长时间。如果作业不能在规定的时间内满足，则不允许这些作业进入系统，从而进一步确保QoS。

文献［15］中Tan等人对面向信任服务的任务工作流调度算法进行了评价，以寻求在截止时间约束下执行时间的最优解。在满足用户约束的情况下，直接信任和推荐相结合的调度算法包括时间、成本等指标。面向信任的工作流调度具有模糊性和多目标的特点，能够分配适合云服务的任务。信任度是根据活动用户和其他每个用户之间的相似度来计算的。计算结果与最小关键路径和贪婪成本模型进行了比较。信任工作流调度具有静态的预定调度，但在动态运行环境中，需要一些自适应的方法来获得最优的调度结果。文献［16］Zhu等人提出了一种用于实时非周期独立任务调度和节能的滚动时域体系结构。作者通过滚动视界，重点研究了虚拟化技术在任务中的应用。这种滚动地平线体系结构按截止日期对即将到来的任务进行排序。通过计算启动时间和执行时间，还可以通过虚拟机处理能量感知调度。如果任务可以分配，则选择能耗最小的虚拟机执行任务，否则拒绝任务。但如果在云计算环境中测试了能量调度，则可以提高能量调度的实时性。文献［17］Sun等人提出了一个集中和分布式多服务器系统中复制延迟优化调度的研究进展。引入低复杂度的调度策略，并在每一个因果策略和非抢占

策略之间建立了随机排序下的时延最优或近似时延最优的低复杂度调度策略。在公共系统设置和延迟度量的情况下测量结果，目的是通过数据位置限制允许主观到达过程、随机任务大小、随机到期时间和异构服务器。引入新的样本路径工具来证明这些结果。文献［18］中Zhan等人提出一种基于双阈值负载均衡控制的云存储资源优化调度方法。利用网格区域配置方法对存储资源进行优化，并采用非线性时间序列重组方法解决了云存储资源分配信息流问题。通过分段插值的自适应优化来计算资源调度的阈值。针对优先级列表，对云存储资源的分区调度进行了实验研究。实验结果表明，该算法具有高效的负载均衡、较高的吞吐量，在资源优化方面具有良好的应用价值。

2.4 分析

本节对这项工作进行了性能分析，以确定这些方法的优缺点。下表给出了这项工作的分析。

序号	题目	优点	缺点
1	[3]Areviewenergy-efficienttaskschedulingalgorithmsincloudcomputing	1云提供商要达到用户满意度;2更好地利用资源。	不考虑每个请求的处理时间。
2	[4]Efficientoptimalalgorithmoftaskschedulingincloudcomputingenvironment	能熟练执行任务,提供更好的结果。	对于大规模计算的复杂任务,没有指定如何选择节点。
3	[5]APSO-basedtaskschedulingalgorithmimprovedusingaload-balancingtechniqueforthecloudcomputin-genvironment	减少任务的平均运行时间,提高资源的访问率。	不考虑每个任务处理时间。
4	[6]Anenhancedschedulinginweightedroundrobinforthecloudinfrastructureservices	1更少的复杂性和更公平地平衡负载;2平均分配工作负荷。	1需要先发制人;2不考虑任务处理时间。
5	[7]Amulti-objectiveoptimizationschedulingmethodbasedontheantcolonyalgorithmincloudcomputing	对绩效和成本的结果和预算成本优化给予反馈。	需要更高的执行时间。
6	[8] AnovelhybridofShortestjobfirstandroundRobinwithdynamicvariablequantumtimetaskschedulingtechnique	减少等待时间、响应时间和部分长任务的饥饿感。	1复杂性和长时间消耗;2不考虑资源上的现有负载。
7	[9]SimulationofoptimizedloadbalancinganduserjobschedulingusingCloudSim	在完成任务的最后期限之前,能量会减少。	较少考虑生产周期和成本。
8	[10]Load-balancingalgorithmsincloudcomputing:Asurvey	生成一个可以改善负载平衡并减少总体完成时间的计划。	调度时不考虑任务的优先级。
9	[11]OptimalschedulingofVMsinqueueingcloudcomputingsystemswithaheterogeneousworkload	1 CPU分配给CPU突发时间最少的进程;2优先考虑的是用户改善负载平衡,而不增加总完成时间。	1难以理解和编码;2重新安排任务以执行负载平衡将增加复杂性和时间。

续表

序号	题目	优点	缺点
10	[12]AHybridTaskSchedulingAlgorithminCloudComputingEnvironment	减少生产周期，提高资源利用率。	仅适用于特定用途。
11	[13]SchedulingStrategyofHierarchicalStorageaboutReplicationinCloudStorage	制作商是优先考虑更适合虚拟机的云资源。	根据最高等级容量分配数据存储，过度提交容量将导致负值。
12	[14]QoSawareVMallocationpolicyincloudusingacreditbasedschedulingalgorithm	已排序的任务队列将尽快完成。	不考虑机器故障、通信开销和动态工作负载。
13	[15]Atrustservice-orientedschedulingmodelforworkflowapplicationsincloudcomputing	命令运行调度程序通知，最适合调度重复性任务。	不适合于复杂的、事件驱动的任务，这种类型的任务可以在整个企业中自动执行任务。
14	[16]Real-timetasksorientedenergy-awareschedulinginvirtualizedclouds	采用集群共享，容量调度器以多级队列的形式将资源划分为池，实现资源利用率和吞吐量的最大化。	唯一的抽象是由管理员设置的反映共享集群成本的队列，如果长任务之后是短任务，则必须等到长任务完成。
15	[17]Ondelay-optimalschedulinginqueueingsystemswithreplications	引入延迟度量来允许随机到达过程、随机任务大小、随机交货时间和异构服务器。	对于由复杂事件驱动的任务来说，这并不容易。
16	[18]Towardsunderstandinguncertaintyincloudcomputingresourceprovisioning	高效的负载平衡、高吞吐量，具有很好的应用价值。	运行时间更长。

3 研究启发

延迟性能在很大程度上取决于不同的调度决策，这些调度决策与多少个虚拟机实例能够并行服务以及哪些实例首先运行有关。另一方面，云计算环境的异构性和动态工作负载特性使得延迟优化调度方案的设计成为一个重要的研究课题。在云计算环境中，用户通常会根据不同类型的虚拟机和不同的任务长度请求多个任务，其中虚拟机类别表示指定的资源集，由CPU、内存和存储器组成。任务长度表示任务预期运行的持续时间。这个严格的特性利用了云计算系统中的虚拟机调度问题，与通常的调度问题有很大的不同，在通常的调度问题中，一个任务只需要一个一维的资源，即与CPU资源有关的任务调度问题。为了解决所有这些问题，未来的工作将集中在开发和实现几种在线算法，以优化此类排队云系统中的虚拟机调度，降低每个任务的延迟性能。

4 结果与讨论

在本节中，将使用仿真来测量SJF-RL、SJF-MMBF、SRDQ、LBMM和Min-Min调度的结果。其中SJF-RL是一种结合基于SJF缓冲和强化学习的调度算法的方案，SJF-MMBF是一种结合了SJF

缓冲和最小-最小最佳拟合调度算法的低复杂度在线方案，SRDQ是一种使用动态变量任务量子的最短任务优先和RR算法的调度程序。LBMM一种以最小-最小调度算法为基础的负载平衡方法。

本节比较了SJF-MMBF、SJF-RL、SRDQ、LBMM和Min-Min调度的延迟性能和资源利用率性能。其中保证率被描述为延迟保证的工作数量与完成的工作数量之比，最大保证率表示延迟保证工作的最高数量。

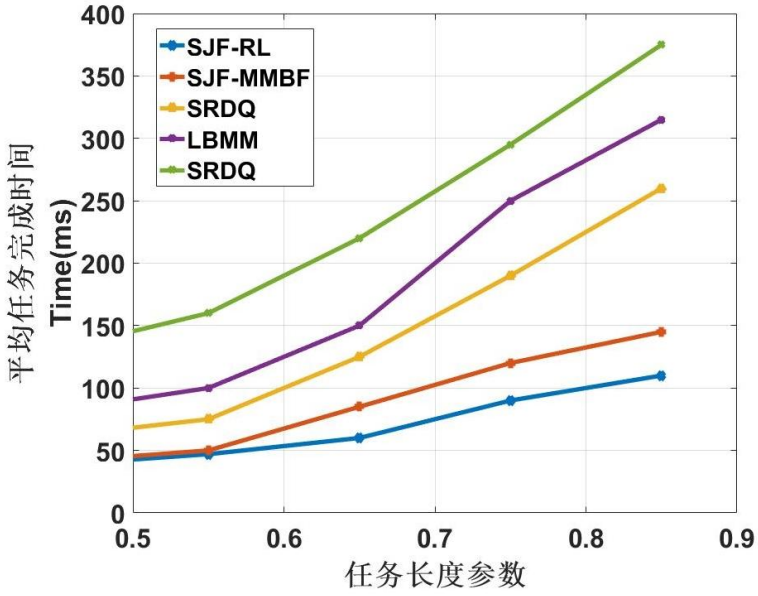


图1　延迟性能与任务长度参数

如图1所示，SJF-RL的性能优于SJF-MMBF、SRDQ、LBMM和Min-Min调度等方法，在一类任务的各个参数中，SJF-RL给出了最低的平均任务完成时间。SJF-RL算法的参数为0.9时，最佳平均任务完成时间为112ms，而其他方法（如SRDQ、LBMM和Min-Min调度）则需要较高的任务完成时间，分别为145ms、262ms、312ms和369ms。

图2显示了在几种类型中，相对于几种调度方法（例如SJF-RL、SJF-MMBF、SRDQ、LBMM和Min-Min调度）的平均任务托管率的性能比较结果。从结果可以得出结论，提出的SJF-RL算法的参数为0.9，具有最高的平均托管率85.15%，而其他方法（如SRDQ、LBMM和Min-Min Scheduling）则需要更高的任务完成时间，分别为80.15%，75.62 %，70.15%和66.92%。由于所提出的工作决策是通过强化学习进行的，因此平均工作托管率提高了比率。

图3显示了SJF-RL、SJF-MMBF、SRDQ、LBMM和Min-Min调度等调度方法在一类任务不同参数下的资源利用率。从图3可以看出，当参数为0.9时，所提出的SJF-RL算法的资源利用率为83.51%，而SJF-MMBF、SRDQ、LBMM和Min-Min调度方法的资源利用率分别降低了81.56%、73.62%、69.21%和65.81%。

5　总结与未来工作

本文分析了云计算任务的负载平衡和调度的多种算法及其存在的问题。回顾工作可知，以往提出的各种调度算法在其应用方式上各有优缺点。延迟性能在很大程度上取决于不同的调度决策，这些调度决策与多少个虚拟机实例能够并行服务以及哪些实例首先运行有关。另一方面，云计算环境的异构性和动态工作负载特性使得延迟优化调度方案的设计成为一个重要的研究课题。为了解决所有这些问题，未来的工作将集中在开发和实现几种在线算法，以优化此类排队云系统中的虚拟机调度，以降低每个任务的延迟性能。未来的工作将集中于资源的有效利用；提高总吞吐量和缩短任务完成时间。所有这些都将降低运营成本，吸引更多用户使用云计算。

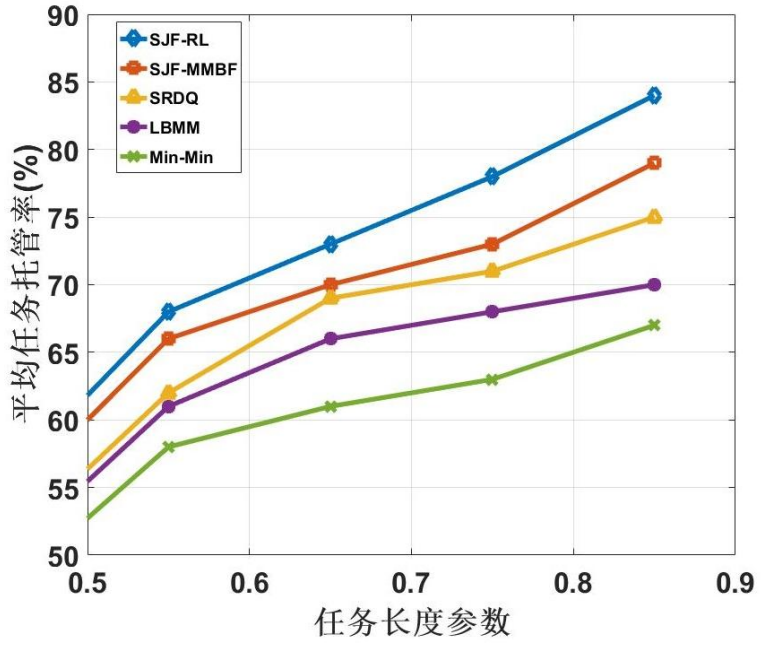


图2　平均任务托管率与任务长度参数

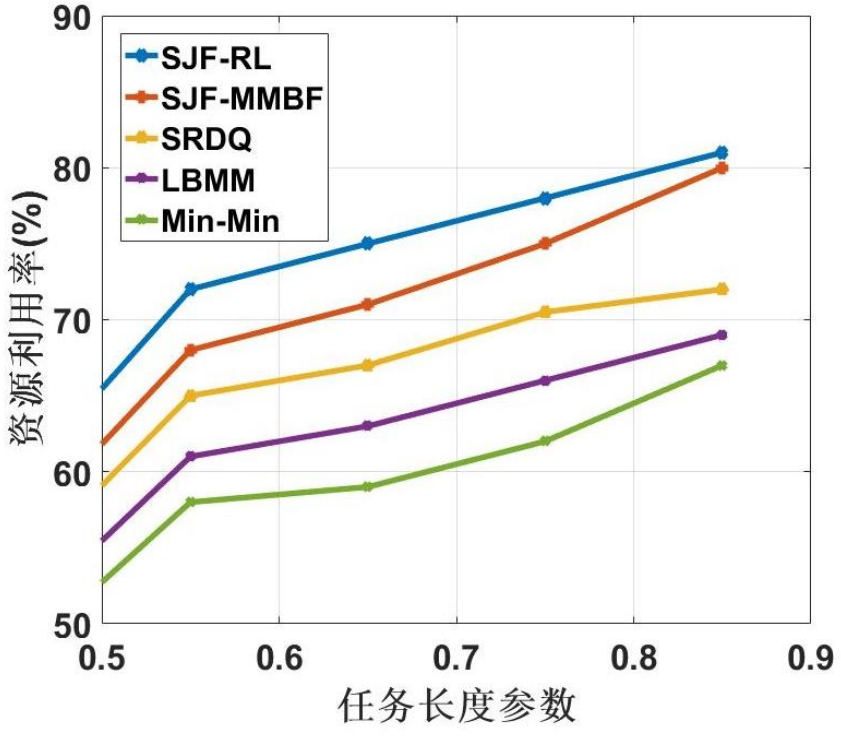


图3　资源利用率与任务长度参数

参考文献:

[1] MellP and GranceT. , The NIST Definition of Cloud Computing. Nat. Inst. Stand. Technol. ：Gaithersburg，MD，USA，2011.

[2] Razaque A , Vennapusa N R , Soni N , et al. Task scheduling in Cloud computing［C］// Long Island Systems，Applications & Technology Conference. IEEE，2016.

[3] Atiewi S , Yussof S , Ezanee M , et al. A review energy-efficient task scheduling algorithms in cloud computing［C］// Long Island Systems，Applications & Technology Conference. IEEE，2016.

[4] Agarwal, Jain, S. D. , Efficient optimal algorithm of task scheduling in cloud computing environment, International Journal of Computer

Trends and Technology (IJCTT)，2014，9(7).

[5] Ebadifard F , Babamir S M . A PSO-based task scheduling algorithm improved using a load-balancing technique for the cloud computing environment [J]. Concurrency and Computation: Practice and Experience，2018，30(1).

[6] Basker R. , Rhymend UthariarajV. , and Chitra DeviD. , An enhanced scheduling in weighted round robin for the cloud infrastructure services, International Journal of Recent Advance in Engineering & Technology，2014，2(3).

[7] Zuo L , Shu L , Dong S , et al. A Multi-Objective Optimization Scheduling Method Based on the Ant Colony Algorithm in Cloud Computing[J]. IEEE Access，2017，3:2687-2699.

[8] Elmougy S , Sarhan S , Joundy M . A novel hybrid of Shortest job first and round Robin with dynamic variable quantum time task scheduling technique[J]. Journal of Cloud Computing，2017，6(1).

[9] Velde V , Rama B . Simulation of optimized load balancing and user job scheduling using CloudSim[C]// IEEE International Conference on Recent Trends in Electronics. IEEE，2018.

[10] Qader, Nooruldeen, Nasih, , et al. Load-balancing algorithms in cloud computing: A survey [J]. Journal of Network & Computer Applications，2017.

[11] Guo M. , Guan, Ke, W. Q. , Optimal scheduling of VMs in queueing cloud computing systems with a heterogeneous workload, IEEE Access，2018,6:15178-15191.

[12] Al-Arasi, Saif, A. R. A. , HTSCC: A Hybrid Task Scheduling Algorithm in Cloud Computing Environment, International Journal of Computers & Technology，2018，17(2)

[13] Dong-Ju Y , Qing L I . Scheduling Strategy of Hierarchical Storage about Replication in Cloud Storage[J]. Computer ence，2017.

[14] Pranav P , Rizvi N , Jha R . QoS aware VM allocation policy in cloud using a credit based scheduling algorithm [C]// The International Conference on Communication and Computing Systems（ICCCS-2016). 2016.

[15] Tan W, Sun Y, Li LX, Lu G, Wang T, A trust service-oriented scheduling model for workflow applications in cloud computing, IEEE Systems Journal，2014,8(3).

[16] Zhu X. , Yang L. T. , Chen H. , Wang J. , Yin S. and Liu, X. , Real-time tasks oriented energy-aware scheduling in virtualized clouds，IEEE Transactions on Cloud Computing，2014,2(2.

[17] Sun Y. , KoksalC. E. , and ShroffN. B. , —On delay-optimal scheduling in queueing systems with replications, CoRR, vol. abs/1603. 07322，2016.

[18] Tchernykh A. , Schwiegelsohn U. , Alexandrov V. and Talbi, E. G. , Towards understanding uncertainty in cloud computing resource provisioning，Procedia Computer Science，2015，51.

[作者简介]

周梦丽（1993-），女，硕士，研究生，主要研究方向为云安全，e-mail：zml12080205@163. com；

刘文彦（1986-），男，博士，助理研究员，主要研究方向为网络空间防御和云安全；

陈福才（1974-），男，教授，研究员，主要研究方向为网络安全。

有限租期内时序数据驱动的设备最优停时决策

陈熙沅，中国人民大学统计学院

[1]张明培育 微软（亚洲）互联网工程院邓英俊* 天津大学应用数学中心

摘　要：本文研究了基于设备状态数据和停时决策的预防性维护策略，用以提升设备租赁期内的生产收益和减少系统失效带来的维修费用。具体而言，本文在最优停时问题的框架下考虑维修决策，并引入折旧因子考虑预期收益对当前决策的影响，形成总体决策收益函数的合理设计。进一步地，为建立从高维时序状态数据到最优停时决策之间的映射，本文采用多层感知机来输出设备在有限时间段内每一检测时间点上的停止决策概率，并通过逆向归纳和随机梯度下降来寻找最优的神经网络参数。为验证本文提出的停时决策策略和优化算法，我们在NASA的CMAPSS涡扇发动机仿真退化数据集上进行仿真实验。经实证发现，本文提出的数据驱动的停时决策在测试数据集上较经验最优停时（即失效时间）的平均离差小于25个时间单位，具有较好的决策效果。

关键词：预防性维护、最优停时、深度神经网络、多层感知机、逆向归纳、剩余寿命

Optimal Stopping Decision with Time Series Data in a Finite Lease Period

——Xiyuan chen, School of Statistics, Renmin University of China

Abstract: This paper studies the predictive maintenance strategy based on the equipment condition data and stopping decision to improve the profits generated during the equipment lease period and reduce the maintenance cost caused by system failure. Specifically, this paper considers the maintenance decision under the framework of the optimal stopping problem, and introduces the depreciation factor for the expected profits on current decisions, so as to propose a reasonable decision target function. Furthermore, in order to establish the mapping from the high-dimensional time series data to the optimal stopping decision. this paper uses a multi-layer perceptron to output the stopping probability of the device at each detection time point in the finite time period, and finds the optimal neural network parameters by backward induction and stochastic gradient descent. In order to verify the stopping decision strategy and optimization algorithm proposed in this paper, we carried out simulation experiments on the CMAPSS turbofan engine simulation degradation data set from NASA. It is found that the mean deviation of the data-driven stopping decision proposed in this paper is less than 25 time units compared with the empirical optimal stopping time (i. e. , the failure time) in the test data set, reaching a good decision performance.

Key words: predictive maintenance; optimal stopping; deep neural network; multi-layer perceptron; backward induction; remaining useful life

Mingpeiyu Zhang， Microsoft Search Technology Center Asia

Yingjun Deng*， Center for Applied Mathematics， Tianjin University

1　引言

当今时代工业发展的重要性不容小觑，工业4.0的概念最早是由德国提出的，工业4.0时代是利用信息化变革促进产业升级换代的时代，也是继

基金项目：国家自然科学基金资助项目（71701143），天津市人工智能重点项目(19ZXZNGX00050)，粒子输运与富集技术国防科技重点实验室开放基金。

通讯作者：Email: yingjun.deng@tju.edu.cn

蒸汽时代、电气时代、信息化时代之后的智能化时代。其三大主题包括智能工厂，智能生产，智能物流。其关键点在于实现物质原材料和信息的等价关系，使得制造业最终会走向信息产业。与之相对应的，为了贯彻落实新发展理念，推动工业发展的＂效率变革＂，坚定不移的走工业强国的道路，我国也在2015年提出了制造强国战略的第一个行动纲领《中国制造2025》。工业大数据技术的重要性与日俱增，其本质目的从复杂数据集中发现新的规律和模式，挖掘有价值的新信息，从而促进制造型企业的产品创新、提升经营水平和生产运作效率以及拓展新的商业模式。

近些年，尤其从1990年开始，越来越多的厂家从自购设备转向租借设备[18]。原因主要有两个，一是购买设备成本过高； 二是随着技术的发展，设备更新换代的速度越来越快。在租赁期内设备由于老化，人为操作不当等原因仍会面临维修问题。如果等到设备故障再进行维护，工厂不仅仅需要承担生产延误带来的损失，还需要对设备租借方进行赔偿，这无疑会使工厂损失惨重。这时，制定最优的预测性维修策略就显得尤为重要。最优维护策略的制定可以使设备在租赁期间可以进行有效的主动维护[19]，即在设备出现故障前进行维护，可以有效提高企业的生产效率，对于工厂而言，提高生产效率是增加企业利润的必要途径，维护策略的制定也被认为是提高工业整体效益的必要手段，这也贯彻落实了国家工业发展战略对于每个小企业的要求和期待。此外，现代化工业设备大都备有设备传感器，可以实时返回设备的状态数据信息，这些状态数据的获取也为分析设备运行状态、及时维护、防止设备因发生故障而造成的巨额损失提供了必要的技术支持。

设备传感器返回的状态数据具有以下两个性质：实时性和高维性。其一，数据的实时性来源于传感器在每个监测点都可以返回的设备状态数据，我们希望以此为依据做出最优决策，因而最优停时的决定是一个不断更新的过程，因此动态规划[21]被广泛应用在这类问题之上。其二，设备传感器返回的状态数据是高维的，研究者们在想要有效利用这种数据的过程中往往会面对“维度灾难”[8]，因为问题的复杂程度会随着数据量的增大而成倍增长。为了建立从高维状态数据到决策向量的映射，根据设备状态数据做出维护决策，深度强化学习技术在近年得到广泛的关注。

为了更好的解决上述内容中提到的问题，我们在与本文相关的以下三方面研究做了一定调研。

基于设备状态的维护（Conditional-based Maintenance）：维护策略通常分为两种，基于时间的维护和基于状态的维护。目前关于基于时间的维护策略已经较为完善，例如［1-3］。同样，在低维情景下，很多学者也对基于设备实时状态的维护策略进行了研究，如，Jardine[4]总结了近些年来，基于设备状态的维护模型的发展历程与研究成果，系统介绍了模型分类，算法设计，数据采集和数据处理方法。Peng[5]也为人们的研究提供了各种技术和算法的概括总结，不同的是，他讲预测模型分为四类：物理模型，知识模型，基于数据驱动模型和组合模型。Ahmad和Kamaruddin[6]介绍了基于时间的维护和基于状态的维护在工业中的应用，也介绍了最新的状态监测技术。Shin和Jun[7]回顾了基于状态维护的方法，并重点讨论了基于状态维护在其定义，优点，缺点，相关国际标准，程序和技术等方面的问题，成为人们后续研究的重要依托和参考。

深度强化学习（Deep Reinforcement Learning）：该技术是通过学习周围环境中的信息，得到最优的方法策略，尤其在处理复杂程度较高的问题时会有更好的效果。它不仅运用在自然科学社会科学之上，也广泛应用于工程技术[9]，例如，在Wang等人[10]也介绍了当前在维修领域常用的五种深度神经网络；Silver等人[11]提出了AlphaGo；Wang等人[12]提出了竞争性神经网络体系；Tamar等人[13]设计了值迭代神经网络；Huuhtanen和Jung[14]用卷积神经网络来检测光伏板的变化以此进行合理维护；Lee等人[15]利用神经网络构造一个策略模型用于预防重力加速器在运行之中产生故障；Ong等人[16]一起从基于环境的传感器设备网络出发，提出了一种用于设备维护的无模型深度强化学习算法；Zhang等人[17]提出了用于维护的六种机器学习算法。

动态规划（Approximate Dynamic Programming）：为解决最优停时问题，研究者们大多会采用近似动态规划[22]的算法，动态规划的思想可以帮助人们把需要进行最优决策的问题拆分成一个

一个小问题，在不知道当前信息的情况下，根据过去的已知信息做出优化决策，决策做完之后即会获得一个当前信息的反馈，据此再做出下一阶段的决策。为了解决高维最优停时问题，研究者们将动态规划思想与深度学习技术[23]相结合。进一步，为了区别于深度学习“黑箱子”的特性，有研究者提出将动态规划与决策树融合，提出一种新的可解释的最优停时算法[24]。

本文的研究目的是为了帮助租赁设备的工厂制定一套适当的维护策略，使得在租期内工厂的收益最大化。具体来讲，我们利用了多层感知机[20]建立端到端的决策模型[21]，再根据设备的高维状态数据，做出决策，得到最优决策策略。本文的贡献主要在于以下两点：其一，实现了深度学习技术和最优停时模型[21]的有机结合；其二，将优化目标，即收益函数，融入多层感知机的训练过程，由此得到的感知机可以帮助我们做出最优的维护决策。

在后续章节中，我们对于问题场景，解决问题的技术方案以及仿真实验过程都进行了具体阐述。第二章中主要介绍了问题的场景设置，详细阐述了合理制定收益函数的过程。第三章具体描述了最优停止时间的确定，引入神经网络近似决策函数，并利用逆向规划学习神经网络参数。第四章是应用上述技术方案解决具体场景下问题的仿真实验。在第五章中，我们给出本文的结论以及未来的研究方向。

2 高维时序数据驱动的最优停时决策

2.1 场景和数据描述

在本文中，我们假定工厂为订单需求，租赁一批带有传感器的现代化设备。所有设备在未开始运转时都是完好无损的。已知租期有限，在租赁期间传感器会以单位时间间隔对设备进行周期性检测，检测方式为完全检测，即每次检测都可以获得设备传感器返回的所有状态数据信息，不存在数据缺失情况，检测到的数据具有高维和不确定性两个特质。假设设备的历史信息也是可以获得的，根据历史数据可以得到设备生命周期内的所有状态数据和故障时间。工厂希望可以在租赁期间尽量减少设备损坏，因为损坏设备不仅需要进行修理还需要依照合约向租借方支付巨额赔偿，所以需要制定一套最优维护策略，在设备损坏之前对其进行维护，具体的维护原则如下：如果进行维护，那么设备停止运转不再产生利润直到租赁结束；如果不进行维护，那么设备将继续正常运转直到下一个监测点。

2.2 基于设备状态的收益函数

假设设备正常工作时，设备在有限租期内的状态可以用状态连续时间离散的马尔科夫过程$\mathbb{X}=(X_n)_{n=0}^{N}$来描述，$\mathbb{X}=(X_n)_{n=0}^{N}$取值在$\mathbb{R}^d$上的马尔科夫过程，其中$X_i$表示第i次观测状态。设备在单位时间产生单位利润。在租期N结束前的每个检测时间点$n\in\{0,1,2,...,N\}$，都可能进行预防性维护。我们的目标是基于设备当前状态设计一个合理的目标函数来描述设备收益。为了具体说明设计过程，引入如下记号：

N是设备检测的有限时间段；ζ是用于描述设备失效时间的随机变量。在实验之前是未知的，但对于每台设备来讲，若设备可以不受限制的自然运转，那么ζ就是一个可以观测到的定值。如果观测时刻在设备损坏时间之前，则若在此刻维护，设备将不再工作，保持停止状态直到租赁合约结束；否则设备损坏无法继续工作；τ是决策策略需要决定的在租期内进行维护的时间点，$\tau=1,2,...,N$；X_n是描述设备在时刻n的状态，若设备在时刻ξ损坏了，则设备保持状态X_ξ；C_f是如果在设备失效时间ζ之前没有及时停止设备并进行维护，工厂需要支付的赔偿费用；q是折旧因子，已知设备在时刻n处正常运转，其在n到$n+1$时刻仍然工作的概率。

若设备在有限的租赁期N内发生故障，则设备在时刻n产生的收益为：

$$\mathrm{g}_\mathrm{n}^{\pi}=\begin{cases}n & n<\zeta\\ \zeta-C_\mathrm{f} & n\geq\zeta\end{cases} \tag{2-1}$$

在考虑折旧影响之后，公式（2-1）可以改写为：

$$g(n,X_n)=nq^n\square_{\{n<\xi\}}+(\xi-C_f)q^\xi\square_{\{n\geq\xi\}} \tag{2-2}$$

但是这里需要注意的是设备可能在整个租赁期间都不发生任何故障，可以一直产生利润直到时刻N。此时我们认为设备在N时刻产生的利润为：

$$\mathrm{g_N}=N*\mathrm{q^N} \tag{2-3}$$

整合公式（2-2）和（2-3）后可以得到在整个

租赁时间段N内的收益函数：

$$g(n,X_n)=\begin{cases} nq^n\square_{\{n<\xi\}}+(\xi-C_f)q^{\xi}\square_{\{n\geqslant\xi\}} & 0<n<N \\ Nq^N & n=N \end{cases} \tag{2-4}$$

2.3 最优停时问题

本课题主要考虑形如：

$$sup\mathbb{E}g(\tau,X_\tau) \tag{2-5}$$

的问题，其中$X=(X_n)_{n=0}^N$是一个在$\mathbb{R}^d$上取值的离散的马尔科夫过程，目标是使得收益函数取值最大，停止时间τ依赖于X的观测值.将观测状态用马尔科夫过程建模具有以下两点合理性：一是几乎所有连续最有停止问题都可以进行时间离散化，因为马尔科夫性的假设不会是问题失去一般性；二是观测的当前状态包括了过去的所有信息，尽管这样会使问题的维度增加。

理论上在有限时间段内，使得收益函数值最小的最优停止时间是可以通过计算每个点停止时的反馈值得到的，是一定可以计算的，困难之处在于如何解决高维度问题.不同于传统解决方法，本文希望利用多层感知机构造一个端对端的决策模型来有效解决最优停时问题，因此我们将决策分解成一系列的0 - 1停止决策，并用多元前馈神经网络来近似.可能的最优停止时间可以通过梯度下降法得到，之后通过迭代比较找到所需要的最优停止时间。

3 基于多层感知机的端到端决策学习

3.1 技术概述

我们希望利用端对端学习[3]的方法制定一套维护策略使得设备维护过程中收益最大化。模型使用问题具有如下形式：

$$\sup_{\tau\in\Gamma} Eg(\tau,X_\tau) \tag{3-1}$$

其中，g: $\{1,2...,N\}\times\mathbb{R}^d$是可测函数，$\Gamma$是所有可能进行预防性维护的时间点的集合。为确保问题（3-1）的定义完善且能够得到其最优值的置信区间，假定g存在二阶矩，即：

$$E[g(n,X_n)^2]<\infty \qquad n\in\{1,2,...,N\}$$

为解决上述最优停时决策问题我们首先要将这个决策分解成一系列的0 - 1决策问题，即在每个时间点是否停止。利用这一系列决策函数表达出最优停时。用神经网络网络来近似决策函数是本课题的创新和重点之处。合理的超参数选择和神经网络搭建会使我们得到在每一点停止的概率，再根据概率大小判断在每一点是否要停止，即0-1决策函数的输出值。

神经网络模型参数的逆向规划则是技术方案中的另一个亮点之处。参数训练采用反向传播和逆向递归相结合的方法：由于在有限时间段的最后一个时间点必然会停止，我们就从最后一个停止时间出发，向前寻找前一个时间段内的最优停止时间，不断向前迭代，最终获得在整个时间段内的最优停止时间。

最后则是参数的具体训练过程，核心思想就是梯度下降法，近似决策函数的神经网络是几乎处处光滑的，而损失函数又是神经网络的一个线性变换，因此可以应用链式求导法则逐层求导进行参数优化。

3.2 基于神经网络的停时表述

在有限时间内确定设备进行维护时间的问题可以转化成在每个检测点是否做维护的一系列问题，因此停止时间也可以由一系列决策函数确定。一般来说，在$t=n$时刻要不要做决策取决于设备在时间点n之前的状态，但基于模型假定设备状态是一个离散型Markov过程，因此决策可以根据函数：

$$f_n:\mathbb{R}^d\to\{0,1\} \qquad n\in\{1,2,...,N\}$$

构造辅助成本问题：

$$V_n=\sup_{\tau\in\Gamma_n} Eg(\tau,X_\tau) \tag{3-2}$$

其中，$n=0,1,2,...,N,\Gamma_n=\{n\leq\tau\leq N\}$

特别的当整个有限时间段内都没有进行预防性维护，到$t=N$时设备会因为租赁到期而停止进行维护。即$\tau_N=N$时，$f_N\equiv 1$

根据给定的$n\in\{0,1,...,N\}$以及一系列取值$\{0,1\}$的可测函数$f_n,f_{n+1},...,f_N:\mathbb{R}^d\to\{0,1\}$以及$f_N\equiv 1$，可以得到n到N内使得损失成本最低的预防性维护时间：

$$\tau_n=\sum_{m=n}^{N} mf_m(X_m)\prod_{j=n}^{m-1}(1-f_j(X_j)) \tag{3-3}$$

其中，$n=0,1...N-1$，Γ_n是所有满足$n\leq\tau\leq N$的停止时间的集合。

为了确定决定是否在n时刻停止的函数f_n，我们采用神经网络f^{θ_n}近似，其中$\theta\in R^q$。由于f^{θ_n}取

值在{0, 1}，不能使用梯度下降法，因此我们引入一个连续神经网络：

$$F^{\theta}:\mathbb{R}^{d} \to (0,1) \qquad \theta \in \mathbb{R}^{q}$$

具体形式如下：

$$F^{\theta} = \psi \circ a_{I}^{\theta} \circ \varphi_{q_{I-1}} \circ ... \circ \varphi_{q_1} \circ a_1^{\theta}$$

其中 φ 是 sigmoid 函数，满足 $f'(x) = f(x)[1 - f(x)]$；对于任意的 $j \in N$，φ_j 是激活函数 ReLU，即 $\varphi_j(x_1, x_2, ..., x_j) = (x_1^+, ..., x_j^+)a_i^{\theta} = A_i x + b_i, i = 1, 2...I$，$\theta \in R^q$ 中包括：

矩阵 $A_1 = [a_{i_1,j}]_{q_1 \times d} \in \mathbb{R}^{q_1 \times d}, A_2 = [a_{i_2,i_1}]_{q_2 \times q_1} \in \mathbb{R}^{q_2 \times q_1}..., A_I = [a_{1,i_{I-1}}]_{1,i_{I-1}} \in \mathbb{R}^{1 \times q_{I-1}}$，

向量 $b_1 \in \mathbb{R}^{q_1}, ..., b_I \in \mathbb{R}$；网络共有 I 层，$q_1, q_2...q_{I-1}$ 分别表示每个隐藏层的节点数目，

向量 $y^1, y^2, ..., y^{I-1}$ 分别表示每个隐藏层的输出，对于任意的 $i = 1, 2, ..., I - 1$，$y^i = (y_1^i, y_2^i, ..., y_{q_i}^i)$ 最终输出为 y。

根据 F^{θ_n} 的表达式可知，输出值 $y \in (0, 1)$ 表示在 n 点出停止的概率，因此可以得到在时刻 n 的损失金额为：

$$E[g(n, X_n)F^{\theta_n}(X_n) + g(\tau_{n+1}, X_{\tau_{n+1}})(1 - F^{\theta_n})] \quad (3\text{-}4)$$

我们的目标是训练模型得到使得损失金额最小的参数 θ_n，当 θ_n 确定之后，可以得到在时刻 n 的决策函数 f^{θ_n}：c

$$f^{\theta_n} = 1_{[0,\infty]} \circ a_I^{\theta_n} \circ \varphi_{q_{I-1}} \circ ... \circ \varphi_{q_1} \circ a_1^{\theta_n}$$

f^{θ_n} 与 F^{θ_n} 之间的区别就在于 f^{θ_n} 取值 {0, 1}，F_{θ_n} 取值（0，1），二者具有如下对应关系：

$$f^{\theta_n} = \begin{cases} 1 & F^{\theta_n} \geq \frac{1}{2} \\ 0 & F^{\theta_n} < \frac{1}{2} \end{cases}$$

3.3 基于逆向递归的参数优化

本文所用最优停时算法采用逆向递归的方法来训练参数，从最终停止时间出发逐一寻找。无论如何设备必须在 N 时刻停止，因此 $f_N = 1$，在此基础上寻找 N-1 阶段的最优停止时间；当 N-1 阶段的最优停止时间确定之后，再寻找 N-2 阶段的最优停止时间；最后一直回到起始阶段。在寻找 f_n 时，$f_{n+1}, f_{n+2}..f_N$ 都已经确定，即 τ_{n+1} 已知，$g(\tau_{n+1}, X_{\tau_{n+1}})$ 自然也是可以确定的值。

当 $n = N - 1$ 时，$\tau_{n+1} \equiv N$；当 $n \leq N - 2$ 时，τ_{n+1} 可以写成：

$$\tau_{n+1} = l_{n+1}(X_{n+1}, X_{n+2}, ..., X_{N-1})$$

其中，可测函数 $l_{n+1}: \mathbb{R}^{d(N-d-1)} \to \{n + 1, n + 2, ..., N\}$ 有如下形式：

$$l_{n+1}^k = \begin{cases} N & n = N - 1 \\ l_{n+1}(x_{n+1}^k, x_{n+2}^k, ..., x_{N-1}^k) & n \leq N - 2 \end{cases} \quad (3\text{-}5)$$

$\{x_n^k\}_{n=0}^{N}, k = 1, 2, ...K$ 是设备独立的 K 次状态观测值，根据每次的观测值可以得到第 k 次的损失函数：

$$\phi_n^k(\theta_n) = g(n, x_n^k)F^{\theta_n}(x_n^k) + g(l_{n+1}^k, x_{l_{n+1}^k})(1 - F^{\theta_n}) \quad (3\text{-}6)$$

取 K 次损失函数的平均值可以得到与公式（3-4）近似的值，即训练的损失函数：

$$E(\theta_n) = \frac{1}{K}\sum_{k=1}^{K}\phi_n^k(\theta_n) \quad (3\text{-}7)$$

4 基于 CMAPSS 涡扇发动机仿真退化数据的仿真实验

4.1 数据准备和实验方案

为了验证上述模型，我们选用 CMAPSS 涡扇发动机中的 001 组仿真数据，CMAPSS 数据集是由 NASA 卓越故障预测研究中心提供的。数据集由多元时间序列构成，包括训练集和测试集两个数据集。每个时间序列都来自不同的发动机引擎。这些数据可以被认为是来自同一个车队的同一类型发动机。每台发动机的初始磨损程度不同，制造工艺不同。训练模型时，我们将训练集中记录的同一设备的 100 次实验数据按照 7：3 的比例，分成训练集和测试集两组。

为了探究本文中利用多层感知机建立的最优停时决策模型是否能有效解决租赁场景下的维护决策问题，我们设计了如下实验方案：

1）为确定实验结果对于不同超参数的敏感程度，我们选取 3 层神经网络，训练周期均为 20；在隐层节点个数取值 d，2d，4d 时，探究模型对于隐层节点个数的敏感程度，其中 d 是数据的维度。

2）对上述实验结果，即预测设备停止的时间，和设备历史损坏时间进行回归性分析。

4.2 实验结果与讨论

当选用三层神经网络，每个隐层节点数分别取 d，2d，4d 时，进行多次试验，将设备的历史损坏时间与决策产生的最优停时进行对比。综合图 1 中的四次实验结果可以看出，在每次实验中，设备每次的最优停时并没有随着隐层节点数的变化

而产生变化，无论节点数为d，2d还是4d，设备的最优停时都是相同的，因此隐层节点个数对于模型的输出没有影响。更进一步，分别看图1中的四次实验，每次实验中都有超过三分之二的时间节点与设备的历史损坏时间节点重合，因此我们认为模型的决策效果与真实情况相对接近，即决策模型的训练效果较为理想。

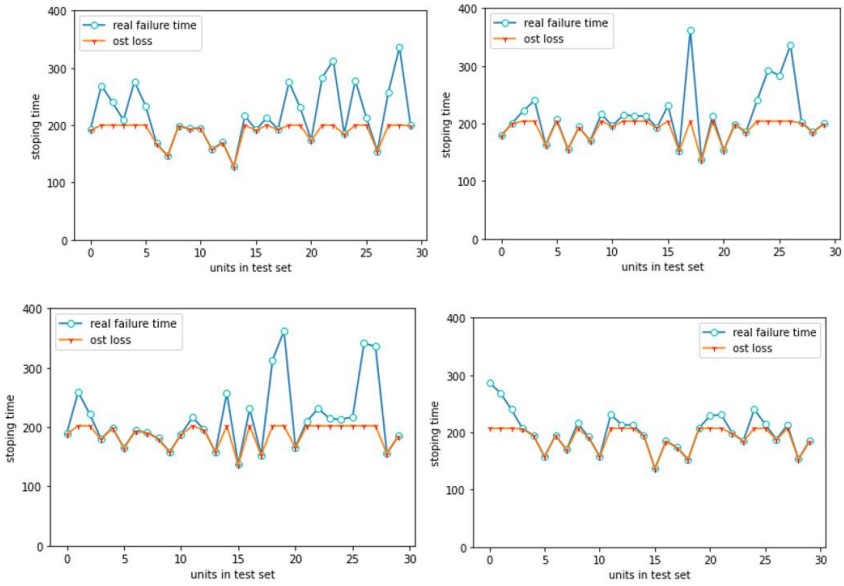


图1 四次实验均显示最优停时与隐层节点数无关

在构造收益函数的时候我们假定设备在单位时间产生单位收益，考虑设备折旧产生的损失（由折旧因子q产生）以及设备因没有合理维护而损坏支付的赔偿费用（假设为100个单位）五次实验的收益情况如图2所示。从图2中我们可以看出，设备的收益集中在75个单位附近，这说明每次实验设备都在损坏前进行维护，并没有出现赔偿状况。

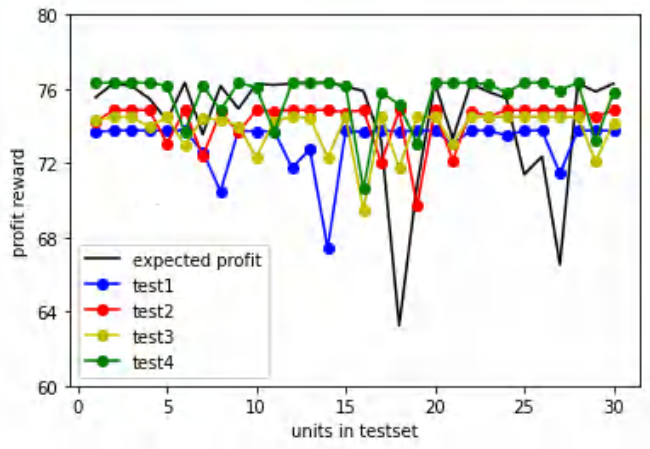


图2 五次实验的收益变化情况

为了更精准的展现真实值和决策值之间的误差，我们分别求得二者的均方误差（MSE），均方根误差（RMSE），平均绝对误差（MAE）以及R平方（R Squared）结果如图2所示。虽然从图1中看超过三分之二的节点处决策停时与历史设备损坏时间相近，但是图2中MSE较大，这可能是由于停时本身较大，集中在200附近，若一次实验设备的损坏时间很大就会对MSE的值产生很大影响。

5 结束语

本文利用多层感知机构建了一个端对端的最

test	MSE	RMSE	MAE	R Squared
1	1781.47	42.2	18.2	0.36
2	1418.37	37.66	16.9	0.32
3	1426.63	37.77	17.23	0.41
4	2063.67	45.43	22.73	0.19

图 3　四次实验结果和预测结果的回归性分析数据

优停时决策模型，从而确定一个使得有限租约时间内工厂收益最大的维护策略。研究过程中的最大难点在于如何进行高维时序状态数据下的决策，更具体地讲就是如何建立一个从设备高维状态数据到最优停时的映射。在该场景下，“维度灾难”大大增加了问题的复杂程度，通常的统计工具难以处理。因此我们引入了深度神经网络，并利用多层感知机作为处理工具，实现了数据降维和停止维护策略的制定。

最优停时决策的目标是使得工厂收益最大化，因为如何优化收益函数，如何在收益函数中体现基于数据基于状态的特点是未来研究的方向之一，初步设想是把对于设备剩余寿命的预测值引入到收益函数的设计之中，这也意味着我们的决策过程将更复杂；此外，为进一步提升实验结果的精确度，我们还将尝试对模型中的参数贴近实际场景设计，例如赔偿费用，一方面可以进行基于经验的枚举法修正，另一方面可以选择延续基于设备状态的建模尝试，更加合理化的参数选择会消除主观因素对于实验效果的影响，同样具有研究价值和意义。

参考文献:

[1] Sherif Y S , Smith M L . Optimal Maintenance Models for Systems Subject to Failure— A Review[J]. Nav. Res. Logist. , 2010, 28(1): 47-74.

[2] Wang H. A survey of maintenance policies of deteriorating systems [J]. European Journal of Operational Research, 2002, 139.

[3] Alaswad S , Xiang Y. A review on condition-based maintenance optimization models for stochastically deteriorating system [J]. Reliability Engineering & System Safety, 2017, 157.

[4] Jardine A K S , Lin D , Banjevic D . A review on machinery diagnostics and prognostics implementing condition-based maintenance[J]. Mechanical Systems & Signal Processing, 2006, 20 (7):1483-1510.

[5] Peng Y , Dong M , Zuo M J . Current status of machine prognostics in condition-based maintenance: a review[J]. International Journal of Advanced Manufacturing Technology, 2010, 50(1-4):297-313.

[6] Zequeira R I, Berenguer C. Optimal scheduling of non-perfect inspections[J]. IMA Journal of Management Mathematics, 2006, 17 (2): p. 187-207.

[7] Shin J H, Jun H B. On condition based maintenance policy [J]. Journal of Computational Design and Engineering, 2015, 2 (2): 119-127.

[8] Becker S , Cheridito P , Jentzen A , et al. Solving high-dimensional optimal stopping problems using deep learning[J]. Arxiv, 2019.

[9] Hao X , Zhang G , Ma S . Deep Learning[J]. International Journal of Semantic Computing, 2016, 10(03):417-439.

[10] Deng, Li. Three Classes of Deep Learning Architectures and Their Applications: A Tutorial Survey[J]. APSIPA Transactions on Signal and Information Processing, 2013.

[11] Dongbin Zhao, Derong Liu, LewisF. L. , et al. Special Issue on Deep Reinforcement Learning and Adaptive Dynamic Programming [J]. IEEE Transactions on Neural Networks & Learning Systems, 2018, 29(6):2038-2041.

[12] Wang Z , Bapst V , Heess N , et al. Sample Efficient Actor-Critic with Experience Replay[J]. 2016.

[13] Tamar A , Wu Y , Thomas G , et al. Value Iteration Networks[J]. 2016.

[14] Huuhtanen T , Jung A . Predictive maintenance photovoltaic panels via deep learning[C]. 2018:66-70.

[15] Lee S W , Tak Y H , Yang H J , et al. Deep learning application of vibration data for predictive maintenance of gravity acceleration equipment[J]. 2020.

[16] Ong K S H , Niyato D , Yuen C . Predictive Maintenance for Edge-Based Sensor Networks: A Deep Reinforcement Learning Approach [J]. 2020.

[17] Zhang W , Yang D , Wang H . Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey[J]. IEEE Systems Journal, 2019, 13(3):2213-2227.

[18] 杨爱峰, 王文婷, 范世庆. 耐用设备租赁及预防性维修的联合优化策略[J]. 合肥工业大学学报:自然科学版, 2015(38): 557.

[19] 张云正,张晓红,曾建潮. 租赁设备的状态维修决策建模与优化[J]. 系统工程理论与实践,2019,39(7):1732-1743.

[20] Abdullah-Al-Mamun M , Alam T . An approach to empirical Optical Character Recognition paradigm using Multi-Layer Perceptorn Neural

Network[C]. 2015 18th International Conference on Computer and Information Technology (ICCIT). IEEE，2016.

[21] BeckerSebastian, CheriditoPatrick, JentzenArnulf. Deep optimal stopping[J]. Journal of Machine Learning Research，2019，20：1-25.

[22] TamakiHisao. Positive-instance driven dynamic programming for treewidth[J]. Journal of combinatorial optimization，2019.

[23] Becker S , Cheridito P , Jentzen A , et al. Solving high-dimensional optimal stopping problems using deep learning[J]. Papers，2019.

[24] Ciocan D , Misic V . Interpretable Optimal Stopping[J]. SSRN Electronic Journal，2018.

[作者简介]

陈熙沅（1998—），女，理学学士

中国人民大学统计学院硕士研究生在读，研究方向为应用统计。

张明培育（1994—），男，理学硕士，微软（亚洲）互联网工程院工程师， 研究方向为机器学习和人工智能。

邓英俊（1986—），男，哲学博士，天津大学应用数学中心讲师，研究方向为数据驱动的预测性维修。

SecMVX：多变体执行脆弱性分析

李秉政[1]，张铮[1]，王晓梅[1]，曲晟[1]，邬江兴[2]
[1]信息工程大学 数学工程与先进计算国家重点实验室，郑州 450001；
[2]国家数字交换系统工程技术研究中心，郑州 450002

摘　要：作为一种主动防御技术，多变体执行（multi-variant execution，MVX）通过并行运行的异构执行体之间一致性检查发现攻击行为，相较于补丁式的被动防御，可在不依赖攻击特征信息的情况下防御已知漏洞乃至未知漏洞威胁。然而该技术在实际部署中，采用软件多样性技术组合生成的变体集合在并行执行时容易引入新的漏洞。首先从形式化描述角度对多变体执行理论安全性进行分析，其次总结针对多变体执行攻击的一般形式和攻击利用技术，分析变体生成技术组合产生新的漏洞原因，提出SecMVX，一种安全多变体执行架构及变体生成技术。基于CVE漏洞和SPEC 2006基准测试集的实验评估表明，SecMVX引入了11.29%的平均时间开销，在保证原有多变体执行安全防御有效性基础上避免了变体生成技术组合不当造成的漏洞。
关键词：多变体执行、软件多样性、网络空间安全

SecMVX: Analysis on the Vulnerability of Multi-Variant Execution

LI Bingzheng[1], ZHANG Zheng[1], WANG Xiaomei[1], QU Sheng[1], WU Jiangxing[2]
1.Information Engineering University State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China;
2.China National Digital Switching System Engineering and Technological R&D Center, Zhengzhou 450002, China

Abstract: As an active defenses technique, multi-variant execution(MVX) can detect attacks by monitoring consistency of heterogeneous variants with parallel execution. Compared with patch-style passive defense, MVX can defend against known and even unknown vulnerabilities without relying on attack feature information. However, Variants generated by the combination of software diversity techniques will introduce new vulnerabilities when they execute in parallel. First, we analyze the security of MVX theory from the perspective of formal description. We summarize the general forms and attack techniques for performing attacks against MVX, analyze the causes of new vulnerabilities arising from the combination of variant generation technologies. We propose SecMVX, a secure MVX architecture and variant generation technology. Experimental evaluations based on CVEs and SPEC 2006 benchmark show that SecMVX introduces 11. 29% of the average time overhead, and avoids vulnerabilities caused by improper combination of variant generation technologies while keeping the defensive ability of MVX.
Key words: multi-variant execution; software diversity; cyberspace security

1　引言

系统级软件开发过程中，由于编程语言（如C和C++等）的局限性和软件开发者的安全知识缺乏等因素，软件不可避免地出现诸如缓冲区溢出、悬空指针和内存泄漏等漏洞。同时，在软件开发过程中，使用第三方库、借鉴开源代码等代码复用方式虽然能够显著提升开发效率，但也带来了软件同质化的问题。OpenSSL是一个开放源代码的软件库包，被广泛应用在互联网程序和网站上，其加密代码中一种名为Heartbleed的严重安全漏洞（CVE-2014-0160）曾引发重大影响。相同的漏洞

基金项目：国家重点研发计划资助项目（Grant No. 2018YF0804003，Grant No. 2017YFB0803204）
通讯作者：张铮（ponyzhang@163.com）

经代码复用方式被广泛传播，给攻击者提供了便利，并给网络安全带来极大风险。

研究人员提出了多种措施来防御软件漏洞。针对控制流劫持攻击，早期研究提出了栈[1]、数据不可执行（Data Execution Prevention， DEP）[2]、地址空间布局随机化（Address Space Layout Randomization， ASLR）[3]等防御方法，能够有效抵御代码注入攻击，但容易被return-to-libcReturn[4]、面向返回编程（Return Oriented Programming，ROP）[5][6]等代码重用攻击手段绕过。控制流完整性防御（Control Flow Integrity， CFI）[7]在运行时严格按照预先定义的控制流图限制程序控制转移，能够抵御代码注入、代码重用攻击和内存信息泄露，但难以在性能和安全性两者间找到平衡。面向数据编程（Data-oriented Programming， DOP）[8]采用非控制数据攻击方式实现图灵完备的攻击，可绕过细粒度的控制流完整性防御。攻防对抗双方此消彼长，被动式防御方法往往容易被新的攻击绕过，无法应对未知漏洞的威胁。

Cox[9]于2006年首次提出将多变体执行（Multi-Variant Execution， MVX）用于解决软件安全问题。作为一种主动防御方式，多变体执行使用软件多样性技术生成变体集合，将程序输入分发至多个功能相同，结构不同的变体并行执行，并设置检查点，通过比较发现变体执行状态的不一致来检测攻击。攻击者必须在不触发表决检测的情况下同时对多变体执行架构中所有变体实施攻击。

近年来，对多变体执行研究主要围绕多变体执行安全防御架构机制的实现以及安全性和性能平衡等方面展开。Stijn等人[10]基于GHUMVEE[11]和ReMon[12]多变体执行架构，加入线程时钟同步并行机制，采用锁步的方式对多线程中部分系统调用排序执行，解决了多变体执行中线程调度和资源争用问题，首次在软件多变体执行架构中支持多线程并行。Österlund S等人[13]首次将软件多变体执行防御技术应用于操作系统内核，将内核内存空间分区，采用地址空间布局随机化来构造两个内核多变体。在执行系统调用时两个内核同时处理，对执行结果进行同步检查以判断是否存在内核内存泄露。Voulimeneas A[14]首次提出了分布式ISA异构平台DMON的多变体执行架构，实现指令集异构执行，并利用arm和x86不同的硬件安全机制，进一步提升多变体执行安全性，但存在通信成本和性能损耗大的问题。与之类似的HeterSec[15]在异构的ISA上实现多变体执行。不同于DMON基于ptrace在用户层对变体进程进行监控和裁决，HeterSec基于Linux内核修改，增加内核动态加载模块支持用户层分布式变体进程的同步和通信。姚东等人[16][17]对多变体执行相关研究进展进行了全面综述，包括变体生成机制、监视器实现机制等关键技术的研究，并提出MVX-CFI，一种基于多变体执行架构的CFI，提高了MVX的执行性能。拟态防御[18][19]是我国科研团队提出的针对未知漏洞后门的主动防御技术，核心思想是动态异构冗余（Dynamic Heterogeneous Redundancy， DHR），相比MVX，拟态防御的动态性和反馈机制使拟态防御系统具有内生的测不准效应，在裁决机制上更加完善。

目前，研究人员主要通过建立威胁模型，以及基于相关漏洞进行攻击验证的方式对多变体执行防御效果进行评估[20]，存在以下不足：一是建立威胁模型时假设条件不充分，对多变体执行理论安全性分析论证不够完备，二是对于多变体执行设计实现中的缺陷讨论不足。变体生成技术的简单组合是否会带来安全增益？多变体执行中是否变体数量越多越安全？本文研究表明多变体执行架构并非牢不可破，并列举了几种可能的攻击利用方式。实际部署中，需要有效组合变体生成技术才能构造安全的多变体执行，发挥其防御优势。具体来说， 本文完成了以下几个方面的工作：

从形式化描述角度对多变体执行理论安全性进行分析；

总结针对多变体执行攻击的一般形式，分析了四种具体的攻击利用技术；

针对攻击利用技术，提出了SecMVX，一种安全的多变体执行架构中变体集合的构造方法；

对SecMVX进行安全性评估和性能开销评估。

2 多变体执行研究概述

2.1 多变体执行形式化描述

由一组功能相同但结构不同的变体并行运行的技术，称之为多变体执行。软件漏洞的本质是

一系列程序逻辑错误引起的程序状态非正常转换（如指针越界，缓冲区溢出等错误），这种错误可以为攻击者所非法利用，从而在未授权的情况下访问或破坏系统。在软件安全领域，多变体执行技术将给定输入分发给所有变体同步执行。假设该程序是功能完备的，则对于正常的输入，不会发生程序状态的非正常转换，能够产生一致的输出；而对于非法输入，由于变体程序结构发生了变化，其程序状态的非正常转换会导致不同变体之间的程序状态不一致，即产生不一致的输出结果。通过合理设置检查点，对多变体输出结果或中间状态的标志进行比较，即可检测攻击行为。

定义检查点集合：$[D_0, D_1, D_2, ...]$，程序运行到检查点处时（如I/O系统调用），触发对程序状态的检查。将程序执行定义为一组状态序列转换集合：$\left[S_0, S_1, S_2 \cdots\right]$。假设变体数为3，多变体执行过程中，在检查点$D_i$处多变体程序的状态表示为：$< S_i^1, S_i^2, S_i^3 >$。从检查点$D_0$到$D_i$，多变体的执行表示为：$[< S_0^1, S_0^2, S_0^3 > , < S_1^1, S_1^2, S_1^3 > , < S_2^1, S_2^2, S_2^3 > ... < S_i^1, S_i^2, S_i^3 >]$。检查点处的状态序列是整个多变体执行状态序列的子集。

在检查点处的状态序列，不同的变体进程$P^1, P^2, P^3, \cdots$状态语义上等价，非语义属性则多样化。假设进程的语义与内存布局无关，则内存布局为非语义属性，可对其进行多样化处理。定义映射函数$Map()$表示在检查点D_i处从变体进程集合$[P^1, P^2, P^3, \cdots]$的状态$< S_i^1, S_i^2, S_i^3, \cdots >$到原始进程$P^0$的状态$S_i^0$的映射。定义转换函数$T^j$和$T^o$，其函数输入为某检查点处变体进程$P^j$和原始进程$P^o$的输入和状态，其输出为程序执行的下一个状态。最后一个检查点的状态代表程序执行的最终状态。

正常执行下的等价性。在多变体执行正常执行的情况下，可以得出以下结论：

$$Map\left(S_0^1\right) = Map\left(S_0^2\right) = \cdots = Map\left(S_0^n\right) = S_0^o \#(1)$$

$$\forall 0 \leqslant i \leqslant N, \forall I \in Normalinputs:$$

$$Map\left(T\left(S_i^1, I\right)\right) = Map\left(T\left(S_i^2, I\right)\right) = \cdots = T\left(S_i^o, I\right) \#(2)$$

假设多变体执行在每个检查点进行同步，上述等式表示正常执行下的等价属性。

等式1表明所有变体都处于等价的初始状态，可以直接映射到原始进程的状态。

等式2表明在给定正常输入的情况下，所有变体中转换函数在每个检查点都与原始进程具有等价的映射关系。

等式2还表明所有变体产生的输出与原始进程相同，从而确保了语义上的等价。

攻击检测：假设多变体执行的初始状态不受攻击，则始终满足等式1。多变体执行系统中，需要构造异构性，使得当攻击者试图对程序发起攻击时，多变体映射关系不满足等式2。公式3表明，如果在检查点D_i之后发生控制流劫持、内存泄露等攻击行为，则由于异构属性导致等式2失效，检查点D_i处多变体状态序列$< S_i^1, S_i^2, S_i^3, \cdots >$的状态必不同。每当多变体状态出现不一致时，即检测到攻击行为。

$$\forall 0 \leqslant i \leqslant N, \forall I \in Inputs:$$

$$\exists 1 \leqslant j \leqslant n, T\left(S_i^j, I\right) \in Attacks \Rightarrow$$

$$Map\left(S_{i+1}^j\right) \neq Map\left(S_{i+1}^k\right), k \neq j \#(3)$$

2.2 变体生成技术

多变体执行通过变体生成技术构造异构性，使遭受攻击的多个变体程序状态产生不一致，从而检测攻击行为。软件多样性[21]是变体生成技术的基础。软件多样性技术通常通过随机化手段，破坏某种类型攻击的假设条件，而对该攻击具有防御效果。软件多样性可以与多变体执行相结合用来构造变体异构性。应用软件多样性技术可以从不同的粒度上生成变体。根据多样性技术分类，常见的变体生成技术有：

1. 指令级多样性：包括指令集异构和指令集随机化两种技术。指令集异构[14][15]一般采用分布式技术在两种不同的指令集架构上（例如，x86_64和aarch64）运行多变体。由于不同指令集架构的体系结构，可执行文件格式、内存空间布局均不同，攻击者无法利用程序漏洞实施攻击，指令集异构的多变体执行同时能防御针对微架构漏洞的攻击。指令集随机化[22][23]则对机器指令进行加密，在解析执行时进行解密，单进程执行时加密算法存在被攻击者破解的风险。多变体执行中则采用不同密钥加密机器指令，使得攻击者无法注入有效代码。

2. 内存级多样性：包括ASLR[3]、栈布局随机化[24]、堆布局随机化[25]、数据随机化[26][27]等技术。ASLR在运行时支持随机化堆和栈的基地址，结合-fPIE编译选项生成位置无关代码，可随机化

代码段及数据段基地址，但不会随机化每个内存段内的内容。堆栈布局随机化对堆栈内的数据变量重新排布，粒度更细，引入数据和代码填充则可进一步增加多样性，同时能够消除ROP攻击依赖的gadget代码[28]。但该技术需要编译器的支持。数据随机化包括数据布局随机化[26]和XOR数据随机化[27]等技术，使攻击者无法读取存放在内存的真实数据。

3. 函数级多样性：包括系统调用号随机化[29]、库函数映射随机化[30]、函数参数随机化[31]、函数块随机化[32]等技术。单独使用系统调用号随机化时系统调用号随机空间较小（Linux-4.19.104中系统调用共436个），容易被攻击者爆破，而多变体执行中不同变体的系统调用号不同，攻击者无法发起有效的系统调用。库函数映射随机化在程序载入时将库函数映射到随机的虚拟地址，需要操作系统支持。函数参数随机化由编译器支持，打乱参数的顺序，使攻击者无法使用函数。函数块随机化增大了ROP攻击的难度。

若变体生成技术d在单独使用时将攻击成功的可能性减小到概率P，则从理论上计算，使用n变体生成技术，能将攻击成功的概率减小到P^n。通常情况下，组合使用多种变体生成技术能进一步提高攻击者的门槛。然而，不恰当的变体生成技术组合在多变体执行环境下反而会破坏安全假设，引入新的攻击面。本文第三章讨论针对多变体执行的攻击，分析变体生成技术的组合使用中存在的攻击面暴露问题，第四章提出构建更安全多变体执行的方法。

3 针对多变体执行的攻击分析

基于上述多变体执行架构防御原理可知，攻击者对多变体执行架构实施攻击的前提是每个变体存在可利用的漏洞。如图1所示，程序中存在栈溢出漏洞，攻击者可从内存对象A越界覆盖写入内存对象C。在多变体执行架构中，采用内存随机化的变体生成技术生成三个内存空间布局不同的变体，其中变体3的内存对象C位于内存对象A低地址处，则攻击者无法对变体3以及包含变体3的多变体集合实施攻击。然而，在由变体1和变体2组成的多变体执行系统中，虽然内存对象C的地址不同，但都位于内存对象A的高地址处，攻击者仍可能发起以下攻击：

1. 并行攻击：攻击者构造输入从内存对象A越界连续写入2个偏移的目标数据，同时覆盖两个变体的内存对象C。

2. 串行攻击：分别执行两次攻击，第一次攻击越界一个单位覆盖变体2的内存对象C，第二次攻击越界2个单位偏移覆盖变体1的内存对象C。

上述攻击将导致内存对象B被覆盖写入，因此攻击成功的前提是对内存对象B的写入不会触发检查。本章基于上述两种攻击类型，分析攻击者常用的攻击原语，结合具体的内存漏洞类型分析4种可能的攻击利用技术。

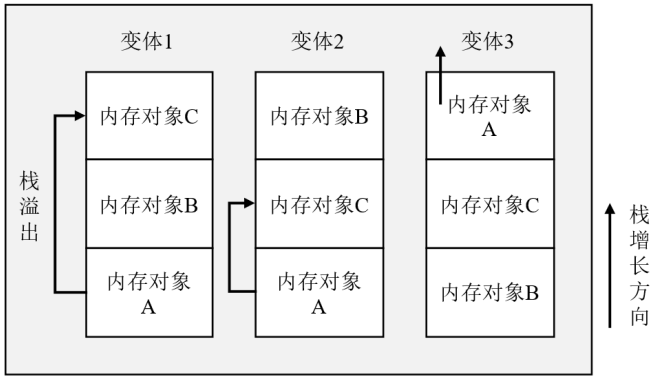


图1 多变体执行栈布局示意图

3.1 攻击原语和攻击目标

如图2所示，攻击者常用的攻击原语有：

1. **向上溢出：**构造超长数据，超出缓冲区边界写入上级缓冲区。上级缓冲区存放的数据、上一条指令的指针或其他程序的输出内容，可被攻击者恶意篡改。

2. **向下溢出：**构造超长数据，触发缓冲区向下溢出，下级缓冲区存放的是下一条指令的指针或其他程序的输出内容，可被攻击者恶意篡改。

3. **偏移写入：**攻击者可在距某个内存对象任意偏移的位置写入任意值。

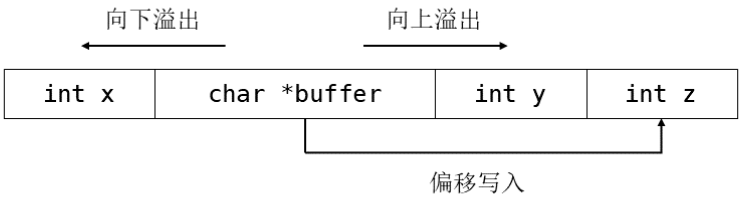


图2　攻击原语示意图

攻击原语通过非法读写内存中的目标数据，触发程序状态改变，获取权限、泄露内存数据或劫持程序并执行恶意代码。可能发生攻击的程序状态变化是多种多样的。以下几种类型的状态改变标志成功实施了攻击：

1. **数据篡改：**将选定的值写入选定的对象，例如，将值5写入creds-> auth_level，实现提权操作。

2. **函数劫持：**将选择的函数的地址写入特定对象，例如，将&grant_access函数地址写入返回地址，实现任意代码执行。

3. **数据属性：**在某些情况下，攻击者构造的数据仅需满足某些属性即达到目的。假设32位字段creds->is_admin用作标志。将任意非零值写入creds-> is_admin即实现提权操作。

实际情况下，攻击者面临诸多限制。例如，由于程序对输入数据格式检查，攻击者必须写入既是函数地址又是有效ASCII字符串的值；由于内存布局限制，攻击者只能使缓冲区溢出固定字节个数，避免非目标数据被篡改导致程序崩溃；由于内存变量限制，攻击者只能写入双字大小的值。

3.2　攻击利用技术

本节讨论针对多变体执行的攻击利用技术。在特定的攻击样例中，组合使用不同的变体生成技术反而会引入新的攻击面。

3.2.1　偏移返回攻击

如图3所示，内存对象A存在溢出，攻击目标是覆盖程序的返回地址，将当前指向func1的返回地址改为指向func2，完成控制流劫持。多变体执行架构中，ASLR使得func2在三个变体的虚拟地址都不同。变体3采用栈布局随机化技术，其栈布局与其他两个变体不同。

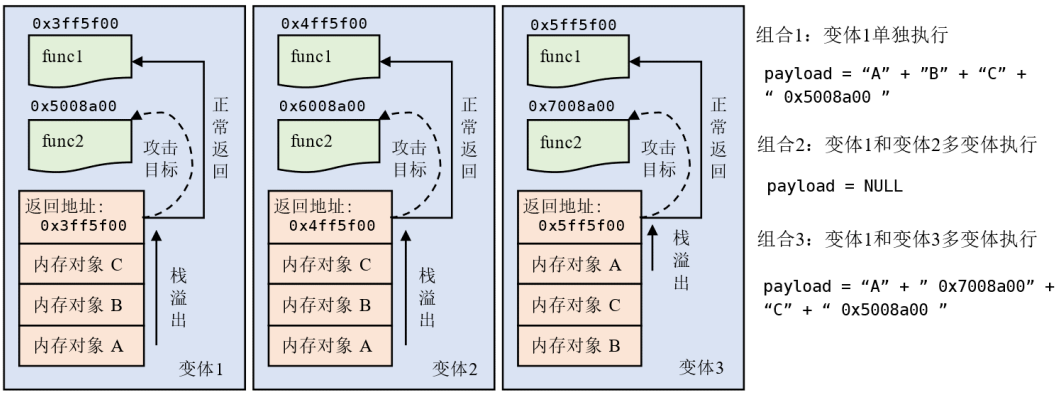


图3　偏移返回攻击示意图

变体组合2中，变体1和变体2的func2映射到不同的地址，栈中返回地址与两个变体中的内存对象A距离为三个单位偏移。由于任意一次攻击只能使用单个有效payload覆盖相同的值到两个变体的返回地址中，攻击者无法同时篡改两个变体的返回地址为func2。该变体组合中，对地址空间

布局进行随机化，而不进行栈布局随机化，可防御基于栈溢出的控制流劫持攻击。

变体组合3中，变体3增加了栈布局随机化，在多变体执行架构中可能引入新的攻击面。变体1和变体3的内存对象A与返回地址相对偏移不同，可以构造单个有效payload，将对应的func2地址写入变体1和变体3的返回地址，实现对多变体执行的并行攻击。上述变体组合中，增加了栈布局随机化的多变体执行反而引入了控制流劫持攻击，我们称之为“偏移返回攻击”。同理，堆布局随机化也会引入新的攻击面。

3.2.2 偏移数据攻击

数据随机化技术在数据写入内存时，与掩码进行异或运算，隐藏真实数据的意义。在内存读取时，将加密数据与掩码进行异或运算还原数值。多变体执行系统中攻击者非法写入的数据与不同变体的掩码进行异或运算还原得到的数值不同，进而产生不一致。然而，在特定情形下，攻击者仅需写入随机值即可满足数据属性，改变程序的状态，实现提权等操作。

假设程序使用整数作为布尔标志，则除零以外的任何值都为“True”。分支在正常运行时判断为“False”，则写入任意非零的随机值即可篡改该分支。攻击者并不需要完全控制多个变体的数据一致，仅需满足某些数据属性即可实现攻击。

攻击者使用随机值覆盖写入存储缓冲区长度变量，大概率会增大缓冲区边界。增大的缓冲区边界将导致进一步的内存破坏攻击。我们称此类攻击为偏移数据攻击。偏移数据攻击会导致程序出现“数据属性”状态改变。交叉检查可防御此类攻击，第4章中将阐述针对偏移数据攻击的防御技术。

3.2.3 数据填充攻击

多变体执行中，由于不同变体的内存布局不同，一次payload将相同数据写入不同变体非对应的内存区域，极大概率会触发表决。当攻击目标为篡改程序用户名时，不同变体的用户名变量地址不同，覆盖一个变体中的用户名的同时可能覆盖另一个变体中的函数指针，当该函数指针被调用时攻击行为即被发现。

数据填充可随机化内存布局，但是在变体中插入数据填充（无意义的0x00）可能会引入新的攻击面。若攻击者payload覆盖变体1目标对象的同时恰好覆盖其他变体中无效的数据填充，则可在不触发检查点检测的情况下独立于其他变体来攻击变体1，进而可通过串行攻击每个变体来攻击整个多变体执行系统，例如在触发程序访问某个函数指针之前依次重写每个变体中的函数指针。

图4所示的多变体执行系统中，与图3不同之处为栈内只有1个内存对象，引入数据填充对变体进行栈布局随机化处理。与3.2.1节分析的攻击类似，利用栈溢出漏洞，可以成功进行攻击。然而在本节样例中，攻击者写入的返回地址覆盖的是变体1中的无效填充。数据填充削弱了该多变体执行系统的防御能力。

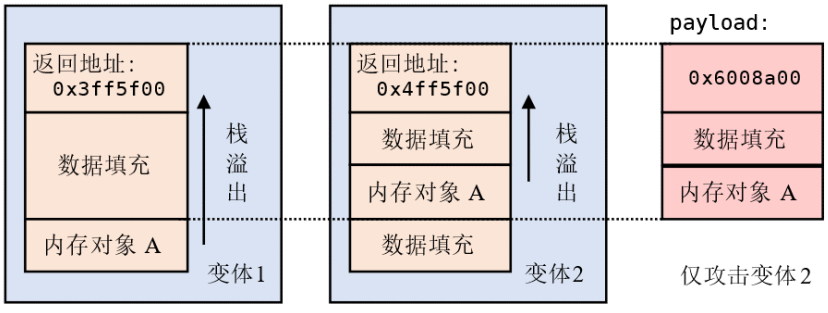


图4 数据填充攻击示意图

3.2.4 SafeStack支持的跨栈攻击

现有的栈保护技术（如SafeStack[33]）将栈分为两个区域，由clang编译器通过静态分析出控制流信息和需要保护的关键数据，并将其移动到“安全栈”，其他变量位于“非安全栈”，实现关键数据的隔离。攻击者无法从非安全栈溢出到安全栈，以此来限制栈缓冲区的溢出，保护栈上的关键数据。

在多变体执行中应用SafeStack等栈保护技术可能引出新的攻击利用技术。应用SafeStack之后，非安全栈上的内存对象与上一次函数调用中的非安全栈对象相邻，而基址指针和返回地址位于安全栈上。攻击者可以从非安全栈溢出覆盖到上一次函数调用的非安全栈，不会因改写基址指针和返回地址而触发表决。

4 探索更安全的多变体执行

针对多变体执行的攻击案例表明，对变体生成技术的不当使用会给攻击者创造新的攻击面。本章讨论如何构造安全的多变体执行系统。由于多变体执行中变体并行执行，并且主流CPU均为多核处理器架构，本文将随机化与确定性布局相结合，提出名为SecMVX的安全多变体执行及变体生成技术，可有效避免3.2节中所述攻击，以防御未知漏洞的攻击。

4.1 安全多变体执行架构

图5所示为变体集合构造示意图。变体1和变体2采用ASLR重定位栈基地址，堆基地址、数据段基地址和代码段基地址；两个变体具有不同的数据随机密钥。变体内部框表示变体内存段内相对布局。变体1和变体2相同颜色的内存段相对地址偏移是相同的。在单个内存段中（栈，堆或代码段），不同变体的内存对象A相同的偏移量溢出必然落在同一目标对象。该组合使得攻击者无法发起3.2.1节中所述偏移返回攻击。

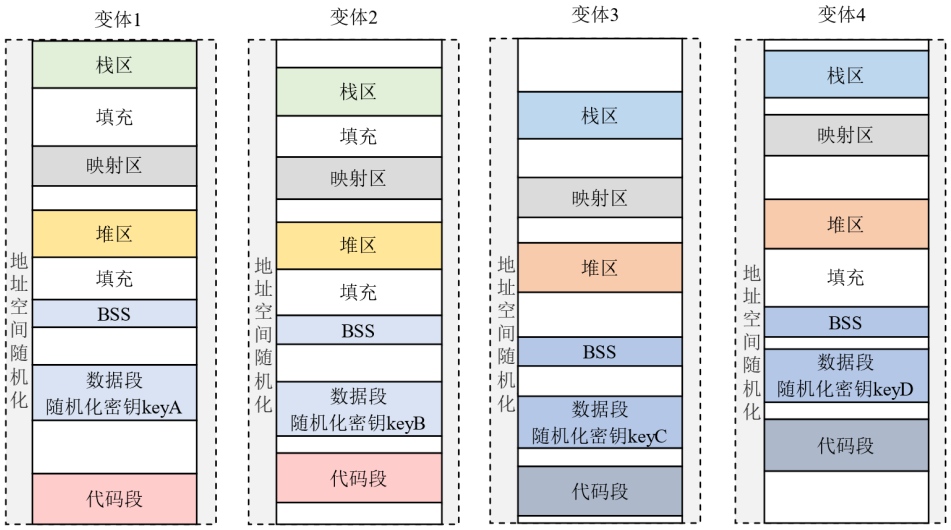


图5 安全多变体组合示意图（ASLR=2，开启PIE编译选项）

如图5所示，增加变体3和变体4，与变体1和变体2一样开启ASLR和数据随机化，内存段内相对偏移固定，采用多样化编译方法，使变体3和变体4堆、栈和数据段内布局与变体1和变体2不同（例如，反向栈、栈布局随机化和堆布局随机化等），进一步提高多变体系统防御缓冲区溢出漏洞的能力。

为防御第3.2.2节中分析的偏移数据攻击，在变体生成过程中对数据随机化增加交叉检查机制，将关键数据（如条件分支判断变量、用户权限变量等）传递给表决器，检查每个变体中的数据值是否一致。假设内存变量is_admin为条件分支判断变量，is_admin=0为普通用户，非零值代表具有管理员权限，攻击者向所有变体写入非零值即可劫持该分支，获取管理员权限。启用交叉检查机制后，相同的值写入所有变体is_admin变量的内存地址处，读取时和不同变体的密钥进行掩码运算后会得到不同的非零值传递给表决器，即检测到不一致。

4.2 实现与实验评估

本文基于MultiCompiler[34]，采用4.1节所述安全多变体执行架构改进了变体生成技术，并在MVEE[34]多变体执行环境中构建了SecMVX。SecMVX的变体生成流程如图6所示，Clang将源代码编译为LLVM IR中间表示，由LLVM Passes在中间层进行堆栈布局随机化和数据随机化的转

换，生成两组孪生变体集合的IR，进而由Clang编译生成二进制文件，并在MVEE多变体执行环境中启用ASLR，构建SecMVX。

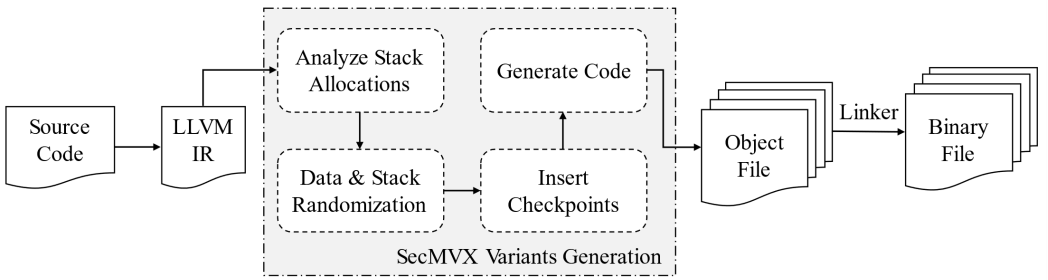


图6　SecMVX变体生成流程图

4.2.1　安全性评估

本文选取开源程序中CVE漏洞对SecMVX进行测试，评估SecMVX的防御效果。表1为相关漏洞的详细信息及测试结果。前四个CVE漏洞在MVEE多变体执行环境中均进行过测试，在两个变体进程并行执行环境中能检测攻击行为，SecMVX同样能防御此类漏洞，这表明SecMVX与MVEE具有相同的防御能力。

表1　基于真实漏洞的评估结果

CVE ID	软件版本号	漏洞类型	漏洞代码	测试版本	MVEE	SecMVX
CVE-2013-2028	Nginx 1.3.9- 1.4.0	栈溢出	http/ngx_http_parse.c	1.4.0	Yes	Yes
CVE-2010-4221	ProFTPD before 1.3.3c	栈溢出	netio.c	1.3.1	Yes	Yes
CVE-2012-4409	mcrypt before 2.6.8	整型溢出	extra.c	2.6.8	Yes	Yes
CVE-2014-0749	TORQUE 2.5.x	栈溢出	lib/Libdis/disrsi_.c	2.5.13	Yes	Yes
CVE-2017-13089	wget before 1.19.2	整型溢出	src/http.c	1.19.1	Yes **I/O syscall error**	Yes **Segmentation fault**

wget网络文件下载工具存在栈溢出漏洞，编号为CVE-2017-13089。由于源码中fd_read函数将HTTP响应数据（用户输入）写入dlbuf时并未检查长度参数contlen是否为负数，攻击者可构造特定long长整型负数值，覆盖写入缓冲区，引发栈溢出漏洞。该漏洞可用于实施控制流劫持攻击。MVEE中采用栈布局随机化和ASLR生成的两变体进程虽然能够在I/O输入输出系统调用中检测输出不一致，但攻击者仍可通过3.2.1节所述偏移返回攻击篡改两个进程的栈内返回地址。而SecMVX则在攻击者试图劫持控制流时即导致孪生变体进程中一个进程发生段错误，从而检测到攻击行为。这表明SecMVX相较MVEE进一步减小了程序漏洞的攻击面。

进一步，我们采用ROP漏洞gadget检测工具

图7　wget程序漏洞代码

```
1                                    //vulnerable function
2                                static bool
3          skip_short_body (int fd, wgint contlen, ...)
4                                    {
5              enum {SKIP_SIZE = 512, ...};
6              wgint remaining_chunk_size = 0;
7      char dlbuf[SKIP_SIZE + 1]; //vul buff
8              while (contlen > 0 || chunked)
9                                    {
10     remaining_chunk_size = strtol (line, &endl, 16);
11                                   //string->long error
12     contlen = MIN (remaining_chunk_size, SKIP_SIZE);
13   ret = fd_read (fd, dlbuf, MIN (contlen, SKIP_SIZE), -1);
14                                   //memory error function
15                                   }
16                                   }
```

Ropper[35]分别扫描基准程序、MVEE多变体执行程序和SecMVX程序二进制代码，对比gadget代码的数量，评估SecMVX在防御ROP攻击时的有效性。获取所有二进制代码的gadget后，我们根据以下判定规则，对多变体执行程序中的gadget进行筛选分类：

1. 源程序二进制代码中，所有gadget均为有效gadget。

2. MVEE二进制代码中，同一个gadget在两个变体中地址均相同，为有效gadget；两个变体gadget地址不同，为无效gadget。

3. SecMVX二进制代码中，同一个gadget在四个变体中地址均相同，为有效gadget；四个变体中，任一gadget地址不同，为无效gadget。

实验采用SPEC CPU Benchmark 2006 CINT程序集部分基准测试程序，原始程序采用gcc-7编译器编译，所有程序编译选项均为-O0，实验结果如表2所示。MVEE和SecMVX均采用内存布局随机化技术生成变体程序，能够消除程序中绝大部分gadget，SecMVX为四个变体程序，相较MVEE两个变体程序，其随机空间更大；NOP代码填充进一步消除了存在的gadget。

表2　ROP gadgets测试结果

Benchmark	ROP gadgets				
	original gcc7.3.0 -O0	MVEE Clang3.8.1 -O0		SecMVX Clang3.8.1 -O0	
		count	reduction	count	reduction
bzip2	2387	37	1.5501%	6	0.2514%
mcf	1152	73	6.3368%	11	0.9549%
gobmk	47361	495	1.0452%	98	0.2069%
sjeng	3879	91	2.3460%	27	0.6961%
h264ref	14701	304	2.0679%	76	0.5170%
astar	1923	33	1.7161%	4	0.2080%

4.2.2　性能评估

为了评估SecMVX对程序运行带来的额外开销，本文采用SPEC CPU Benchmark 2006中的CINT程序集，运行3次基准程序和SecMVX多变体执行程序来测量平均时间损耗和内存开销。实验在Ubuntu18.04系统中运行，CPU型号为Intel Silver 4210，内存容量为32GB，LLVM版本为3.8.1。

如图8所示，采用几何平均数计算SecMVX的平均时间开销为11.29%。在多核环境下，与采用两变体的MVEE多变体执行相比，SecMVX性能损耗净增长幅度较小，同时获得更大的安全增益。在单个测试样例中，存在比平均时间开销高的样例429.mcf，分析该测试样例为访存密集型程序，在运行时需要最大约1.7GB的内存，其访存次数频繁，触发检查点次数远大于其他程序。内存开销方面，对程序运行时内存占用进行统计，SecMVX引入的内存开销均接近于300%。这是由于MVEE和SecMVX多变体执行均采用多进程的方式执行，因此内存开销总体上取决于变体进程的数量。

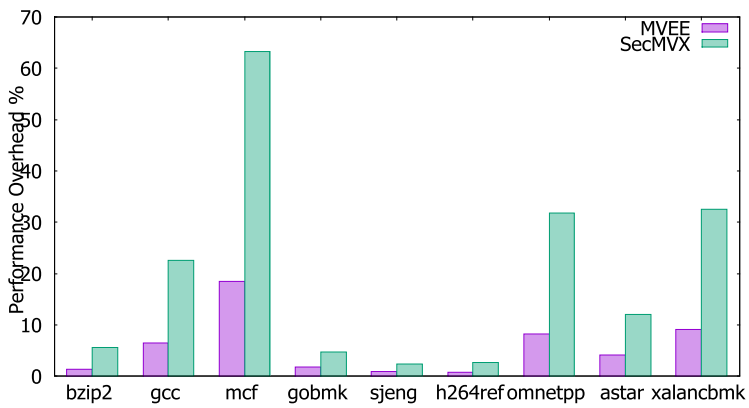


图8　SPEC 2006测试结果

5 总结和展望

本文对多变体执行中变体生成技术组合存在的安全性问题进行了分析，列举几种可能的攻击利用技术，在此基础上提出SecMVX，一种安全多变体执行架构及变体生成技术，可以防御此类攻击。实验结果表明，在一定的性能损耗下，SecMVX能够避免变体生成技术的不当组合引入的漏洞，带来更大的安全增益。

本文研究存在不足之处是，基于现有架构的SecMVX依赖于ptrace等调试方式拦截系统调用设置检查点，在安全性和系统开销的平衡方面没有很好解决；在四个变体的情况下，由于堆栈布局等对象的随机空间较小，存在将某些内存对象放置在不同变体堆栈布局相同位置的可能，在后续工作中将增加随机化检查机制。同时，多变体执行与拟态防御思想相结合来实现软件安全防御，将拟态防御的DHR构造引入到进程执行过程中，在异构冗余的基础之上进行表决，我们称之为拟态2.0，可为拟态防御研究提供新的思路。

参考文献:

[1] Cowan C, Pu C, Maier D, et al. Stackguard: Automatic adaptive detection and prevention of buffer-overflow attacks [C]//USENIX security symposium. 1998，98：63-78.

[2] Molnar I. Method and apparatus for creating an execution shield：U. S. Patent Application 10/420，253[P]. 2004-12-9.

[3] Bhatkar S, DuVarney D C, Sekar R. Address Obfuscation: An Efficient Approach to Combat a Broad Range of Memory Error Exploits[C]//USENIX Security Symposium. 2003，12(2)：291-301.

[4] Tran M, Etheridge M, Bletsch T, et al. On the expressiveness of return-into-libc attacks [C]//International Workshop on Recent Advances in Intrusion Detection. Springer，Berlin，Heidelberg，2011：121-141.

[5] Shacham H. The geometry of innocent flesh on the bone: Return-into-libc without function calls (on the x86)[C]//Proceedings of the 14th ACM conference on Computer and communications security. 2007：552-561.

[6] Roemer R, Buchanan E, Shacham H, et al. Return-oriented programming: Systems, languages, and applications [J]. ACM Transactions on Information and System Security (TISSEC)，2012，15(1)：1-34.

[7] Abadi M, Budiu M, ÚErlingsson, et al. Control-flow integrity principles, implementations, and applications[J]. ACM Transactions on Information and System Security (TISSEC)，2009，13(1)：1-40.

[8] Hu H, Shinde S, Adrian S, et al. Data-oriented programming: On the expressiveness of non-control data attacks [C]//2016 IEEE Symposium on Security and Privacy (SP). IEEE，2016：969-986.

[9] Cox B, Evans D, Filipi A, et al. N-Variant Systems: A Secretless Framework for Security through Diversity [C]//USENIX Security Symposium. 2006：105-120.

[10] Volckaert S, Coppens B, De Sutter B, et al. Taming parallelism in a multi-variant execution environment[C]//Proceedings of the Twelfth European Conference on Computer Systems. 2017：270-285.

[11] Volckaert S, De Sutter B, De Baets T, et al. GHUMVEE: Efficient, effective, and flexible replication [C]//International Symposium on Foundations and Practice of Security. Springer，Berlin，Heidelberg，2012：261-277.

[12] Volckaert S, Coppens B, Voulimeneas A, et al. Secure and efficient application monitoring and replication[C]//2016 {USENIX} Annual Technical Conference ({USENIX}{ATC} 16). 2016：167-179.

[13] Österlund S, Koning K, Olivier P, et al. kMVX: Detecting Kernel Information Leaks with Multi-variant Execution[C]//Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems. 2019：559-572.

[14] Voulimeneas A, Song D, Parzefall F, et al. DMON: A Distributed Heterogeneous N-Variant System [J]. arXiv preprint arXiv: 1903. 03643，2019.

[15] Kim S H, Ravindran B. A Framework for Software Diversification with ISA Heterogeneity [C]//23rd International Symposium on Research in Attacks，Intrusions and Defenses. IEEE，San Sebastian，Spain，2020

[16] 姚东,张铮,张高斐,邬江兴等. 多变体执行安全防御技术研究综述[J]. 信息安全学报,2020
Yao D, Zhang Z, Zhang G, et al. A Survey on Multi-Variant Execution Security Defense Technology [J]. Journal of Cyber Security，2020.

[17] 姚东,张铮,张高斐,邬江兴. MVX-CFI:一种实用的软件安全主动防御架构[J]. 信息安全学报,2020,5(04):44-54.
Yao D, Zhang Z, Zhang G, et al. MVX-CFI: a practical active defense framework for software security [J]. Journal of Cyber Security，2020,5(04):44-54.

[18] 邬江兴. 网络空间拟态防御研究[J]. 信息安全学报，2016 (4)：1-10.
WU J X. Research on Cyber Mimic Defense[J]. Journal of Cyber Security，2016，1(4):1-10.

[19] 仝青, 张铮, 张为华, 等. 拟态防御 Web 服务器设计与实现[J]. 软件学报，2017，28(4).
Tong Q, Zhang Z, Zhang WH, Wu JX. Design and implemention of mimic defense Web server[J]. Ruan Jian Xue Bao/Journal of Software，2017，28(4):883-897.

[20] Bekiroglu B, Korel B. Survivability Analysis of K-variant Architecture for Different Memory Attacks and Defense Strategies [J]. IEEE Transactions on Dependable and Secure Computing，2019.

[21] Larsen P, Homescu A, Brunthaler S, et al. SoK: Automated software diversity [C]//2014 IEEE Symposium on Security and Privacy. IEEE，2014：276-291.

[22] Kc G S, Keromytis A D, Prevelakis V. Countering code-injection attacks with instruction-set randomization［C］//Proceedings of the 10th ACM conference on Computer and communications security. 2003：272-280.

[23] Papadogiannakis A, Loutsis L, Papaefstathiou V, et al. ASIST: architectural support for instruction set randomization［C］// Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. 2013：981-992.

[24] Aga M T, Austin T. Smokestack: thwarting DOP attacks with runtime stack layout randomization［C］//2019 IEEE/ACM International Symposium on Code Generation and Optimization (CGO). IEEE，2019：26-36.

[25] Novark G, Berger E D. DieHarder: securing the heap［C］// Proceedings of the 17th ACM conference on Computer and communications security. 2010：573-584.

[26] Chen P, Xu J, Lin Z, et al. A practical approach for adaptive data structure layout randomization［C］//European Symposium on Research in Computer Security. Springer，Cham，2015：69-89.

[27] Bhatkar S, Sekar R. Data space randomization［C］//International Conference on Detection of Intrusions and Malware，and Vulnerability Assessment. Springer，Berlin，Heidelberg，2008：1-22.

[28] Homescu A, Jackson T, Crane S, et al. Large-scale automated software diversity—program evolution redux［J］. IEEE Transactions on Dependable and Secure Computing，2015，14(2)：158-171.

[29] Rauti S, Laurén S, Hosseinzadeh S, et al. Diversification of system calls in linux binaries［C］//International Conference on Trusted Systems. Springer，Cham，2014：15-35.

[30] Wartell R, Mohan V, Hamlen K W, et al. Binary stirring: Self-randomizing instruction addresses of legacy x86 binary code［C］// Proceedings of the 2012 ACM conference on Computer and communications security. 2012：157-168.

[31] Rajagopalan M, Hiltunen M A, Jim T, et al. System call monitoring using authenticated system calls［J］. IEEE Transactions on Dependable and Secure Computing，2006，3(3)：216-229.

[32] Chen Y, Wang Z, Whalley D, et al. Remix: On-demand live randomization［C］//Proceedings of the sixth ACM conference on data and application security and privacy. 2016：50-61.

[33] Chen G, Jin H, Zou D, et al. Safestack: Automatically patching stack-based buffer overflow vulnerabilities［J］. IEEE Transactions on Dependable and Secure Computing，2013，10(6)：368-379.

[34] Volckaert S. Advanced Techniques for multi-variant execution［D］. Ghent University，2015.

[35] Schirra S. Ropper-rop gadget finder and binary information tool［EB/OL］. 2014. https://scoding. de/ropper/

.

[作者简介]

李秉政（1996—），男，学士，博士生，主要研究方向为网络空间安全，主动防御技术。

张铮（1976—），男，博士，副教授，主要研究方向为网络空间安全，主动防御技术。

王晓梅（1976—），女，博士，副教授，主要研究方向为通信网络，大数据。

曲晟（1996—），男，学士，博士生，主要研究方向为网络空间安全，主动防御技术。

邬江兴（1953– ），男，中国工程院院士，教授、博士生导师，主要研究方向为网络通信与安全。

时间敏感网络安全防护关键技术研究

吴少勇，张磊，庞宏俐，骆汉光，王延松，李振廷
之江实验室，杭州 310012

摘　要：本文概述了时间敏感网络的时间同步、流量调度、高可靠性和资源管理等关键技术，分析了时间敏感网络新网络架构、特有协议以及时间敏感流量所带来的安全威胁，并针对这些风险从网络配置控制器、网络协议、数据面流量三个方面研究了安全防护方案，以增强时间敏感网络的安全性，为新型低时延高可靠业务融合网络提供安全防护指导建议。

关键词：时间敏感网络、配置控制器、时间同步、安全防护

Research on key technology of time sensitive network security protection

Wu Shaoyong, Zhang Lei, Pang Hongli, Luo Hanguang, Wang Yansong, Li Zhenting
Zhejiang Lab, Hangzhou 310012, China

Abstract: This paper summarizes the key technologies of time sensitive network, such as time synchronization, traffic scheduling, high reliability and resource management, analyzes the security threats brought by the new network architecture, unique protocols and time sensitive traffic. Then, aiming at these risks, security protection schemes has been studied through network configuration controller, network protocol, and data plane flow to enhance the security of time-sensitive networks and provide guidance and suggestions for security protection of new type of low delay and high reliability service convergence network.

Key words: time-sensitive network; centralized network configuration; time synchronization; security protection

1　引言

当今世界网络信息技术日新月异，根据全球移动数据流量预测报告显示，到2020年全球IP网络接入设备将达263亿台，其中工业和机器连接设备将达122亿台，相当于总连接设备的一半，同时高清和超高清互联网视频流量将占全球互联网流量的64%。激增的视频流量和工业机器应用，带来了大量的业务拥塞崩溃和数据分组时延。同时，许多网络应用，例如工业互联网中的数据上传和控制指令下发、远程机器人手术、无人驾驶、VR游戏等，需要将端到端时延控制在1-10毫秒，将时延抖动控制在微秒级，但传统的网络只能将端到端的时延缩短至几十毫秒。面对时延敏感性业务的迫切需求，如何从“尽力而为”到“准时、准确”的控制端到端的确定性低时延，将成为未来网络新的挑战［1］。

时间敏感网络作为IEEE 802.1标准委员会提出的新一代以太网技术，是当前低时延确定性网络实现预期发展目标的关键。通过集成多项关键技

术，对传统以太网协议进行适当修改，保证时间敏感数据传输确定性和高可靠性的同时，实现对非时间敏感数据传输的兼容。时间敏感网络主要应用于工业互联网、车载网、高端装备、移动前传网等领域，这些行业安全需求相对较高，一旦受到破坏，将可能对工业运行甚至社会公共利益造成严重损害［2］。本文首先分析时间敏感网络的架构和关键技术，并从新架构和关键技术角度分析时间敏感网络面临的新安全风险，然后从网络配置控制器、网络协议、数据面流量三个方面研究了安全防护方案，展望了时间敏感网络安全技术未来趋势。

2　时间敏感网络关键技术概述

时间敏感网络是IEEE 802.1在现有以太网标准基础上开发的一套新标准。目前时间敏感网络标准按照功能分类主要包括：时间同步、流量调度、高可靠性和资源管理，已发布详细功能分类标准如表1［3］所示。

表1　时间敏感网络已发布标准（截止2020年9月）

分类	标准编号	标准名称	状态
时间同步	IEEE Std 802.1AS-2020	时间敏感应用的时间同步	2020年01月发布
流量调度	IEEE Std 802.1Qbv-2015	调度流量的增强功能	2016年03月发布
	IEEE Std 802.1Qbu-2016	帧抢占	2016年08月发布
	IEEE Std 802.1Qch-2017	周期队列及转发	2017年06月发布
高可靠性	IEEE Std 802.1Qci-2017	流量过滤及监管	2017年09月发布
	IEEE Std 802.1CB-2017	无缝冗余	2017年09月发布
	IEEE Std 802.1Qca-2015	路径控制和预留	2016年03月发布
资源管理	IEEE Std 802.1Qcc-2018	带宽预留	2017年10月发布
	IEEE Std 802.1Qcp-2018	数据模型	2018年09月发布
	IEEE Std 802.1AB-2016	路径发现	2016年01月发布
	IEEE Std 802.1AX-2020	链路聚合	2020年01月发布
	IEEE Std 802.1Qcx-2020	连接故障管理数据模型	2020年06月发布
	IEEE Std 802.1CM-2018	前传网络配置	2018年05月发布

（1）　时间同步

时间敏感网络相关的流量调度特性需要依赖时间同步功能，时间同步误差可能带来流量调度效果的劣化甚至失效。时间同步的基本原理为：确认时钟之间的主从关系，主、从时钟之间交互同步报文并记录报文的收发时间，从时钟按照主、从时钟之间的往返总延时的一半来调整本地时间，就可以实现其与主时钟的同步［4］。

IEEE 802.1AS是基于IEEE 1588产生的二层协议标准，对二层无关机制进行精简，对二层时间同步机制进行增加，与IEEE1588相比具有更快的启动能力，可以在几秒钟内锁定并进行精准时间。

（2）　流量调度

IEEE 802.1Qbv标准采用时间感知整形（Time-aware shaper，TAS）机制，要求所有时间触发的窗口时间同步，再利用门控列表技术控制不同优先级队列的传输或等待。如图1所示，TAS整形器定义了控制时间敏感网络交换机出口处发送队列阀门的开关机制，计划流量所在队列在预定的时间窗口到达后会被放行传输，而在同个时间窗口中其他非计划流量的队列会被阻止传输，因此排除了计划流量被非计划流量阻塞的可能性［5］。

在流量调度系列标准中，还有适用于实时周期性非定长数据流量模型的抢占式机制，即IEEE 802.1bu和IEEE 802.3br，以及通过排队转发控制每个节点固定转发时延从而实现数据端到端确定性时延的机制，即IEEE 802.1Qch等。

（3）　高可靠性

IEEE 802.1Qci在入口对基于每个流进行过滤和管理，实现时间感知调度程序。主要原理是将以太网流量通过过滤策略形成不同的流，同时为这些流分别分配一个流ID，该流ID会映射一个流Gate，为不同的Gate配置不同的时间片。通过这种方案，能够在有限的时间内为需要保证传输且不能中断的业务流类别授予以太网传输介质的独占使用权，防止网络攻击和流量过载［6］。

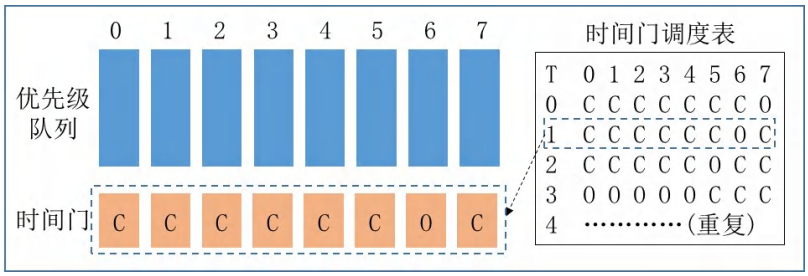


图1　时间敏感网络门控调度机制

在高可靠性系列标准中，还有通过多路冗余管理机制来实现高可用无缝冗余的IEEE 802.1CB标准，以及定义设置冗余路径的IEEE 802.1Qca标准等。

（4） 资源管理

IEEE 802.1Qcc中定义的时间敏感网络的配置模型分为全集中式配置模型、混合式配置模型以及全分布式配置模型3种，考虑到工业互联网场景下需要融合SDN体系架构，通常选择采用如图2所示的全集中式配置模型。

图中Talker、Listener分别是数据流的发送方和接收方，即工业设备或者应用系统；Bridge可以是不同形态的二层桥接设备；集中式网络配置控制器（CNC）与集中式用户配置控制器（CUC）可作为软件功能模块融合部署于专用服务器上，也可以采用嵌入式系统部署于时间敏感网络其他设备上。管理单元中CUC负责用户对网络需求的翻译及网络信息和设备配置的域间协同；CNC负责实现设备监控管理、网络拓扑发现、流量监控及调优，业务建模及调度模型下发等功能；Bridge单元除了支持时间敏感网络相关转发特性，还支持相关在线测量协议，实时将相关状态上送给管理单元［2］［7］。

时间敏感网络应用的垂直行业比较多，当前业界聚焦于根据垂直行业典型需求来完善技术细节，再进一步形成可指导时间敏感网络设备生产和方案规划部署的技术规范，如用于前传网络的IEEE 802.1CM等。

3　时间敏感网络面临的安全风险

根据IEEE 802.1Qcc定义的时间敏感网络全集中式配置模型，时间敏感网络分为应用层、控制层和网络层三层架构。时间敏感网络安全威胁可分为两类，一是基于传统网络面临的安全和威胁，如拒绝服务攻击、口令猜解、数据库漏洞利用等，但这些安全威胁的影响范围、重要程度等由于时间敏感网络的集中式控制模型而发生了变化。例如就拒绝服务而言，其对CNC所造成的危害程度远远高于其对桥设备的危害。二是时间敏感网络特有设备、协议和接口所带来的安全威胁，如对时间同步协议、时间敏感流量等自身安全漏洞造成的威胁。结合时间敏感网络的新网络架构和特有协议，时间敏感网络主要面临着网络配置控制器、网络协议、数据面流量三大共性安全风险。

（1） 网络配置控制器安全风险

在时间敏感网络中，确保集中式网络配置控制器CNC安全至关重要，因为CNC是整个时间敏感网络的控制核心，一旦受到攻击，会直接危害整个TSN网络，下述安全威胁会影响到时间敏感网络的CNC：

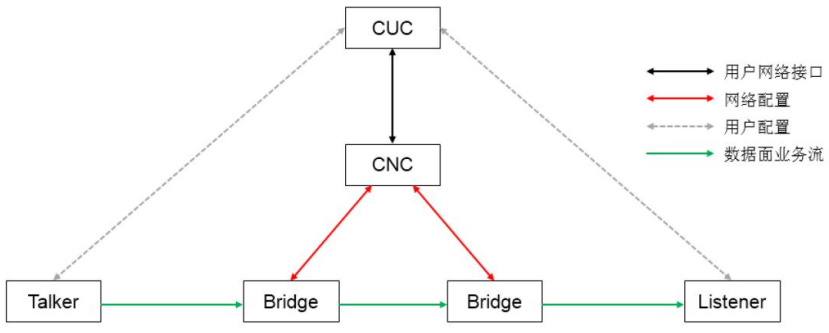


图2　全集中式配置

1）流量调度表冲突：新下发的流量调度表可能与已有的表项发生冲突，导致预先部署的流量调度表失效。

2）恶意流策略注入：攻击者可以通过劫持CUC，构造并发送一些恶意流量调度表项，实现数据窃听等恶意行为。

3）欺骗：攻击者可以通过伪装成管理员或应用程序的手段，篡改CNC上的敏感数据，获取网络拓扑结构、流量调度表等信息，甚至完全控制CNC。通过CNC地址欺骗的方式，攻击者可以伪装CNC，从而获得整个时间敏感网络的控制权。攻击者甚至可以伪装成一个时间敏感网络桥设备，对目标网络进行侦听。

4）抵赖：管理员或应用程序可以否认其曾经构造的恶意流量转发策略。

5）操作系统漏洞：由于CNC需要运行在操作系统上，因此操作系统的漏洞会导致CNC面临安全威胁。攻击者可以利用操作系统的漏洞，如默认密码，后门账户，开放端口、服务和协议等，去销毁或替换系统组件或整个系统，这样就会严重影响CNC的正常运行。

6）软件漏洞：由于CNC是以软件的形式呈现给用户，因此软件自身漏洞会导致CNC面临安全威胁。

7）信息泄露：攻击者可以得到敏感的系统信息（例如配置数据，用户证书等），进一步伪装成一个合法用户，通过CNC向网络注入伪造信息流，以获取更多的网络数据。

本章给出实例进一步描述网络配置控制器的安全风险，如图3所示，时间敏感网络的流量调度策略由CNC下发的流量调度表决定，管理员预先配置一个流量调度表项如下：从Talker1到Listener1的传输路径为绿色路径Path1（Talker1 -> Bridge4 -> Bridge1 -> Bridge2 -> Bridge3 -> Listener1），分配的时隙为t0，队列为TC5；一段时间后CNC下发新业务Talker2到Listener2，传输路径为紫色路径Path2（（Talker2 -> Bridge1 -> Bridge2 -> Bridge3 -> Bridge6 -> Listener2），分配的时隙为t1，由于CNC设计错误、恶意流策略注入、欺骗、漏洞等原因，给新业务Talker2到Listener2分配的队列同样为TC5。由于这两个业务的队列都是TC5，且Path1和Path2在Bridge1、Bridge2上有相同的出端口，在这些出端口上，如果t0时隙收到Talker2到Listener2的报文，或者t1时隙收到Talker1到Listener1的报文，均会由于Bridge传输错误时隙的业务报文，导致正常的业务流延长传输时间，不能提供确定性时延，在缓存队列满的情况甚至丢弃报文。

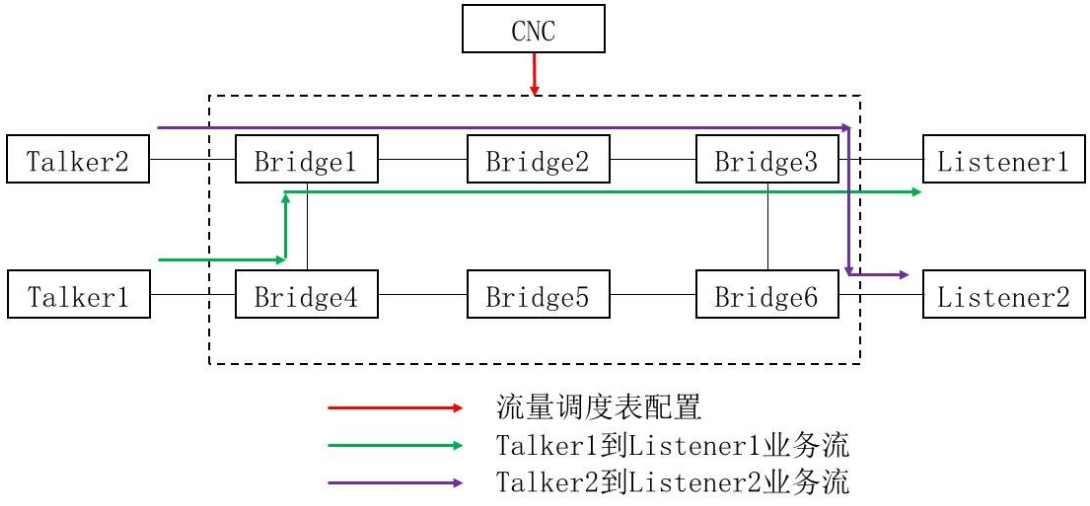


图3　网络配置控制器流量调度表失效

（2） 网络协议安全风险

1）时间同步技术（IEEE 802.1AS）作为时间敏感网络的基础，控制着整体网络内各时间感知系统之间的误差，用于保障其他时间敏感网络协议服务的正常运行，对整体网络性能至关重要。正因为时间同步在时间敏感网络中的重要地位，其遭受恶意攻击的概率和带来的后果也将显著增加。时间同步技术易受到的攻击手段主要包括

三类：

恶意时钟源攻击：时间同步协议的核心在于超主时钟，超主时钟的选举是在系统初始化时期由各端口上报自身时钟属性从而选举产生的。在此过程中，没有对上报信息进行审核验证，而恶意设备可以利用这一点，通过篡改自身时钟属性来获取超主时钟身份。一旦超主时钟被恶意设备夺取，系统时钟稳定性将受到威胁；

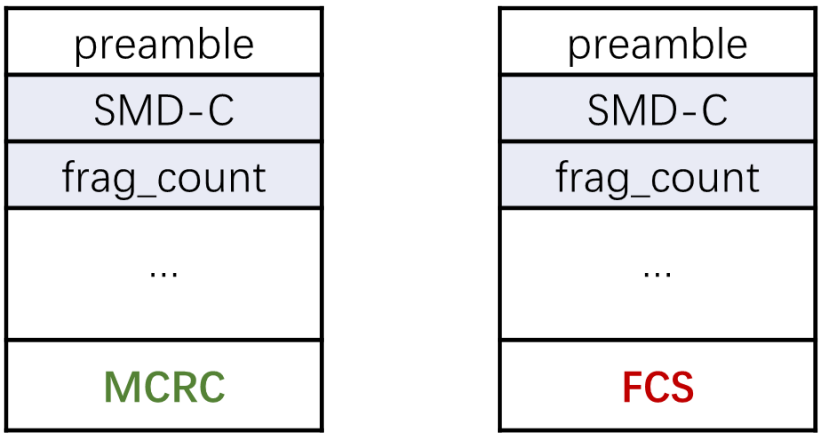
间接攻击：恶意设备接入整个网络，恶意软件通过入侵中间设备，从而拒绝服务，或者注入错误信息干扰时间同步。拒绝服务及其他攻击会引发时钟同步精度的劣化［8］；

物理干扰：通过对链路、超主时钟设备进行恶意攻击致其瘫痪，导致剩余网络内需要对超主时钟进行重新选举，造成剩余网络中时钟震荡，需要较长时间收敛。

2）IEEE802.1CB 中定义了冗余转发路径和冗余帧，因此可被用于时间敏感网络，并在数据报文层面上增强网络的高可靠性，这一机制也可以使提高时间敏感网络的可用性大大提高。

尽管该协议通过发送端指定不同的发送路径发送报文，并通过在数据帧中添加新字段来被接收方识别的方式，在报文意外丢失或数据缺损方面可以提供较大的可靠性提升，但是在针对恶意攻击的方面仍旧相对薄弱。特别是当链路意外接入恶意设备或转发路径上的设备被恶意攻击时，通过虚拟报文转发路径并发送大量伪造或篡改过的报文，从而干扰接收端正常判断。这种攻击方式将会对整个网络中报文收发造成影响。

3）帧抢占协议（IEEE802.1Qbu/IEEE802.3br）中，低优先级报文的传输将会被高优先级报文打断，从而保证高优先级报文的延时。被打断的低优先级报文会在接收端被收集，重新排序以及合并。这种协议容易面临如下两种恶意攻击手段：



a. 被抢占报文中间部分　　b. 被抢占报文末尾帧

图4　IEEE802.3br 被抢占报文格式

伪造报文：IEEE802.3br 协议报文通过 SMD+frag_count 来标识该帧在数据报文中的位置，如上图4所示，而恶意攻击能够通过伪造结束报文，即创造以 FCS 为结尾的报文来强制伪造传输结束现象。接收端收到该结尾的报文后，将开始报文拼接，得到的将是被截断的不完整报文。

截获报文：当传输路径上面的设备恶意侵占时，会阻塞转发或打乱报文转发次序，即删除帧或打乱帧到达顺序。尽管 IEEE802.3br 中规定使用 frag_count 来表示帧次序同时拥有一定的乱序接收能力，一旦被攻击设备且顺序错乱超过 3 个帧，frag_count 就无法及时修正，从而导致该被抢占报文的所有数据帧丢失。

（3） 数据面流量安全风险

TSN 促进了 IT 网络和 OT 网络的融合，以往 IT 方面遇到的网络安全威胁也将逐渐渗透到 OT 领域。时间敏感网络面临的威胁主要有两个方面：

第一，通过影响高优先级时间敏感流量的正常传输，从而影响其实时性和可靠性，恶意流量攻击是一种破坏网络服务的黑客方式，虽然具体

的实现方式千差万别，但是其表现方式有以下几种：

1）制造大流量无用数据，造成通往被攻击主机的网络拥塞，使被攻击主机无法正常和外界通信。此类攻击典型的是Smurf攻击，其通过发送伪装的ICMP数据包，将目的地址设为某个网络的广播地址，源地址设为要攻击的目的主机，使所有收到此ICMP数据包主机都将对目主机发出回应，使被攻击主机在某一段时间内收到成千上万恶意流量包，从而无法正常接受时间敏感网络数据包，导致实时业务受到影响。

2）利用被攻击主机提供服务或传输协议上处理重复连接的缺陷，反复高频发出攻击性重复服务请求，使被攻击主机无法及时处理其它正常请求。此类攻击典型的是TCP SYN泛洪攻击，使本地TCP服务长期处于半打开连接状态，导致有限的资源被消耗从而无法传输正常的时间敏感流。

第二，通过中间人攻击或者中继型攻击等方法获取OT网络的重要信息或者控制受害者，主要表现方式有以下几种：

1） 欺骗攻击，通过DNS欺骗或者ARP欺骗等手段，伪装攻击者成为目标主机，从而获得受害者的重要信息或者导致受害者的控制信息不能到达正确的目的主机，更进一步会注入恶意数据来控制受害者，从而导致生产环境无法正常工作，更加严重的，有可能导致生产环境的瘫痪。

2） VLAN中继型攻击，交换机的VLAN交互功能在一定程度上防止了广播风暴，间接保护了网络安全。在无路由的网络中，两台计算机的VLAN无法进行数据交换。而动态中继协议（俗称DTP）则主要负责两个网络之间的中继功能。若交换机中有端口能进行TRUNK，则此端口同样具备中继能力。VLAN的中继型攻击主要使用了DTP功能。攻击者主要是进行数据伪装，将假冒的DTP消息发送到要攻击的目的交换机，欺骗交换机成为一个数据中继站，而该交换机得到此消息之后随即打开802.1Q的中继功能，于是攻击者就有可能利用此交换机和特定VLAN的网络通信，从而获取特定VLAN网络里的重要信息，导致信息泄露，对网络安全造成危害。

4 时间敏感网络安全防护方案

4.1 网络配置控制器安全防护

（1） 认证

CNC对应用层、交换机、网关、端设备进行身份认证，以确保相应应用层、交换机、网关、端设备是真实的，不是伪造的。常用的身份认证机制，包括但不限于：基于用户名/密码的身份认证，基于预置密钥（pre-shared key）的身份验证，基于证书的身份验证。

（2） 流量调度表冲突检查

CNC新生成的流量调度表可能会与原有的表项相冲突，导致原有表项的失效。因此，CNC应对相应流量调度表（插入/更新/删除流量调度表）进行管理，从而避免表项冲突。

（3） 操作系统加固

操作系统加固能够最大程度消除安全风险，使操作系统更加安全。设计一系列操作，如正确配置系统和网络组建，删除无用文件，删除所有不必要软件程序，更新补丁，格式化硬盘，只安装服务器必须的功能，禁用来宾账户，重命名管理员账户等。

（4） 授权

应用层程序和管理员访问CNC时需要遵守访问控制策略。常用的访问控制机制，包括但不限于：白名单/黑名单，访问控制列表，基于角色的访问权限控制等。

（5） 软件漏洞检测

不应存在已公布漏洞，或具备补救措施防范漏洞安全风险，不应存在恶意程序，不应存在未声明的功能和访问接口。

（6） 安全管理

安全管理是指对系统平台、资源的访问控制，避免非授权使用或修改相关安全策略。安全管理可以对用户进行审计、控制错误密码尝试次数、最小化系统平台所需要的配置、强制执行操作系统的安全策略。安全管理可以对网络中的各类信息数据进行整合分析，用来支撑相应攻击检测等功能。

（7） 安全审计

CNC应提供日志和审计功能，记录控制器曾经执行过的操作，并支持安全审计。

4.2 网络协议安全防护

（1） 未知时钟源攻击

目前IEEE 802.1AS中时钟同步针对时钟源信息未经校验，恶意设备可通过修改自身时钟源参数：例如priority或时钟精度，时钟ID等，使其在超主时钟选举中胜出。此外，时间同步报文中sequence id会按照报文发送顺序逐次增长，而恶意攻击者利用通过篡改sequence id报文，可以使下游设备丢弃后续的真实时间同步报文，并对网络时间造成影响。针对此类恶意时钟源攻击，在协议中插入额外的安全信息对同步报文来源进行甄别，可以过滤掉非安全来源的时间同步报文。这种将安全信息插入报文的方法已经被应用在1588的安全防御并收录在1588v2协议附录中［9］。其安全信息由来源信息、目的信息、密钥id、随机id和计数器组成，主要的防御机制如下：

1）信息完整性检查：在报文发送端插入完整性检查结果（ICV），报文接收端根据密钥计算ICV，并将其与报文中原有的ICV对比。由于攻击者无法得知计算ICV所需的私有密钥，因此任何插入错误报文的攻击都会别甄别出来。

2）重复报文保护机制：为防止报文被恶意重复发送，安全信息中的随机id和计数器将发挥重要作用，即为安全来源id中的时间同步报文维护一个计数器，每次消息来到时加二，任何低于或等于该计数器值的报文将会被丢弃。

此外，1588v2安全协议还将对安全信息中的密钥id进行鉴别，保证各报文中的密钥id一致，同时对即将过期的密钥id进行刷新。

（2） 同步报文发送干扰探测

IEEE 802.1AS中，由于恶意设备接入网络，或者网络设备被入侵导致同步消息被滞留、劫持或发送错误报文，这些攻击行为会引发下游网络中时间同步信息发送不及时，影响同步报文正常发送周期，进而导致本地系统时钟长时间震荡而无法收敛到预期范围内。

根据该攻击的行为模式，可以采取如下防范措施：无论插入或是劫持报文，均可以简单地归类为改变同步报文发送间隔。因而这种攻击可以通过网络中时钟同步报文流量监控和分析来探查出网络异常［10］：

使用QoS监控网络中时钟同步报文流量，并送入分析模型识别异常。Allan方差在IEEE 802.1AS附录中被用于本地时钟精度优劣的评判［4］，该模型也可以作为评判延时异常的工具。通过将时间同步系统报文流量监控与长期统计数值相结合，就能敏锐识别出系统在时延方面的异常改变。

除Allan方差模型之外，也可以采取置信区间的统计学方法，通过判断网络延时落在置信区间外的概率情况来判断网络中时间同步报文是否异常。

（3） 物理层面攻击防御

破坏超主时钟、网络链路或关键节点的桥设备，引发时间同步网络内转发路径的重新规划或超主时钟重新选取，将会对高可靠、高稳定性网络构成严重威胁。因此，IEEE 802.1AS-Rev中引入超主时钟“热备份”技术［11］，如图3所示，其中红色链路为正常转发路径，GM1为超主时钟；绿色链路为冗余备份链路，GM2对应设备为备份超主时钟：

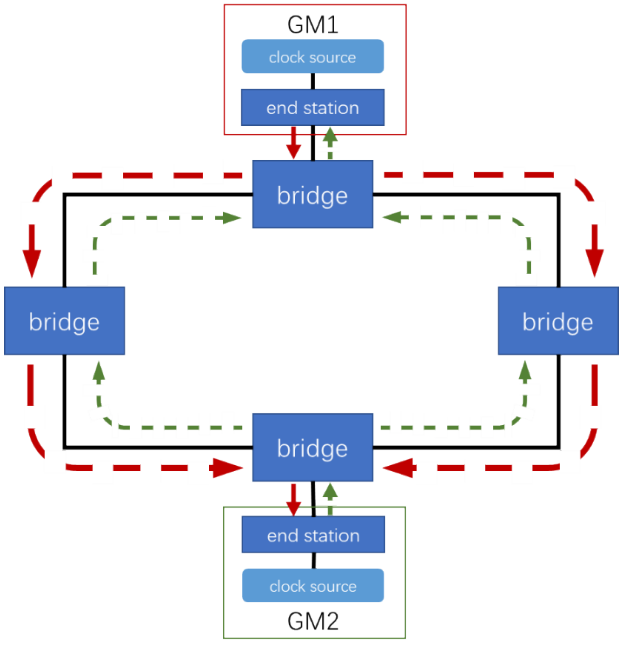


图5 冗余时钟链路

IEEE 802.1AS-Rev中说明时间同步网络中可以存在多个子域，同一设备可同时在不同子域的不同转发路径中。这种“冗余超主时钟+冗余链路”技术带来的优点是，当图示中的超主时钟故障后，可立即使能另一子域中的超主时钟，避免

因再次选举造成时间敏感网络中各设备时钟震荡而引入的系统稳定性问题。

（4） 针对恶意设备发布虚假,篡改报文攻击

此种攻击可出现在时间同步中模拟虚假同步消息，也可出现在IEEE802.3CB和IEEE802.1Qbu中仿冒报文格式破坏接收端消息完整性，主要特征是通过被侵占的设备或意外接入的恶意设备来仿冒虚假报文。针对这种攻击采取在端口与端口之间建立可信链路的方案来解决：在链路建立之前，为相邻端口增加特定参数，通过添加白名单的方式识别安全设备。

4.3 数据面流量风险安全防护

时间敏感网络协议族中的IEEE 802.1Qci［6］类似防火墙的机制，它可以对转发前的数据进行筛选和过滤，对特定标识的数据帧加以控制。此协议在数据链路层就对数据进行了处理，对比于以前基于高层的安全防护机制，效率更高，对正常时间敏感网络流的影响更小。

IEEE 802.1Qci定义的PSPF（Per-stream filtering and policing）机制主要分成4个模块：模块1根据流特征来标识一个流，确定流的Stream ID，流特征包括VLAN，目的MAC和源MAC等信息；模块2通过标识出来的流的Steam ID找到对应的流过滤表项，指定门控表索引Gate ID，和流量表索引Meter ID，并且提供帧统计功能和最大包过滤功能，最大包过滤功能可以通过设置Maximum SDU size来过滤掉超过指定包大小的数据包；模块3会根据模块2指定的门控表索引来查找门控表项，门控表项的内容包括门控时间表，数据流映射的内部优先级（IPV），如果某时间段内的门打开，则数据流可以通过，否则数据流被丢弃，通过IPV映射以后，对应的流可以获取自己的内部优先级，为后续基于优先级的转发做好准备；模块4会根据模块2指定的流量表索引来查找流量表项，流量表项的内容包括MEF 10.3定义的流量控制算法所需参数，通过模块4以后，数据流的流量会被限流，流量会小于流量表项定义的流量。Stream ID、Gate ID和Meter ID的定义如图6所示。

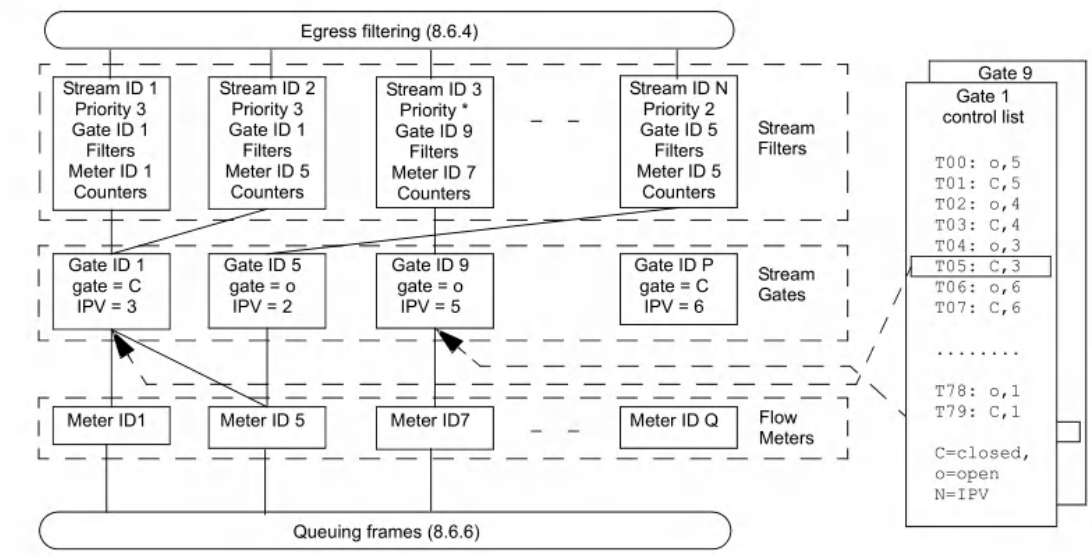


图6 IEEE802.1Qci机制

通过IEEE 802.1Qci提供的4个模块可以在数据链路层对上文提起的恶意流量攻击进行防护，感知网络中的恶意流量来识别流量的特征。

对于章节3中提到的针对目的主机的Smurf攻击，可以通过目的MAC地址来识别恶意流量，识别恶意流后可以采取以下措施进展安全防护：

（1）Qci流量截断：可以通过模块3的门控时间表来暂时关闭恶意流量，当感知到恶意流量消失以后，重新打开门控；

（2）降低优先级：通过模块3的IPV机制将内部优先级映射到最低优先级0，从而不影响其它正常流的转发，当感知到恶意流量消失以后再恢复优先级；

（3）Qci流量限制：通过模块4的流量算法来限制特征流量，将其限制在一个合理的水平，即使恶意流量短时间内快速增加也无法对其他特征

流以外的流量产生很大的影响。

针对恶意流量感知可以利用模块2的帧统计功能来进行辅助实现，Qci的帧统计功能提供如下计数器：

（1）满足模块1定义的特征流的帧数量；

（2）通过门控的帧数量；

（3）因为门控关闭被丢弃的帧数量；

（4）通过Maximum SDU size过滤的帧数量；

（5）无法通过Maximum SDU size过滤的帧数量；

（6）因为流量表操作被丢弃的帧数量。

对于针对目的主机的TCP SYNC泛洪攻击，可以通过源MAC地址来识别恶意流量，识别恶意流后可以采取上述的Qci流量截断方法，隔离来自恶意主机的流量。

对于上文提到的欺骗攻击和VLAN中继型攻击，可以通过源MAC地址或者VALN来识别攻击主机，识别攻击者以后可以采取Qci流量截断的方法，限制攻击者访问特定的VLAN网络。针对欺骗攻击例如ARP欺骗攻击，还可以使用静态ARP缓存表或者使用ARP服务器，通过服务器来查找ARP转换表来响应其他机器的广播。针对VLAN中继型还可以将所有的中继端口上使用特定的VLAN ID，并且禁止所有闲置的交换机端口或移至无用的VLAN网络之内。

4.4 总结

针对以上分析的恶意攻击，相对应的安全防护手段可以总结为：

表2 恶意攻击对应的安全防护方案

网络配置控制器攻击	网络协议攻击			数据面流量攻击		
	未知时钟源	恶意设备虚构构或篡改报文	物理攻击	高频数据流或服务请求	VLAN中继攻击	欺骗攻击
认证和授权 安全管理和安全审计 操作系统加固 软件漏洞检测 流量调度表冲突检查	信息完整性检查 重复报文保护 密钥检查刷新	统计分析流量异常 端口可信链路	热备份	流量截断 降低优先级 流量限制	流量截断	静态ARP缓存表 指定VLAN ID

5 时间敏感网络安全技术趋势

（1）安全标准的完善：目前IEEE 802.1安全工作组已经推出通用的安全和隐私等系列标准，专注于数据加密、认证等功能，但是时间敏感网络由于其实时性要求高，需要评估现有安全标准的开销对时间敏感流的时延性能影响，并针对时间敏感网络推出满足低时延高可靠的安全协议与标准，并进行更多的测试验证和逐步完善。

（2）未知漏洞的防范：在新的网络开放场景下，基于先验知识的传统防御手段难以应对各种针对位置漏洞和后门的攻击，需要转变防御思路，从被动迈向内生安全的主动防御，研究拟态安全等内生安全防御技术，通过时间敏感网络的流量调度算法、可编程芯片、多路径传输等要素，从主动性、变化性和随机性实现拟态内生安全环境，大幅度增加未知漏洞的攻击难度和成本。

6 结束语

随着工业互联网、5G前传网络、车载网等产业逐渐成熟，安全问题将是时间敏感网络在全面实际应用开展过程中面临的主要挑战之一。本文回顾了时间敏感网络发展历程，介绍了时间同步、流量调度、高可靠性和资源管理等关键技术，分析了时间敏感网络新网络架构、特有协议、传输流量所带来的安全风险，然后针对这些风险从网络配置控制器、网络协议、数据面流量三个方面研究了安全防护方案，以增强时间敏感网络的安全性。未来需要密切关注时间敏感网络国际国内技术进展，重点进行时间敏感网络安全标准、测试床及验证环境的研发，研究未知漏洞的防范机理，为新型低时延高可靠业务提供更安全的融合网络。

参考文献:

[1] 第三届未来网络发展大会组委会．未来网络发展白皮书．2019.

[2] 工业互联网产业联盟．时间敏感网络(TSN)产业白皮书V1.0版．2020.

[3] Time-Sensitive Networking (TSN) Task Group. https://1. ieee802.org/tsn/

[4] IEEE. 802.1AS-2020-IEEE Standard for Local and Metropolitan Area Networks--Timing and Synchronization for Time-Sensitive Applications[S]. 2020.

[5] IEEE. 802.1Qbv-2015-IEEE Standard for Local and metropolitan area networks -- Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic[S]. 2015.

[6] IEEE. 802.1Qci-2017-IEEE Standard for Local and metropolitan area networks--Bridges and Bridged Networks--Amendment 28: Per-Stream Filtering and Policing[S]. 2017.

[7] IEEE. 802.1Qcc-2018-IEEE Standard for Local and Metropolitan Area Networks--Bridges and Bridged Networks -- Amendment 31: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements[S]. 2018.

[8] Treytl, Albert, et al. "Traps and pitfalls in secure clock synchronization. " 2007 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication. IEEE，2007.

[9] IEEE. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems，IEEE Standard 1588™-2008，2008.

[10] TreytlA. , G. Gaderer, LoschmidtP. and KerönN. ，"Investigations on Security Aspects in Clock Synchronized Industrial Ethernet" ，Precise Time and Time Interval (PTTI) Systems and Applications Meeting，2006，pp. 232-240.

[11] IEEE. IEEE P802.1AS-Rev/D8.1 Draft Standard for Local and Metropolitan Area Networks—Timing and Synchronization for Time Sensitive Applications, IEEE Std 802.1AS™，2019.

[作者简介]

吴少勇 1981年生，男，硕士学历，之江实验室高级工程师，主要研究领域为时间敏感网络、网络安全、未来网络等。

张磊 1984年生，男，硕士学历，之江实验室高级工程师，主要研究领域为时间敏感网络、无线网络、未来网络等。

庞宏俐 1995年生，女，硕士学历，之江实验室工程师，主要研究领域为时间敏感网络、时间同步技术等。

骆汉光 1988年生，男，博士学历，之江实验室助理研究员，主要研究领域为网络安全、未来网络等。

王延松 1970年生，男，硕士学历，之江实验室教授级高工，主要研究领域为工业互联网、SDN/NFV、路由器、光传输、网络安全、5G等。

李振廷 1983年生，男，硕士学历，之江实验室高级工程师，主要研究领域为无线通信、工业互联网、确定性网络和网络安全等。

基于异构可重构架构的任务映射集负载均衡策略

祁晓峰，高彦钊，陈磊，虎艳宾
战略支援部队信息工程大学，河南 郑州 450002

摘　要：异构可重构计算架构是计算领域革命性研究和技术。面对海量计算任务，采用传统的任务映射策略难以保证可重构系统性能与可靠性。本文针对提高可重构系统可靠性、可用性和最大化系统网络通信利用率的问题，设计基于粒子群优化的任务映射集负载均衡策略（PSOMapSet）。实验结果表明，本文提出的任务映射集负载均衡策略能够有效提高系统网络通信利用率，减少任务集合之间的通信延迟，并且在循环映射过程中分担计算器件的计算应力，增强了系统的可靠性和可用性。

关键词：可重构计算、任务映射、负载均衡

Heterogeneous Reconfigurable Computing Architecture Based Load Balancing Using Task Mapping Set Strategy

Qi Xiaofeng, Gao Yanzhao, Chen Lei, Hu Yanbin
Information Engineering University,Zhengzhou 450002,China

Abstract: Heterogeneous Reconfigurable Computing architecture is the revolutionary research and technology in High Performance Computing. The traditional task mapping algorithms cannot assure the large-scale system's reliability and availability. In order to enhance the system reliability and availability and maximize the utilization of network communication, we proposed a PSO based task mapping set algorithm (PSOMapSet) for load balancing. The results proved that our algorithm enhanced the utilization of the network communication, reduced the communication delay between tasks, shared the stress for the loop computation and strengthened the reliability and availability of the system.

Key words: reconfigurable computing; task mapping; load balancing

1　引言

近年来，基于软硬件协同计算的异构可重构架构逐渐成为体系结构的主流方向［1］。异构可重构架构能够满足通用计算和专用计算的平衡，具有高可用性和强健壮性［2］。可重构计算器件的基本单元从以卷积神经网络（CNN）卷积计算为核心的乘加计算处理单元（PE）［3］，到以脉冲神经网络（SNN）为核心的事件驱动型神经网络处理单元［4］［5］。神经网络处理单元“存算一体”的特性和脉冲神经网络事件驱动计算模式的特点使得个体PE功能简洁，PE阵列的能耗极大降低，PE阵列的规模将越来愈大。PE之间的互连关系不仅决定芯片的功能和性能，也决定着芯片的效能和安全。PE互连的灵活性和可靠性已经超出PE本身性能，成为系统架构新的核心［6］。为了保证系统数据交互性能，最大化网络通信利用率，PE阵列的互连结构从基础的网络接口单元

（NIU），到Mesh-NOC，再到分层的三维NOC结构[7]。未来的互连结构势必需要通过软件定义的互连结构［8］实现系统对灵活性、可靠性和可用性的要求。

本文针对基于异构可重构计算架构的PE互连体系结构，大量处理单元被大量任务轮流访问，一方面大规模器件阵列必然导致器件失能成为常态［9］，另一方面阵列的网络通信利用率成为制约系统性能的主要因素［10］，对系统灵活性、可靠性和可用性造成的问题，研究基于异构可重构架构任务映射负载均衡策略。

任务映射策略是多系统约束的最优化问题，解决这类问题的方法主要分为两类，模调度方法［11］和启发式方法［12］。然而以上方法针对传统可重构架构，未考虑大规模互连模型的器件失能以及网络通信利用率成为影响系统性能的主要问题，难以支撑新型互连计算结构。针对以上问题，本文在第二节提出问题模型；第三节设计一种基于资源优先和粒子群优化的任务映射集负载均衡策略；第四节通过实验证明该策略的优缺点；第五节对全文进行总结。

2　问题模型

2.1　系统模型

如图1所示，一种简单的异构可重构系统架构由一个任务管理器（Task Manager），1个CPU，一组8*8的PE阵列和与PE对应的一组8*8的NIU互连结构组成。任务管理器负责分发任务，CPU负责维护计算资源状态和计算调度，PE阵列负责任务计算。PE之间的数据交互通过对应的NIU完成。NIU负责数据传输协议的解析和转换，包括对路由机制、查表方式、业务类型的控制［13］。任务管理器将获取计算器件状态的请求和任务信息分别发送给CPU和PE阵列，过程（1）（5）。CPU将自身状态和监控的PE阵列状态的信息传回任务管理器，过程（2）。CPU发送指令到PE阵列进行PE阵列的状态维护或执行计算或存储任务，过程（3）。PE阵列将状态信息返回给CPU，过程（4）。PE阵列将计算数据输出至任务管理器，过程（6）。对于大规模异构可重构系统结构，在图1所示的架构基础上，对任务管理器、CPU和PE阵列进行扩展。形成多Task Manager、Task Manager与CPU一对多、CPU与PE阵列一对多的关系。

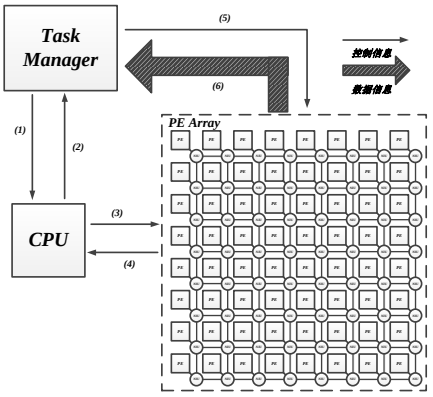


图1　简单的异构可重构系统架构

处理单元PE是可重构器件的基本计算单元，通过精心设计PE逻辑电路可以实现CNN的卷积计算，SNN的脉冲计算等计算模型。以PE实现CNN的卷积运算为例，本文采用图2所示为PE阵列分层互连结构。PE结构简单，其内部组成如图3所示，包括控制器（Controller），生成器（Generator），乘加模块（MAU）和RAM阵列。控制器的功能包括接收命令，启动控制信息，管理PE元数据，将控制信号转化为接口信号，启动计算过程，对编解码、流控、同步、服务质量、循环展开等过程的控制。生成器负责接收和解析控制器的指令，从本地或外部RAM阵列读取数据送往MAU，将计算结果写入本地或外部RAM阵列。MAU负责完成不同的运算任务。RAM阵列负责解析生成器和NIU发来的命令，完成读写操作。

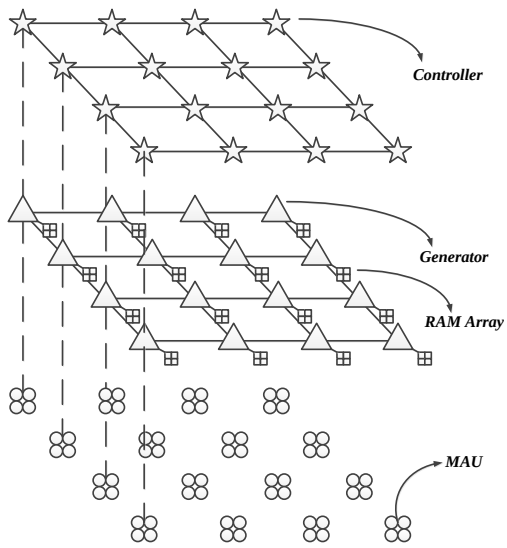


图2　PE阵列的分层互连结构

PE工作流程如图3所示，控制器将计算规则传递给生成器，过程（1）。算式生成器将数据传递到MAU计算，过程（2）。MAU将计算结果返回生成器，过程（3）。生成器获取RAM阵列状态，当允许写操作时，将数据写入RAM阵列，过程（4）。外部PE通过NIU访问本地PE时，生成器获取RAM阵列状态过程（5）。当数据可读，从RAM读取数据并输出，过程（6）。当本地PE需要访问外部PE时，由生成器向NIU发送请求，过程（7）。

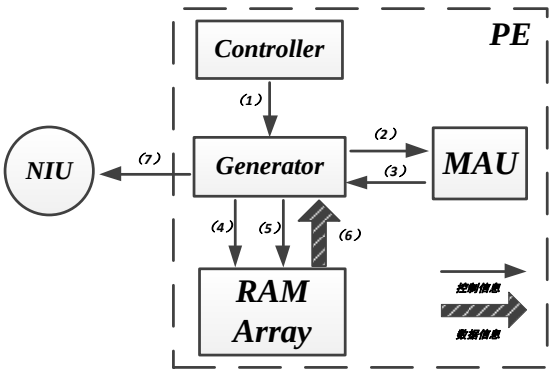


图3　PE计算流程

2.2　任务映射模型

任务映射根据任务数据流图（DDG）的每个任务节点，按照任务计算存储所需资源和逻辑关系，映射到PE阵列。任务映射前需要确定计算存储资源是否可用，计算存储资源是否已经充分利用。基于以上场景，设计任务映射模型。

假设PE在空间中占1个面积单位，且PE形状为正方形，即PE的长度和宽度均为单位1。不同的任务需要的计算资源大小不同，即需要使用一个或多个PE，组成PE子阵列。假设任务所需PE子阵列的面积为S, $S \in N^*$。故，PE子阵列包含PE个数为S。

以任务集alpha_big［14］为例，其数据流图如图4所示。数据流图 $G = \{T_i^t, E_i^j\}, i, j, t \in N^*$。$T_i^t$ 是任务节点，下角标i是任务唯一的标识，t表示任务在第t个重构周期执行。E_i^j 是有向边，表示数据流方向为从源任务i到目的任务j。E_i^j 的标量值 $|E_i^j|$，为源任务i到目的任务j的通信开销。

本文默认任务所需PE子阵列中的PE是聚合的，不是离散的，且PE子阵列的形状总是矩形的。根据PE子阵列的矩阵形状，任务 T_i^t 至少具有5个维度的参数，即 $T_i^t = \{i, m, n, t, p\}$。i为任务的唯一标识，m为任务所需PE子阵列的宽度，n为任务所需PE子阵列的长度，t为任务执行的第t重构周期，p为任务在PE阵列中的映射位置。其中，PE子阵列的面积S = m*n。

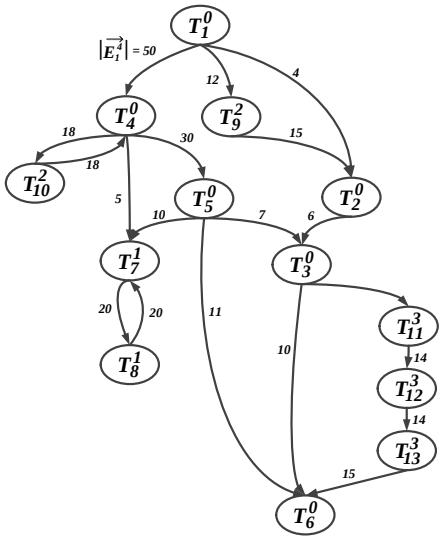


图4　alpha_big任务数据流图

3　基于粒子群优化任务映射集合负载均衡策略

在有限网络带宽资源条件下进行任务映射优化，应尽量把任务映射在有相关输入数据存储的PE上或近邻PE上执行。本文简化系统模型，默认将任务映射在相关输入数据存储的PE上，即把数据存储在相关输入数据PE对应的RAM阵列上，从而节省网络通信延迟。执行计算任务时，大量PE将被大量任务轮流访问，任务映射的结果对任务间通信和处理单元负载影响巨大。通过合理配置任务映射的位置，能够最大化网络带宽的利用率。同时，通过负载均衡策略，降低计算存储器件的应力老化程度，提高数据的可靠性和可用性。

3.1　基于曼哈顿距离的任务切分

PE之间相互访问，必须由其对应的NIU路由寻址，且访问路径唯一。NIU的路由路径越短，PE之间的数据传输效率越高。为降低PE之间的通信距离，本文的任务映射策略按照曼哈顿距离的最短路由路径，将任务切分重新组织，再映射到PE阵列。

曼哈顿距离是一种距离计算方法，属于闵可

夫斯基距离的一个子集。假设数值点P和Q的坐标，$P=(x_1,x_2,\cdots,x_n)\in R^n$, $Q=(y_1,y_2,\cdots,y_n)\in R^n$。闵可夫斯基距离为：

$$\left(\sum_{i=1}^{n}|x_i-y_i|^p\right)^{\frac{1}{p}}$$

当p = 1时，是曼哈顿距离；当p = 2时，是欧式距离。PE阵列简化后如图5所示，NIU通过上下左右四个方向互连。故在计算NIU路由路径距离时，相比于欧式距离，曼哈顿距离更加合理。

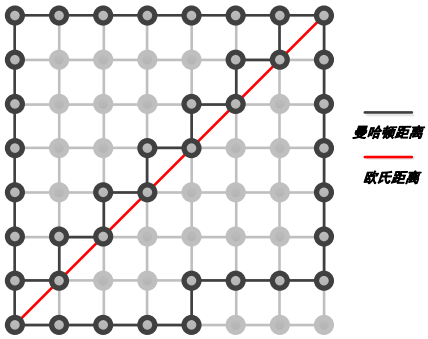


图5　曼哈顿距离和欧式距离

在任务内部，为使NIU之间的路由路径最短，本文提出基于曼哈顿距离的最短路径任务切分。根据矩形任务$T_i^t=\{i,m,n,t,p\}$，其任务个数为S = m*n。任务内部PE通信的总路径$Path_{sum}$为：

$$Path_{sum}=\frac{mn(m+n-1)}{2} \tag{1}$$

又知$m\in N^*, n\in N^*$, S = mn，即：

$$Path_{sum}=\frac{S(\frac{S}{n}+n-1)}{2} \tag{2}$$

通过导数方法求函数极值，当$n=\sqrt{S}$，或$m=\sqrt{S}$（m和n等效）时$Path_{sum}$取得最小值。在实际映射中，由于$m\in N^*, n\in N^*$，故取与$\sqrt{S}$最近正整数为m值或n值时，$Path_{sum}$取得最小值。

本文定义任务切分函数mht（）。函数mht（）的逻辑流程如图6所示。

任务切分操作过程（1）输入任务参数，计算新任务的宽度w值，即最接近$\sqrt{S}$值的正整数。（2）计算新任务的高度h值，即S/n的值向下取整。（3）判断S - w*h = 0是否为真。若真，则将新任务T（w，h）写入任务集合，并返回任务集合；若假，则进行步骤（4）计算剩余S，并跳转到过程（1）循环。

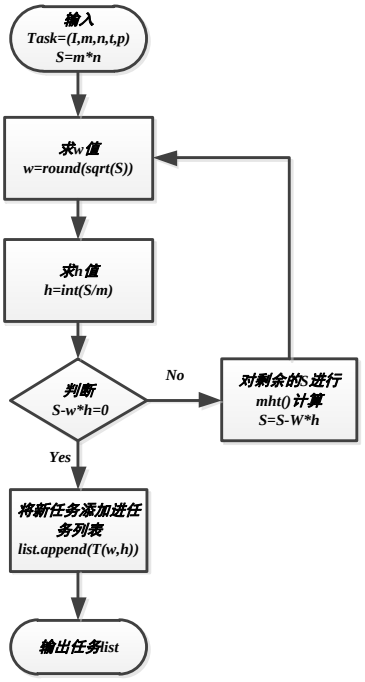


图6　任务切分过程

任务切分操作后，原任务实际上将会被拆分成一个尽可能大的矩形任务以及0个或多个小矩形任务。例如，对于需要12个PE的任务T_{exp1}，经过mht（T_{exp1}）计算，可知当w = 3, h = 4时，任务内部的通信总路径最短；对于需要14个PE的任务T_{exp2}，经过mht（T_{exp2}）计算，输出任务集合$\{T_{output1},T_{output2}\}$，$T_{output1}$的w = 4, h = 3，$T_{output2}$的w = 1, h = 2。此时任务内部的通信总路径最短。

3.2　基于资源优先的任务映射

将具有一定逻辑关系和外在约束的任务集合映射到PE阵列上，属于多约束条件下的最优化求解问题。对于k个任务，任务映射本质是在k维解空间寻找满足系统约束的最优解。为便于计算，本文规定系统约束为（1）任务映射的PE子阵列可用。（2）任务映射位置不超出PE阵列总空间。（3）最大化利用计算资源。

根据以上约束，本文提出资源优先的任务映射策略，算法流程如图7所示。输入任务参数，（1）按照基于曼哈顿距离的最短路径，将任务切分。（2）根据任务参数t，将同一执行重构周期的任务分为一组，每组任务按照组任务总面积之和由大到小排序，并依次循环下面步骤。（3）若所有组任务全部映射完成，则输出映射方案；否则，按照重构周期组排序，对每组组内任务按照任务

面积由大到小排序，并依次循环下面步骤。（4）若组内任务全部映射完成，则进行下一重构周期组的任务映射；否则，按照任务排序，对PE阵列从左到右，从上到下扫描可用的PE，并将任务映射到PE阵列。（5）若任务映射成功，则进行下一任务的映射；否则，输出空集的映射方案。

3.3 基于粒子群优化的任务映射

基于资源优先的任务映射策略至多生成一个任务映射方案，该映射方案是映射方案解空间内的一个局部最优解，未能最大化网络通信利用率。同时，对于大规模循环迭代的任务，海量数据在同一PE上重复进行相同计算操作，不可避免造成计算器件应力老化和不同计算器件负载不均匀的问题。针对上述问题，本文提出基于粒子群优化映射集合的负载均衡策略。

粒子群优化（PSO）是一种模仿鸟群寻食行为的启发式算法［15］。PSO从随机解出发，经过迭代寻找最优解，同时适应函数评价解质量。使用粒子群优化算法求解任务映射方案，粒子群模型与任务映射方案模型对应如表1所示：

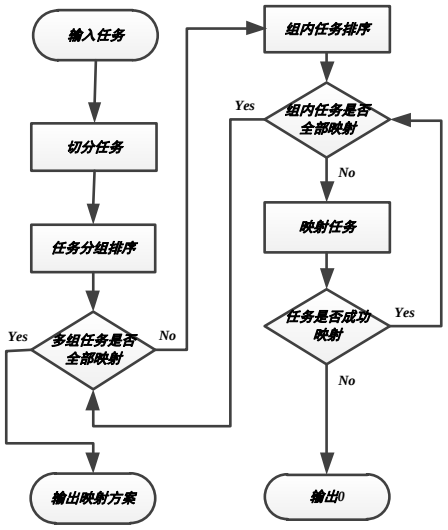


图7 资源优先的任务映射流程

表1 粒子群模型与任务映射模型参数对比

粒子群模型	粒子群参数	任务映射模型	任务映射参数
粒子	$Particle_i, i \in 1,2,\cdots size$	任务映射方案	$Placement_i, i \in 1,2,\cdots size$
粒子个数	Size	映射方案个数	Size
粒子维度	Dim	任务个数	K
粒子位置	$\{p_1,p_2,\cdots,p_{dim}\}$	映射方案的任务集合	$\{(x_1,y_1),(x_2,y_2),\cdots,(x_k,y_k)\}$
粒子速度	$\{v_1,v_2,\cdots,v_{dim}\}$	映射方案任务集合的位置变化速度	$\{(v_{x1},v_{y1}),(v_{x2},v_{y2}),\cdots,(v_{xk},v_{yk})\}$
粒子最优位置	$\{p_1,p_2,\cdots,p_{dim}\}_{best}$	映射方案最优任务集合	$\{(x_1,y_1),(x_2,y_2),\cdots,(x_k,y_k)\}_{best}$
粒子最优位置适应值	$Par\_fit(\{p_1,p_2,\cdots,p_{dim}\}_{best})$	映射方案最优任务集合的适应值	$TM\_fit(\{(x_1,y_1),(x_2,y_2),\cdots,(x_k,y_k)\}_{best})$
迭代次数	iter_num	迭代次数	iter_num
粒子最大速度	v_{max}	映射方案任务集合的位置的最大变化速度	(v_{max},v_{max})
粒子群最优位置	$\{p_1,p_2,\cdots,p_{dim}\}_{gbest}$	映射方案集合的最优任务集合	$\{(x_1,y_1),(x_2,y_2),\cdots,(x_k,y_k)\}_{gbest}$
每次迭代的最优适应值	$Par\_fit(\{p_1,p_2,\cdots,p_{dim}\}_{gbest})$	每次迭代的最优映射方案适应值	$TM\_fit(\{(x_1,y_1),(x_2,y_2),\cdots,(x_k,y_k)\}_{gbest})$

为提高PSO任务映射算法的效率，本文PSO初始化阶段，将粒子的初始值指定为基于资源优先任务映射策略生成的映射方案，而不是随机生成的任务映射方案。基于PSO的任务映射流程如图8所示。输入任务参数，（1）对任务按照基于资源优先的任务映射策略进行初始化操作。（2）按照迭代次数循环迭代。（3）在每一次迭代中，根据映射方案最优任务集合和映射方案集合的最优任务集合，对所有映射方案进行速度和位置的更新。（4）记录符合系统约束的映射方案。（5）迭代结束后，输出任务映射方案集合。

3.4 映射集合的循环映射策略

在任务循环映射过程中，在每组重构周期之间，将符合系统约束的所有映射方案循环映射到PE阵列上，从而使PE共同分担计算应力，实现负载均衡和抗应力老化的能力，映射集合的循环映射效果如图9所示。图中所示8*8的PE阵列的不同循环周期，彩色矩形是经过任务切分后的任务集

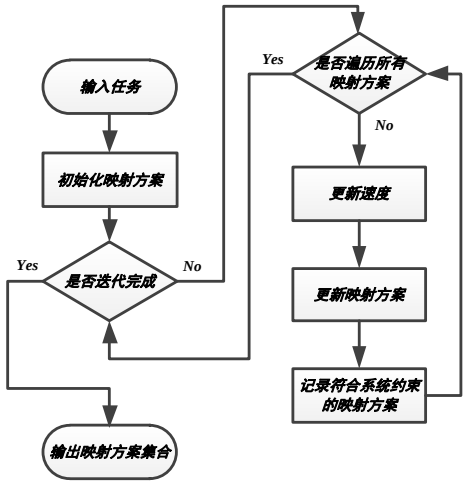


图8　基于粒子群优化的任务映射流程

合。执行PSOMapSet算法，生成了12种任务映射方案在循环映射周期（period）中轮流使用。在实验中，对于6*6的PE阵列，PE阵列面积与任务总面积比值接近1，由于PSOMapSet策略随机性强，因此在PE阵列面积有限的场景下，PSOMapSet难以找到更优解。对于10*10的PE阵列，任务映射向下兼容8*8的PE阵列上的最优解，故不再列出更多映射方案。

4　实验结果

本文在Window 10操作系统，Core i5 CPU上进行仿真实验。通过Python仿真建模，模拟PE阵列模型、任务映射模型和粒子群模型。PE阵列取5种规模，分别为6*6、8*8、10*10、64*64、1024*1024的PE阵列。任务数据集通过TGFF生成器[16]配置参数生成，根据不同规模PE阵列，随机生成不同规模的任务流图，任务个数分别为8个任务，50个任务和500个任务。仿真实验的对比任务映射策略包括，基于任务优先（TF）的任务映射算法，基于任务优先的任务映射集的负载均衡（TF_Rotate）策略[3]和基于资源优先（RF）的任务映射算法，基于粒子群优化任务映射集（PSOMapSet）的负载均衡策略。对任务映射策略的评价从（1）映射方案的网络带宽利用率，即任务间的通信总距离，（2）算法复杂度及运行时间，（3）PE阵列负载均衡效果等维度展开。本文设定粒子群的迭代次数为2000，粒子数量为200。在仿真过程中，由于单机计算性能受限，PSOMapSet算法在可接受运行时间（本文设定为300s）内，可在6*6、8*8、10*10的PE阵列，8个任务的规模下生成任务映射集。对于更大数量规模，如64*64，1024*1024的PE阵列和任务数量的任务映射模型中，任务映射可通过在数据中心的云计算资源上软件定义配置调度实现。

4.1　任务互连通信距离

在8个任务，6*6，8*8和10*10的PE阵列任务映射规模下，生成最优映射方案如图10所示，映射方案的互连通信总距离如图11所示。

由图可知，随着PE阵列规模与任务总面积之比降低，映射方案解空间内的局部最优解增加，由粒子群优化生成的任务映射方案更易优于规划调度的任务映射算法。粒子群优化算法在迭代过

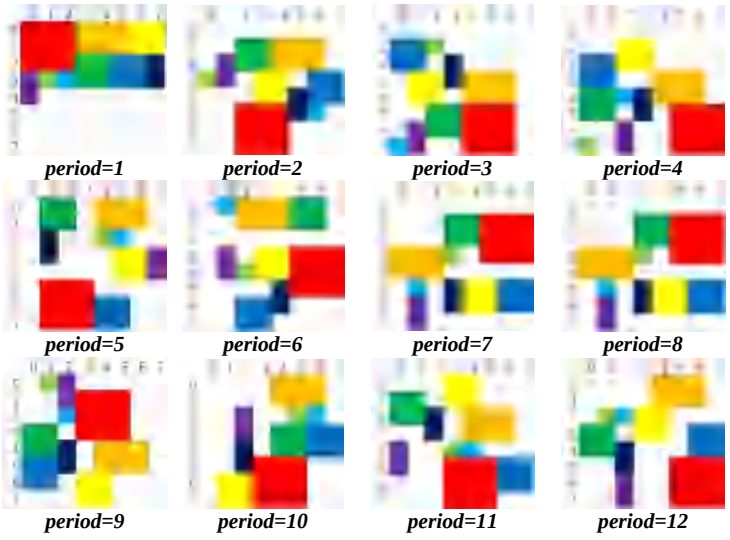


图9　映射集合的循环映射

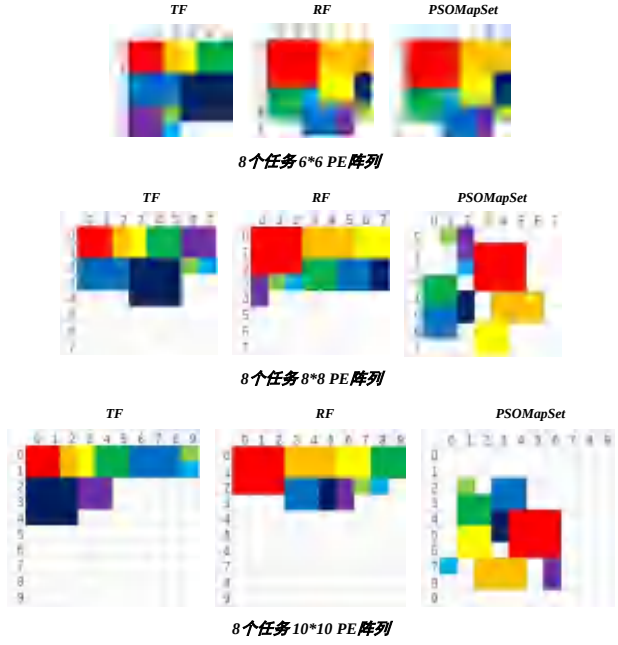


图10　最优任务映射方案

程中能够有效跳出局部最优解，通过启发式的搜索发现更优解，甚至最优解。基于粒子群优化的任务映射方案在任务互连通信总距离上，相比与任务优先的任务映射算法和资源优先的任务映射算法，互连通信总距离相应平均减少15.71%，11.26%。

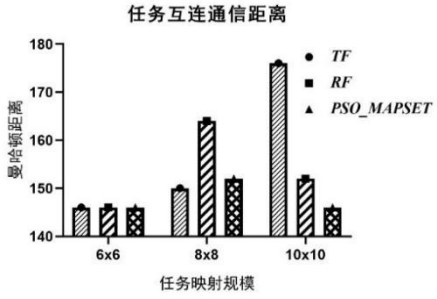


图11　任务互连通信总距离

4.2　算法运行时间

在不同任务规模，不同任务映射策略生成任务映射方案的时间如图12所示。规划调度的任务映射算法其算法复杂度由任务规模的大小k决定，为O(k)。本文PSOMap4Set算法的复杂度由任务规模k、迭代次数iter_num、映射方案数量size共同决定，为O(k*iter_num*size)。PSOMapSet算法由于多组迭代循环，算法占用大量执行时间。但是可以预见的是，一方面未来系统级调度映射将由并行分布式的云端计算资源实现，软件定义互连的任务映射算法运行时间对系统的影响将会因分布式计算而变得越来越短；另一方面，随着任务规模的增加，算法一次性执行时间相比于大规模任务的循环执行的通信延迟，其对互连体系结构计算性能的影响将越来越低。

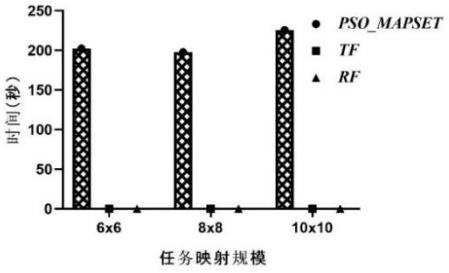


图12　算法执行时间

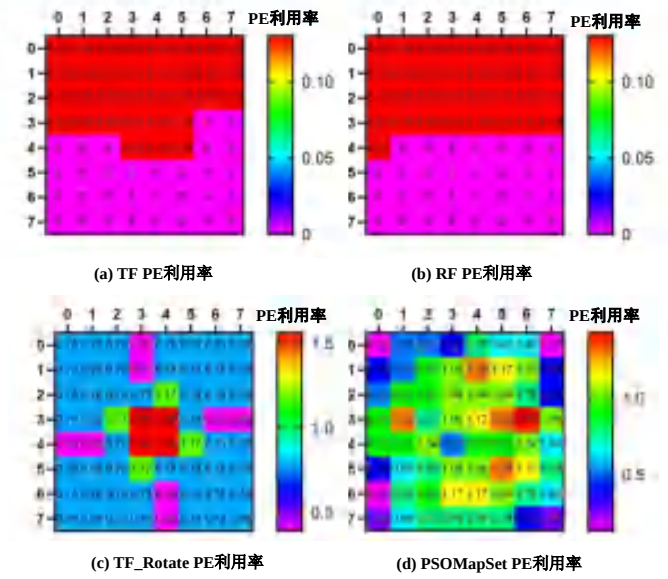
4.3　PE阵列负载均衡效果

Master主要负责任务的控制调度，其计算任务较少，故执行计算的应力负担小，本文未考虑Master的负载均衡问题。对于Worker的计算应力问题，本文假设一组重构周期的任务集合循环映射迭代次数为12，任务集合元素个数为8，在8*8的PE阵列的任务映射规模上，比较任务循环映射迭代周期内，每个PE执行计算任务的次数与所有待执行计算任务PE的任务计算百分比，计算PE利用率，分别如图13（a），（b），（c），（d）所示。本文算法的PE利用率比基于模调度的任务映射算法TF，RF的PE利用率提高47.26%，与RF_ROTATE映射集负载均衡策略的PE利用率保持相同。

对比图14（a），（b），可以看出PSOMapSet策略的PE利用率分布更加分散，对比图14（c），（d），PE利用率的最大值小于RF_ROTATE策略，说明PSOMapSet策略可以使PE阵列有效平均分担计算应力，提高PE阵列的负载均衡和抗应力老化能力。

5　结束语

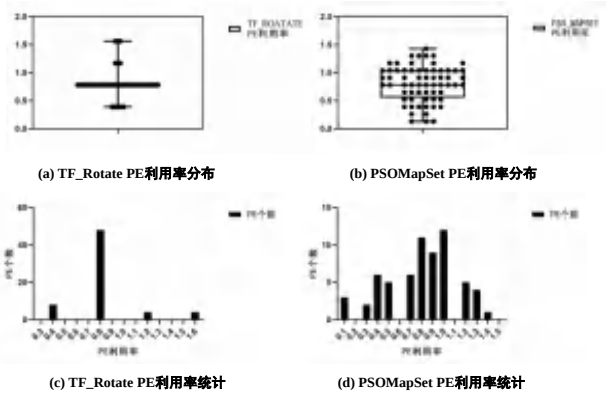
本文针对在异构可重构架构上，任务映射到PE阵列的可靠性、可用性，以及最优化互连通信利用率的问题，提出基于资源优先和粒子群优化的任务映射集负载均衡策略。实验结果表明，本文算法能够有效提高系统互连通信利用率和系统



(a) TF PE利用率　(b) RF PE利用率

(c) TF_Rotate PE利用率　(d) PSOMapSet PE利用率

图13　PE利用率



(a) TF_Rotate PE利用率分布　(b) PSOMapSet PE利用率分布

(c) TF_Rotate PE利用率统计　(d) PSOMapSet PE利用率统计

图14　PE利用率分布统计

可靠性和可用性。

在下一步研究中，除了对本文研究的任务映射策略的优化，以及对逻辑模型内部机制各环节的实现以外，需要解决本文为了计算而对系统模型进行简化约束的几个问题。首先，本文的NIU互连模型是基于二维的Mesh-NOC，未来将出现更高维的NIU互连结构，需要研究高维互连结构对系统通信的影响；其次，本文默认PE阵列式同构的，而在实际场景中对大规模PE阵列扩容时，必定存在异构PE阵列互连的场景需求，故需设计异构器件协议控制、解析、转换等机制。最后，本文仿真实验受时间和条件限制只在单机上进行，未能和云计算资源紧密配合，故在算法适配的系统规模有限。

参考文献:

[1] 尹首一,郭珩,魏少军. 人工智能芯片发展的现状及趋势[J]. 科技导报,2018,36(17):45-51.

[2] 邬江兴. 拟态计算与拟态安全防御的原意和愿景[J]. 电信科学,2014,30(07):2-7.

[3] 尹首一,欧阳鹏,唐士斌,涂锋斌,李秀冬,郑时轩,陆天翼,谷江源,刘雷波,魏少军. Thinker:可重构混合神经网络计算芯片[J]. 人工智能,2018(02):34-45.

[4] 杜宇阳,刘忠轩,宋继强. 英特尔Loihi神经拟态芯片:引领智能计算新突破[J]. 人工智能,2018(02):60-71.

[5] Jing Pei, Lei Deng, Sen Song, Mingguo Zhao, Youhui Zhang, Shuang Wu, Guanrui Wang, Zhe Zou, Zhenzhi Wu, Wei He, Feng Chen, Ning Deng, Si Wu, Yu Wang, Yujie Wu, Zheyu Yang, Cheng Ma, Guoqi Li, Wentao Han, Huanglong Li, Huaqiang Wu, Rong Zhao, Yuan Xie, Luping Shi. Towards artificial general intelligence with hybrid Tianjic chip architecture[J]. Nature: International weekly journal of science,2019,572(7767).

[6] 吕平,刘勤让,邬江兴,陈鸿昶,沈剑良. 新一代软件定义体系结构[J]. 中国科学:信息科学,2018,48(03):315-328.

[7] 李晨,马胜,王璐,郭阳. 三维片上网络体系结构研究综述[J]. 计算机学报,2016,39(09):1812-1828.

[8] Mao Y , Oak J , Pompili A , et al. DRAPS: Dynamic and Resource-Aware Placement Scheme for Docker Containers in a Heterogeneous Cluster[J]. 2018.

[9] OuniBouraoui, MtibaaAbdellatif. Modules placement technique under constraint of FPGA forbidden zones [M]. Inderscience Publishers, 2015.

[10] 阳王东,王昊天,张宇峰,林圣乐,蔡沁耘. 异构混合并行计算综述[J]. 计算机科学,2020,47(08):5-16+3.

[11] 陆天翼. 面向可重构计算的存储优化算法研究[D]. 清华大学,2018.

[12] Neetesh K , Prakash V D . A Hybrid Heuristic for Load-Balanced Scheduling of Heterogeneous Workload on Heterogeneous Systems [J]. The Computer Journal(2):276-291.

[13] 吕平,董春雷,刘冬培,张文建,汪欣. 基于FPGA的软件定义流量发生器[J]. 通信学报,2018,39(S2):66-71.

[14] Filho J G , Chau W J . Exploring the problems of placement and mapping in NoC-based reconfizurable systems[J]. 2013:1-4.

[15] Eberhart R , Kennedy J . A new optimizer using particle swarm theory [C]// Mhs95 Sixth International Symposium on Micro Machine & Human Science. IEEE, 2002.

[16] Dick R P , Rhodes D L , Wolf W . TGFF: task graphs for free[C]// Hardware/Software Codesign, 1998. (CODES/CASHE '98) Proceedings of the Sixth International Workshop on. IEEE, 1998.

[作者简介]

祁晓峰（1992—），男，硕士，研究实习员，主要研究方向是高性能计算，人工智能。

高彦钊（1984—），男，博士，助理研究员，主要研究方向是高效能计算，人工智能。

陈磊（1987—），男，硕士，工程师，主要研究方向是高性能计算。

虎艳宾（1978—），女，硕士，助理研究员，主要研究方向是高速电路设计，内生安全。

中文短文本分类技术研究综述

刘硕[1]，潘世东[2]，王庚润[1]，李英乐[1]

[1]中国人民解放军战略支援部队信息工程大学 河南省 郑州市 450000；

[2]江南计算技术研究所 江苏省 无锡市 214000

摘　要：随着信息技术的迅速发展，网络上产生了海量的中文短文本数据。利用中文短文本分类技术，在低信息量的数据中挖掘出有价值的信息是当前的一个研究热点。中文短文本相较于长文本，存在字数少、歧义多、特征稀疏和信息不规范等特点，导致使用传统文本分类技术效果不佳。本文首先介绍中文短文本分类技术的研究现状，然后围绕中文短文本分类的基本流程和关键技术进行阐述，并对文本预处理、文本表示、特征扩展和分类算法做出详细地介绍。最后，对中文短文本分类技术未来发展的趋势进行展望。

关键词：短文本分类、特征扩展、文本表示、分类器

A Survey on Chinese Short Text Classification Technology

LIU Shuo[1], PAN Shidong[2], WANG Gengrui[1], LI Yingle[1]

1.People's Liberation Army Strategic Support Force Information Engineering University, Zhengzhou 450000, China;

2.Jiangnan Institute of Computing Technology, wuxi,214000, China

Abstract: With the rapid development of information technology, massive amounts of Chinese short text data have been generated on the Internet. Using Chinese short text classification technology to dig out valuable information from low-information data is a current research hotspot. Compared with long texts, Chinese short texts have the characteristics of fewer words, more ambiguities, sparse features, and irregular information, which leads to poor results using traditional text classification techniques. This paper first introduces the current research status of Chinese short text classification, and then elaborates on the basic process and key technologies of Chinese short text classification, and gives a detailed introduction to text preprocessing, text representation, feature expansion and classification algorithms. Finally, the future development trend of Chinese short text classification technology is prospected.

Key words: Short text Classification; Feature extension; Text representation; Classifier

1　引言

随着互联网用户的增加和社交网络的快速发展，生活中产生了大量的短文本数据，例如手机短信、微博、商品描述、评论信息和新闻主题等。如何利用短文本分类技术挖掘出隐藏的信息，并应用到垃圾短信分类［1]、邮件过滤［2]、话题追踪［3]、情感分析［4]、舆情监测［5］和用户个性化推荐［6］等实际任务中，是目前自然语言处理（NLP）领域研究的热点之一。

相较于中文长文本，短文本分类技术面临以下几个难题：1）短文本中存在大量的表情符号和网络用语，导致中文分词任务面临严重的歧义消除和未登录词识别问题；2）短文本字数少，特征稀疏，造成文本信息难以提取和表达，从而导致使用传统的文本分类技术难以获得较好的分类效果；3）短文本数据规模大、更新速度快，在短时间内网络中会产生海量的低价值短文本数据，因而对数据处理的性能提出了更高的要求。

下文中，第2节将阐述中文短文本分类技术的研究现状，第3节将重点介绍中文短文本分类的处理流程和相关技术，第4节介绍评价指标与实验，

基金项目：国家自然科学基金(61803384)。

第5节则讨论了短文本分类的研究难点及对策，第6节进行总结与展望。

2 研究现状

中文短文本由于字数少、歧义多、特征稀疏以及信息不规范等特点，导致在中文分词、文本表示、特征提取和分类模型搭建等任务上存在诸多难题。同时，使用传统的文本分类技术难以取得较好的分类效果。接下来，本节将详细介绍中文短文本分类技术的研究现状。

中文短文本分类的首要任务是分词，其分词结果将直接影响分类效果。由于中文字词之间没有明确的分隔符，且短文本具有多歧义和不规范等特点，导致中文分词任务面临严重的歧义消除问题和未登录词识别问题。针对此类问题，刘泽文等［7］提出基于条件随机场（CRF）的中文分词算法，该算法先使用CRF模型对短文本进行初步分词，再利用词典分词算法对前一步的分词结果进行修正，从而提高分词的准确率。张子睿等［8］提出基于BiLSTM-CRF混合模型的中文分词算法，此算法通过利用BiLSTM模型来获取上下文的信息，再利用CRF模型完成分词任务，从而提高分词的性能。

由于短文本特征较为稀疏，完成分词任务后，往往需要对短文本进行特征扩展，常用的特征扩展方法分为基于外部知识库的特征扩展和基于文本自身内容的特征扩展。李湘东、巴志超等［9-11］通过借助维基百科和知网等外部知识库，对短文本特征进行语义扩展，从而解决了短文本特征表示不足问题。吕超镇、邵云飞、张浩等［12-14］通过借助LDA模型对短文本进行主题预测，然后将预测的类别特征词扩展到短文本中以提升短文本特征表示的效果。黄梦婷等［15］使用非负矩阵分解的方法，对短文本进行特征扩展，提高了短文本分类的准确性和算法的鲁棒性。靳一凡等［16］采用频繁项挖掘算法构建频繁项集，然后将相关的新特征添加到短文本的特征空间中，来提高短文本的分类效果。

为了让机器能够处理自然语言，需要进行文本表示。采用传统的文本表示方法（例如词袋模型），往往会导致语义表示不充分和“维度灾难”。2013年，Mikolov等［17-18］提出了word2vec词向量模型，该模型采用单层神经网络将高维度的one-hot向量转换成低维度的词向量，能够较好的考虑上下文语义信息，同时可以避免维度“灾难”问题。但由于word2vec是一种静态的词向量表示方式，其中词与向量是一一对应的关系，无法解决多义词的表示问题。2018年，Peters等［19］针对Word2vec的缺点提出来一种动态的词向量表示模型ELMo，其根据每个词的上下文信息来推断与之对应的词向量，通过结合前后文语境对多义词进行理解，从而很好的解决了多义词的问题。

为了进一步提高文本表示能力，研究人员提出了预训练模型，这类模型首先对大规模的语料数据进行训练，获得一个通用的“语言理解”模型，然后根据具体的NLP任务对模型进行微调，以实现更好的表示效果。2018年，Devlin等［20］提出了基于深度双向Transformer的Bert预训练模型，该模型在众多NLP任务上取得优异表现。2019年，Zhang等［21］提出了一种基于Bert和知识图谱融合的ERNIE预训练模型，该模型利用知识图谱中的多信息实体作为外部知来改善语言表征。Yang等［22］基于Transformer-XL提出了XL-Net预训练模型，该模型采用自回归训练方法，通过对输入序列进行“排列组合”，让模型实现双向上下文信息的学习，同时又可以避免Bert模型中mask方法带来的两阶段训练数据分布不一致问题。Lan等［23］提出了ALBERT预训练模型，该模型采用矩阵分解和跨层参数共享技术对Bert模型进行参数缩减，在维持Bert性能的情况下，降低了其空间复杂度，提高模型的训练速度，同时具有较好的扩展性。

中文短文本分类任务中最核心的一步，在于分类模型的构建。常用的分类模型分为基于传统机器学习的分类模型和基于深度学习的分类模型。基于传统机器学习的分类器模型有K-近邻算法［24］、朴素贝叶斯［25］、决策树［26］和支持向量机［27］等，这些分类模型结构简单，但分类效果不好。目前，基于深度学习的分类模型比较流行，例如TextCNN、Bi-LSTM、FastText、胶囊网络、注意力机制和Bert等。李志杰等［28］提出了LSTM-TextCNN融合的短文本分类模型，TextCNN模型能够获取短文本的局部特征，而LSTM模型能够获取短文本长距离的特征，将这两

种模型进行融合，能够提升短文本分类的效果。尹春勇等［29］提出一种改进胶囊网络的文本分类模型，该模型使用两层卷积操作对文本的局部特征进行提取，然后再使用胶囊网络对整体特征进行提取。贾旭东等［30］提出了一种多头注意力机制与胶囊网络融合的文本分类模型，该模型在多标签分类任务上有良好的表现。丁辰晖等［31］提出了一种知识图谱与注意力机制融合的短文本分类模型，该模型在短文本分类任务中具有优秀的效果。段丹丹等［32］使用Bert模型对中文短文本进行分类，实验表明，此方法优于TextCNN模型的分类效果。

3　处理流程及相关技术

中文短文本分类流程主要分为文本预处理、文本特征表示和分类器构建三个阶段，如图1所示。在预处理阶段需要进行中文分词，并去除停用词等；在文本特征表示阶段会将前一步得到的词转化成机器能够识别运算的数字或向量，然后进行特征扩展，提取出更多对分类有重要作用的特征；最后，将上一步得到的特征表示输入到分类器模型中，即可完成分类任务。整个流程如图1所示：

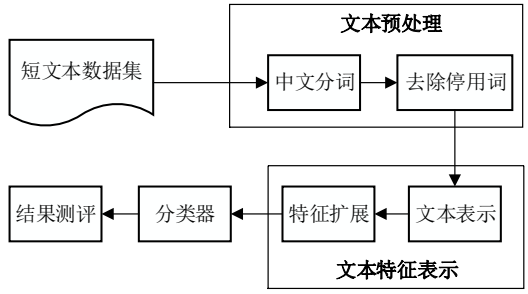


图1　中文短文本分类流程

3.1　文本预处理

文本预处理阶段的主要任务是分词和停用词的剔除。

分词技术是众多NLP任务的前提和基础，其分词结果的好坏直接影响了下一环节任务的质量。然而，相较于以英文为代表的拉丁语言，中文句子没有以空格为界的“天然”分隔符，导致中文分词任务会更加复杂与困难。黄昌宇等［33］总结中文分词的难点，主要有词的界定、歧义消除和未登录词识别。在中文短文本分词任务中，歧义消除和未登录词识别问题更加明显。

中文分词主要有基于词典匹配、基于统计学习和基于语义理解的算法。基于词典匹配的分词算法是根据一定的匹配策略进行分词，其优点是分词速度快，但不能很好的解决歧义消除和未登录词识别问题。基于统计学习的分词算法常把分词任务转变成序列标注任务，其中条件随机场模型［34］较好地融合了隐马尔可夫模型［35］和最大熵模型［36］的优点，能够综合考虑上下文之间的特征，并解决了标记偏见的问题，是传统序列标注分词模型中效果较优的算法。随着深度学习在中文分词领域的应用，采用神经网络与条件随机场相结合的分词模型能大大提升分词的准确性，同时也能更好的解决歧义消除和未登录词识别的问题，其中具有代表性的是Bi-LSTM+CRF模型［37］。

完成分词任务后，需要进行停用词过滤工作。该过程主要是过滤掉文本中无实际意义的词语，例如助词、语气词以及副词等，这类词不仅对分类任务没有帮助，反而会影响分类效果。

3.2　文本特征表示

3.2.1　特征表示

众所周知，机器是无法识别和理解自然语言的。文本表示就是把自然语言转换成数字或向量，进行建模，让机器能够认识且更好地理解文本信息。常用的文本表示方式有以下几种：

1　独热编码

独热编码又名一位有效编码。在文本表示中，首先根据文本内容构建一个固定顺序的词表，在出现词对应的词表位置用1表示，没出现的词用0表示。独热编码的优点是原理简单，使用方便。缺点是没有考虑语义相关信息，且容易导致维度“灾难”等。

2　词频-逆文档频率

词频-逆文档频率（TF-IDF）是一种词频加权的文本表示方法，可以评估一个词在文章中的重要程度。其算法思想是一个较少出现的词在某篇文章中却多次出现，说明该词对这篇文章很重要。词频（TF）表示某个词在在文章中出现的频率；逆文档频率（IDF）表示一个词的普遍重要性的程

度。词频-逆文档频率的计算公式如下：

TF-IDF=TF×IDF （1）

其中：

TF= （2）

IDF=log（） （3）

词频-逆文档频率的优点是简单快速，使用词频来衡量词的重要程度。缺点是忽略了词与词的位置信息以及词与词之间的相互关系。

3.word2vec

2013年，Mikolov等人针对神经网络语言模型[38]（NNLM）的弊端，提出了词向量文本表示方法word2vec。该方法仅采用只有一个隐藏层的神经网络，把one-hot向量形式的输入映射成一个n维度的词向量。

word2vec包含CBOW和Skip-gram两种词向量训练模型，如图2所示。CBOW连续词袋模型，即通过上下文的K个单词来预测中心词w（t），而Skip-gram模型是通过中心词w（t）来预测上下文K个单词。word2vec虽然能够较好的考虑上下文语义信息，同时可以避免维度“灾难”问题。但由于词与向量是一对一的固定表示，会导致无法根据特定任务进行词向量的动态优化和多义词的准确表示。

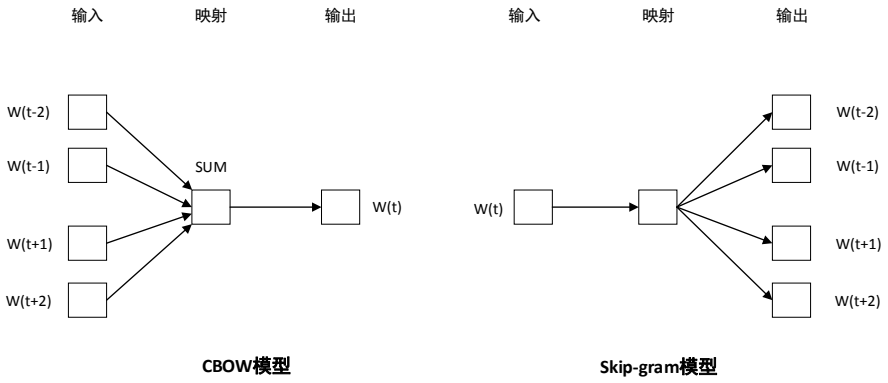


图2 word2vec模型

4.ELMo

由于word2vec模型无法很好地解决多义词的表达问题，2018年，Peters等人提出了ELMo动态词向量表示模型，该模型采用双层双向的LSTM，其结构如图3所示。

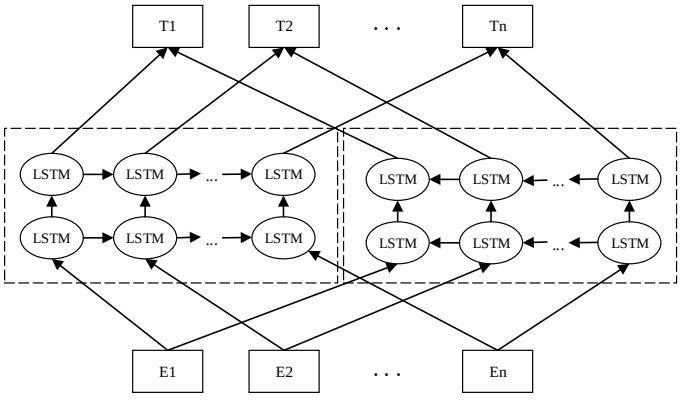


图3 ELMo模型

为了解决多义词的表示问题，ELMo模型采用两阶段的训练过程。第一阶段，采用语言模型进行预训练；第二阶段，根据单词的上下文语义信息对词向量进行调整。通过这种训练方式，EMLo模型很好地解决了多义词的表示问题。由于ELMo预训练模型在特征抽取上选用的是LSTM，而不是

特征提取效果更好的Transformer，导致其效果不如后来的GPT模型和Bert模型。

5 Bert

Bert是基于深度双向Transformer的预训练模型，其结构如图4所示。首先通过对大规模的语料数据进行训练，获得一个通用的“语言理解”模型，然后在具体的NLP任务中对Bert模型进行微调，以实现更好的表示效果。

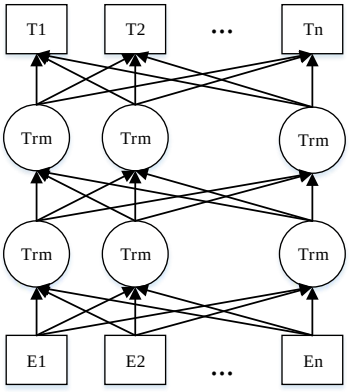


图4 Bert模型结构

Bert模型的核心部分是Transformer Encoder，如图5所示。Transformer Encoder模型的输入是一句话的字嵌入表示和其对应的位置编码信息，模型的核心层是一个多头注意力机制。注意力机制最初应用在图像特征提取任务上，比如人在观察一幅图像时，并不会把图像中每一个部分都观察到，而是会把注意力放在重要的部分，后来研究人员把注意力机制应用到了NLP任务中，并取得了很好的效果。多头注意力机制就是使用多个注意力机制进行单独计算，以获取更多层面的语义信息，然后将各个注意力机制获取的结果进行拼接组合，得到最终的结果。Add&Norm层会把Multi-Head Attention层的输入和输出进行求和并归一化处理后，传递到Feed Forward层，最后会再进行一次Add&Norm处理，输出最终的词向量矩阵。

3.2 2 特征扩展

由于中文短文本字数少，特征稀疏等特点，造成文本特征表示能力不强，从而导致使用传统的文本分类技术效果不好。解决该问题的方法主要分为两大类，一是基于外部语料库来扩展特征，

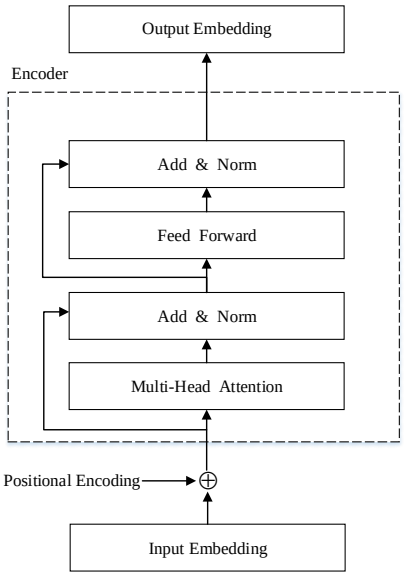


图5 Transformer Encoder模型结构

二是基于短文本自身内容扩展特征。

基于外部语料库的特征扩展方法，主要是通过引用维基百科、百度文库和知网等外部知识库，对短文本内容进行扩充，从而来解决短文本特征表示能力不足的难题。但这种方法需要根据不同的分类任务，构建与之相关的外部语料库，存在泛化能力不强，普遍适用性差等缺点。

基于短文本自身内容扩展特征的方法，不需要借助外部语料库，主要通过对短文本自身的语义、词频等进行分析，改进文本特征表达方法，来实现对短文本特征的扩展。例如借助LDA主题模型，将与短文本主题最相近的词汇加入到特征集中，或通过词向量语义计算，对短文本中重要的特征赋予更高的权重等。

3.3 分类器模型

分类器搭建是短文本分类任务中最关键的一步，其分类器的优劣直接决定了分类结果的好坏。目前短文本分类模型主要分为基于传统机器学习的分类模型和基于深度学习的分类模型。

3.3.1 传统机器学习分类算法

基于传统机器学习的分类算法主要有K-近邻算法、朴素贝叶斯、决策树和支持向量机等。各种算法的原理和优缺点对比如表1所示。

3.3.2 深度学习分类模型

随着深度学习在计算机视觉和语音识别领域取得突破性进展后，研究人员开始将深度学习技术应用到NLP领域。近年来，使用深度学习模型

表1　传统机器学习分类算法对比

算法	原理	优点	缺点
K-近邻	通过特征空间中K个最近距离的样本类别，来预测待分类样本的类别。	原理简单；模型训练速度快；适用于样本容量较大的文本分类。	计算复杂度高；样本数据不均衡时，分类效果不好。
朴素贝叶斯	利用贝叶斯公式根据特征的先验概率计算出后验概率，然后选择具有最大后验概率的类别作为预测样本类别。	算法简单；估计参数很少；对缺失数据不敏感。	需要先验概率；假设特征之间相互独立，导致分类效果不好。
决策树	在已知各种情况发生概率的基础上，通过构成决策树来求取净现值的期望值大于零的概率，来对样本进行分类。	计算复杂度低；结果易于理解。	容易出现过拟合问题；处理连续性数据比较困难；忽略了数据集中属性之间的相关性。
支持向量机	通过求解能够正确划分样本的最大间隔超平面，来对样本进行分类。	分类精准度高，灵活性强；可以解决非线性问题。	对缺失数据比较敏感；训练速度比较慢；寻求合适的核函数相对困难。

进行短文本分类任务，已经取得了很好的分类效果。目前，主要流行的深度学习分类器模型有CNN、RNN、LSTM、Bi-LSTM和FastText等。

1　卷积神经网络

2014年，Kim等［39］基于卷积神经网络CNN提出了TextCNN模型，如图6所示，该模型能够有效地提取文本的局部特征，在短文本分类任务中已经取得较好的分类效果。

TextCNN模型由输入层、卷积层、池化层、全连接层和输出层组成。在输入层，主要是将训练好的词向量送到模型中；卷积层会使用多个不同的卷积核，进行局部特征的提取工作；池化层会对卷积层输出的特征进行再次提取，同时实现降维的效果，减少训练参数；全连接层将池化层提取的全部特征连接在一起，然后使用softmax函数，输出相应分类类别的概率分布。

2　循环神经网络

循环神经网络（RNN）是一个擅于处理序列数据的模型，如图7所示。它能够将上一时刻的隐藏状态传递到当前时刻，并和当前时刻的输入一起决定下一时刻的输出，具有一定的记忆能力。但会存在梯度消失和梯度爆炸问题。

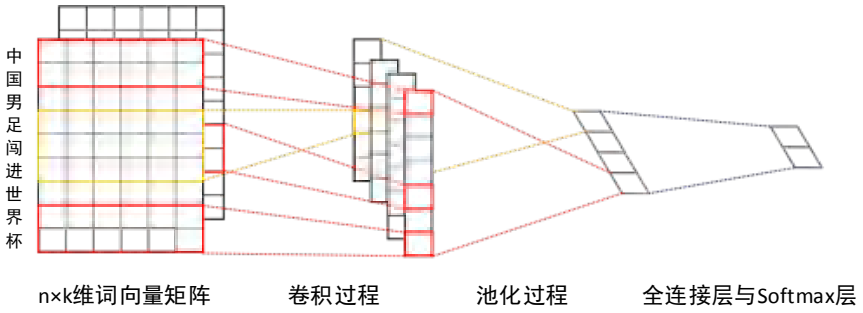


图6　TextCNN模型

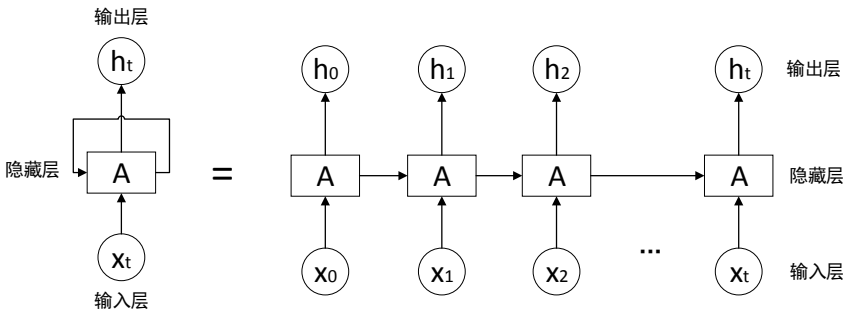


图7　RNN模型

长短期记忆网络（LSTM）是RNN的一个变体，如图8所示，它能够有效缓解RNN模型存在的梯度问题。相比RNN，LSTM增加了细胞状态（cell state），并通过门结构（gates）来决定保留或丢弃细胞状态中的哪些信息。

LSTM的三个门结构分别是输入门、遗忘门和输出门。遗忘门f_t决定要从细胞状态中丢掉哪些不重要的信息；输入门i_t决定那些信息需要被记录下来，并完成细胞状态的更新；输出门o_t会决定将细胞状态中哪些信息输出。

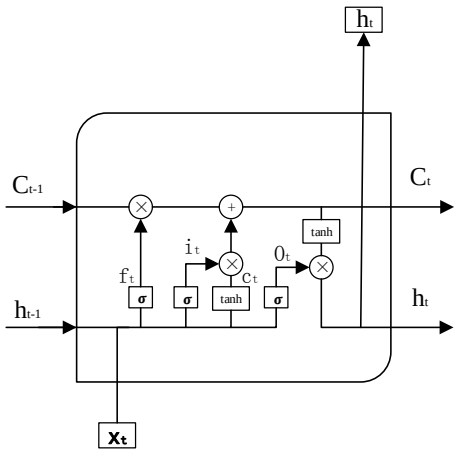


图8　LSTM模型

3　FastText模型

FastText［40-41］是Facebook公司开源的一款能够进行词训练和文本分类的模型，其架构与word2vec工具中CBOW模型相似，CBOW模型预测的是中心词，而FastText预测的是类别标签。该模型结构简单，仅通过浅层网络即可取得和深度网络相媲美的准确度，同时训练速度更快。

FastText模型由输入层、隐藏层和输出层组成，如图9所示。在输入层，FastText模型首先对输入的词序列进行词粒度的N-gram处理以获取更多额外的特征，然后对所有的词向量进行求和取均值。通过隐藏层的处理后，在输出层采用分层的Softmax函数进行分类。相比标准softmax，层次softmax通过构造霍夫曼树，能够大大提高分类速度。

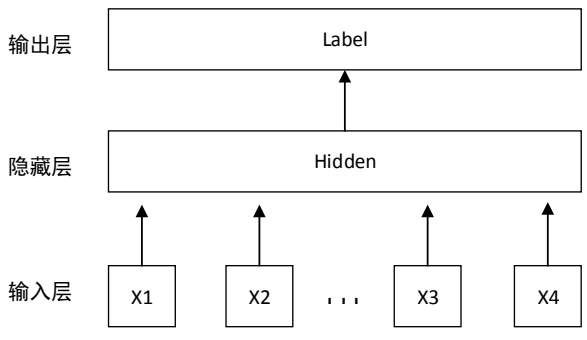


图9　FastText模型

4　Bert模型

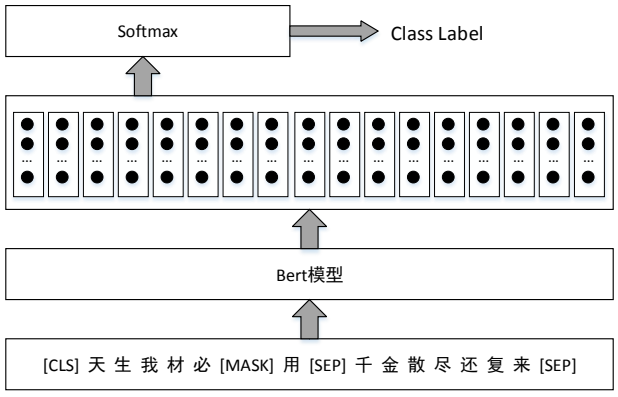


图10　基于Bert的短文本分类模型

基于Bert的中文短文本分类，只需要在Bert预训练模型的基础上添加一个Softmax层，就可以实现分类任务，并且能够取得非常优秀的结果，其模型结构如10图所示。该模型的输入是一个长度小于512的文本序列，且序列由一个或两个句子组成。通常以［CLS］为开头，用［SEP］分隔成两个句子。对于中文短文本分类任务，通常取［CLS］的隐藏状态代表整个句子的语义，然后经过Softmax层的处理即可完成分类任务。

4　评价指标与实验

4.1　评价指标

通常采用准确率、精确率、召回率和F1值来衡量一个文本分类模型的好坏。表2为二分类结果的混淆矩阵，共有以下4种情况：TP（True Positive，真阳性）表示预测是正类，实际也是正类的文本个数；FP（False Positive，假阳性）表示预测是正类，实际是负类的文本个数；TN（True Negative，真阴性）表示预测是负类，实际也是负类的文本个数；FN（False Negative，假阴性）表示预测是负类，实际是正类的文本个数。

表2　二分类结果混淆矩阵

真实	预测	P	F
	P	TP	FP
	F	FN	TN

1　准确率

准确率（Accuracy，简称Acc值）指的是结果预测正确的样本数量占全部结果的样本数量，表示公式如下：

Acc= （4）

2　精确率

精确率（Precision，简称P值）又称查准率，是较于预测结果而言的，指的是正类样本预测结果正确的数量占全部预测结果为正类的样本数量，表示公式如下：

P= （5）

3　召回率

召回率（Recall，简称R值）又称查全率，指的是正类样本预测结果正确的数量占全部正类样本预测的数量，表示公式如下：

R= （6）

4　F1值

F1值是精确率和召回率的调和平均值，由于精确率和召回率很难全面反映分类结果的好坏，故引入F1值综合性评判指标，表示公式如下：

F1= （7）

4.2　实验数据

本次实验选用THUCNews新闻标题数据集作为实验数据，它是一个中文类的数据集，每个标题字数约为20字，适合做中文短文本分类任务。我们从中选取了10类别的数据，分别为时政、财经、体育、科技、教育、娱乐、游戏、社会、股票和房产，每个类别选取了12000条数据，其中，10000条作为训练集，验证集和测试集各为1000条。

4.3　实验结果分析

为了对比TextCNN、BiLSTM、FastText和Bert四个分类模型在中文短文本上的分类效果，我们进行了实验验证。其中，dropout均设置为0.5，学习率均设置为0.001。然后，从准确率、精确率、召回率和F1值四个评价指标进行对比，结果如表3所示。

表3　实验结果对比

模型	准确率Acc (%)	精确率P (%)	召回率R (%)	F1值(%)
TextCNN	90.89	90.92	90.89	90.90
BiLSTM	90.79	90.78	90.79	90.78
FastText	91.86	91.90	91.86	91.88
Bert	94.33	94.36	94.33	94.34

从实验结果可以得出基于预训练模型Bert的分类效果最佳，并在四种评价指标上均高于其它三个模型，FastText模型的分类效果在四种评价指标上比TextCNN和BiLSTM高出约1%，其分类速度也比较快。目前，基于Transformer的预训练模型在短文本分类任务上能取得较好表现，主要是由于其采用了Attention机制，能够较好地提取短文本中的特征信息。

5　研究难点及对策

中文短文本由于字数少、特征稀疏以及信息不规范等特点，导致使用传统的文本分类技术难以取得好的分类效果。本节对中文短文本分类技术存在的问题和目前较好的解决方法进行总结。

1）由于短文本多歧义和信息不规范等特点，会造成中文分词任务中歧义消除和未登录词识别问题更加突出。基于序列标注的分词算法可以有效的解决歧义消除和未登录词识别等问题，随着深度学习算法在中文分词任务中的应用，采用深度学习模型和序列标注相结合的分词模型会取得更优的分词效果，像BiLSTM-CRF分词模型的准确率可达到95%以上。

2）由于短文本字数少以及特征稀疏等特点，导致难以提取出大量能够区分文本类别的特征。针对短文本特征表示不足的问题，常采用特征扩展的方式对文本特征进行补充。特征扩展的方法分为基于外部语料库来扩展特征和基于短文本自身内容扩展特征。基于外部语料库的特征扩展方式，通过引用维基百科、百度文库和知网等外部知识库来构建特征集，然后对短文本进行特征扩展；基于短文本自身内容的特征扩展方式，不需要借助外部语料库，主要是通过对短文本自身的

语义、词频等进行分析，改进文本特征表达方法，来实现对短文本特征的扩展。

3）由于短文本具有特征稀疏等特点，导致采用传统的文本表示方法会带来“维度灾难”等问题。Mikolov等提出了基于词嵌入的文本表示方法word2vec，此方法可以很好的解决“维度灾难”问题。由于word2vec无法对多义词进行准确表示，Peters等针对此问题提出了ELMo动态词向量表示模型，该模型通过结合前后文语境对多义词进行理解，从而很好的解决了多义词的表示问题。2018年，Devlin等提出了基于深度双向Transformer的Bert预训练模型，该模型采用Transformer能够很好的捕捉更长距离的依赖，提高了文本表示的效果。2019年，许多研究人员在Bert模型的基础上进行改进，提出了许多效果更好的预训练模型，例如ERNIE、spanBERT、RoBERTa、XLNET和ALBERT等。

4）采用传统的机器学习分类算法进行中文短文本分类，往往难以取得较好的分类效果。随着深度学习算法在计算机视觉和语音识别领域取得优秀的表现，研究人员将深度学习算法应用到中文短文本分类任务中，像目前流行的TextCNN、Bi-LSTM、FastText和胶囊网络等深度学习分类模型，都能取得较高的分类准确率。随着Bert模型及其变体的诞生，将预训练模型应用在短文本分类任务上能够取得更好的分类效果，这也是中文短文本分类技术未来发展的主流方向。

6 结束语

近年来，利用短文本分类技术挖掘数据中隐藏的信息是当前NLP领域的研究热点。本文首先分析了中文短文本分类技术的研究现状；然后按照短文本分类流程，对文本预处理、文本特征表示和分类算法等进行了详细的介绍；最后，介绍了中文短文本分类技术的研究难点和应对方案。在接下来的研究中，中文短文本分类技术可以在以下几个方面进行深入探索：

1）通过对短文本进行特征扩展或特征权重调整，来解决短文本特征稀疏等问题。目前，特征扩展方式主要分为基于外部知识库的特征扩展和基于自身内容的特征扩展，未来可以融合两者的优点进行短文本特征扩展。

2）对现有的基于深度学习分类模型和算法进行优化、融合或提出新的分类算法，来提高短文本分类的效果。

3）鉴于预训练模型在各种NLP任务上取得优秀的表现，充分挖掘其优势并将其应用在中文短文本分类任务上，是一个很好的研究方向。

参考文献：

[1] 曾剑秋,许海源. 垃圾短信分层分级治理研究[J]. 社会治理，2017(05)：49-55.

[2] 赵俊生,候圣,王鑫宇,尹玉洁. 基于集成学习的图像垃圾邮件过滤方法[J]. 计算机工程与科学，2020，42(06)：1049-1059.

[3] YaJun Du,YongTao Yi,XianYong Li,XiaoLiang Chen,YongQuan Fan,FangHong Su. Extracting and tracking hot topics of micro-blogs based on improved Latent Dirichlet Allocation［J］. Engineering Applications of Artificial Intelligence，2020，87.

[4] Kilimci, İlhan Omurca, S. Z. Extended Feature Spaces Based Classifier Ensembles for Sentiment Analysis of Short Texts. Information Technology And Control, 2018，47(3)：457-470.

[5] 赵浚淇. 基于自动分类的网络舆情监测方法研究[J]. 软件导刊，2016，15(03)：133-135.

[6] Chen H. Personalized recommendation system of e-commerce based on big data analysis［J］. Journal of Interdisciplinary Mathematics，2018，21(5)：1243-1247.

[7] 刘泽文,丁冬,李春文. 基于条件随机场的中文短文本分词方法[J]. 清华大学学报(自然科学版)，2015，55(08)：906-910+915.

[8] 张子睿,刘云清. 基于BI-LSTM-CRF模型的中文分词法[J]. 长春理工大学学报(自然科学版)，2017，40(04)：87-92.

[9] 李湘东,刘康,丁丛,高凡. 基于«知网»的多种类型文献混合自动分类研究[J]. 现代图书情报技术，2016(02)：59-66.

[10] 李湘东,阮涛,刘康. 基于维基百科的多种类型文献自动分类研究[J]. 数据分析与知识发现，2017，1(10)：43-52.

[11] 巴志超,朱世伟,于俊凤,魏墨济. 基于语义扩展的数字文献自动分类方法研究[J]. 现代情报，2015，35(09)：70-74.

[12] 吕超镇,姬东鸿,吴飞飞. 基于LDA特征扩展的短文本分类[J]. 计算机工程与应用，2015，51(04)：123-127.

[13] 邵云飞,刘东苏. 基于类别特征扩展的短文本分类方法研究[J]. 数据分析与知识发现，2019，3(09)：60-67.

[14] 张浩,钟敏. 基于Sentence-LDA主题模型的短文本分类[J]. 计算机与现代化，2019(03)：102-106.

[15] 黄梦婷,张灵,姜文超. 基于非负矩阵分解的短文本特征扩展与分类[J]. 计算机科学，2019，46(12)：69-73.

[16] 靳一凡,傅颖勋,马礼. 基于频繁项特征扩展的短文本分类方法[J]. 计算机科学，2019，46(S1)：478-481.

[17] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space［J］. Computer ence，2013.

[18] Mikolov T. Distributed Representations of Words and Phrases and their Compositionality［J］. Advances in Neural Information Processing Systems，2013，26：3111-3119.

[19] Peters M E , Neumann M , Iyyer M , et al. Deep contextualized word representations[C]// Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational

[20] Devlin J , Chang M W , Lee K , et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [J]. arXiv: 1810. 04805.

[21] Zhang Z , Han X , Liu Z , et al. ERNIE: Enhanced Language Representation with Informative Entities[J]. arXiv:1905. 07129.

[22] Yang Z , Dai Z , Yang Y , et al. XLNet: Generalized Autoregressive Pretraining for Language Understanding[J]. arXiv:1906. 08237.

[23] Lan Z , Chen M , Goodman S , et al. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations [J]. arXiv: 1909. 11942.

[24] 黄贤英,熊李媛,刘英涛,李沁东. 基于类别特征改进的KNN短文本分类算法[J]. 计算机工程与科学,2018,40(01):148-154.

[25] Qiang G . An Effective Algorithm for Improving the Performance of Naive Bayes for Text Classification [C]// Second International Conference on Computer Research & Development. IEEE, 2010.

[26] 刘春磊,梁瑞斯,邸元浩. 基于TFIDF和梯度提升决策树的短文本分类研究[J]. 科技风,2019(24):231.

[27] 王义真,郑啸,后盾,胡昊. 基于SVM的高维混合特征短文本情感分类[J]. 计算机技术与发展,2018,28(02):88-93.

[28] 李志杰,耿朝阳,宋鹏. LSTM-TextCNN联合模型的短文本分类研究[J]. 西安工业大学学报,2020,40(03):299-304.

[29] 尹春勇,何苗. 基于改进胶囊网络的文本分类[J/OL]. 计算机应用:1-7 [2020-08-17]. http://kns. cnki. net/kcms/detail/51. 1307. TP. 20200706. 1614. 018. html.

[30] 贾旭东,王莉. 基于多头注意力胶囊网络的文本分类模型[J]. 清华大学学报(自然科学版),2020,60(05):415-421.

[31] 丁辰晖,夏鸿斌,刘渊. 融合知识图谱与注意力机制的短文本分类模型[J/OL]. 计算机工程:1-8[2020-08-17]. https://doi. org/10. 19678/j. issn. 1000-3428. 0056734.

[32] 段丹丹,唐加山,温勇,袁克海. 基于BERT的中文短文本分类算法的研究[J/OL]. 计算机工程:1-12[2020-08-17]. https://doi. org/10. 19678/j. issn. 1000-3428. 0056222.

[33] 黄昌宁,赵海. 中文分词十年回顾[J]. 中文信息学报,2007(03):8-19.

[34] Peng F, Feng F, Mccallum A. Chinese Segmentation and New Word Detection Using Conditional Random Fields[J],2004.

[35] McCallumAndrew, FreitagDayne, Fernando C. PereiraN. . Maximum Entropy Markov Models for Information Extraction and Segmentation[C]// Seventeenth International Conference on Machine Learning. Morgan Kaufmann, 2000.

[36] Low J K , Ng H T , Guo W . A maximum entropy approach to Chinese word segmentation [J]. Proceedings of the Fourth Sighan Workshop on Chinese Language Processing, 2005.

[37] 张子睿,刘云清. 基于BI-LSTM-CRF模型的中文分词法[J]. 长春理工大学学报(自然科学版),2017,40(04):87-92.

[38] Bengio, Yoshua, Ducharme, et al. A Neural Probabilistic Language Model. [J]. Journal of Machine Learning Research, 2003.

[39] Kim Y . Convolutional Neural Networks for Sentence Classification [J]. Eprint Arxiv, 2014.

[40] Joulin A , Grave E , Bojanowski P , et al. Bag of Tricks for Efficient Text Classification [C]// Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. 2017.

[41] Bojanowski P , Grave E , Joulin A , et al. Enriching Word Vectors with Subword Information [J]. Transactions of the Association for Computational Linguistics, 2017, 5:135-146.

[42]

[作者简介]

刘硕（1996—），男，硕士研究生，主要研究方向为自然语言处理、短文本分类。

潘世东（1987—），男，硕士，工程师，主要研究方向为信息处理。

王庚润（1987—），男，博士，助理研究员，主要研究方向为电信网安全、数据处理。

李英乐（1985—），男，博士研究生，副研究员，主要研究方向为通信大数据挖掘。

基于高速SDI芯片封装基板的全链路信号完整性仿真与分析

毛英杰[1]，张波[3]，虎艳宾[2]，汪欣[3]

[1]天津芯海创科技有限公司，天津，100000；

[2]国家数字交换系统工程技术研究中心，河南郑州，450000；

[3]天津市滨海新区信息技术创新中心，天津，100000

摘　要：本文介绍了国内目前高速封装基板的设计现状，阐述了全链路信号完整性仿真的分析方法。基于频域S参数的理论，采用分段提取的方式，获得SDI（Software Defined Interconnection）芯片传输通道上各段的S参数，然后搭建全链路系统进行S参数级联仿真，从频域角度评估高速链路全通道的性能。同时，利用SDI芯片的发送端模型，从时域的角度分析眼图，评估信号经过高速链路各个阶段的变化情况。SDI芯片回片后，在实验室环境下测试眼图，并和全链路仿真结果对比，检验了仿真方法的正确性，构建出一套完整的全链路信号完整性仿真分析和验证方法，为后续更大规模、更高速率芯片产品的研制提供支撑。

Simulation and analysis of full link signal integrity based on high speed SDI chip package substrate

Mao Yingjie, Zhang Bo, Hu Yanbin, Zhang Bo

Abstract: In this paper, the design status of high-speed packaging substrates in China is introduced, and the analysis method of full-link signal integrity simulation is described. Based on the theory of frequency domain S parameters, the subsection interconnection was used to obtain the S parameters in each segment of the transmission channel of SDI (Software Defined Interconnection) chip. Then, the full-link system was built for S-parameters cascade simulation, so as to evaluate the full channel performance of the high-speed link from the perspective of frequency domain. At the same time, the SDI chip's transmitter model is used to analyze the eye diagram from the perspective of time domain and evaluate the changes of signals through various stages of high-speed link. After the SDI chip came back, we tested the eye diagram in the laboratory environment and compared with the full-link simulation results, the correctness of the simulation method is verified, and a complete set of full-link signal integrity simulation analysis and verification method is constructed, which provides reliable support for the subsequent development of larger scale and higher speed chip products.

一　国内高速封装基板设计现状

随着集成电路的发展，芯片的制程工艺越来越先进，在单位空间内能够集成更多的晶体管。在计算机、互联网和大数据等行业，为了提升系统的性能，芯片往往需要具备更高的工作频率和更快的信号传输速率，将大大增加高速封装基板的设计难度和复杂度。封装基板的主要功能是为Die芯片提供一个载体平台，对信号和电源进行分配，以减小不必要的信号延迟和电压损耗，传导并散发芯片工作产生的热量，为芯片和其他器件的互连提供牢固可靠的机械支撑，达到保护芯片的目的。由于产品功能提升的需求和工艺技术的进步，集成电路芯片封装有多种不同的形态和内部结构，根据Die芯片和基板之间的互连方式主要分为：引线键合（WB，Wire Bonding）、倒扣焊接

（FC，Flip Chip）和硅通孔（TSV，Through Silicon Via）等。受信号传输长度、电性能等限制，引线键合的封装形式，其引线会引入较大的寄生电感和延时，不利于高速高频信号传输。另外，引线键合的封装需要引线间保持相对于线径更宽的安全间距，降低了Bonding线的密度，减少单位空间内可引出的I/O信号线数量，无法进行大规模集成。硅通孔是连接硅晶圆两面并与硅基体和其他通孔绝缘的电互连结构，采用TSV技术能够提高系统集成密度，方便实现系统级的异质集成，为系统设计提供了模块化的可能性，但是工艺实现难度大，成本较高，被广泛应用于2.5D/3D等封装产品。

对于需要传输高频信号的高速封装基板，考虑单位空间集成密度、互连长度及信号完整性等指标要求，通常采用凸点倒扣球栅阵列的封装形式（FC-BGA，Flip Chip-Ball Grid Array），如图1所示。信号I/O通过Bump的方式分布在Die芯片表面，由于Bump的直径和高度相对较小，因此在单位空间内分布更多的I/O凸点，减小Die芯片和封装基板的连接长度。在基板内部，传输线可以按照设计要求控制阻抗，能够大幅提升高速高频信号的传输性能。封装时，通过将带有Bump凸点的Die芯片进行180°镜像翻转，使芯片有源区域面对基板，完成Bump与基板的焊接，实现芯片与基板的互连，将芯片上的I/O信号通过封装基板扇出到底部的Ball球。Bump凸点间距较小，FC-BGA的封装形态有利于I/O信号的高密度集成，封装基板分层设计，能够降低芯片厚度，并满足大功耗和高散热的需求。目前国内的封装基板设计，高速Lane数量少，而且速率大都在6.25Gbps以下，无法满足用户对高速网络和大带宽通信产品的强烈需求。

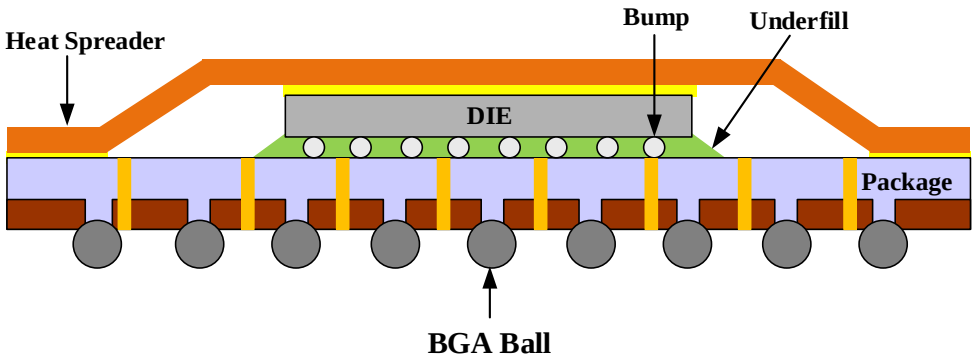


图1　凸点倒扣球栅阵列封装示意

本文基于的高速SDI芯片封装基板，高速Lane数量达到32条，单Lane最高速率可达10.312Gbps，向下支持多个不同的速率。SDI芯片功耗大，封装基板空间小，高速线数量多，信号分布密集，技术指标要求高，需要在有限的空间内综合考虑信号布局、布线、层叠划分、电源功耗、散热及SI/PI仿真优化等。同时，在系统传输的整个通道上，封装基板的走线长度只占很小一部分，为了降低系统设计风险，提升SDI芯片的性能，除了对封装基板的传输线性能进行信号完整性优化仿真外，对整个测试系统的传输线进行全链路的仿真和评估也尤为重要。

二　全链路信号完整性仿真介绍

信号传递最为理想的情况是接收端接收到的状态和发送端发送的状态完全相同，但实际上信号在整个链路的传输需要经过封装、PCB走线、过孔及连接器等部分才能到达接收端。因为链路传输线的内阻、互连结构的影响等，信号从发送端传递到接收端时，不可避免的会出现幅值衰减、相位变化及波形畸变等现象。由于信号、互连结构、电源系统等因素相互作用，最终使信号产生扭曲畸变的现象称为信号完整性（Signal Integrity，SI），主要包括噪声、干扰及其造成的时序影响等。在高速信号传输的过程中，上升/下降沿时间很短，信号在传输过程中会受到损耗、反射和串扰等因

素的影响，信号传递的准确性不能被忽略，由此带来的信号完整性问题也需要重点关注解决。

一个完整的全链路系统，主要由DIE芯片、封装和PCB等部分组成，如图2所示。随着芯片复杂程度和工作频率的不断提高，只单独考虑某一环节的性能无法满足对系统风险的评估判断，必须全面协同考虑芯片、封装和PCB板的相互作用。全链路条件下信号传输所经过的路径包含：芯片到封装、封装到PCB子板、子板到连接器等。全链路信号完整性仿真，需要分别提取链路各部分的模型，然后搭建一个完整的系统模拟整个链路的情况，利用仿真工具进行求解。

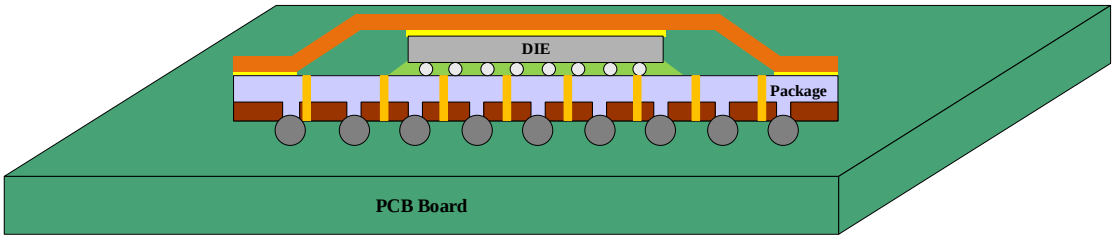


图2　Die-封装-PCB全链路结构示意

对于全链路的信号完整性仿真，主要是利用仿真工具对各部分的模型进行提取整合，分析全链路信号的插入损耗（Insertion Loss）、回波损耗（Return Loss）、串扰（Crosstalk）及时序等参数指标，确保信号从发送端经过传输通道到达接收端时，依然能保持一定的准确性和完整性，满足高速信号的传输要求。同时，利用芯片的有源仿真模型，例如IBIS-AMI，SPICE等，从时域角度分析信号在链路各部分的眼图变化，对高速全链路系统进行可靠评估，降低设计风险。

在速率GHz量级的高速信号传输系统中，传输线的插入损耗、阻抗不连续导致的反射和相邻传输线之间的串扰，会引起一系列的信号完整性问题，是系统设计所面临的主要挑战。本文通过高速SDI芯片封装基板设计，结合天津市滨海新区信息技术创新中心的SDI芯片测试板仿真，先分段提取基板和测试板的S参数，然后搭建全链路环境级联仿真，从频域无源的角度评估传输线性能。同时，利用SDI芯片Serdes TX发送端的模型，从时域有源的角度评估信号经过各部分的眼图变化情况，分析高速通道全链路的性能。

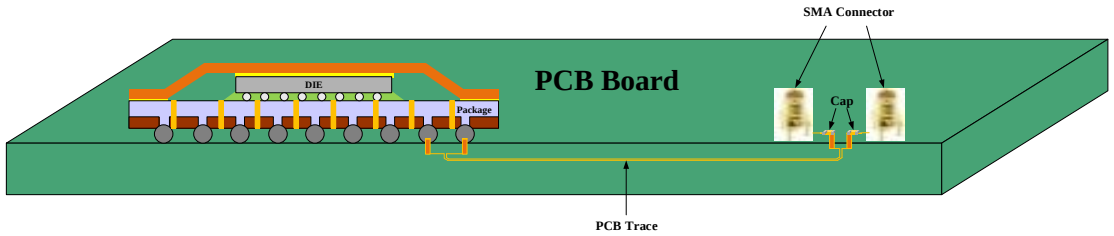
三　全链路各部分S参数提取

为了评估高速SDI芯片Serdes TX发送端的电参数是否满足协议规范，SDI芯片测试板采用SMA同轴连接器的方式引出对外接口，可直接用同轴线缆对接高带宽示波器和误码仪，测试芯片的性能。同时，在测试板的TX高速通道上，按照协议约束串接0.1uF的隔直电容。将SDI芯片封装基板和测试板视为一个全链路系统，如图3（a）所示，对系统内各模块进行划分，共分为四部分：SDI芯片Serdes TX发送端、高速封装基板走线、测试板PCB走线和SMA同轴连接器。建立全链路仿真环境时，利用SDI芯片Serdes TX发送端的Spice模型模拟产生数据源，封装基板、测试板和SMA同轴连接器的模型统一采用S参数格式，兼容业界标准，如图3（b）所示。其中，SMA同轴连接器的S参数模型由厂家提供，下面主要介绍SDI芯片封装基板和测试板的S参数模型提取方法。

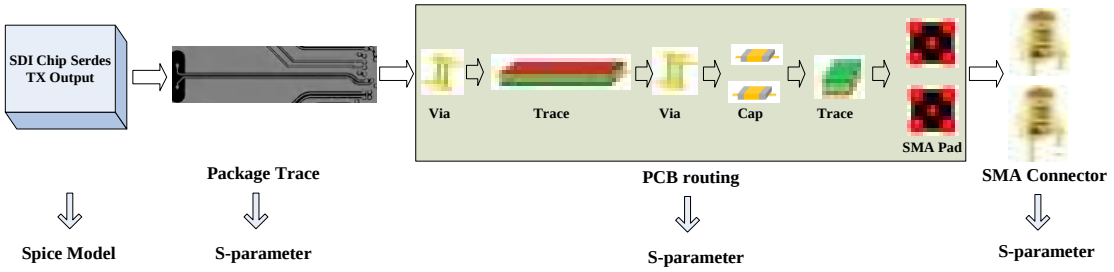
3.1　SDI芯片高速封装基板S参数仿真

SDI芯片具有32条高速lane，支持RapidIO 3.1、FC-AE-ASM 4.0、10GBase-KR、1000Base-X四种协议交换交换，并支持四种协议之间的混合协议转换和数据交换，单lane最高速率10.312Gbps，交换能力为320Gbps。由于SDI芯片高速线数量多，在封装基板信号布局时，将高速线均匀分布在四周，保证出线方向一致，减少基板层数。为了保证SDI芯片在封装基板的走线具有较小的插入损耗，基板材料选用低损耗因子的高速板材，在相同的走线长度下，信号的衰减幅度更小。SDI芯片封装基板的信号完整性仿真，主要通过对高速线走线长度、差分过孔的孔径、反焊盘、出线方式等进行调整，优化差分过孔的阻抗，提升传输线的性能。

SDI芯片的32条高速Lane在基板上走线基本



（a） 全链路拓扑示意



（b） 全链路各模块划分

图3 SDI芯片封装基板-测试板全链路拓扑及模块划分

一致，为评估封装基板走线的最大时延和最大损耗，选取基板走线相对较差的2条Lane作为仿真模型，如图4所示。

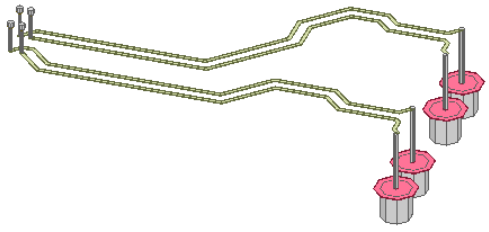


图4 SDI芯片封装基板高速线模型

将模型导入仿真软件，设置差分对组合、端口激励等仿真参数，然后整体检查模型，运行仿真提取S参数，结果如图5所示。从结果可以看出，仿真截止频率25GHz，在基频5.125GHz范围内，封装基板插入损耗小于-0.5dB，回波损耗大于-20dB，性能较好。

3.2 SDI芯片测试板S参数仿真

SDI交换芯片测试板将32条高速lane全部通过SMA同轴的方式引出，方便进行测试，如图6所示。

为了评估SDI芯片测试板性能最差的高速线，选取测试板上长度最长的走线进行仿真评估。仿真时，将选中的高速线从整块PCB中剪切下来，可大大缩短仿真时间。然后，设置叠层参数、差分对、产生端口、仿真约束等设置。仿真结束后，查看S参数结果如图7所示。

从结果可以看出，仿真截止频率25GHz，在基频5.125GHz范围内，SDI芯片测试板插入损耗小于-1.5dB，回波损耗大于-15dB，性能较好。

四 全链路级联仿真

4.1 全链路S参数级联仿真

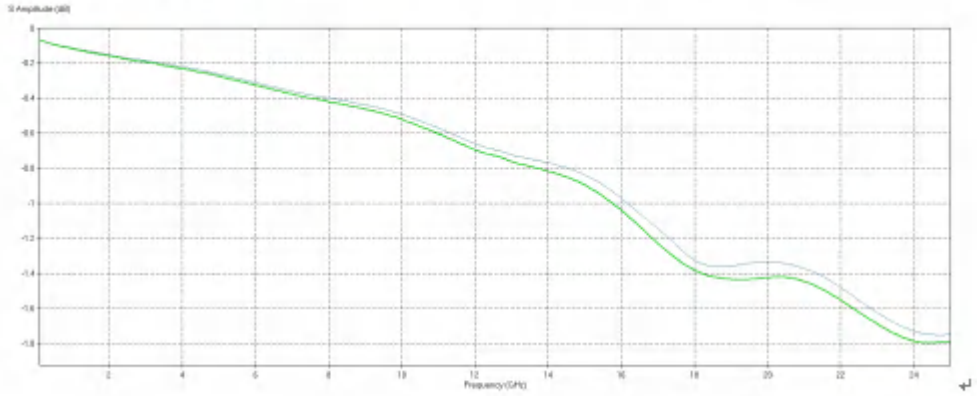
将SDI芯片的封装基板和测试板的S参数进行级联，搭建如图8所示的全链路拓扑。将链路各部分的S参数代入，分析评估全链路整体的传输性能，结果如图9所示。由于封装基板和SMA同轴的插入损耗均较小，从图9结果可以看出，全链路S参数级联后，插入损耗变化不大，说明封装和SMA同轴对信号传输的影响较小，二者引起的信号衰减很微弱。

4.2 全链路眼图级联仿真

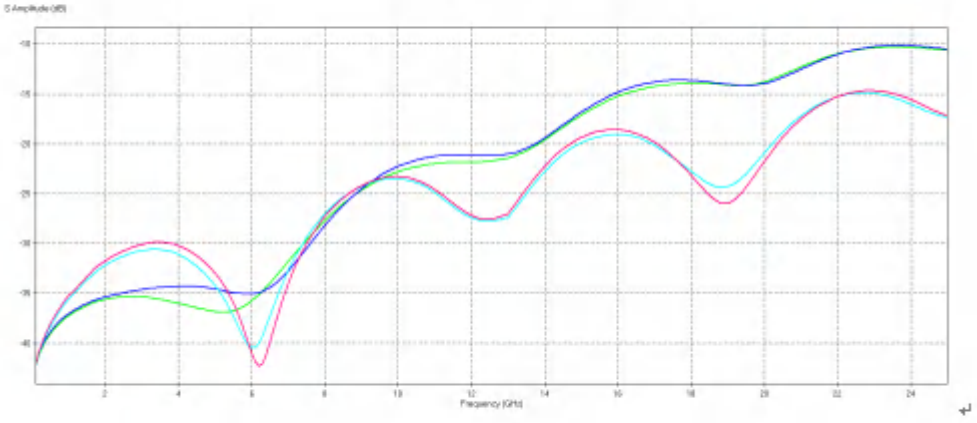
将SDI芯片发送端的模型代入全链路，搭建如图10所示的拓扑结构。利用发送端的模型模拟信号源，以6.25Gbps速率为例，在全链路不同的位置放置示波器探针分析各部分的眼图情况，如图10所示。

从各个位置的眼图仿真结果可以看出：

□ 封装基板TX走线的插损很小，信号经过前后，幅度基本不变。但是，基板TX和测试板连接



(a)插入损耗



（b）回波损耗

图 5 SDI 芯片封装基板 S 参数

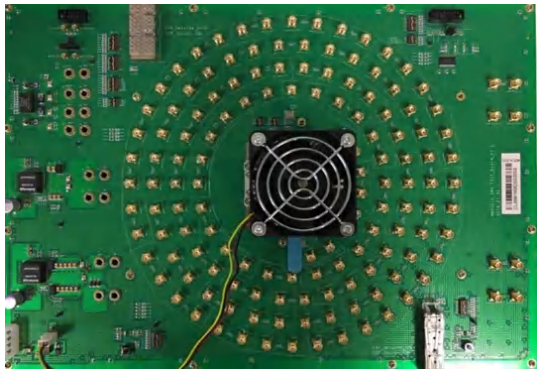


图6　SDI 交换芯片测试板实物

处，二者的 S 参数不同，信号从发送端经过封装后，阻抗不连续导致反射增加，EyeDiff_Probe2 位置眼图幅度噪声增大；

□ 由于 SDI 芯片测试板 PCB 传输线插入损耗的影响，信号在经过测试板前后的信号摆幅明显降低。在 SDI 全链路传输通道上，相对于封装基板，测试板的走线较长，是信号传输幅度衰减的主要原因；

□ 封装基板 TX 走线的插入损耗很小，信号经过前后基本无影响，跟频域 S 参数的仿真结果一致。

通过上述分析，在 6.25Gbps 速率条件下，信号经过全链路通道后，眼图的眼高、抖动、幅度噪声等均有较大余量，SDI 芯片的全链路系统性能较好，可以满足高速信号传输要求。

五　全链路级联实验室测试

SDI 芯片封装完成回片后，为了测试实际性能，利用 SDI 芯片测试板在实验室环境下测试 6.25Gbps 速率时，SMA 同轴连接器位置的眼图，并和全链路仿真 EyeDiff_Probe3 位置的眼图进行对比，如图 11 所示。结果显示，在测试板 SMA 同轴位置，幅值、眼高、眼宽和抖动等眼图指标，实测和仿真基本吻合。

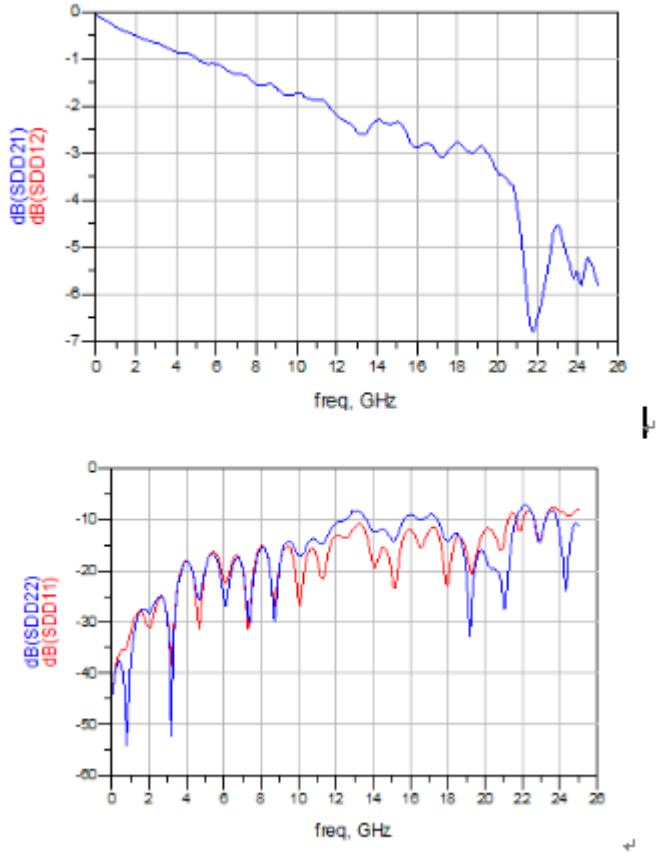


图 7 SDI 芯片测试板 S 参数

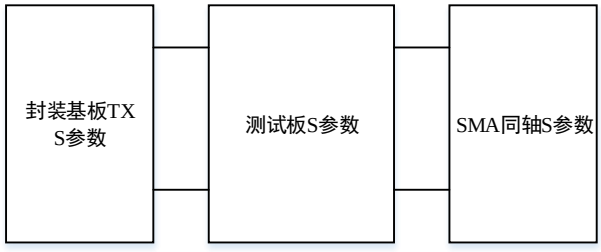


图8　全链路 S 参数级联拓扑

在全链路仿真时，为了模拟信号的传输环境，需要在末端加入 50Ω 的端接电阻，否则将被视为末端开路。而实际测试时，在 SMA 同轴位置直接引出测试线缆，没有考虑 50Ω 端接的影响，所以仿真和实测结果依然有差异。通过实测眼图结果的对比，从频域和时域两个维度均验证了全链路信号完整性仿真方法的可信度和正确性。

六　结束语

当前大规模集成芯片发展迅速，高速封装基板面临的信号完整性问题越来越严峻，不能只关注系统某一部分的传输性能提升，需要从整体出发对全链路的信号完整性进行合理设计和仿真优化，以降低项目风险。本文概述了国内高速封装基板的设计概况，介绍了全链路信号完整性仿真的实现方法，结合 SDI 芯片高速封装基板和测试板设计，搭建全链路仿真拓扑，从频域和时域分析高速链路通道的性能，有力保障了 SDI 芯片产品一次性设计成功。同时，在实验室环境下，用高速示波器进行眼图测量，仿真和实测结果吻合，证明了全链路信号完整性仿真的有效性。从仿真到测试形成闭环，构建了一套全链路信号完整性仿真分析和验证方法，为后续更大规模、更高速率产品的研制提供参考。

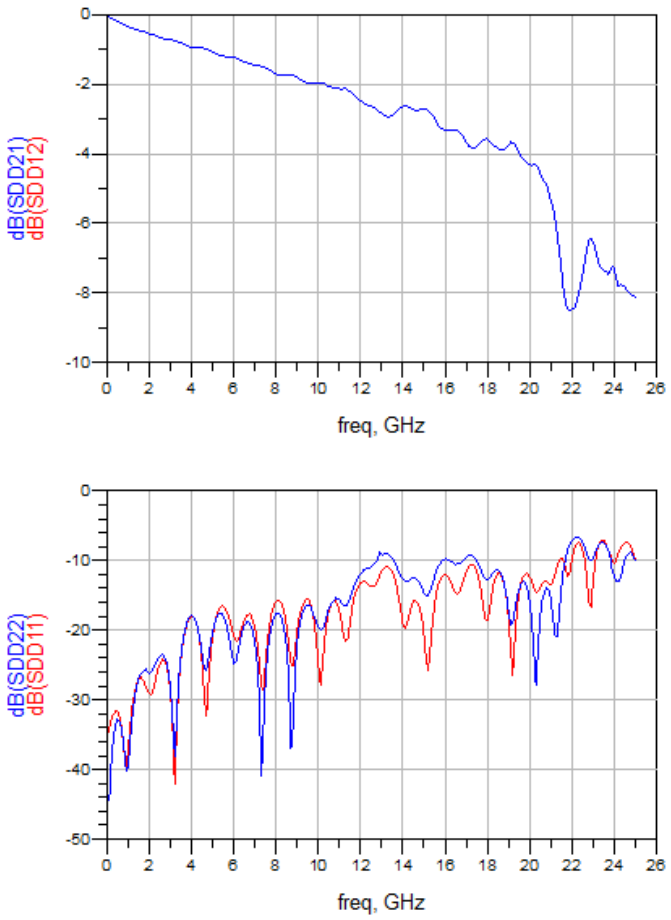


图9　全链路S参数级联仿真

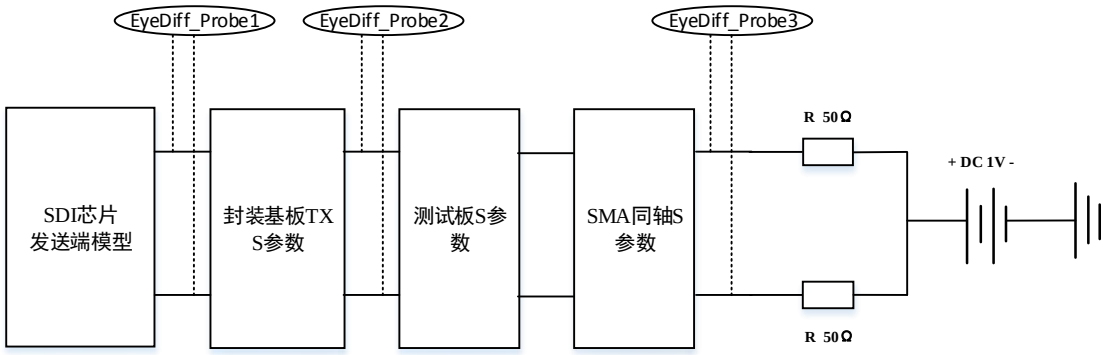


图9　全链路时域仿真级联拓扑

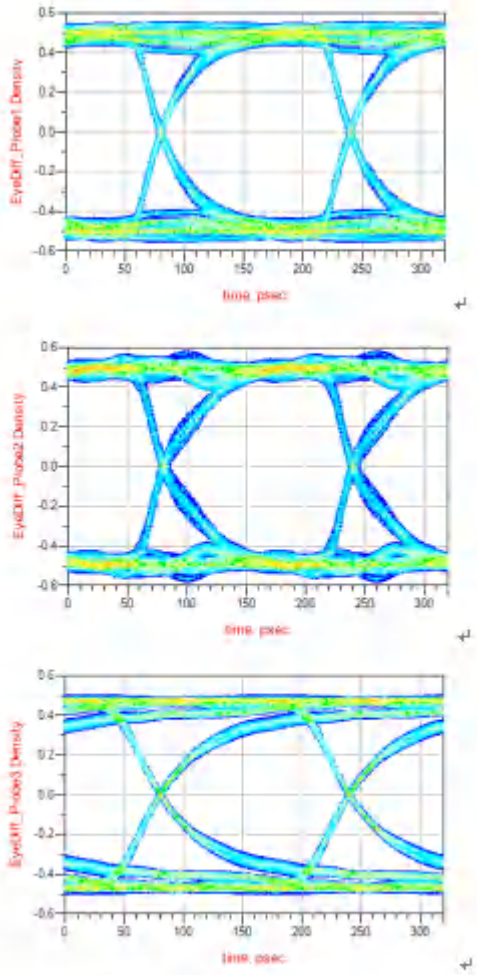
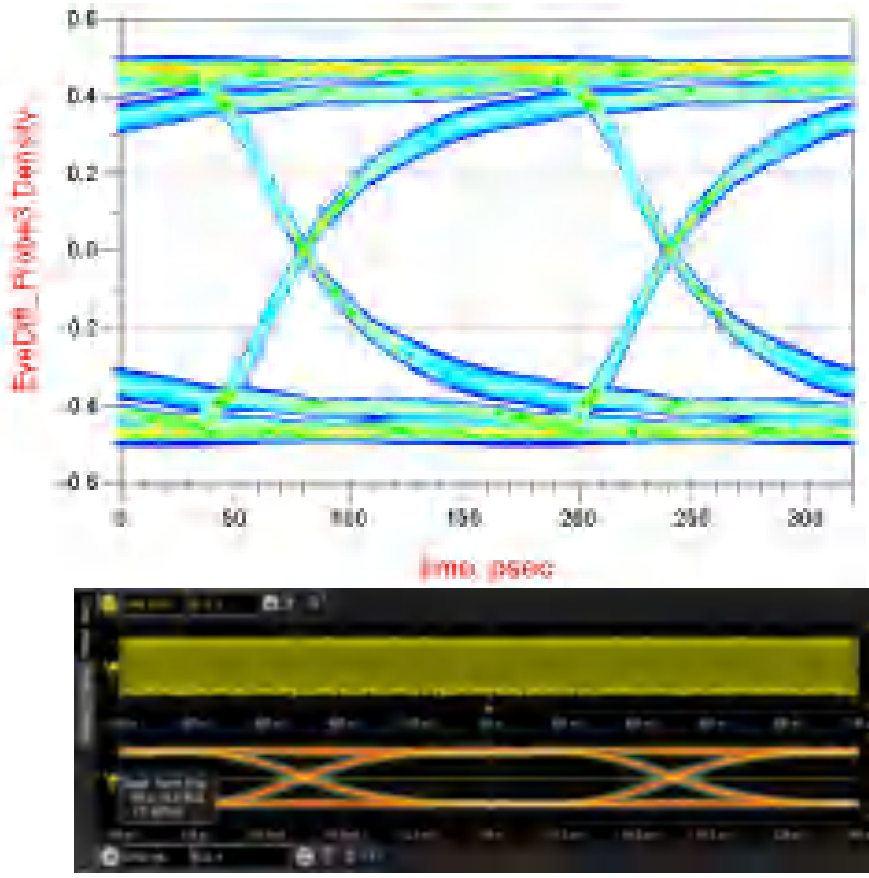


图 10 全链路各位置眼图



仿真眼图

实测眼图

图 11　测试板 TX SMA 同轴位置眼图对比

参考文献:

[1] 吕平,刘勤让,邬江兴,等,新一代软件定义体系结构[J]. 中国科学:信息科学,2018(3)

[2] EricBogatin,信号完整性与电源完整性分析(第 3 版)[M]. 电子工业出版社，2019

[3] Keysight,是德科技高速数字信号光电测试平台

[4] 张涛,ADS 高速电路信号完整性应用实例[M]. 电子工业出版社，2015

[5] Keysight，E5071C ENA Network Analyzers User’s Guide

[6] 房丽丽,章传芳,ANSYS 信号完整性和电源完整性分析与仿真实例[M]. 中国水利水电出版社，2018

[7] Stephen H. Hall,Howard L. Heck,高级信号完整性技术[M]. 电子工业出版社，2015

[8] Joel Dunsmore,微波器件测量手册 [M]. 电子工业出版社，2014

[9] 朱辉、冯云,实用射频测试和测量[M]. 电子工业出版社，2016

[10] 于争,信号完整性揭秘:于博士 SI 设计手记[M]. 机械工业出版社，2013

[11] 毛军发、唐旻,高速集成电路互连[M]. 科学出版社，2017

[12] 陈留国,PCB 高速信号完整性的分析[J]. 信息通信，2015(5)

[13] 谢拥军、刘莹、李磊、丁海强、雷振亚,HFSS 原理与工程应用[M]. 科学出版社，2009

[14] 李可为,集成电路芯片封装技术(第 2 版)[M]. 电子工业出版社，2013

[15] 毛忠宇、杨晶晶、刘志瑞、李生,信号、电源完整性仿真设计与高速产品应用实例[M]. 电子工业出版社，2018

[16] SalmanEmre,G. FriedmanEby,高性能集成电路设计[M]. 电子工业出版社，2015

[作者简介]

毛英杰（1990-），男，硕士，中级工程师，主要研究方向：高速封装基板/测试板设计、SI/PI 仿真及测试。

张波（1986-），男，硕士，高级工程师，主要研究方向：大型复杂系统的硬件设计开发，高速板卡设计。

虎艳宾（1978-），女，硕士，助理研究员，主要研究领域：高速电路设计、芯片封装及测试系统平台设计。

汪欣（1986-），男，硕士，高级工程师，主要研究方向：高速交换电路设计、大规模集成电路设计。

一种拟态web应用程序的安全性测试方法

曲晟，张铮，邢福康，邵昱文，季新生
信息工程大学，郑州 450001

摘　要：由于目前对web应用程序的安全性测试主要是通过安全测试员手动输入或利用漏洞扫描工具完成的，手动测试耗时、费力、对安全测试员的能力要求较高；而漏洞扫描工具的功能欠优化、与测试的内容可适配程度低、也没有针对web应用程序的输入进行测试。本文介绍了一种模糊测试的方法，半自动的对web应用程序的输入进行安全性评估测试。针对拟态web应用程序的安全性测试方法，首先就是对web应用程序进行拟态化改造，然后研究web应用程序的模糊测试方法，最后对定制的拟态web应用程序中可能存在安全漏洞的输入进行模糊测试和结果分析，并通过对比模糊测试的结果评估得到定制拟态化改造后web应用程序的安全性提高。

关键词：模糊测试、拟态化改造、web应用程序、安全漏洞

A Security Test Method for Mimic Web Applications

QU Sheng, ZHANG Zheng, XING Fukang, SHAO Yuwen, JI Xinsheng
Information Engineering University, Zhengzhou 450001, China

Abstract: Since the current security testing of web applications is mainly done through manual input by security testers or using vulnerability scanning tools, manual testing is time-consuming, laborious, and requires high security testers' capabilities; while vulnerability scanning tools lack functionality Optimization, low adaptability to test content, and no testing for web application input. This article introduces a method of fuzzing, which semi-automatically evaluates and tests the security of web application input. For the security testing method of mimic web applications, the first is to modify the web application mimic, then the fuzzing testing method of web applications is studied, and finally the input that may have security vulnerabilities in the customized mimic web application is fuzzed Test and result analysis, and through the comparison of the fuzzing test result evaluation, the security of the web application after the customized mimic transformation is improved.

Key words: fuzzing testing; mimic modification; web application; security vulnerabilities

1　引言

随着进入web3.0时代，网站功能越趋复杂，尤其是互联网的广泛普及和应用，如电子政务、电子商务、网络办公、网络媒体以及虚拟社区的出现，正深刻影响着人类生活、工作的方式[1]。与此同时，web安全的重要性也在不断提升。网站功能越复杂就意味着更多的漏洞暴露于互联网之上，政府、企业各类组织所面临的web安全问题越来越多样化、复杂化，黑客威胁正在飞速增长，给企业的信息网络造成严重的破坏甚至于不可挽回的损失。因此提高web应用程序的安全性是解决web安全问题的重中之重。将web应用程序进行定制的拟态化改造可以提升web应用程序的安全性。

基于拟态防御模型构建的拟态web服务器能够在较小开销的前提下防御测试中全部攻击类型[2]。此安全性测试是由专业的安全测试人员对拟态web应用程序进行扫描探测、漏洞挖掘、攻击植入等测试项，一般是模拟黑客恶意入侵的方式对拟态web应用程序进行测试，不仅对测试者的执

基金项目：国家自然科学基金资助项目（No. 61521003），国家重点研发计划资助项目（Grant No. 2018YF0804003，Grant No. 2017YFB0803204）。

通讯作者：张铮（ponyzhang@163.com）

行力要求很高，测试成本高，还难以被大规模应用。而模糊测试通过向拟态web应用程序提供非预期的输入并监控输出中的异常来发现拟态web应用程序中缺陷，整个执行过程依靠工具进行自动化或半自动化测试。模糊测试技术具有自动化程度高、可用性好、误报率低，、对目标程序源码没有依赖等优点[3]。在对web应用程序进行定制的拟态化改造后，通过模糊测试的方法可以快速的对拟态web应用程序的安全性做出评估。因此对拟态web应用程序提出了一种基于模糊测试的安全性测试方法研究，并运用到拟态web应用程序中。

本文第2节介绍web拟态化改造的相关工作以及拟态化改造后对web应用程序的安全性进行模糊测试的相关工作；第3节介绍拟态web应用程序进行模糊测试的具体方法实施；第4节通过对拟态web应用程序的模糊测试的分析，证明拟态web应用程序的安全性；最后讨论今后可能的研究方向和前景，并总结全文。

2 相关工作

在拟态web改造后，需要对拟态web应用程序的安全性进行测试。现有的对拟态web应用程序安全性测试的方法主要是通过安全测试员手动的、凭借自身经验来进行安全性测试，只从直观的安全防御效果来评判拟态web应用程序是不充分、不全面、不科学的，缺少具有全面的安全性测试来评估拟态web应用程序的安全性[4]。模糊测试是发现软件安全漏洞最广泛部署的技术之一，一般是一个自动或半自动的过程，这个过程包括反复操纵目标软件并为其提供处理数据[5]。它是一种通过提供非预期的输入并监视异常结果来发现软件故障的方法。因此将web应用程序安全性模糊测试的方法运用在拟态web应用程序中，可以全面的评估安全性。

2.1 web拟态化改造相关工作

根据2019年web应用安全报告可以得出，web应用安全依然是互联网安全的最大威胁来源之一。一方面，传统的SQL注入、XSS、命令行注入攻击等传统攻击手段和各种新爆出的web漏洞随时在考验着web应用安全方案的健壮性、灵活性；另一方面，随着大数据技术和流量产业的成熟，爬虫也成为一个不容忽视的存在，伴随而来的数据泄露、流量作弊等问题也让web应用不堪重负。拟态web服务器可以针对以上问题，做出安全的web应用程序提供给厂商、用户使用。通过定制的web应用程序拟态化改造可以修复web应用层中PHP源代码、SQL数据库等安全漏洞。本小节简要介绍web应用程序定制的拟态化改造及web服务器自动化部署的内容。

2.1.1 Web应用程序拟态化改造

Web应用程序的拟态化改造主要针对Web应用程序中的SQL语句和PHP代码。

首先是对SQL语句的拟态化改造。数据库指令异构化模块包含两个子模块：SQL保留字指纹化模块、注入指令过滤模块。SQL保留字指纹化模块对web应用程序中SQL保留字进行指纹化处理，实现web应用程序SQL指令的特征化；注入指令过滤模块依据指纹对数据库读写操作指令进行过滤，剔除攻击者注入的非法指令。

其次是对PHP代码的拟态化改造，基于指令集随机化的防御思想是修改服务器端代码解释器的词法或语法， 并对程序源代码进行一致性的随机化变换。攻击者不了解当前服务器端采用何种随机化变换规则， 因此由外部注入的恶意代码与可执行代码不一致， 无法成功执行[6]。

2.1.2 Web服务器拟态化改造

对web服务器拟态化的改造是基于对已有的防御技术和网络攻击本质的分析，提出的基于“动态异构冗余”结构的拟态防御技术.拟态防御相比主流的防御技术有着更大的防御面，尤其能够防御基于未知漏洞的攻击[2]。如图1所示，为拟态web服务器改造的3模冗余的拟态防御模型。以此模糊为基础，对web服务器进行自动化部署的拟态化改造。

执行体集包含三个执行体池，每个执行体池实现服务器软件、操作系统等多层次的异构化。对于拟态web服务器的自动化部署主要是在服务器软件和php的自动安装执行。通过python中的fabric库编写自动化部署脚本，实现对拟态web服务器执行体池中每个执行体的自动化部署。

2.2 模糊测试相关工作

2.2.1 web应用程序模糊测试

web应用程序模糊测试是一种特殊形式的网络协议模糊测试。网络协议模糊测试可将任意的网

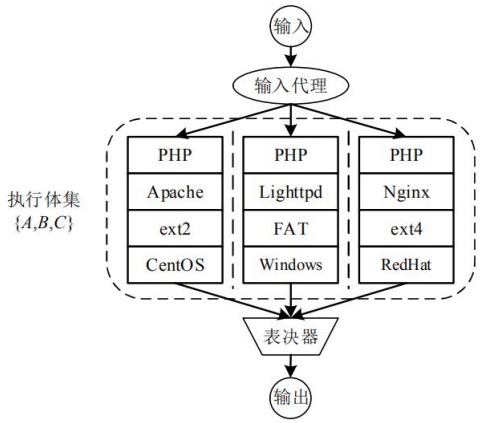


图 1　拟态 web 服务器改造的 3 模冗余的拟态防御模型

络包进行变异，而 web 应用程序模糊测试关注的是遵循 HTTP 规范的包，对 HTTP 请求-响应协议中的信息进行模糊 [7]。web 模糊测试有不同的输入，这些不同的输入将成为模糊测试活动的目标。发送给 web 服务器的请求数据的任何部分都可以认为是一个模糊测试的输入（HTTP 协议的请求方法，请求统一资源定位符，HTTP 协议版本，所有的 HTTP 头以及发送的数据等，每个部分都有合适的模糊测试的输入）。对 web 应用程序的模糊测试流程，如图 2 所示。

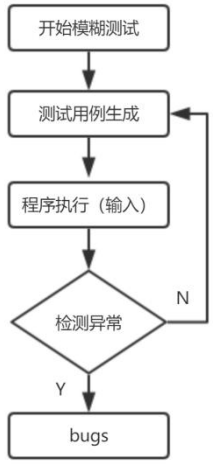


图 2　web 应用程序的模糊测试流程图

HTTP 协议中有 8 种 web 页面的请求方法，其中最常见的是 GET 和 POST，当对 web 页面请求时，这些请求方法都可以用作 web 应用程序模糊测试的输入。当对请求统一资源定位符模糊测试时，URL 的每个部分都可以被模糊化。所有请求的头都可以模糊化，发送数据的内容也可以模糊化。因此我们对 web 应用程序的模糊测试主要就是找到 web 应用程序所有的输入：web 页面、目录、web 页面所支持的方法、web 表单、名-值对、头信息、cookie 等。

web 应用程序易于遭受许多类型漏洞的攻击，所有这些漏洞都可以通过模糊测试来加以识别：拒绝服务（DoS），跨站点编写脚本（XSS），SQL 注入，目录遍历，弱认证，缓冲区溢出，远程代码注入，跨站点请求伪造（CSRF）等。

对 web 应用程序模糊测试的结果进行异常检测时，根据 HTTP 状态码、web 服务器错误消息、中断连接、日志文件、事件文件、调试器等信息，可以识别潜在的安全漏洞条件。

2.2.2　拟态 web 应用程序模糊测试

根据上述对 web 应用程序模糊测试的介绍，提出了针对拟态 web 应用程序的安全性测试方法——模糊测试。在开始对一个拟态 web 应用程序进行模糊测试之前，首先要建立目标环境，然后为目标环境选择输入向量。模糊测试通常要求能够快速、连续地向目标程序发送大量的输入。一个单一输入的事件序列包括：在本地产生输入，将输入发送到目标应用，允许目标应用对输入进行处理，监视输出结果。因此，运行一个模糊器所需要的时间也就由该序列中运行速度最慢的环节所决定。当对一个本地应用进行模糊测试时，整个测试过程的瓶颈是 CPU 周期以及硬件的读/写时间。在现代计算机硬件速度的支持下，这些时间可以减少到最小，因此模糊测试对于研究拟态 web 应用程序的漏洞是一个可行的方法。对于拟态 web 应用程序模糊测试而言，瓶颈点在于由模糊器向目标应用程序所进行的网络包的传输。

从远程地点加载一个 web 页的过程，当浏览一个 web 页时，页面的显示速度是由以下三个速度来决定的：本地计算机，该页所在的服务器，以及位于二者之间的 Internet 连接。因此，当对一个拟态 web 应用程序进行模糊测试时，通过去除其他两个变量来提高网络通信的速度是非常重要的。

将非拟态 web 应用程序和拟态 web 应用程序分别部署到本地服务器上，并用本地服务器的 Burpsuite 渗透测试工具，导入 SQL.txt、Traversal.txt 等 payloads 来进行 web 应用程序的模糊测试。这些 payloads 是在 WFuzz 模糊测试软件的模糊测试集的

基础上，经过搜集一些手动安全测试的用例，最终完善成为进行拟态web应用程序模糊测试的payloads。最后，通过分析拟态前后web应用程序的模糊测试结果来对比判断定制的拟态化改造的有效性。

我们测试一个web应用程序——DVWA。DVWA是一个很容易受到攻击的PHP/MySQL web应用程序，在非拟态化的DVWA中有4种安全等级：Low、Medium、High、Impossible，其中Low级别是较为容易被利用的web安全漏洞。因此我们将对low级别的SQL Injection和File Upload进行模糊测试，来发现web应用程序的PHP代码和SQL数据库的安全漏洞。再对定制拟态化改造后的DVWA进行相同的模糊测试，查看PHP代码和SQL数据库的安全漏洞是否被修复。

3 拟态web应用程序模糊测试方法

拟态安全防御原理旨在提高系统的安全性[4]，拟态web应用程序正是在研究了传统web应用程序及其经常面临的安全漏洞的基础上，结合拟态安全防御的基本思想研制而成的基于修复web应用层漏洞的web应用程序。因此，web应用程序的安全性测试成为测试工作中的关键。

3.1 模糊测试原则

基于拟态防御原理的web应用程序作为新型的web安全防御手段，为验证其防御效果的有效性，保证测试结果的完整性和客观性，测试过程中制定以下原则：

（1）web应用程序必须在拟态化改造后保持功能正常；

（2）受测的web应用程序和拟态web应用程序所处硬件环境和软件环境一致；

（3）模糊测试均在web服务器本地进行；

（4）模糊测试的有效载荷一致。

3.2 模糊测试通用算法

模糊测试与安全性相关。从理论上讲，测试过程中除了关注安全漏洞之外，还需要关注其他。但是，在实践中使用的技术往往各不相同。在设计测试工具时，通常需要访问源代码和了解一些关于web请求和响应的知识。这样的假设通常会推动测试工具的开发，使其具有与模糊测试工具不同的特性，模糊测试工具更有可能被用在其他方面。

Valentin[8]等人提出了一个模糊测试的通用算法，如表1所示；

表1 模糊测试的通用算法

通用算法：模糊测试
Input：C, t_{limit}
Output：B //一个有限的缺陷集
1 B←Ø
2 C←PREPROCESS(C)
3 while $t_{elapsed} < t_{limit} \wedge$ CONTINUE(C) do
4 conf←SCHEDULE(C, $t_{elapsed}$, t_{limit})
5 tcs←INPUTGEN(conf)
6 B', execinfos←INPUTEVAL(conf, tcs, O_{bug})
7 C←CONFUPDATE(C, conf, execinfos)
8 B←B∪B'
9 return B

Preprocess函数：可以执行各种操作，比如将插装代码插入put，或者测量种子文件的执行速度。

Schedule函数：接收当前的Fuzz configuration集合，当前时间telapsed和超时时间tlimit作为输入，并选择用于当前模糊迭代的模糊配置。

Inputgen函数：接受一个模糊配置作为输入，并返回一组具体的测试用例tcs作为输出。

Inputeval函数：接受一个模糊配置conf、一组测试用例tcs和一个Obug（bug oracle）作为输入。然后输出发现的一组bug B '，并运行关于每个模糊的信息（execinfos），这些信息可用于更新模糊配置。

Confupdate函数：接受一组模糊配置C，当前配置conf，并将关于每个模糊运行execinfos的信息作为输入。

Continue函数：接受一组模糊配置C作为输入，并输出一个布尔值，指示是否应该进行新的模糊迭代。这个函数对于建模白盒模糊器非常有用，当没有更多路径可以发现时，白盒模糊器就会终止。

3.3 模糊测试过程

在第2.2节中，描述了web应用程序的模糊测试的流程。本小节将对web应用程序（DVWA）模糊测试的具体过程做出阐述。

首先对web应用程序中的所有输入进行查找和统计，然后获取URL查询字符串，请求主体及HTTP cookie等相关参数，如图3所示；并对这些

参数进行模糊测试，分析模糊测试结果得出漏洞是否存在。

```
GET /vulnerabilities/sqli/?id=§1§&Submit=Submit HTTP/1.1
Host: 192.168.134.131
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.134.131/vulnerabilities/sqli/
Cookie: PHPSESSID=r9o4f1bp4fn5h98q47550apvk5; security=low
Connection: close
Upgrade-Insecure-Requests: 1
```

图 3　web 应用程序的请求

对图 3 的请求进行分析，id 参数可以作为模糊测试的对象。此请求是 SQL 注入的输入框，目前用基于生成的模糊测试有效载荷集对 id 参数进行模糊测试，对响应的结果进行分析得出是否存在安全漏洞的结论。

3.4　模糊测试结果

对 OS 命令行输入模块、SQL 查询模块和、件上传模块和 URL 路径遍历分别进行了基于生成的模糊测试。在对它们的模糊测试结果进行异常检测时，根据 web 应用程序响应字符串的长度信息、HTTP 状态码、web 服务器错误消息等，来判断 web 应用程序是否存在安全漏洞。

Web 应用程序模糊测试结果，如表 2 所示；

表 2　Web 应用程序模糊测试结果

测试内容	测试类别	状态码异常占比/%	响应长度异常占比/%	是否存在漏洞
Web 应用程序模糊测试	OS 命令行注入	0	19.3	是
	SQL 注入	0	11	是
	文件上传	0.9	0.9	是
	路径遍历	0	13	是

拟态 web 应用程序模糊测试结果，如表 3 所示；

表 3　拟态 web 应用程序模糊测试结果

测试内容	测试类别	状态码异常占比/%	响应长度异常占比/%	是否存在漏洞
拟态 Web 应用程序模糊测试	OS 命令行注入	0	1.6	否
	SQL 注入	0	4	否
	文件上传	0.9	0.9	否
	路径遍历	0	0	否

4　结果分析

在第 3 节，对 web 应用程序和拟态 web 应用程序的模糊测试结果进行归纳和汇总。异常状态码就是在对输入的参数进行模糊测试的过程中 web 服务器发生错误而导致的无服务状态，此类状态的响应码为 500 及以上，异常响应长度是在确定正常输入的响应长度后，与正常响应长度的值相差较大的响应长度被认为是异常响应长度 [9]。由此分析 web 应用程序拟态化前后的 OS 命令行注入、SQL 注入、文件上传、路径遍历的漏洞存在情况。

OS 命令行注入，普通 web 应用程序和拟态 web 应用程序都没有出现异常的状态码，但是普通 web 应用程序的响应长度出现异常的占比远高于拟态 web 应用程序。这是因为基于拟态防御原理的冗余策略，采用不同的操作系统作为服务端，OS 命令对于不同的操作系统有大概率是不一样的，少部分对于 web 应用程序的安全无影响的命令可以执行，经过拟态 web 服务器的裁决机制就会将这类命令过滤。

SQL注入，普通web应用程序和拟态web应用程序都没有出现异常的状态码，但是同样是普通web应用程序的响应长度出现异常的占比远高于拟态web应用程序。因为拟态web应用程序的SQL语句采用了指纹化，只有SQL语句符合该指纹化才能正常查询，否则会被过滤。

文件上传，普通web应用程序和拟态web应用程序都出现了相同占比的状态码异常和响应长度异常情况，但是经过对异常的有效载荷进行手动复现时，只有普通web应用程序成功复现了漏洞，而拟态web应用程序没有复现。因为拟态web应用程序的web服务器采用拟态防御原理中异构策略，不同的web中间件存在着不同的安全漏洞，将这些存在安全漏洞的web中间件构成一个集合，采用裁决机制会将属于其中一个中间件的漏洞无法复现。路径遍历漏洞也是同样的策略，拟态web应用程序中，每个执行体的网站路径会有不同文件，在路径遍历的响应中会产生不同长度的响应，而不同长度的响应会触发裁决，使得无法进行路径遍历。

经过对模糊测试的结果进行分析，得到了web应用程序可以让漏洞复现的相关有效载荷。OS命令行注入通过命令；ls即可暴漏此路径下的所有文件名；SQL注入通过命令’or 1=1—’即可暴漏出数据库的相关信息；文件上传通过模糊上传文件的后缀名即可得出phpinfo.php.jpg可以暴漏php的相关配置信息。而这些可以触发web应用程序安全漏洞的有效载荷，在模糊测试拟态web应用程序时均为正常。

5 总结与展望

通过对非拟态web应用程序和定制的拟态化web应用程序的输入框的模糊测试，进而分析模糊测试的结果，得出定制的拟态化web应用程序可以修复web应用层、SQL数据库，PHP源代码的web安全漏洞。

由于模糊测试的测试集是基于生成的，在输入上出现异常状态码的情况相对较少，为今后在基于变异的模糊测试集的研究上提出了新的要求。在下一步工作中，需要完成对表决器中各种日志的分析，通过对日志的分析进行分类，利用机器学习完善模糊测试集和表决器的现有算法。

参考文献:

[1] 巨腾飞, 王楠, 赵少飞．从黑客思维谈 Web 渗透性测试[J]. 网络空间安全，2020，9(4)：1.
JU T F, WANG N, ZHAO S F. Web permeation test from hacker thinking[J]. Cyberspace security，2020，9(4)：1.

[2] 仝青, 张铮, 张为华, 等．拟态防御 Web 服务器设计与实现[J]. Journal of Software，2017，28(4).
TONG Q, ZHANG Z, ZHANG W H, WU J X. Design and implementation of mimic defense web server. Ruan Jian Xue Bao/ Journal of Software，2017，28(4)

[3] 李红辉, 齐佳, 刘峰, 等．模糊测试技术研究[J]. 中国科学: 信息科学，2014，44(10)：1305-1322.
LI H H, QI J, LIU F, et al. Research on Fuzzy Testing Technology [J]. Science in China: Information Science，2014，44(10)：1305-1322.

[4] 张铮, 马博林, 邬江兴．web 服务器拟态防御原理验证系统测试与分析[J]. 信息安全学报，2017 (1)：13-28.
ZHANG Z, MA B L, WU J X. The Test and Analysis of Prototype of Mimic Defense in Web Servers[J]. Journal of Cyber Security，2017 (1)：13-28.

[5] Fan R, Chang Y. Machine learning for black-box fuzzing of network protocols [C]//International Conference on Information and Communications Security. Springer，Cham，2017：621-632.

[6] 马博林, 张铮, 陈源,等．基于指令集随机化的抗代码注入攻击方法[J]. 信息安全学报，2020(4).
MA B L, ZHANG Z, CHENG Y, et al. The Defense Method for Code-Injection Attacks Based on Instruction Set Randomization[J]. Journal of Cyber Security，2020(4).

[7] Sutton M. 模糊测试: 强制性安全漏洞发掘[M]. 机械工业出版社，2009.
Sutton M. FUZZING:Brute Force Vulnerability Discovery[M]. China Machine Press，2009.

[8] Manès V J M, Han H S, Han C, et al. The art, science, and engineering of fuzzing: A survey[J]. IEEE Transactions on Software Engineering，2019.

[9] de Graaf M. Intelligent fuzzing of web applications[J]. Software Engineering. Universiteit van Amsterdam: Digital Academic Repository－UBA，University of Amsterdam，2009.

[作者简介]

曲晟（1996—），男，学士，博士生，主要研究方向为网络空间安全，主动防御技术。

张铮（1976—），性别，博士，副教授，主要研究方向为网络空间安全，主动防御技术。

邢福康（1996—），男，学士，博士生，主要研究方向为网络空间安全，Web应用安全。

邵昱文（1997—），男，学士，博士生，主要研究方向为网

络空间安全，漏洞挖掘。

季新生（1968—）男，博士，教授，主要研究方向为网络空间安全，无线通信。

Differentially authorized deduplication system based onblockchain

ZHAO Tian, LI Hui, YANG Xin, WANG Han

Shenzhen Key Lab of Information Theory & Future Network Arch., Shenzhen 518055, China Shenzhen Graduate School, Peking University, P. R. China

Key words: Convergent key; Deduplication; Blockchain; Differential authoriza- tion

Abstract. In cloud storage， the deduplication technology encrypted with the con- vergent key is one of the important data compression technologies， which can effectively save storage space， as well as improve the utilization of space and bandwidth. In order to further refine its usage scenarios with various user per- missions and enhance its data security， we propose a blockchain-based differen- tial authorization deduplication system. The proposed system simplifies the ex- isting differential authorization process， and realizes credible authority manage- ment and dynamic update of authority through the blockchain which is tamper- resistant and convenient to trace the information with new blocks. Besides， the operations of all legitimate users can be recorded with the blockchain to ensure the traceability of operations.

1 Introduction

In recent years， with the development of cloud storage technology， more and more user data are uploaded to the cloud server， and many copies of the data are repeatedly stored by different users， resulting in unnecessary overhead of a large amount of storage space. This makes the deduplication an urgent problem to be solved.

However， if file data is directly stored in the cloud storage server， it is easy to face a series of risks such as data theft. Therefore， we should store the ciphertext of the data in the cloud storage server.

In traditional encryption and decryption algorithms， the keys are generated inde- pendently by users leading to various ciphertexts of the same data， which makes the deletion of duplicate data difficult. If the cloud storage server generates the key and encrypts the data uniformly， once the cloud storage server is maliciously attacked or the storage server itself becomes untrustworthy， the user's data security cannot be guar- anteed.

In order to achieve deduplication on the basis of ensuring data security， a deduplica- tion system based on convergent keys has been proposed in [1]. The encryption key $H(F)$ is obtained by Hashing the data， then the data is encrypted using it. The conver- gent encryption makes the consistent ciphertext of the same file or data block， and the cloud storage server or external attackers cannot see the original data. It not only guar- antees the confidentiality of the data， but also facilitate the cloud storage server to per- form data deduplication due to the consistent key derived based on the content of the original data.

The encryption method of the convergent key is

vulnerable to offline brute force cracking leading that semantic security cannot be guaranteed [2-3]. In recent years, re- searchers in deduplication for convergent encryption have proposed a series of im-

provements. Bellare et al. [4] proposed an information lock encryption scheme, which optimized key calculations and encryption methods. Puzio et al. [5] proposed the first repetition based on double-layer encryption in the deduplication scheme. The inner

layer uses the convergent encryption scheme mentioned above, and the outer layer is outsourced to a trusted third party. In addition, Bellare et al. [6] also proposed the Du- pLESS scheme, which adds an additional key to the convergent key generation process to invalidate the dictionary attack. Li et al. [7] proposed to use a deterministic secret sharing scheme instead of convergent encryption.

The above schemes are designed to improve their data security issues, but they do not fully consider how to build a credible authority when there are authority differences between users. To solve this problem, we propose a differential authorization dedupli- cation system based on blockchain. We combine the blockchain technology with the deduplication system. The user's public key and private key signed permissions are written into the blockchain to ensure the security of user permissions through the im- mutable modification of the blockchain and maintain each user's permission table.

When the permission is changed, blockchain directly generate a new block to cover the original permission, which is convenient for the dynamic modification of the permis- sion. On the other hand, the traceability of the blockchain can record each user's oper- ation on the data to ensure its traceability.

The rest of this paper is organized as follows. We introduce the traditional differential authorization deduplication system and its problems in section 2. Then our differential authorization deduplication system based on blockchain is proposed in the section 3, followed by the performance analysis in section 4 and experimental simulations in sec- tion 5. Finally, conclusion and future works are drawn in section 6.

2 Traditional differential authorization deduplication

In this section, we introduce the traditional differential authorization deduplication system, including the main process of file upload and download, as well as its problems.

2.1 Traditional structure

Let us consider a practical application scenario. In a company, subordinate relation- ships exit in different users leading to various permissions. For this reason, we put for- ward higher requirements for deduplication. Differential authorization deduplication should be implemented. Users with higher authority can upload and download data, while users with lower authority cannot download and access the data of high-level users.

The traditional differential authorization system [8] mainly provides different per- mission sets for different users. It introduces a private cloud server to maintain the per- mission table, and adopts the hybrid cloud architecture to realize the deduplication of differential authorization.

The system is mainly composed of three parts: the public cloud server S-CSP (stor- age server provider) responsible for storing encrypted user data, the private cloud server responsible for maintaining the user permission table, and the user who uploads and downloads files.

The specific workflow is as follows:

System initialization stage. Define the tag of file F as$\varnothing = \square$ ( ), and a label corresponds to a unique file data. Each permission p of the system has a corresponding

$_F$permission key . Define the token of file F as $\varnothing' = \square$ (,), that is, only

users with permission p can access file F.

Assuming that the permission set owned by user U is , its corresponding permis- sion key { }, $\in$ will be sent to the private cloud server acting as a permission check server. The private cloud server main-

tains a table to store each user’s public key

and its corresponding permissions.

File upload. The file upload process is shown in Fig. 1：

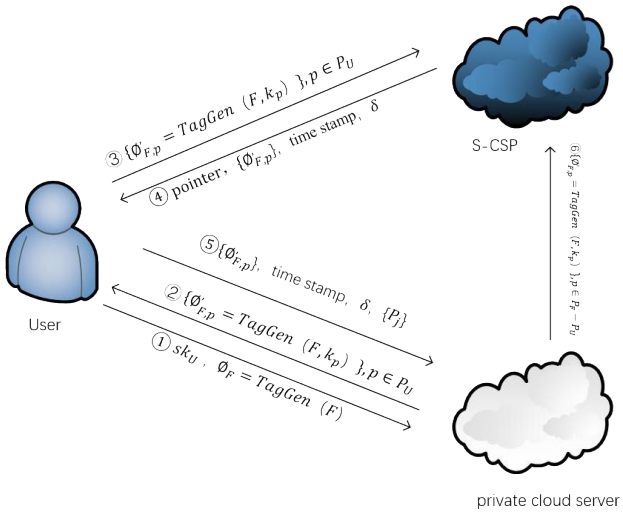


Fig. 1 File upload

Suppose that the file owner *U* wants to upload a file *F* for access by users with per- mission { }, ∈. First, the user needs to use his private key to verify his identity with the private cloud server. If the verification is passed, the user needs to

F,p send the tag ∅ = □ ( ) of the file F to it. The private cloud server will return all initial file tokens {∅' = □ (,) }, ∈ that match the user's permissions, then the user will send these tokens to the S-CSP.

F,p If duplicate files are found in S-CSP during upload, S-CSP first runs POW （Proof of Ownership） algorithm [8] to verify the user’s ownership of the file. If it passes, it will return a file pointer to the user. A signature δ and a time stamp are appended to the token {∅' } and returned to the user. The user sends the token and the permission

F,p set { } of the file *F* to the private cloud server for verification. After the verification is passed, the private cloud server will calculate all the file tokens {∅' =

□ (,) }, ∈ − and return to S- CSP. The permissions of file F at this time are the union of the permissions of and other owners of the file.

F,p F,p If S-CSP does not find duplicate files when uploading, a signature δ and a time stamp are appended to the token {∅' } and returned to the user. The user sends the token to the private cloud server for verification. After passing the identity verification, the private cloud server will calculate all the file tags {∅' } within the authority

and return to the S-CSP, then the user can upload the data encrypted by the convergent

key to the S-CSP.

File download. The user sends a file download request to the S-CSP, and the S-CSP will verify its permissions. If it cannot download, S-CSP will return the download fail- ure. If it can download, it will return the encrypted data. The user uses the locally saved convergent key to decrypt the file. and get the original data.

However, this data deduplication solution has some problems：

The first is the security assurance issue of the private cloud server. If the private cloud server is attacked and the user and corresponding permissions are tampered with, the system will not operate normally.

The second is the dynamic change of permissions. Once a file is uploaded, its per- missions are difficult to modify flexibly. When the user and file permissions change, for example, a user no longer has file permissions, the system cannot modify permis- sions in time.

3 Differentially authorized deduplication system based on blockchain

In order to solve the above problems, this paper designs and implements a differential authorization deduplication system based on blockchain. In more complex specific ap- plication scenarios, such as several companies working together to develop a project. They need to implement data deduplication on the same cloud storage server, which requires the system to credibly record the behavior of users to facilitate accountability. On the other hand, it is not only necessary to implement differential authorization

of files according to different user identities， but also to be able to make changes in time when users or file permissions change. Using the immutability and traceability of the blockchain to ensure the security of user permissions and accountability of behavior， it

can meet the requirements well. At the same time， when the user's file management authority changes， we can write the authority change into a new block. Because the blockchain is based on the record in the newly generated block， we can achieve dynamic changes in permissions.

The system is divided into three parts， the public cloud server S-CSP that stores en- crypted data， the users who upload and download files， and the blockchain that saves permissions and upload and download records.

Blockchain is a distributed ledger， first proposed by Satoshi Nakamoto and used in the Bitcoin currency transaction system. The blockchain network system maintains an orderly data block that keeps growing without a center. Each data block has a timestamp and a pointer to the previous block. Once the data is on the chain， it cannot be changed. Blockchain can be analogous to a distributed database technology. By maintaining a chain structure of data blocks， it can maintain a continuously growing， non-tamperable data record ［9］.

At present， there are many researches on the consensus algorithm of the block chain. In this system， we use the block chain based on the PoV （Proof of Vote） consensus ［10］ to construct the blockchain in the system. The PoV consensus can well avoid double- spending attacks， selfish mining， witch attacks and other attack methods， and can well guarantee the security of user permissions. There are four types of nodes in this system， including commissioners responsible for voting， butlers responsible for accounting and production blocks， butler candidates， and ordinary user nodes that can apply to become butler candidates. This system allows concurrent roles to a certain extent， as shown in Fig. 2.

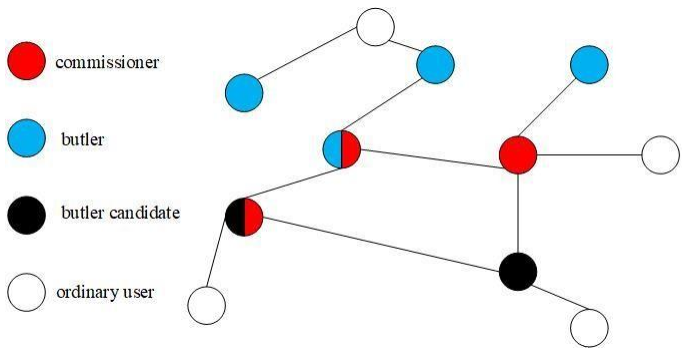


Fig. 2 PoV network model

As mentioned in section 1.1， data deduplication achieved by convergent encryption can be performed in file-level data and block-level data respectively. In order to further

save storage space and efficiently use bandwidth， we can encode the file into n data blocks {Bi} . When the files are not the same but the content is not much different， use the data block deduplication check to complete the deduplication.

We will separately discuss the upload and download of file-level data and block- level data.

3.1 File-level data upload and download

System initialization. Define the file F label as ∅ = □ （ ） ， and a label cor- responds to a unique file data. The user's permission set is ∂ = { 1 ⋯ □}， where we define its number from small to large as the permission from high to low. Those with high-level permissions can access files uploaded by people with low-level permissions and modify file permissions. In the initial state of the system， the blockchain will have an authority table signed with the negotiated highest authority owner *P* private key to de-

clare the authority level of each user. Any legal user in the system can use its public key to check the authority. At the same time， the blockchain will also store the label

∅ of the file F and the encrypted file permission， which is convenient for S-CSP

query. Suppose user U wants to upload a file with permission ， the user's private key is ， and S-CSP is initially blank.

File-level data upload， as shown in Fig. 3.

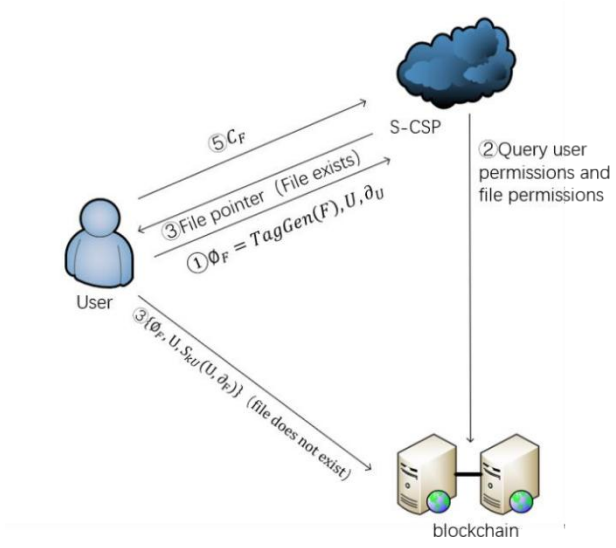


Fig. 3 File-level data upload

First， the user U sends to the S-CSP the tag ∅ = □（） of the file to be up- loaded encrypted by the private key ， the user name *U* and its own authority ， and the S-CSP will query whether there is the tag ∅ = □（） of the file and the permission of the file on the blockchain. If the file exists and the permission is lower than or equal to user permission ， the user needs to verify that he owns the file with S-CSP using the POW algorithm. At this time， the server will return a pointer to

the user indicating that the server already has the file and the user no need to upload repeatedly.

If the file does not exist or has no right to access the file， the user needs to write the uploaded relevant information to the blockchain， namely the file tag ∅ ， the user name

U， the user name *U* signed by the private key and the file can be access level

{∅ ， （ ， ）. After verifying the identity of the user and S-CSP， the blockchain writes the record into a new block， and then the user can send the data of the file en- crypted by the convergent key to the S-CSP.

File-level data download， as shown in Fig. 4：

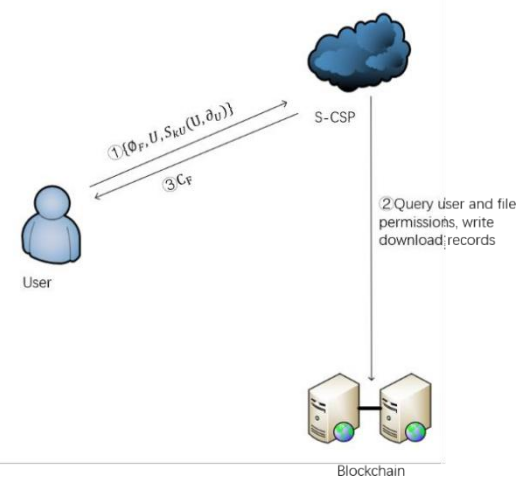


Fig. 4 File-level data download

First， the user sends the tag ∅ = □（） of the file to be downloaded to the S- CSP. The S-CSP will query whether the file exists. If the file does not exist， the S-CSP will return a prompt message to the user.

If the file exists， the user U sends the file tag ∅ ， and the user name *U*， user name and authority information encrypted by own private key ：{∅ ， U， （ ， ）} to S-CSP， S-CSP uses the user's public key to decrypt to obtain authority， and then

uses the public key of the highest authority to query the authority table to confirm whether the authority matches.

After passing， the S-CSP will send a confirmation message to the blockchain. After the blockchain verifies the identity of the user and S-CSP， the download record is saved in the block， and the S-CSP returns the file ciphertext encrypted by the convergent key to the user. The user uses the locally stored convergent key to decrypt the file.（Li et al.

have also done related research on the storage of convergent keys [11], but this is not the focus of this article. For simplicity, this article uses the traditional method of

saving locally).

3.2 Block-level data upload and download

The block-level data upload and download process is similar to the file-level data upload and download process, as follows:

System initialization. It is roughly the same as the system initialization requirements in section 3.1, except that the file is divided into {*Bi*} data blocks for upload and down- load.

Block-level data upload. The user first sends the file *F* and all the tags ∅ and

{∅} of the data block {Bi}, the user name *U*, and the own authority to the S-CSP. S-CSP will query whether there is a label for the file. If label ∅ of the file exists,

it will turn to the processing flow of the file label in the previous section. The user can prove that he owns the file through POW, and then S-CSP returns the corresponding pointer to inform the user that the file already exists. If the data block exists, the user needs to use the POW algorithm to verify to the S-CSP that he owns the data block. At this time, the server will return a pointer to the user indicating that the data block al- ready exists in the server, and there is no need to upload it repeatedly. S-CSP will add a new record to the blockchain, that is, the label of the newly added file corresponding to the data block, which is convenient for the repeatability check of the next file and data block.

If the data block does not exist, the user needs to write the uploaded relevant infor- mation to the blockchain, namely the file tag ∅, the data block tag ∅, the user name *U*, the user name signed by its own private key and the file can be accessed permission level: {∅, ∅, U, (,)}. After verifying the identity of the user and the S-CSP, the blockchain writes the information into the block, and then the user can send the data of the data block encrypted by the convergent key to the S-CSP.

Block-level data download. First, the user sends the label of the file to be down- loaded to S-CSP: S-CSP will query whether there is a label for the file on the blockchain. If the file does not exist, S-CSP will return a prompt message to the user. If the file exists, the user sends the file label, data block label, user name and authority infor- mation encrypted by own private key: {∅, ∅, (,)} to S-CSP. S-CSP uses the user's private key to decrypt to obtain the authority, and then uses the public key of

the highest authority to query the authority table to confirm whether the authority matches. S-CSP will send a confirmation message to the blockchain, after passing the identity verification, the download record is saved in the block. The S-CSP returns the data block ciphertext encrypted by the convergent key to the user, and the user uses the locally stored convergent key to decrypt the data block. Then the user can restore the original data file *F*= {*Bi*}.

The encryption method of the convergent key has its inherent flaw, that is, it cannot resist offline dictionary attacks. Specifically, if external attackers know the ciphertext

and can infer the file set {*F*}, they can directly generate the convergent key and encrypt the corresponding files for comparison. If the ciphertext is the same, attackers successfully obtained the original data file.

In response to this defect, this article proposes to use block-level data upload when storing high-privilege files, and use double-layer encryption for more important data blocks (in the application scenario of deduplication, the number of different data blocks in the file is less), that is, use another *Hash* function to generate the convergent key in the outer layer of the encrypted data block to encrypt again. After S-CSP receives the data, it sends a message to all users higher than the authority, calculates the convergent key of the *Hash* function and saves it. When the user needs to download a file, the convergent key is used to decrypt the outer layer, and then the original

convergent key is used to decrypt the inner layer. This mode can be turned on or off according to the requirements of the system application scenario.

4 Theoretical analysis of system performance

In this section， we analyze the performance of the system， including functional anal- ysis and safety analysis.

4.1 System functional analysis

4.1.1 Differential access control

The above process is the main process of uploading and downloading data of the system. The specific differential authorization is embodied in the user uploading a file， which can be easily written into the file's permissions， including reducing file permissions or increasing file permissions. For example， a fourth-level permission can upload a fifth-level permission file for low-privilege access， it can also upload a high-level file such as the second-level authority. Users with a level greater than or equal to the second-level authority can access the file， and at the same time， the third-level user has no right to access the file， which realizes the system's differential authority access control.

4.1.2 Dynamic changes in permissions

The system can also solve the problem of dynamic changes in permissions.

When the user authority changes， we can update the record in the block， that is， gen- erate a new block record. According to the traceability of the blockchain， when all nodes confirm the block， the user will have the new authority. At this time， if the au- thority level is increased， the user can access and download the high-level authority file， If the authority is reduced， the user cannot access the original authority file.

When the file authority changes， the high-level authority or the file uploader can send information to the blockchain again， rewrite the file authority level， and realize the dy- namic management of file authority.

Compared with traditional data deduplication solutions， our proposed solution has obvious advantages in dynamic management and modification of permissions. The rapid generation of the blockchain ensures that when the permissions change， it can be modified quickly and form a consensus confirmation between users.

4.2 System security analysis

The security of permissions is mainly guaranteed by the security of the blockchain， that is， the tolerance of attacking nodes.

Define the number of blockchain nodes as n and the number of attackers as f， then our tolerance for attacking blocks is：

$f \leqslant \square - 1$

2

That is， it can tolerate no more than half of the nodes being attacked， which better guarantees that the permissions cannot be tampered with.

At the same time， the double-layer encryption scheme for confidential data blocks mentioned in Section 3 can also better prevent offline dictionary attacks. The encryp- tion method we use for convergent encryption is AES （Advanced Encryption Standard） encryption， and the key length is 256bit. AES encryption has good resistance to brute force cracking. If 10，000 collision attacks are executed every nanosecond， it will take $1.8*10^{56}$ years to crack［12］.

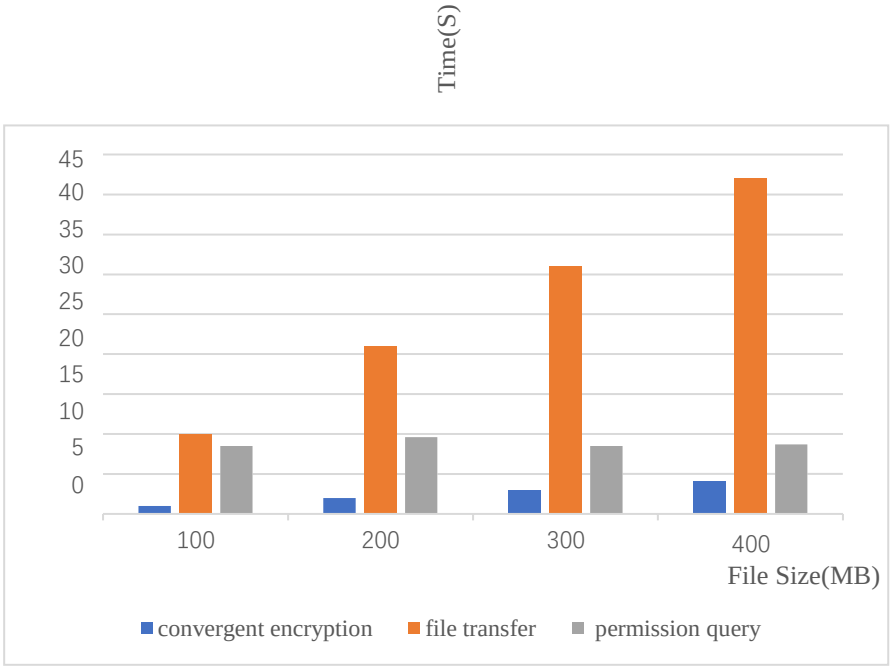
5 Experimental simulation

We implemented the model of the system and ran it in our own experimental envi- ronment. The code was implemented in C++， the server operating system was Ub- untu16.04， the CPU brand was AMD， the frequency was 1.7GHz， the memory was 8GB， and the network bandwidth was 1000Mb/ s.

We test the performance of this model. The main measurement indicators are the ratio of the time used for permission query， file transfer， file label generation， and con- vergent encryption when the upload and download file sizes are different.

We use AES encryption as the encryption algorithm for convergent encryption. The key is a 256-bit hash value generated by the SHA-256 algorithm; the file label is gen- erated by the SHA-1 algorithm. When the permission changes， we can manually set the new block generation time to 1s， which is the time required for the permission change.

We tested the performance of the traditional solution and our system separately when the file size is 100MB， 200MB， 300MB， and 400MB. As shown in Fig. 5 and Fig. 6.



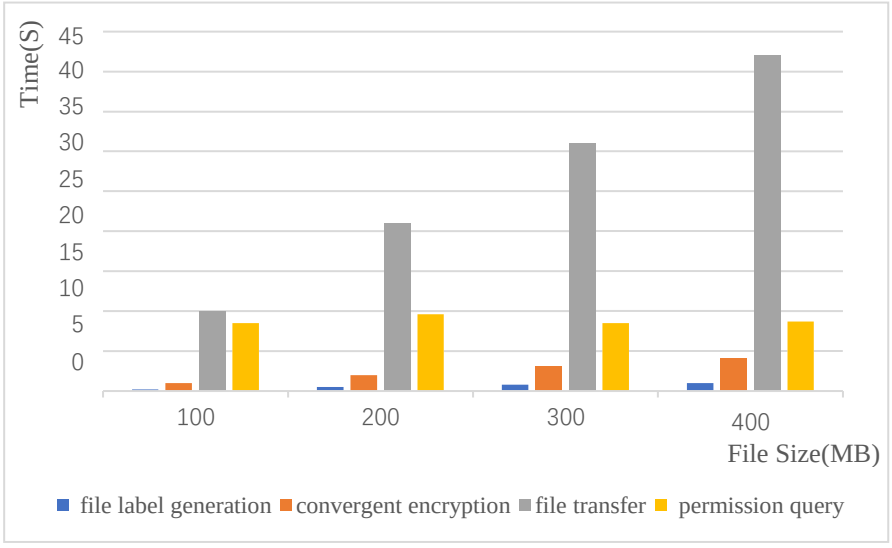


Fig. 5 Permission query time and main process time of the traditional solution

Fig. 6 Permission query time and main process time of our system

Because the authorization query time is too short， the figure shows the total time used for 10，000 queries. It can be seen from the experimental results in Fig. 5 and Fig. 6 that compared to data transmission and convergent encryption， the time required for user

permission query and record writing to the blockchain is shorter. Our system does not significantly increase system overhead while achieving differential authorization dedu- plication.

At the same time， we also tested the enhancement scheme proposed in section 3. The main performance indicators are the ratio of the time used for double-layer encryption of important data blocks， file label generation， convergent encryption and file transfer. In this enhanced scheme， the outer layer encryption uses the AES encryption algo- rithm， and the key is a 128-bit hash value generated by the MD5 hash algorithm. The inner layer encryption uses the AES encryption algorithm， and the key is a 256-bit hash

value generated by the SHA-256 algorithm.

We tested the situa- encryption） accounted tion where the number of for different proportions important data blocks of the total number of da-（requiring double- layer ta blocks. For convenience， set the number of important data blocks to 1， the size of 100MB， and the size of ordinary data blocks to 100MB. The experimental results are shown in Fig. 7.

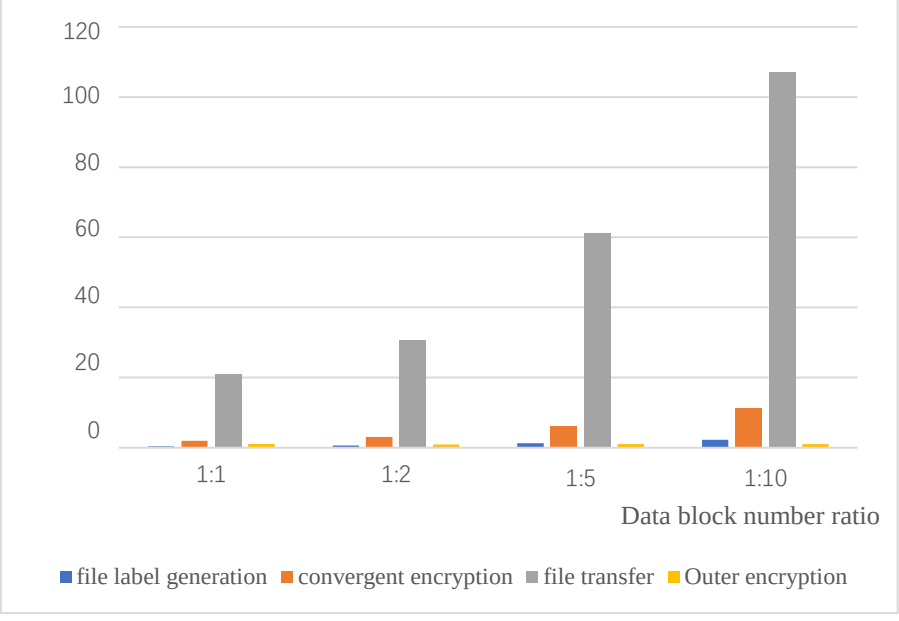


Fig.7 The time required for outer encryption and the main process time

It can be seen from the experimental results that when the important data is relatively small， our enhancement scheme will not significantly increase system overhead while improving data security.

6 Conclusions and future work

The Differential authorization deduplication system based on blockchain system pro- posed in this paper writes the user permission table and file permissions into the block- chain， using the immutable modification of the blockchain， and it better solves the vul- nerability of public cloud servers and private cloud servers. Without significant increase in system overhead， the user's authority management has been successfully realized.

At the same time， because the blockchain record is easy to write， and the latest writ- ten information will prevail. Therefore， it can solve the prob-

lem that the permissions of users and files in the original system cannot be dynamically modified in time， and re- alize the management of dynamic changes in user and file permissions.

In the follow-up work， we can continue to improve the existing convergent key gen- eration method or explore other encryption schemes to replace the existing convergent key encryption to overcome its vulnerability to offline dictionary attacks.

References:

[1] Douceur J R, Adya A, Bolosky W J, et al. Reclaiming space from duplicate files in a serv- erless distributed file system［C］. international conference on distributed computing sys- tems，2002：617-624.

[2] Liu J, Asokan N, Pinkas B, et al. Secure Deduplication of Encrypted Data without Addi- tional Independent Servers［C］. computer and communications security，2015：874-885.

[3] Liu X, Sun W, Lou W, et al. One-tag checker: Message-locked integrity auditing on en- crypted cloud deduplication storage［C］. international conference on computer communica- tions，2017：1-9.

[4] Bellare M, Keelveedhi S, Ristenpart T, et al. Message-Locked Encryption and Secure Deduplication［C］. theory and application of cryptographic techniques，2013：296-312.

[5] Puzio P, Molva R, Onen M, et al. ClouDedup: Secure Deduplication with Encrypted Data for Cloud Storage［C］. ieee international conference on cloud computing technology and science，2013：363-370.

[6] Bellare M, Keelveedhi S, Ristenpart T, et al. DupLESS: server-aided encryption for dedu- plicated storage［C］. usenix security symposium，2013：179-194.

[7] Li J, Chen X, Huang X, et al. Secure Distributed Deduplication Systems with Improved Reliability［J］. IEEE Transactions on Computers，2015，64(12)：3569-3579.

[8] Halevi S, Harnik D, Pinkas B, et al. Proofs of ownership in remote storage systems［C］. computer and communications security，2011：491-500.

[9] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system［R］. Manubot，2019.

[10] Li K, Li H, Hou H, et al. Proof of Vote: A High-Performance Consensus Protocol Based on Vote Mechanism & Consortium Blockchain［C］. high performance computing and com- munications，2017：466-473.

[11] Li J, Chen X, Li M, et al. Secure Deduplication with Efficient and Reliable Convergent Key Management［J］. IEEE Transactions on Parallel and Distributed Systems，2014，25(6)：1615-1625.

[12] Stallings W, Brown L, Bauer MD, Bhattacharjee AK, et al. Cryptography and network security: principles and practice［M］. Pearson Education India，2003.

About the authors

ZHAO Tian is a Postgraduate of Peking University Shenzhen Graduate School. The main research directions are big data applications， blockchain， and distributed storage （Email：1801213424@pku. edu. cn）

LI Hui ［corresponding author］ He is currently a professor and PhD supervisor at Peking Uni- versity Shenzhen Graduate School. Research fields include cyberspace security and blockchain technology， artificial intelligence and future network systems， distributed storage coding theory and systems， intelligent big data analysis and data standards. （Email：huilihuge@163. com）

YANG Xin received the B. Eng. degree from the Department of Computer Science and Engi- neering， South China University of Technology， in 2016. She is currently pursuing the Ph. D. degree with the School of Information Science， Peking University. She is also the student of the Peng Cheng Laboratory. Her research interests include cyber security， future network architec- ture， and distributed storage systems.（Email：yangxin2016@pku. edu. cn）

WANG Han received the BEng degree from the Department of Communication Engineer- ing，JiLin University of Technology， in 2017. She is currently working toward the PhD degree in the School of Information Science， Peking University. Her research interests include distrib-

基于学习的物理层智能认证综述

张永斌[1，2]，金梁[2，3]，李凯[2]

[1]东南大学网络空间安全学院 南京 211111；

[2]网络通信与安全紫金山实验室，南京 211111；

[3]国家数字交换系统工程技术研究中心，河南 郑州 450002

摘　要：物理层认证技术近年来已经成为无线网络身份安全的一个重要发展方向，机器学习方法的蓬勃发展为物理层认证技术朝着智能认证发展提供了可靠的增强技术。本文首先简单介绍了物理层认证方法的原理，然后阐述了近年来基于机器学习的物理层认证方案的研究成果，把这些方案按照机器学习的特征进行了分类，并描述了其特点，这些方案相比于传统的物理层认证方案有着更好的可靠性，鲁棒性，以及精确性。最后，文章探索了物理层认证在未来通信安全中的挑战与发展。

关键词：物理层认证、智能认证、机器学习

Overview of Physical Layer Intelligent Authentication Based on Learning

ZHANG Yongbin[1,2], JIN Liang[2,3], LI Kai[2]

1.School of Cyberspace Security, southeast University, Nanjing 211111,China;

2.Purple Mountain Laboratories,Nanjing 211111,China;

3.National Digital Switching System Engineering & Technological R&D Center,Zhengzhou 450002,China

Abstract: Physical layer authentication technology has become an important development direction of wireless network identity security in recent years, and the vigorous development of machine learning methods provides reliable enhancement technology for physical layer authentication technology. This article briefly explains the principles of physical layer authentication methods, then introduces the recent research of physical layer authentication schemes based on machine learning, classifies these schemes according to the characteristics of machine learning, and describes their characteristics. These schemes have better reliability, robustness, and accuracy than traditional physical layer authentication schemes. Finally, the article explores the challenges and development of physical layer authentication in future communication security.

Key words: Physical authentication; Intelligent Authentication; machine learning

1　引言

由于无线介质的开放广播性质，任何的非法接收者都能在覆盖范围内对无线电信号进行接收[1]。在使用标准化传输和常规化安全方案的无线网络下，非法窃听者拦截和窃听信息变得相当简单[2]。此外，随着5G的蓬勃发展与6G的提出，物联网相关应用的数量显著增长，无线基础设施与物联网支持的垂直工业应用之间也在不断融合发展，这些都为攻击者提供了丰富的欺骗机会，因此无线通信安全问题很是重要，其中实现可靠的终端身份认证更是重中之重。

随着接入设备的不断增加，高层认证所需要进行的密钥管理变得更加困难，越来越复杂的网络架构也使得高层认证的兼容性变得更差。与此同时，无线通信网络不断朝着高度异构的方向发展，其动态拓扑结构使得传统密码学的安全机制难以保证高度的安全性。再者，无线通信终端设

基金项目：国家自然科学基金面上项目资助，编号61871404

备不断朝着更轻便，更快捷的方向发展，利用传统密码学虽然能够实现通信的保密性，但它需要额外的计算而且产生了延迟[3]，此外，数据加密和解密都需要消耗一定的时间[4]，这些都与设备发展的目标相违背。而且随着量子计算机的出现与发展，其强大计算的能力使得破解密钥成为可能。基于以上这些原因，物理层认证技术得到了认可与研究。

物理层认证技术充分利用了物理层特有的属性，通过利用物理层属性特征来验证消息发送方的身份，能做到与高层协议架构有较好的兼容性和通用性。物理层认证技术还能够与高层的加密技术做到共存，这相当于在原本加密的基础上又增强了安全性。与复杂的计算加密技术不同，物理层认证利用空间中物理层信息的相关性和时空唯一性来进行确认发送端的身份，所需要的计算和通信开销更低。而且非法窃听者难以预测和模仿信号的物理特性，所以物理层认证具有较好的安全性和保密性[5]。

2　物理层认证方法原理介绍

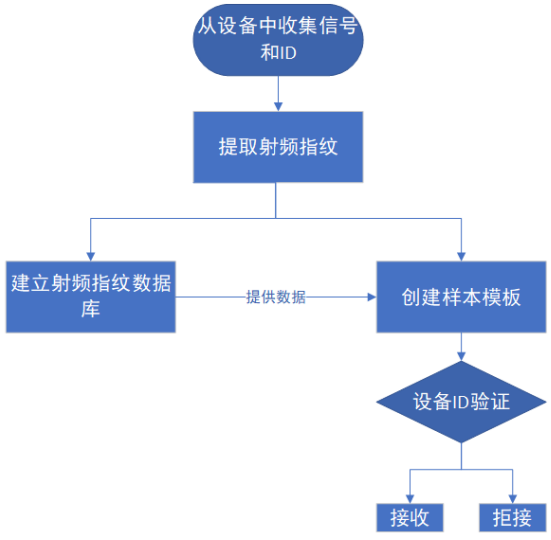
根据接收信号提取的特征不同可以将物理层认证分为两种机制：基于发射机特征的机制和基于无线信道特征的机制。发射机的特征主要代表为发射机模拟电路的射频指纹特征，简称为射频指纹。无线信道的特征反映的是无线信道的响应和周围环境。其中，基于射频指纹机制是对具有不完美信号特征的无线设备硬件进行识别，基于无线信道特征机制是利用无线信道的多样性，唯一性和随机特性这些属性对设备进行识别认证。

2.1　基于射频指纹的认证机制

射频指纹反映的是发射机的物理特征，基于射频认证机制的原理就是利用各个发射机的物理特征不同来进行身份验证。文章［6］阐述了发射机的元件容差是产生射频指纹的主要原因，并证明了应用射频指纹识别发射机的可行性。目前，射频指纹技术主要是基于瞬态信号或者调制域信号来实现指纹数据的获取[7]。但是这种技术对检测器件的性能要求非常高，而且测量指纹的过程需要高精度的信号仪器来进行提取分析信号间的微小差异，成本开销比较高昂。

文献［8］把基于射频指纹的认证机制看成一个典型的模式分类问题，分为提取和认证两个步骤。如图1所示，在提取阶段，系统会收集并存储设备的信号，提取每个设备的RF指纹，将其注册并存储在库中，并与相应设备的ID号链接起来以形成射频指纹数据库。在认证阶段，通过收集待认证设备的通信信号来提取相应的射频指纹，将结果与指纹数据库中的比较指纹进行比较。

为解决无线网络通信中未经授权的网络访问和欺诈攻击问题，Reising使用射频指纹技术来增强WAP安全性[9]，文章提出通过降维分析和设备ID身份验证来检测冒充授权设备的恶意设备，实验证明此方法对72种欺骗攻击的拒绝率超过93%。文献［10］通过采用基于硬件的安全原语从一个全新的角度研究了多媒体身份验证的潜力，作者在真实硬件上的实验结果表明，使用硬件安全性方法获得的多媒体身份验证具有很高的安全性，而且所需要资源更少，消耗的功耗更低。

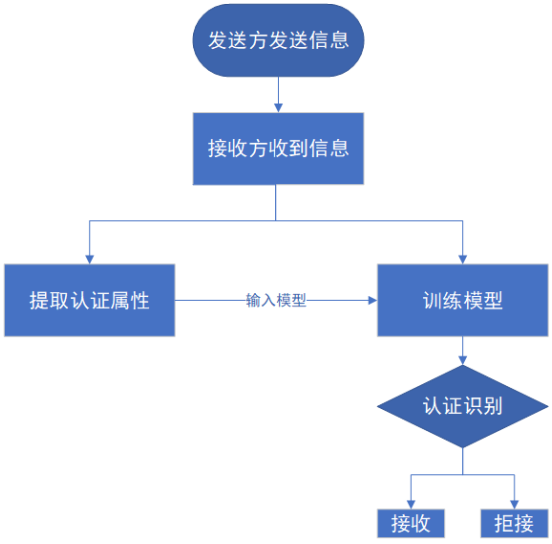


2.2　基于无线信道特征的认证机制

随着物理层认证的不断发展，基于无线信道的认证的机制也在不断被提出与创新。文献［11］提出将无线信道的多样性，唯一性和随机性作为认证属性，根据属性的变换或者人为构造的属性特征建立假设检验，然后实现身验证。在目前基于无线信道特征的认证技术方案中，常用的信道特征有接收信号强度指标（Received Signal Strength Indicator，RSSI），接收信号强度（Received Signal Strength，RSS），信道状态信息（Channel State Information，CSI），信道脉冲响应

（Channel Impulse Information， CIR），信道频率响应（Channel Frequency Information， CFR）等。

文献［12］认为现有的无线信道身份验证方法可分为提取特征和进行认证两个单独的步骤。如图2所示，在提取阶段，接收方在收到发送方发来的消息后，从中提取出信道特征。在认证阶段，接收方使用训练完的学习模型检测发送方是否为欺骗攻击者，从而对发送方进行身份的识别与认证。



物理层认证的属性随着物理层安全的不断发展也在被不断开发与优化。文献［13］过将无线信道的多径延迟特性集成到基于信道脉冲响应（CIR）的物理层认证框架中，提出了一种新的物理层认证增强方案，为简化认证决策规则提出了一种二维量化方法用于预处理通道变化。文献［14］对具有不同程度硬件损坏的大规模MIMO系统，提出了一种新的基于信道的认证方案，研究了在第五代（5G）及以后的网络中具有非理想硬件的大规模多输入多输出（MIMO）系统中的发射机认证。

3 物理层认证增强技术

虽然物理层认证改善了兼容性，计算延迟等一系列问题，但是仍然具有一些局限性，这些局限性主要体现在：1）通信链路属性的不完美估计和变化会使得物理层特征表现出较强的波动性，不断变化的传播和干扰条件让通常采用的用于物理层身份验证的静态假设测试在时变的通信通道中面临着巨大的挑战。2）物理层特征相对固定的取值范围，使得物理层认证难以在终端设备密集的场景下部署。为了满足不断发展的终端设备身份鉴别的性能需求，物理层认证技术需要进一步增强其可靠性和稳健性。

基于学习的方法为现有的物理层身份验证方法和物理层安全性能提供了许多其本身不具备的优势。在许多情况下，智能模型的强大学习能力可以补偿信道参数估计的不完善之处，所以用于无线通信的智能物理层身份验证方案也在被不断提出与完善。同时机器学习所带来的成本不仅很低，还为设计出高效且性能优秀的认证系统模型带来了可靠的增强性技术。

3.1 基于射频指纹的智能认证

在2.1提到将射频指纹认证分为了提取和认证这两个步骤，而基于学习的认证中常见的可靠性增强技术在射频指纹认证的提取和认证这两个阶段都能进行应用，如表1所示。对于在提取射频指纹过程中诸如信道条件，设备移动性和噪声的变化等因素导致射频信号的信噪比级别发生变化的问题，文献［15］提出了一种使用卷积神经网络（CNN），通过降维和去相关以及聚类的新型集成方法，从授权设备收集的射频跟踪来训练CNN。

表1

认证阶段	学习方法	特点	参考文献
特征提取阶段	强化学习	为了学习独特的指纹并将其用于检测未经授权的射频设备	[15]
认证分类阶段		基于攻击样本学习分类模型，实现可靠认证	[16-17]
	机器学习	将选定的特征提供给几种机器学习算法进行分类	[18]
提取和认证统一	强化学习	将特征提取和无线发射机识别两个部分统一为一个深度学习框架进行优化	[19]

在认证分类阶段，也可以利用机器学习来给物理层认证提供增强技术，通过学习合法终端和攻击终端的样本，建立样本分类模型，实现可靠性高的物理层认证。文献［16］提出了一种多采

样卷积神经网络（MSCNN），从选定的不稳定的关注区域中提取射频指纹以对ZigBee设备进行分类。文献［17］则提出一种基于深度学习的分类器，该分类器可以学习低功率无线电的硬件缺陷来建立分类模型。不同于仅依靠时域信号和相应特征的传统方法应用在分类阶段，文献［18］提出使用能量瞬变信号，从能量瞬变中提取统计特征来进行分类，然后作者分别使用四种流行的机器学习算法：K邻近（kNN），判别分析（DA），支持向量机（SVM）和神经网络（CNN）来进行训练，实验发现内置超参数优化分类方法的分类效果更好。

传统的射频指纹识别系统大多是通过提取信号中的人工特征来进行训练和识别的，这不可避免地导致提取的特征不完整，进而产生无法准确描述无线发射机的所有个体特征的问题。与上述分为训练和认证两个阶段的射频认证方法不同，文献［19］采用基于卷积神经网络的射频指纹识别方法，将特征提取和无线发射机识别两个部分统一为一个深度学习框架进行优化。在深度学习的框架内联合优化特征表示和分类器，以最大化联合协作的性能。

3.2 基于无线信道特征的智能认证

在无线信道特征的物理层认证中，机器学习主要应用于认证阶段，接收方通过机器学习算法训练认证分类模型来对发送方的身份进行识别与认证。机器学习算法可以根据其功能和结构进行分类，不同场景下使用不同类别的学习方法。文章从两个角度阐明机器学习技术在无线信道特征的物理层认证中的应用：参数/非参数学习和监督/非监督/强化学习，如表2和表3所示：

3.2.1 参数学习方法

参数学习方法已日趋成熟，例如逻辑回归，线性判别分析，感知器和朴素贝叶斯，而这些都需要特定形式的训练功能。当选择合适的训练样本进行训练时，参数学习方法可以比非参数学习方法更准确，更简单并且所需要的训练样本也更少。文章［20］在假设所有无线节点都是静态的情况下，通过利用发射机的RSSI进行认证增强，提出了一种基于逻辑回归的认证方案。文献［21-23］提出了基于深度学习的物理层认证方案，通过训练合法发送方与非法发送方的样本特征数据，建立样本分类模型，从而实现性能更可靠的物理层认证。在以上认证方案中，参数学习方法可以基于训练函数来对物理层特征属性进行建模，尽可能的避免复杂的时变环境所带来的不确定性。

3.2.2 无参数学习方法

无参数学习方法可以从时变环境中动态学习，而无需假设有关的训练模型。这为物理层身份验证提供了更好的灵活性，尤其是在那些需要实时计算和几乎瞬时的身份验证方案中。文献［24］提出基于核机器学习的物理层身份验证方案，通过降低基于多属性的身份验证系统的维数，并将最终的身份验证过程建模为线性系统，从而降低了身份验证过程的计算复杂度；更重要的是，文章提出的内核学习算法可跟踪时变属性，可以迅速适应时变环境。文献［25］提出基于聚类学习的物理层认证方案，通过聚类物理层特征进行分类，提高物理层认证的可靠性。但是，与参数学习方法相比，非参数学习方法需要更多的训练数据，而这可能导致过度拟合。

表2

学习类型	学习方法	特点	参考文献
有参数学习	回归学习	学习样本属性,建立评估模型,增强认证可靠性	[20]
	深度学习	基于攻击者样本精确判决阈值,提高认证准确性	[21−23]
无参数学习	核函数学习	降低系统维数,建立线性模型,提高认证可靠性	[24]
	聚类学习	基于样本结构进行自适应分类,增加认证可靠性	[25]

3.2.3 监督学习方法

监督学习算法和非监督学习算法之间的主要区别在于，监督学习需要对给定的输入预测相应的输出，而非监督学习算法是从无标记数据中学习数据的统计规律。与大多数现有的基于已知无线信道模型的基于假设检验的物理层身份验证方案不同，文献［26］提出的身份验证系统使用逻辑回归消除了对已知信道模型的假设，适用于更

通用的无线网络。文献［27］使用从无线节点继承的接收信号强度（RSS）的空间相关性来检测欺骗攻击，在训练数据方面使用支持向量机（SVM）来进一步提高确定攻击者数量的准确性。在智能认证方案中，在监督还是无监督机器学习算法之间进行选择通常取决于认证问题和手头训练数据的数量，当训练数据和合法通信会话的相应输出很容易获得时，监督学习算法更合适。

3.2.4 无监督学习方法

在探索无监督方法时，接收方的学习者仅接收未标记的训练数据，并对所有看不见的点进行预测[28]，比如K-均值聚类[29]。当合法发送方的样本远比非法发送方的样本多的时候，可以引入无监督的学习算法来进行智能身份验证。文献［25］提出了一种基于多通道属性的欺骗检测机制，然后利用改进的聚类算法进一步降低复杂度。为了满足物理层认证增强方法，进一步提高对攻击样本的敏感程度，提高物理层认证的精度，降低学习的复杂度，需要把机器学习算法从有监督的学习策略逐步向半监督甚至无监督的学习策略转换[30]。

3.2.5 强化学习方法

强化学习不需要精确的输入和输出以及精确的参数更新。文献［31］根据接收信号强度指标提出了一种基于Q学习的认证方案，来检测无线网络中欺骗攻击的物理层身份验证，实现最佳测试阈值并提高认证准确性，但这是一种静态身份验证方案，而且当获取信道属性特征的可用资源和时间受到限制时可能不适用。随着网络智能化水平的不断提高，对于无法获得训练样本的通信场景需要强化学习的技术来提高物理层认证的可靠性和精确性。

表3

学习类型	学习方法	特点	参考文献
监督学习	回归方法	学习属性可靠性评估模型,增强认证可靠性	[26]
	支持向量机	基于样本来精确判决阈值,提高认证准确性	[27]
无监督学习	聚类方法	提高攻击样本的敏感程度,提高认证的精度	[25,29]
强化学习	Q学习	进行预测最佳的测试阈值,提高认证准确性	[31]

4 总结与展望

本文中，先是回顾了无线物理层认证技术研究的原理，然后介绍了有关基于机器学习的物理层安全认证的研究方案，这些方案解决了传统的物理层认证方案一些局限性问题，比如通过构建自适应系统，使得认证能够在时变信道环境中执行；通过机器学习算法，构建出更好的分类模型，为认证提供更好的可靠性，鲁棒性以及认证性能等等。

由于通常采用的物理层身份验证是静态假设，这很难在时变性强的通信通道中保证认证的可靠性与准确性，而且由于不断变化的传播和干扰条件的存在，现有物理层身份验证方案的性能可能会受到所使用的通信链路属性的不完美估计和变化的严重影响，而基于学习的智能身份验证方法的提出与创新为解决以上问题带来了新的方案。

随着网络智能化水平的不断提高，基于机器学习的物理层认证增强技术有望得到大规模的部署。而且随着5G的大规模应用，物理层智能认证技术也为以物联网移动边缘计算（MEC）为例的各种应用的安全性提供了保证。利用具有学习特性的智能认证，能够逐渐实现具有自适应、自主、自生长特点的安全保证，为未来对于6G智能终端身份鉴别提供了可行性，也使得满足6G网络对超低时延、超可靠通信和用户隐私性的需求成为可能。

参考文献:

[1] WangX., HaoP. and HanzoL., "Physical-layer authentication for wireless security enhancement: current challenges and future developments," in IEEE Communications Magazine, vol. 54, no. 6, pp. 152-158, June 2016, doi: 10. 1109/MCOM. 2016. 7498103.

[2] ZouY., ZhuJ., WangX. and HanzoL., "A Survey on Wireless Security: Technical Challenges, Recent Advances, and Future Trends," in Proceedings of the IEEE, vol. 104, no. 9, pp. 1727-1765, Sept. 2016, doi: 10. 1109/JPROC. 2016. 2558521.

[3] XiaoY., ChenH. -H., SunB., WangR.,and S. Sethi, "MAC security and security overhead analysis in the IEEE 802. 15. 4 wireless sensor networks,

" EURASIP J. Wireless Commun. Netw.，2006，doi: 10. 1155/WCN/2006/93830.

[4] ApostolopoulosG.，PerisV.，PradhanP.，and SahaD. “Securing electronic commerce: Reducing the SSL overhead，”
IEEE Network, vol. 14, no. 4，pp. 8－16，Jul. 2000.

[5] 黄开枝, 金梁, 钟州. 5G 物理层安全技术——以通信促安全[J]. 中兴通讯技术，2019，25(4)：43-49.
HUANG K Z, JIN L, ZHONG Z. 5G physical layer security technology: enhancing security by communication[J]. ZTE Technology Journal，2019，25(4)：43-49.

[6] 袁红林,胡爱群. 射频指纹的产生机理与惟一性[J]. 东南大学学报(自然科学版)，2009，39(02)：230-233.

[7] 张继明. 无线网络中物理层身份认证研究[D]. 华中科技大学，2013.

[8] GuoX.，ZhangZ. and ChangJ.，"Survey of Mobile Device Authentication Methods Based on RF Fingerprint，" IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)，Paris，France，2019，pp. 1-6，doi: 10. 1109/INFOCOMWKSHPS47286. 2019. 9093755.

[9] ReisingD. R., TempleM. A., and JacksonJ. A.. "Authorized and Rogue Device Discrimination Using Dimensionally Reduced RF-DNA Fingerprints. "IEEE Transactions on Information Forensics & Security. vol. 10，no. 6，pp. 1180-1192，2015.

[10] ShahrakM. Z., YeM., SwaminathanV. and WeiS. "Two-way real time multimedia stream authentication using physical unclonable functions，" 2016 IEEE 18th International Workshop on Multimedia Signal Processing (MMSP)，Montreal，QC，2016，pp. 1-4，doi: 10. 1109/MMSP. 2016. 7813398.

[11] 任品毅,徐东阳. 无线物理层认证技术:昨天、今天和明天[J/OL]. 中兴通讯技术：1-18[2020-08-21]. http://kns. cnki. net/kcms/detail/34. 1228. TN. 20200722. 1043. 002. html.

[12] QiuX., DaiJ. and HayesM.，"A Learning Approach for Physical Layer Authentication Using Adaptive Neural Network，" in IEEE Access，vol. 8，pp. 26139-26149，2020，doi：10. 1109/ACCESS. 2020. 2971260.

[13] LiuJ. and WangX.，"Physical Layer Authentication Enhancement Using Two-Dimensional Channel Quantization，" in IEEE Transactions on Wireless Communications，vol. 15，no. 6，pp. 4171-4182,June2016，doi: 10. 1109/TWC. 2016. 2535442.

[14] ZhangP., TalebT., JiangX. and WuB.，"Physical Layer Authentication for Massive MIMO Systems With Hardware Impairments，" in IEEE Transactions on Wireless Communications，vol. 19，no. 3，pp. 1563-1576, March 2020，doi: 10. 1109/TWC. 2019. 2955128.

[15] BasseyJ., AdesinaD., LiX., QianL., AvedA. and KroeckerT.，"Intrusion Detection for IoT Devices based on RF Fingerprinting using Deep Learning，" 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)，Rome，Italy，2019，pp. 98-104，doi: 10. 1109/FMEC. 2019. 8795319.

[16] YuJ., HuA., LiG. and PengL.，"A Robust RF Fingerprinting Approach Using Multisampling Convolutional Neural Network，" in IEEE Internet of Things Journal，vol. 6，no. 4，pp. 6786-6799，Aug. 2019，doi: 10. 1109/JIOT. 2019. 2911347.

[17] DasR., GadreA., ZhangS., Kumar and JS.. M. F. Moura，"A Deep Learning Approach to IoT Authentication，" 2018 IEEE International Conference on Communications (ICC)，Kansas City，MO，2018，pp. 1-6，doi: 10. 1109/ICC. 2018. 8422832.

[18] EzumaM., ErdenF., AnjinappaC. K., OzdemirO. and GuvencI.，"Micro-UAV Detection and Classification from RF Fingerprints Using Machine Learning Techniques，" 2019 IEEE Aerospace Conference，Big Sky，MT，USA，2019，pp. 1-13，doi: 10. 1109/AERO. 2019. 8741970.

[19] ZongL., XuC. and YuanH.，"A RF Fingerprint Recognition Method Based on Deeply Convolutional Neural Network，" 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)，Chongqing，China，2020，pp. 1778-1781，doi: 10. 1109/ITOEC49072. 2020. 9141877.

[20] XiaoL., WanX., LuX., ZhangY. and WuD., "IoT Security Techniques Based on Machine Learning: How Do IoT Devices Use AI to Enhance Security?," in IEEE Signal Processing Magazine, vol. 35, no. 5，pp. 41-49，Sept. 2018，doi: 10. 1109/MSP. 2018. 2825478.

[21] WangN., JiangT., LvS. and XiaoL.，"Physical-Layer Authentication Based on Extreme Learning Machine，" in IEEE Communications Letters，vol. 21，no. 7，pp. 1557-1560, July2017，doi: 10. 1109/LCOMM. 2017. 2690437.

[22] Liao R. -F.; Wen H.; Wu J.; Pan F.; Xu A.; Jiang Y.; Xie F.; Cao M. Deep-Learning-Based Physical Layer Authentication for Industrial Wireless Sensor Networks. Sensors 2019，19，2440.

[23] Baldini, G, Giuliani, R, Dimc, F. Physical layer authentication of Internet of Things wireless devices using convolutional neural networks and recurrence plots. Internet Technology Letters 2019；2：e81. https://doi. org/10. 1002/itl2. 81

[24] FangH., WangX. and HanzoL.，"Learning-Aided Physical Layer Authentication as an Intelligent Process，" in IEEE Transactions on Communications，vol. 67，no. 3，pp. 2260-2273, March 2019，doi: 10. 1109/TCOMM. 2018. 2881117.

[25] XiaS., LiN., XiaofengT. and FangC.，"Multiple Attributes Based Spoofing Detection Using an Improved Clustering Algorithm in Mobile Edge Network，" 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)，Shenzhen，2018，pp. 242-243，doi: 10. 1109/HOTICN. 2018. 8605953.

[26] XiaoL., WanX. and HanZ.，"PHY-Layer Authentication With Multiple Landmarks With Reduced Overhead，" in IEEE Transactions on Wireless Communications，vol. 17，no. 3，pp. 1676-1687, March 2018，doi: 10. 1109/TWC. 2017. 2784431.

[27] YangJ., ChenY., TrappeW. and ChengJ.，"Detection and Localization of Multiple Spoofing Attackers in Wireless Networks，" in IEEE Transactions on Parallel and Distributed Systems，vol. 24，no. 1，pp. 44-58，Jan. 2013，doi: 10. 1109/TPDS. 2012. 104.

[28] FangH., WangX. and TomasinS.，"Machine Learning for Intelligent Authentication in 5G and Beyond Wireless Networks，
" in IEEE Wireless Communications, vol. 26, no. 5，pp. 55-61，October 2019，doi: 10. 1109/MWC. 001. 1900054.

[29] JiangC., ZhangH., RenY., HanZ., ChenK. and HanzoL.，"Machine

Learning Paradigms for Next-Generation Wireless Networks，" in IEEE Wireless Communications，vol. 24，no. 2，pp. 98-105，April 2017，doi: 10. 1109/MWC. 2016. 1500356WC.

[30] 夏仕达,徐珺,陶小峰. 面向6G智能终端身份鉴别技术[J]. 物联网学报,2020,4(01):131-138.

[31] XiaoL. , LiY. , HanG. , LiuG. and ZhuangW. ，"PHY-Layer Spoofing Detection With Reinforcement Learning in Wireless Networks，" in IEEE Transactions on Vehicular Technology，vol. 65，no. 12，pp. 10037-10047,Dec. 2016，doi: 10. 1109/TVT. 2016. 2524258.

基于物联网星型网络下的物理层密钥生成方法

曹志[1，2]，金梁[3]，韩乾[2]

[1]东南大学 网络空间安全学院，江苏 南京 211111；

[2]网络通信与安全紫金山实验室，江苏 南京 211111；

[3]国家数字交换系统工程研究中心，河南 郑州 450002

摘　要：随着计算能力的飞速发展，物联网设备在星型网络下的传统加密手段存在风险，且密钥分发，更新等过程复杂，其安全性面临巨大挑战。物理层安全手段可以改变这一现状，故一系列的密钥生成方法应运而生。本文提出了基于物联网星型网络下物理层密钥生成方法，通过对协商后的保密序列进行校验，判断是否改变量化阶段的门限值或者修改纠错码码率，来提高协商后的密钥一致性。仿真结果证明了本文所提出的方法较传统物理层密钥生成方法具有显著的优势。

关键词：物联网星型网络、物理层安全、密钥生成

Physical layer key generation method based on IoT star network

CAO Zhi[1,2], JIN Liang[3], HAN Qian[2]

1.School of Cyber Science and Engineering, Southeast University, Nanjing, Jiangsu, 211111;

2.Purple Mountain Laboratories, Nanjing, Jiangsu, 211111, China;

3.National Digital Switching System Engineering & Technological R&D Center, Zhengzhou, Henan, 450002

Abstract: With the rapid development of computing power, the traditional encryption methods of IoT devices under star networks are at risk, and the key distribution and update processes are complicated, and their security faces huge challenges. The physical layer security measures can change this situation, so a series of key generation methods have emerged. This paper proposes a method for generating a physical layer key based on the star network of the Internet of Things. By checking the negotiated secret sequence, it is judged whether to change the threshold value of the quantization stage or modify the error correction code rate to improve the ability of reconciliation stage. The simulation results prove that the method proposed in this paper has significant advantages over traditional physical layer key generation methods.

Key words: IoT star network; Physical Layer Security; Secret Key Generation

1　引言

随着无线通信技术飞速发展，物联网必将成为未来发展的关键。它是在互联网基础上延伸和扩展的网络，是将各种信息传感设备与互联网结合起来而形成的一个巨大网络，实现在任何时间、任何地点，人、机、物的互联互通，实现真正的“万物互联”[1]。物联网的快速发展必然会带来新的挑战，特别是安全问题，但是技术的更新也同样会带来解决问题的全新思路。

传统的通信加密手段是基于上层协议的公钥加密体制，密钥的强度源自于密钥的长度，而密钥生成分发和更新机制是基于设备固有密钥和上层协议来实现，这样的经典加密和密钥分发方式并不适合物联网设备的指数型增长的场景需求。随着量子计算机的快速发展，通过“计算安全”来间接实现“绝对安全”的手段已经出现破绽；而随着物联网终端数量的指数型增长，使得传统密钥分发更新机制变得异常繁重，需要大量的密钥分发和管理的开销，不适合用于资源受限的物

基金项目：国家自然科学基金面上项目资助（No. 61871404）。

联网场景。传统的通信安全的定位一直是服务于通信，依托于通信，不能有效地解决无线通信安全问题，而无线物理层安全，利用无线信道的随机性以及合法通信双方信道的唯一性和互易性，从无线信道的物理属性入手，挖掘其中的内生安全因素并解决通信过程中的安全缺陷[2]。其中利用无线信道特征生成物理层密钥的方法，适应物联网的场景需求，受到了越来越多研究者的关注。

物联网设备在星型网络下的传统加密手段存在风险，且密钥分发，更新等过程复杂，其安全性面临巨大挑战。为了解决该问题，结合上文所述的物理层密钥提取方法的众多优势，本文提出了基于物联网星型网络下物理层密钥生成方法，通过对协商后的准保密序列进行校验，判断是否改变量化阶段的门限值重新进行量化，从而在量化阶段减少错误比特数量，或修改纠错码码率，来提高纠错性能以实现协商后的密钥一致性。量化过程使用双门限单比特方法，门限设置简单；协商过程使用Polar码，只需一次传输协商信息，避免设备之间的多次信息交互；校验阶段使用水平垂直校验码，简单高效，易于实施；通过本文设定的用户码可以实现星型网络下的独立密钥分发和组密钥分发。

本文的其余部分组织如下。第二节介绍了物理层密钥提取过程及星型网络拓扑系统模型。在第三节提出基于物联网星型网络下物理层密钥生成方法。第四节对本文提出方案进行仿真分析。第五节总结全文。

2　物理层密钥提取及系统模型

2.1　物理层密钥提取

上世纪九十年代Maurer等学者开展利用公共随机信道提取密钥的一系列研究[3]-[5]，并提出了公共信道密钥提取模型，如图1所示，合法通信双方Alice和Bob从公共信道中获取共同参数并通过协商等步骤生成物理层密钥，而整个密钥生成过程中Eve可以窃听。

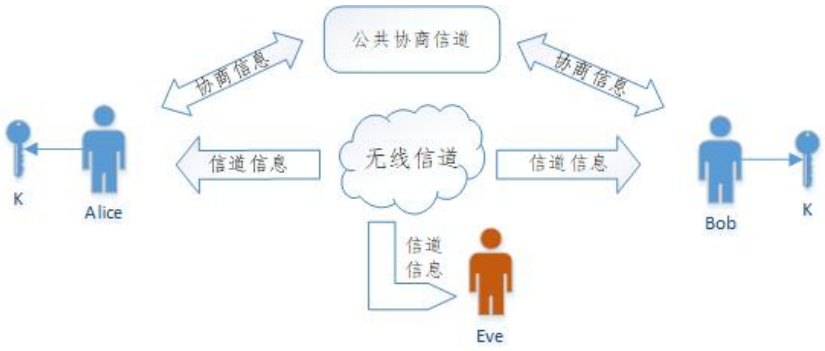


图1　公共信道密钥提取模型

物理层密钥生成流程分为如下四个步骤：信道探测，量化，信息协商，保密增强，如图2所示。详细工作原理见第三节。

2.2　系统模型

针对物联网中基于星型网络拓扑的密钥生成系统进行建模，如图3所示。A_1，A_2，…，A_N，Bob为合法通信网络节点，其中Bob为中心节点，其它节点需要与中心节点生成相同密钥，以保持与中心节点之间的安全交互。在该网络中存在一个窃听者Eve试图窃取保密信息，为了隐藏自身，仅被动窃听，不发送任何干扰信息，且距离任意一个合法节点的距离均不小于半个波长，所以窃听信道信息与合法信道信息不相关。整个模型中所有设备皆配备了单收发天线。节点之间的信道模型为复高斯信道，相干时间T内信道不变，中心节点与不同节点之间的信道（h_1，h_2，…，h_n）是相互独立，且满足互易性。

基于上述模型，本文提出了基于物联网星型网络下物理层密钥生成方法。中心节点与其它节点可以独立实施本文方法实现密钥的提取。

3　基于物联网星型网络下的物理层密钥生成方法

物联网星型网络场景与普通通信场景的不同，部分密钥生成方法不太适合物联网星型网络场景，因此本文提出了基于物联网星型网络下的物理层

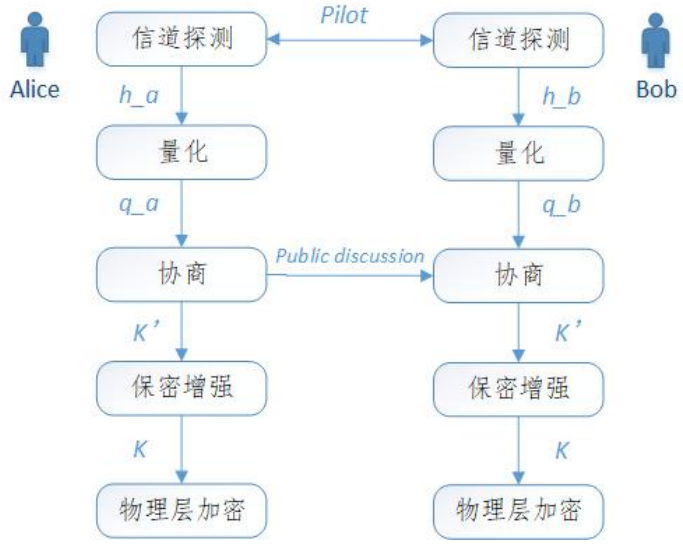


图 2　密钥提取过程

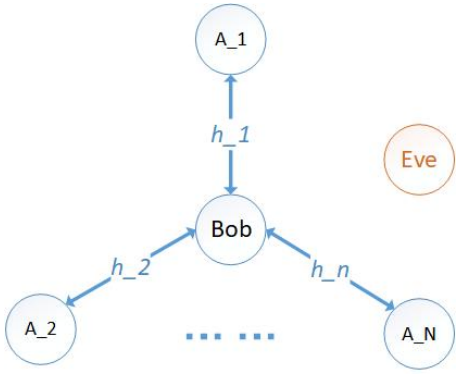


图 3　系统模型

密钥生成方法。以其中一对用户为例，其中 Bob 为中心节点，而 Alice 为子节点，方案步骤如图 4 所示。其它子节点与中心节点之间的密钥生成采用相同的方法步骤。该方案通过用户码的设置可以实现独立密钥分发以及组密钥分发。

3.1　独立密钥生成

当物联网星型网络下的各个子节点需要与中心节点保持独立私有的安全交互时，独立密钥就变得不可或缺，本文方法可以实现星型网络下的子节点与中心节点之间生成独立的物理层密钥。

3.1.1　信道探测阶段

信道探测是从无线信道中获取物理信息的手

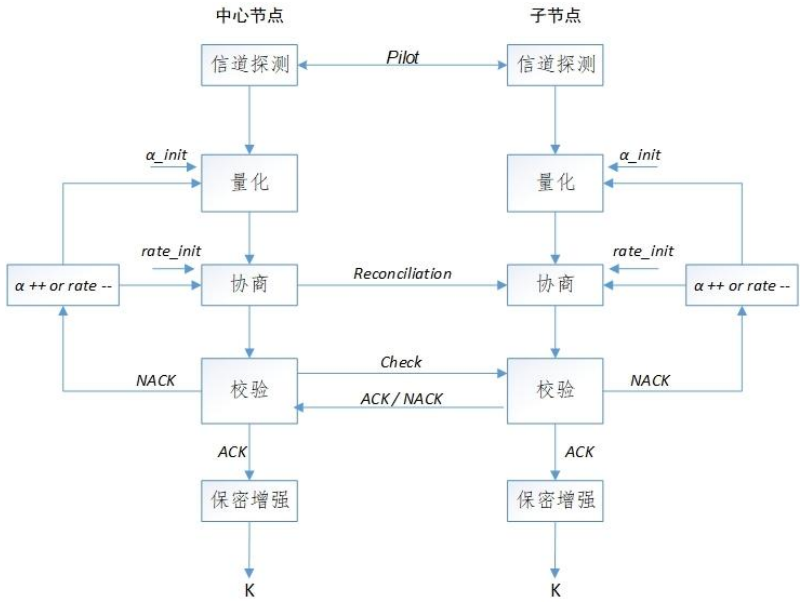


图 4　方案流程

段，而信道特征则是物理信息的表征，常见的参数有接收信号强度（RSS）[6][10]，信道冲激响应（CIR）[7]，信号包络[8]，到达角度（AoA）[9]等。现有的无线设备均可以采集RSS信息，且获取方法简单，因此本文方法利用RSS采样值来提取密钥。文献［10］对采样值进行分数插值，实现了双方信号估计值的准同步，以获取更加同步精准的信道信息；而文献［6］对发送导频的方式进行了设计，在WiFi繁忙条件下使用蓝牙随机跳频来发送导频；但是本文避免使用复杂过程，信道探测使用最简单的方式：当通信双方需要生成物理层密钥时，Alice与Bob相互发送导频估计信道，Bob估计出信道参数h_b，Alice估计出信道参数h_a。

3.1.2 量化阶段

本文量化阶段使用双门限的单比特量化方法[11]，摒弃了单门限量化方法[12]的弊端。其上下门限的设置通过采样值的统计信息（μ，σ）以及量化门限参数α进行调整。上下门限设定为：$\begin{cases} q^{+} = \mu + \alpha * \sigma \\ q^{-} = \mu - \alpha * \sigma \end{cases}$，量化函数为$c(x) = \begin{cases} 1, & x \geqslant q^{+} \\ drop, & q^{-} < x < q^{+} \\ 0, & x \leqslant q^{-} \end{cases}$，通过舍弃位于门限值之内的采样点来主动避免量化错误。通过对量化区间更加细致的划分可以获得多比特的量化结果，使用格雷码给区间分配码字效果更优[11][13]，但是与此同时也会带来更多的错误，需要更多的处理，因此本文仅仅使用双门限的单比特量化，并且双方共同维护α_*list*，根据标记*Flag_q*来调整α的取值，进而调整量化门限。

3.1.3 信息协商阶段

上下行信道具有互易性，但由于噪声，收发机硬件差异以及量化误差等影响，导致了合法通信双方各自量化后的保密序列存在一定的差异，需要通过信息协商方法来纠正密钥比特中的不一致比特。现有的信息协商方法主要有两类，第一类是Cascade[13]，winnow[14]等多次交换校验信息来纠错的方法及其改进方案；第二类是利用纠错编码来实现协商的方法[15]-[20]。

本文信息协商参考纠错码与异或操作相结合的方法[15]，使用Polar码来实现。中心节点会随机生成用户码*S*，*S*的长度根据量化后比特长度*Length_Q*以及码率*rate*来确定，通过速率匹配来实现。*Length_S* = *rate***Length_Q*，因此本文密钥长度仅与量化后保密序列长度有关，而用户码的长度则会随着*rate*与*Length_Q*的变化而变化。码率根据校验反馈自适应调节，由下文4.1节仿真可知量化门限的设置会导致密钥长度的减少，而本文码率调节并不会降低密钥长度，但是会使得用户码S缩短，同样会影响公共协商信息的保密性能[21]。本文为了简单实现，优先考虑降低码率*rate*，后考虑提升α。通过标记*Flag _r*来调整码率*rate*的取值，协商流程如图5所示。对于不同子节点，中心节点生成不同的用户码。

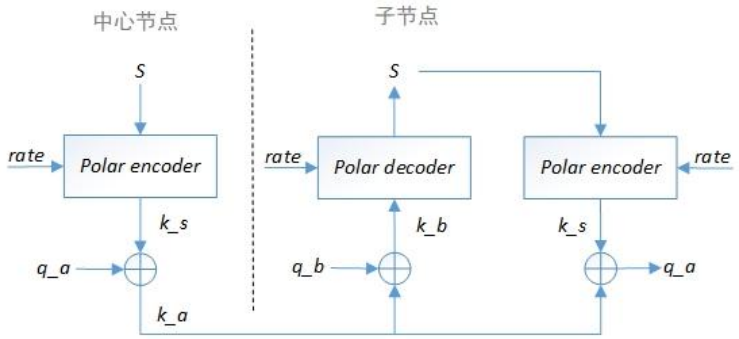


图5 协商过程

3.1.4 校验阶段

本文在以往密钥生成的基础上增加了校验判断环节，利用水平垂直校验来对协商后的信息进行校验，并通过校验返回值来调整α或者*rate*，如图6所示。校验阶段只需要将协商后的信息进行简单的串并转换以及奇偶校验即可，简单高效，而且只需交互一次，只需要返回信息ACK/NACK即可。

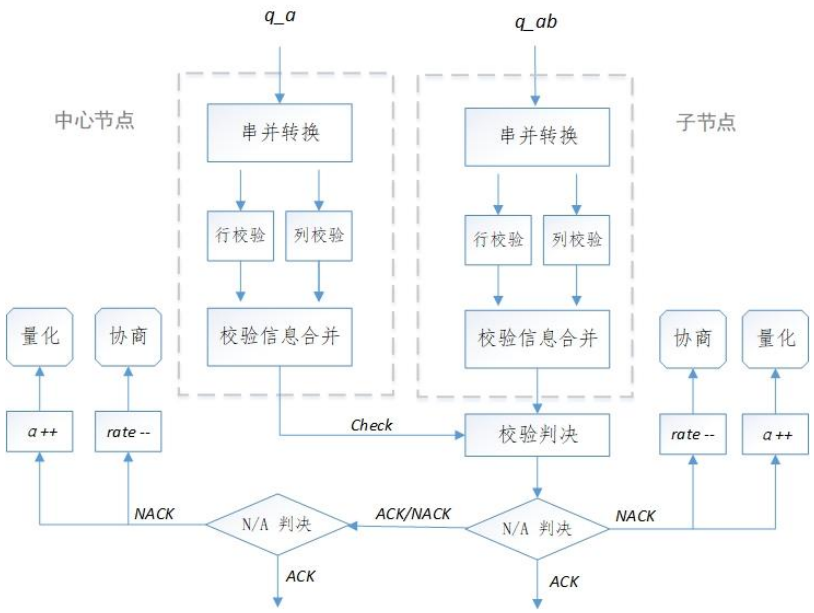


图 6　校验流程

3.1.5　保密增强阶段

信息协商以及校验过程中需要发送校验信息，而这会导致部分原有信道特征信息不在安全。保密增强的目的就是要去除协商过程中暴露的信息，使得最终密钥安全可靠。常用的手段是通过 Hash 函数 [22] 来实现，与此同时 Hash 函数还会压缩协商后的密钥比特，并生成固定长度的密钥比特。本文亦是采取 Hash 函数来实现。

3.2　组密钥生成

独立密钥可以满足子节点与中心节点之间独立安全的通信，但是物联网星型网络中存在组密钥通信的情况，用以满足中心节点对网络下的子节点的统一调度，并且可以实现子节点之间的安全通信，由星型拓扑扩展成网络拓扑，实现更一般的通信模型。本文方法同样可以实现组密钥的生成，方法与独立密钥基本相同，信息协商方面参考文献［20］。

信道探测，量化阶段以及保密增强与独立密钥生成基本一致。信息协商过程由中心节点广播协商信息，并通过校验来实现；广播信息由组用户码以及中心节点与各信道的量化值生成。

方案步骤如下，中心节点随机生成一个组用户码 S_g，并进行 polar 编码生成 K_g；而中心节点各个量化值 $\{q_1, \cdots, q_n\}$ 与之异或生成一组协商信息集合 $\{K_1, \cdots, K_n\}$，并以此作为协商信息广播给各个自节点。各子节点根据自身量化值 $q_i \in \{q_1, \cdots, q_n\}$ 与广播信号异或并通过 polar 解码获得组用户码 S_g，后以同样的步骤生成 K_g，再次与广播信息其它字段 $K_j \in \{K_1, \cdots, K_n;\ j \neq i\}$ 进行异或，由此可以获得其它节点的量化值 $\{q_1, \cdots, q_{i-1}, q_{i+1} \cdots, q_n\}$。最后所有节点获得中心节点的各个量化值 $\{q_1, \cdots, q_n\}$。

以上协商过程中如果出现了协商失败必定会导致组密钥生成失败，所以在各子节点协商之后需要进行校验。码率调节以及量化的自适应调节皆需要调整组用户码，以此用来重新生成合适的广播信息，使协商失败节点重新获得量化值。中心节点维护所有节点对应的组用户码以及修改后版本，待所有校验结束后，广播各节点通知丢弃部分修改后的量化值，由标记 Flag_d 指示，使得所有用户获得相同的量化值集合，之后通过 Hash 函数来实现保密增强，并获得固定长度的组密钥。以图 7 为例，中心节点 Bob，子节点 A_1，A_2，A_3，其中协商过程 A_3 第一次协商失败，后续通过自适应调整而协商成功，其它子节点皆是第一次协商就成功。

信道探测以及量化后，各节点获得不同量化值。Bob 设置组用户码 S_g 通过 Polar 码生成 K_{S1} 并异或估计量化值生成广播信息 $B=\{K_{S1} \oplus q1,\ K_{S1} \oplus q2,\ K_{S1} \oplus q3\}$；A_1 节点与 A_2 节点皆第一次协商成功获得正确的 $\{K_{S1}, q1, q2, q3\}$ 和 $\{K_{S1}, q1, q2, q3\}$；而 A_3 则校验失败，后与 Bob 重新

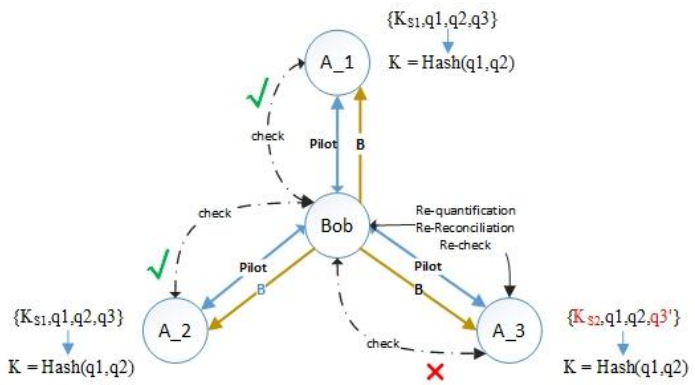


图 7　四节点组密钥生成模型

协商后成功获得｛K_{S2}，q1，q2，q3’｝；校验结束后Bob广播通知各子节点舍去A_3量化值；最终星型网络下的所有节点将｛q1，q2｝通过Hash函数生成组密钥。

3.3　方案分析

本文方法重点的改进体现在量化与信息协商阶段，将量化阶段与协商阶段相结合，根据协商结果的校验值进行自适应码率调节或自适应量化。以下从四个方面对本文提出的方法进行了分析：（1）从信道探测角度来看，RSS易于探测，满足物联网设备的性能。（2）从量化角度来看，量化门限的设置只需要简单的计算即可，α的值也是通过维护的α_list来获取，自适应量化的实现也十分简单，只需要自增α_list指针即可。（3）从协商角度分析，每次协商过程只需要一次信息交互即可，避免设备之间的频繁交互；码率的简单自适应调整同样简单，也为协商后的密钥一致率提供了额外的保障。（4）校验阶段只需要简单的串并转换以及奇偶校验即可，简单高效，而且只需交互一次，且只需要返回信息ACK/NACK；整个实现过程适合物联网星型网络下的物理层密钥生成。方案的整体实现过程如表1所示。

表 1　方案实现过程

步骤一 信道探测：星型网络下的所有节点依次发送导频序列，测量出RSS估计值，节点0为中心节点，节点$i \in \{1,\cdots,N\}$为子节点，一共N个子节点。
步骤二 量化阶段：中心节点与子节点根据α设置量化门限，分别获得量化结果Q_{0i}和Q_{i0}，α取值于α_list，由$Flag_q$指示。
步骤三 信息协商：中心节点随机生成用户码S_i或者组用户码S_g，并通过Polar码编码生成K_{si}和K_g，通过$K_i = K_{si} \oplus Q_{0i}$或者$K_i = K_g \oplus Q_{0i}$生成广播信息$\{K_1,\cdots,K_N\}$并通过公共无噪信道发送给所有子节点；子节点通过$K_{i'} = K_i \oplus Q_{i0}$操作以及解纠错码获得$S_i$或者$S_g$，后续重新编码生成$K_{si}$或者$K_g$，并执行$K_{si} \oplus K_i$或者$K_g \oplus K_i$操作，获得协商后的保密序列$r_i = Q_{0i}$或者$r_i = \{Q_{01},\cdots,Q_{0N}\}$；其中Polar编码的码率由$rate$自适应调节，$rate$取值于$rate_list$，由$Flag_r$指示。
步骤四 校验阶段：中心节点通过水平垂直校验生成校验码发送给子节点，子节点同理生成校验码并比对，判断是否协商成功，并反馈$'ACK/NACK'$；若校验结果为$'NACK'$则会相应的更改$Flag_q$和$Flag_r$的值，并重新进行量化或协商；若校验结果为$'ACK'$则进入**步骤五**。
步骤五 保密增强：(1)生成独立密钥时，中心节点与子节点对协商后的信息r_i进行哈希函数处理，获得独立密钥K；(2)生成组密钥时，中心节点广播通知子节点需要舍弃的量化值序号，由$Flag_d$指示，中心节点与子节点对协商后的信息r_i剔除$Flag_d$所指示的量化值而获得r_g，对其进行哈希函数处理，获得组密钥K_G。

4　仿真分析

本节主要对前文所提到的密钥生成方案进行仿真分析，测试该方案在复高斯信道下的性能。仿真研究：□ 自适应量化门限调节；□ 自适应码率调节；□ 本文提出方法与参考方法对比。

以其中任意一子节点Alice和中心节点Bob一对通信双方为例，窃听者为Eve，收发皆单根天线。导频序列包含256个符号，α取自于双方共同配置α_list＝［0.1，0.2，0.3，0.4，0.5，0.6，0.7，0.8，0.9］，常规配置下α = 0.2，特定仿真条件下

选取所有取值。上下门限为：$\begin{cases} q^{+} = \mu + \alpha * \sigma \\ q^{-} = \mu - \alpha * \sigma \end{cases}$，而量化函数：$c(x) = \begin{cases} 1, & x \geqslant q^{+} \\ drop, & q^{-} < x < q^{+} \\ 0, & x \leqslant q^{-} \end{cases}$。polar码编码码率 *rate* 取自于 *rate_list* = ［3/4，2/3，1/2，2/5，1/3，1/4］，根据仿真条件取值。水平垂直校验过程将协商后的密钥序列向量转化为矩阵块，按照行列分别进行奇偶校验，生成行、列校验，后合并成校验信息发送给校验方。

4.1 自适应量化门限调节

量化门限对量化结果产生重要的影响，本文通过α来改变量化门限。下文对α的不同值对量化后保密序列的误比特率的影响进行了仿真，设定了3种信道条件 *SNR_dB*，9种α取值，如图8所示。从仿真结果可以发现，随着α的增大，量化后的误比特率明显下降，这是由于随着α的增大，上下门限之间的间距就增大，即代表被舍弃的采样数据的数量增大，从而量化后的值一致率更高。但是与此同时量化后的保密序列的长度也会随之下降，如图9所示。该调节方法是一种利用效率换取一致率的折中，所以如何选择α就变得十分重要，故本文提出了自适应选择的方法。

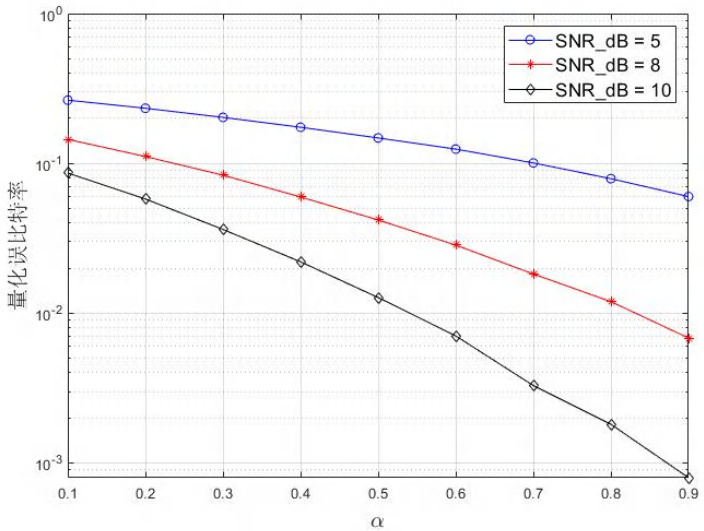


图8　α对量化结果的影响

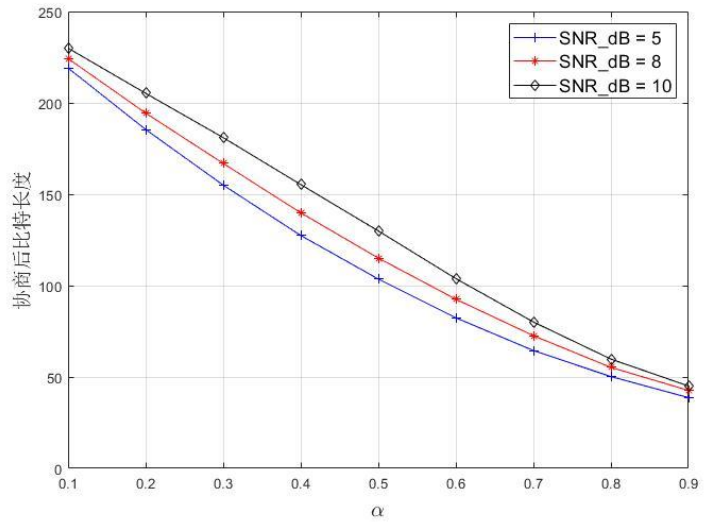


图9　α对量化后保密序列长度的影响

以下对自适应量化与固定α = 0.2的量化进行了对比仿真，由于需要协商后校验信息作为反馈来调节量化门限，所以协商过程固定码率rate = 1/3，使得α为单一变量，仿真结果如图10所示。由于本文采用了自适应量化方法，主动通过调节量化门限来避免错误比特，虽然减少了密钥比特的长度，见表2所示，但是较固定的量化门限可以获得更低的误比特率，大约有3dB的信噪比增益。

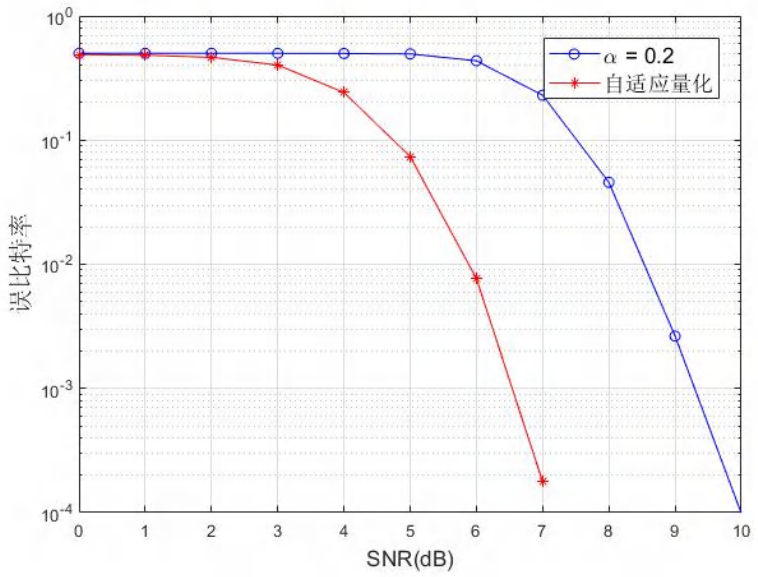


图10　自适应量化与固定$\alpha = 0.2$量化对比

表2　协商后保密序列长度对比

SNR(dB)	≤3	4 ~ 5	6 ~ 8	＞8
α	0.8	0.8	0.2 ~ 0.8	0.2
Length_k(bits)	—	40~50	60 ~ 180	210 ~ 230

表中所示的保密序列长度为协商成功情况下的保密序列长度；其中“—”代表无法成功协商。

4.2　自适应码率调节

本文采用了自适应码率调节方法，校验失败后通过主动通过降低码率来增强Polar码的纠错性能以保障协商后保密序列的一致性。以下对自适应码率与非自适应码率两种场景下的协商后保密序列的误码率进行了仿真分析，仿真中固定α = 0.2，非自适应码率场景固定rate = 1/2，自适应码率场景下rate取自*rate_list*。如图11所示，相比于非自适应码率场景，采用自适应码率可以获得更低的误码率，这是因为自适应码率会根据反馈结果来降低码率来提高Polar编码的纠错性能。

4.3　与参考方法对比

由于噪声，收发机硬件差异以及量化误差等影响，导致了合法通信双方各自量化后的密钥比特存在一定的差异，需要通过信息协商来纠正密钥比特中的不一致比特。下文以cascade方法作为参考方法一；以文献［16］，文献［21］等不使用校验过程的纠错码（Polar）协商方法作为参考方法二与本文方法进行了仿真对比，如图12所示，同时标注了量化后的误比特率，用来凸显各种协商方法对于性能提升的差异。

（1）当信道情况较差时，Alice和Bob量化后保密序列具有较大的差异，皆超出了三种方法的纠错能力，特别是使用Polar码的本文方法以及参考方法二并不会降低量化后的误比特率，相反，对比量化后的误比特率可以发现，这两种方法的使用可能会增加误比特率；而参考方法一只可以减少部分量化后的误码率，完全达不到密钥一致率的要求；三种方法的误比特率皆维持50 %左右水平，都不能实现密钥提取。

（2）随着信道条件的改善，量化后的保密序列的误比特率也随之降低，但仍然不能直接用来生成密钥，需要进行协商操作。由于参考方法一的纠错性能有限，经过协商后的误比特率只是相比于量化后的保密序列误比特率有所降低，完全达不到密钥一致率要求；与其相比，本文方法以

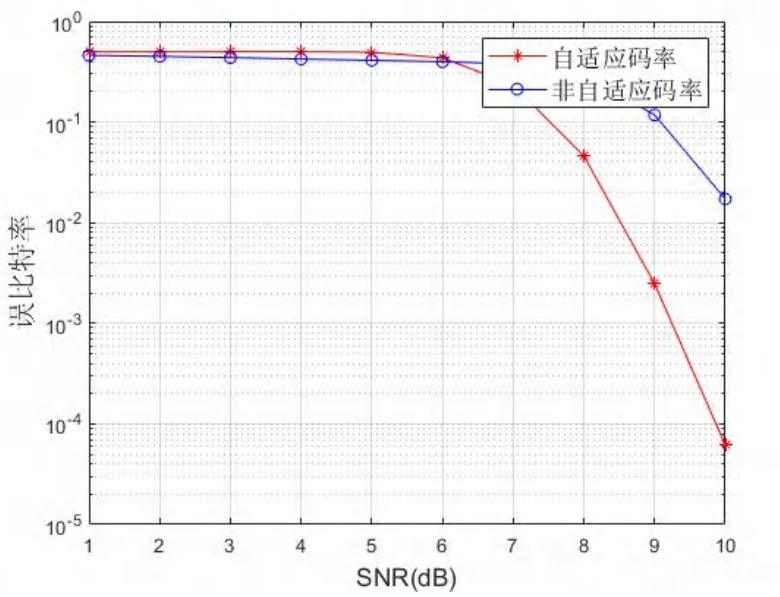


图 11　自适应码率与非自适应码率对比

及参考方法二使用的Polar码展现出了其优秀的纠错性能，可以获得较参考方法一更低的误码率；进一步来说，基于本文自适应量化门限调整和码率的调节，使得本文方法较参考方法二具有更低的协商后保密序列误码率。

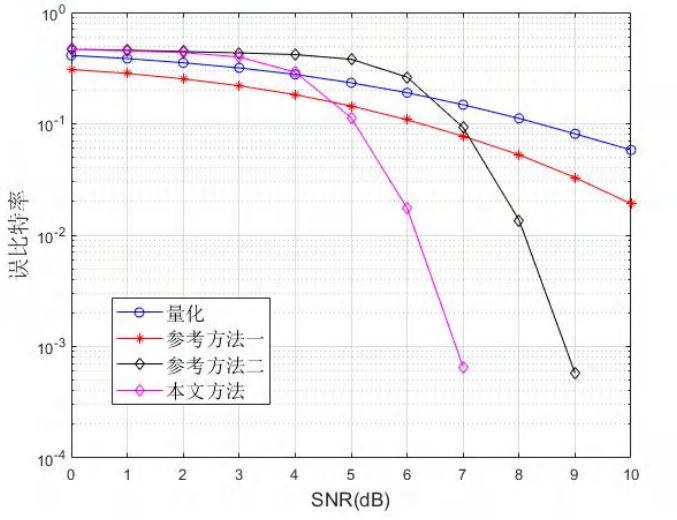


图 12　本文方法与参考方法比较

以下内容对本文所提出的方法在组密钥生成中的应用进行了仿真，并以参考方法二组密钥生成作对比，体现了本文方法密钥生成效率的优势，如图13所示。性能指标为每次信道密钥生成率，取值范围为0~1，表明每次探测成功生成密钥的概率，反映了密钥生成的效率。仿真中用户节点数为3，子节点数为2。仿真结果可知本文方法由于使用了自适应的量化门限和码率调节，较参考方法二可以获得3～4 dB的信噪比增益，即可以获得更优的密钥生成效率。

5　结束语

本文针对物理层密钥生成，提出了一种基于物联网星型网络下物理层密钥生成方法，通过对协商后保密序列的校验信息做判断来调整量化门限或者纠错码码率，提高系统的可靠性；信息协

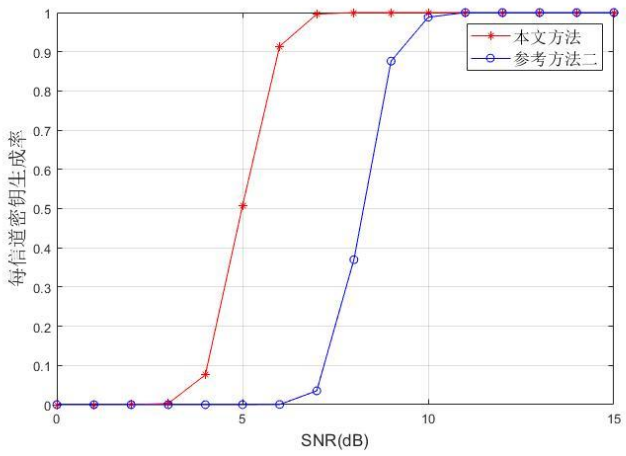


图 13　组密钥生成效率

商过程利用 Polar 来实现，将通信双方的量化差异转化为传输过程中的造成误差，通过纠错码来全部纠错；协商后的校验信息使用水平垂直校验码来生成，校验方对校验码进行判断并反馈判断结果，用以决定 Polar 码码率及量化门限。仿真结果显示，本文方法较参考方法有更优的性能，并且量化门限和码率的自适应调节也保证了协商后保密序列的长度以及协商和校验信息的保密性。后续工作将围绕在量化门限和码率之间的最优分配开展，进一步优化系统的性能。

参考文献:

[1] 季新生,黄开枝,金梁,等. 5G 安全技术研究综述[J]. 移动通信，2019,43(1)：34-39.

[2] 黄开枝,金梁,钟州 . 5G 物理层安全技术——以通信促安全[J]. 中兴通讯技术,2019,25(04):43-49.

[3] Maurer U. Secret key agreement by public discussion from common information [J]. IEEE Transactions on Information Theory，1993，39(3)：733-742.

[4] Ahlswede R, Csiszar I. Common randomness in information theory and cryptography. I. Secret sharing [J]. IEEE Transactions on Information Theory It，1993，39(4):1121-1132.

[5] Ahlswede R, Csiszar I. Common randomness in information theory and cryptography. II. CR Capacity [J]. IEEE Transactions on Information Theory，1998，44(1)：225-240.

[6] Premnath S N, Gowda P L, Kasera S K, et al. Secret key extraction using Bluetooth wireless signal strength measurements[C]// Eleventh IEEE International Conference on Sensing. IEEE，2014.

[7] Wang Q, Xu K, and Ren K. Cooperative secret key generation from phase estimation in narrowband fading channels [J]. IEEE Journal on Selected Areas in Communications，2012，30(9)：1666-1674.

[8] Zhang J, Marshall A, Hanzo L. Channel-Envelope Differencing Eliminates Secret Key Correlation: LoRa-Based Key Generation in Low Power Wide Area Networks[J]. IEEE Transactions on Vehicular Technology，2018:1-1.

[9] Jiao L, Tang J, Zeng K. Physical Layer Key Generation Using Virtual AoA and AoD of mmWave Massive MIMO Channel[C]// 2018:1-9.

[10] Patwari N, Croft J, Jana S, et al. High-Rate Uncorrelated Bit Extraction for Shared Secret Key Generation from Channel Measurements[J]. IEEE Transactions on Mobile Computing，2009，9(1):17-30.

[11] Premnath S N, Jana S, Croft J, et al. Secret Key Extraction from Wireless Signal Strength in Real Environments [J]. IEEE Transactions on Mobile Computing，2013，12(5):917-930.

[12] Aono T, Higuchi K, Ohira T, et al. Wireless secret Key generation exploiting reactance-domain scalar response of multipath fading channels[J]. IEEE Trans Antennas and Propagation，2005，53(11)：3776-3784

[13] Han, PengB. , WuS. , WangC. , WangX. , B. LoRa-Based Physical Layer Key Generation for Secure V2V/V2I Communications [J]. Sensors 2020，20，682.

[14] Buttler W T, Lamoreaux S K, Torgerson J R, et al. Fast, efficient error reconciliation for quantum cryptography [J]. Physical Review A，2003，67.

[15] Junqing Z, Trung D, Roger W, et al. Securing Wireless Communications of the Internet of Things from the Physical Layer, An Overview[J]. Entropy，2017，19(8).

[16] 肖帅芳,郭云飞,白慧卿,金梁,黄开枝 . 面向物联网准静态信道的中继协作密钥生成方法[J]. 电子与信息学报,2018,40(01):50-56.

[17] 肖帅芳,郭云飞,黄开枝,金梁 . 面向物联网多跳中继系统的协作密钥生成方法[J]. 通信学报,2018,39(03):86-94.

[18] 戴峤 . 基于有噪私密信道的物理层密钥分发技术研究[D]. 解放军信息工程大学,2013.

[19] Peng L, Li G, Zhang J, et al. Securing M2M Transmissions Using Nonreconciled Secret Keys Generated from Wireless Channels[C]// 2018 IEEE Globecom Workshops (GC Wkshps). IEEE，2018.

[20] Li G, Hu L, Hu A. Lightweight Group Secret Key Generation Leveraging Non-Reconciled Received Signal Strength in Mobile

Wireless Networks［C］// 2019 IEEE International Conference on Communications Workshops (ICC Workshops). IEEE，2019.

[21] Li G, Zhang Z, Yu Y, et al. A Hybrid Information Reconciliation Method for Physical Layer Key Generation［J］. Entropy，2019，21（7）：688.

[22] Bennett C H, Brassard G, Crepeau C, et al. Generalized privacy amplification［J］. IEEE Transactions on Information Theory，1995，41(6)：1915-1923.

TOE技术及应用研究综述

任芸莉，刘冬培，韩国栋
解放军战略支援部队信息工程大学，河南郑州 450002

摘　要：在互联网的推动下，以太网速率迅速增长，达到现在的10Gb/s，因而导致CPU负担过重。TOE技术的出现很好的解决了这个问题，它将占用大量资源的TCP/IP协议进行卸载，使CPU可以把更多的资源放在处理上层应用上，提高CPU的处理能力和处理效率。本文从TOE的相关背景、TOE的实现方法、TOE的应用、TOE存在的问题以及TOE未来工作方向五个方面对TOE技术的研究现状进行了总结，并对比了不同实现方法的优缺点，分析了其适用的条件，并思考了TOE技术的进一步研究方向。

关键词：TOE技术、TCP/IP协议、开销、卸载、以太网

Summary of TOE Technology and Application Research

Ren Yunli, Liu Dongpei, Han Guodong
Information Engineering University, Zhengzhou, Henan, 450002

Abstract: Driven by the Internet, the Ethernet rate has increased rapidly, reaching the current 10Gb/s, which has caused an excessive burden on the CPU. The emergence of TOE technology has solved this problem very well. It will offload the TCP/IP protocol, which takes up a lot of resources, so that the CPU can put more resources on processing upper-layer applications, and improve the processing power and efficiency of the CPU. This article summarizes the research status of TOE technology from five aspects: TOE related background, TOE implementation method, TOE application, TOE problems and TOE future work direction. What's more, it not only compares the advantages and disadvantages of different implementation methods but also analyzes its applicable conditions, and thinks about the further research direction of TOE technology.

Key words: TOE technology; TCP/IP protocol; overhead; offloading; Ethernet

1　引言

1.1　背景

互联网的日益普及刺激了互联网上传输数据的爆炸性增长和传输速度的急剧提高。TCP/IP协议历来是处理网络上数据传输的标准，它提供了端系统间数据的可靠传输、多路并传、流量控制和拥塞控制等功能，从某种意义上说，它已成为存储和集群领域的首选协议。卸载指的是把CPU上的一些工作转移到其他位置来减轻CPU的负担。传统上，对于TCP/IP协议栈的处理是由CPU完成的，图1所示为传统网络数据包的处理流程，但是尽管CPU的性能不断提高，它仍然不能满足网络速率迅速增长的需求。CPU大量的资源都用在了处理网络协议，而不是处理上层的应用，这会使服务器整体性能降低。一种称为TCP/IP卸载引擎（TCP/IP Offload Engine，TOE）的技术应运而生，它引入了一种新的网络接口体系结构，将TCP/IP

协议栈的处理工作部分或全部交由外置硬件加速卡处理，不仅能提高网络处理带宽，还能在一定程度上防止软件攻击，保证协议处理过程中数据安全。[1]

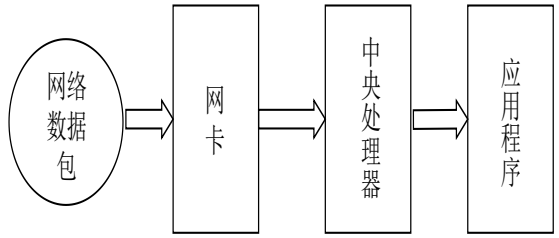


图1　传统网络数据包处理过程

TCP/IP卸载引擎体系结构图2所示，它是根据操作系统的网络体系结构扩展而成的。

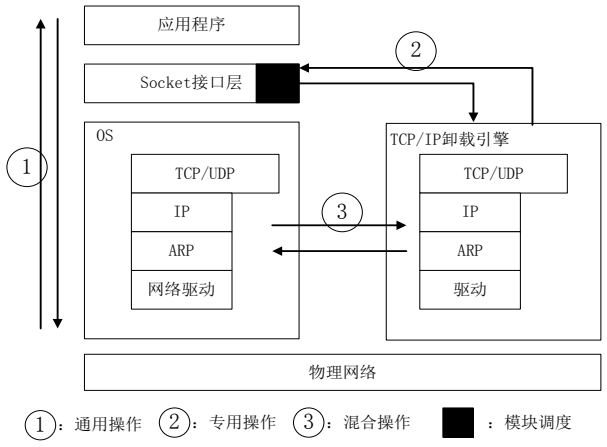


图2　TCP/IP卸载引擎体系结构

通过图2可以说明：（1）□代表传统的数据处理过程，表明该结构可以在普通网卡上处理数据，说明图2结构有通用性；（2）在□处的箭头表明，TCP/IP协议栈处理流程不是由CPU处理，而是通过串口模块交到卸载引擎处理，说明这个结构有专用性；（3）□表明把协议栈的处理分为两部分，一部分交给卸载引擎，而剩下一部分由CPU处理，这说明这个结构有扩展性。图中两种协议栈的处理流程在原则上是一致的。二者最大区别在于操作系统里的网卡驱动是针对普通的以太网卡，而卸载引擎里的驱动则是针对实现了卸载功能的专用网卡[2]。

1.2　TOE技术的优势

用传统的处理方法处理TCP/IP协议会使系统产生庞大开销，其中主要是协议处理、数据拷贝和中断处理这三个方面，TOE技术可以减少这三方面的开销。

1.2.1　协议处理

对TCP/IP协议的处理占用了CPU资源的很大一部分，在协议处理时，CPU的作用是进行通用计算而不是输入输出操作，因此可以使用TOE技术将协议从CPU上卸载下来，这样可以解决对TCP/IP协议栈的处理使CPU负担过重的问题。

1.2.2　数据复制

应用程序在收发数据时，同一数据要复制好几份。普通的网卡只能降低发送数据的复制量，而有TCP卸载引擎的网卡不仅可以降低发送数据的复制量，还能够降低接收数据的夫质量，它在接收数据时直接把数据复制到应用程序的缓冲区，而没有先把它复制到服务器的缓冲区，这种方式避免了网卡和服务器间不必要的数据往复拷贝。实测证明，对于文件服务器和以内容服务为主的服务器应用环境来说，使用TCP减负引擎网卡（TNIC）代替普通网卡，相当于为服务器增加了一个CPU[3]。

1.2.3　中断处理

I/O中断的操作也很难满足大量的协议数据处理的需求，大量I/O中断操作迅速增加了数据包的处理量，这对于服务器的中断任务是一个负担。TNIC与普通网卡的工作原理不同，前者让每个应用程序完成一次完整的数据处理进程后才触发一次中断，后者处理一个数据包就要触发一次中断。很明显，TNIC可以对于减轻服务器在中断上的负担有很大作用，也消除了过于频繁的中断事件对服务器的过度干扰。

2　TOE实现方法

2.1　卸载方式

根据TCP负载分担的策略，TOE技术可以分为全卸载和部分卸载（又称数据路径卸载）两种。用户可以根据应用要求进行选择。

2.1.1　全卸载方式

全卸载方式是把TCP/IP协议栈全部的操作从CPU卸载下来，由硬件来处理，不需要CPU的参与。为了实现全卸载，在应用层和网卡之间要有一个接口层，可以对应用程和TOE网卡的消息进

行格式转换。

2.1.2 部分卸载

10G以太网环境下，TCP连接在很长一段时间内都比较稳定，很少出现丢包，因此，对于较为可靠的数据传输环境，可以考虑采用部分卸载的方式。网络中，数据收发过程给CPU造成了很大的负担，因此可以通过部分卸载来重点减轻CPU的压力，改善应用的处理性能。部分卸载是将TCP/IP中的数据处理交给硬件，而其他部分例如TCP连接管理等交给CPU去处理。连接的建立、关闭与传统的TCP/IP协议栈处理过程相同。

2.1.3 性能分析

对于TOE性能的评价一般从延时、吞吐量、CPU使用率三个方面来考虑。

FengW[4]等人对万兆以太网适配器的性能进行评估，与非TOE实现的10.37μs相比，TCP卸载引擎（TOE）可实现约8.9μs的点到点延迟（改进约14.2%）；并且TOE的吞吐量高达7.6 Gbps，而非TOE堆栈的吞吐量为5 Gbps（提高了约52%）。结果表明使用TOE的适配器与未用TOE的相比，使用TOE的适配器获得了更好的性能，数据传输速度更快，尤其是对于大文件来说效果更加显著。

朱[5]等人从部分卸载切入，最终达到完全卸载，测试了使用TOE与未用TOE吞吐率的差别，在理想情况下，综合卸载比率、处理速度等因素从理论上得出吞吐率的最优解。

陈[6]等人以部分卸载方式实现TOE网卡的软件架构的设计，两台测试机分别是Server和Client，Client端安装的千兆网卡是IntelPRO/1000，Server端则先后安装了TOE网卡和IntelPRO/1000作性能对比测试，测试证明此种新的TOE技术可以更有效地降低主机CPU利用率，提高传输效率。

李[7]等人提出了一种基于FPGA的Smart NIC架构下高性能DNS（域名系统）权威响应流水线，在开发FPGA时不需要卸载网卡所有功能，只需要专注于所加速的服务，简化工作量并且缩短开发周期，实验结果表明，相比通用的DNS权威服务器，其吞吐量接近10 Gb链路线速，响应延迟大大降低；同时FPGA的资源开销较少，具有良好的可扩展性。

薛[8]等人研究了一种基于offload和FPGA的网络数据采集方法，将网络数据的解析工作放到FPGA中进行，解决了操作系统网络协议栈及应用软件对网络数据的解析计算会消耗大量的CPU时间，导致网络数据传输效率急剧下降，甚至出现丢包的问题。

2.1.4 比较

全卸载方式便于特殊应用的二次开发，且其性能优于部分卸载方式，但是实现难度和风险较大且网卡上的协议处理器容易成为新的网络瓶颈[9]。相反，部分卸载的实现难度较小。两者的对比如表1所示。

表1 全卸载与部分卸载的对比

比较方面/类型	全卸载方式	部分卸载方式
性能	更优	优
实现难度与风险	大	小
适用环境	连接时间短、出错率高	可靠的大数据量传输
产品	Adaptec公司的GigENAC7711网络加速卡	Alacritech的1000xl

两种卸载方式各有优势，要根据应用需求进行选择。如果连接时间较短，或者传输容易受到攻击出错的情况下，全卸载的方式性能更优，传输更加安全可靠。如Adaptec提供的GigENAC7711网络加速卡，在网络环境不佳，易出现断链的情况下，可能会有大量数据包丢失，如果采用全卸载处理方案，则可以有效避免。但是，如果环境较为安全，可以选择部分卸载方式，因为其与全卸载方式相比，性能差别不大。如Alacritech的1000xl。

由于TOE网卡主要应用于IP存储和文件服务等需要海量数据传输的方面，因此，与全卸载方式相比，TOE的部分卸载方式应用更加广泛。

2.2 实现方案

2.2.1 基于软件的实现方案

为了实现TOE技术，首先使用了基于软件的方法（软件TOE），该方法基于通用嵌入式处理器。刘[10]等人提出了一种TOE与主机操作系统的接口，实现了低延迟。尽管与在硬件中实现所有操作相比，软件TOE具有易于实现和可扩展性强的优点，但它们的性能较差，因为一般的嵌入式处理器通常比主机CPU慢。Hewlett-Packard开发的评估适配器就是软件TOE实现的一个例子。In-

tel的PRO/1000TIP存储适配器是基于通用嵌入式处理器的TCP/IP和SCSI（小型计算机系统接口）协议的另一个示例，但这些适配器的性能非常差。从理论上讲，可以使用集成在单个芯片中的多个处理器来实现基于软件的卸载引擎，但这种方法存在一定的扩展问题，例如内存争用和缓存抖动。这种多处理器方法受到不同处理器之间争夺到片外存储器路径的内存的影响，数据传输的性能较差[11]。

除此之外，基于嵌入式Linux的软件TOE还有许多问题，比如在嵌入式Linux内核和应用程序之间频繁地进行上下文切换等。因此，Sang[12]等人实现了一种改进的基于软件的TOE，称为HL-TCP100134（高性能轻量级TCP/IP）。HL-TCP是一个没有嵌入式操作系统的独立TCP/IP。采用HL-TCP协议的TOE具有零拷贝发送机制和高效的TCP重传DMA机制的特点，实验结果表明，使用HL-TCP的TOE的CPU利用率几乎为零，而普通千兆以太网的CPU利用率约为23%，证明此方法具有良好性能。Jang[13]等人也证明了HL-TCP的设计使TCP/IP堆栈非常适合嵌入式系统中的高性能网络。

2.2.2 基于硬件的实现方案

基于硬件（hardware-TOE）的应用方法有：使用离散的现成组件构建离散的板级解决方案、专用集成电路的研制和基于FPGA的实现方法等。

2.2.2.1 离散板级集成

离散TOE板级集成解决方案是一种特殊用途的计算机卡上解决方案，其设计适合于系统的扩展总线，以便处理TCP算法。这种离散的板级解决方案优点是过程简单、开发预算小、具有很大灵活性以及可以进行多功能升级。

2.2.2.2 ASIC方法

电路离散板级设计的广泛灵活性可能会使产品设计师的工作更容易，但性能却不是最佳的，而ASIC解决方案不必为提供相同的功能而承担不必要的开销。例如，不需要在ASIC中复制许多未使用的功能。因此，ASIC的操作可以非常高效，这也使开发专用硬件，如ASIC成为目前实现TOE的常用方法。有几种基于ASIC的协议卸载引擎，如Chelsio的Terminator6芯片等。这种基于ASIC的协议卸载引擎芯片表现出了优异的性能，但在灵活性、可编程能力和扩展性方面比较差。为了进一步提高处理的灵活性又保证处理的速度，现在又出现了FPGA固件方案。该芯片灵活性高，但是硬件的成本比较高。

Hoskote[14]等人提出了一种可编程专用硬件引擎的设计，该引擎能够以最小的数据包大小，对饱和的10Gb/s以太网链路进行线速TCP入站处理。实现了针对TCP处理的专门指令，大大减少了每个包的处理时间。该芯片还实现了一种新的硬件数据包动态重排序算法。结果表明，由一个简单而高性能内核的专用引擎所提供的计算性能相当于在同一频率下运行的最先进的通用处理器，在芯片面积和功耗方面都有显著的节省。这样的引擎将成为TCP卸载的全面系统级解决方案的核心。

2.2.2.3 FPGA方法

由于基于ASIC的方法灵活性差，因此，为了进一步提高处理的灵活性又保证处理的速度，基于FPGA的方案被广泛使用。该芯片灵活性高，但是硬件的成本比较高

李[15]等人提出了一种基于硬件的TCP卸载引擎（TOE），它以10Gbps的吞吐率全双工运行，延迟最小。实现了拥塞避免、可配置的MSS（最大报文长度）和时间戳等支持高速网络的功能。

吴[16]等人提出了一种基于FPGA的万兆以太网TCP/IP协议处理架构，利用硬件实现了完整的TCP/IP协议栈，并将所提出的架构应用于实际万兆以太网TCP/IP卸载板卡中，有效的解决了服务器处理万兆网络流量的瓶颈问题，其协议支持ARP、ICMP、UDP、TCP等，经测试，该系统板卡功能完善，尽管其可扩展性差，但具有低响应延时、高文件传输速率和高数据吞吐率的特点，解决了CPU处理高速网路数据的高占用率、高延时的问题。

2.2.2.3 硬件实现方案对比

Mannem[17]等人从多个方面对方法进行比较，现对各方法比较整理为表2。

根据不同的应用环境，可以采用不同的方法实现。基于离散板级的解决方案更加灵活，可以实现新的特性，基于ASIC的解决方案可靠性高处理速度快，基于FPGA的方案灵活性高其性能好。

总的来说，尽管FPGA方法的硬件的开发和初始应用成本比较高， 但是FPGA技术的灵活性高、

表2　离散板级方法与ASIC方法性能的比较

比较方面/方法	离散板级集成	AISC方法	FPGA方法
并行处理	只有一个处理器，不能并行处理	多个处理器，可以并行处理	多个处理器，可以并行处理
片上存储器	存储器容量相对小	存储器容量大	存储器容量大
多个硬件计时器	必须在软件中实现TCP定时器，效率低	效率和准确性高	效率和准确性高
吞吐量	低	较高	高
能否实现万兆以太网	不能实现万兆以太网	可实现万兆以太网	可实现万兆以太网
系统成本(假设无TOE的系统为$3000)	较高	低	高
可靠性	相对较低	较高	高
灵活性	较高	低	高

有较好的可扩展性以及高性能等优势使其有了广泛应用，毫无疑问，采用FPGA方法的TOE提供了更好的性能和投资回报。

2.2.3　混合型

使用FPGA实现混合TOE有硬件资源消耗少，可移植性高等优点，适用于以硬件实现网络传输的应用场景[18]。因此，为了克服软件TOE的低性能和硬件TOE的低扩展性，Jang[19]等人提出了一种基于FPGA的软硬件协同设计和TX/RX路径分离机制的混合TOE体系结构，如图3所示，其中校验和计算等由硬件执行，连接建立、控制和重传等则由软件执行。混合TOE可以通过在硬件上处理时间关键的操作来获得高性能，并且由于可以直接添加TCP/IP的演进和软件中的新功能，所以它们保持了可扩展性。

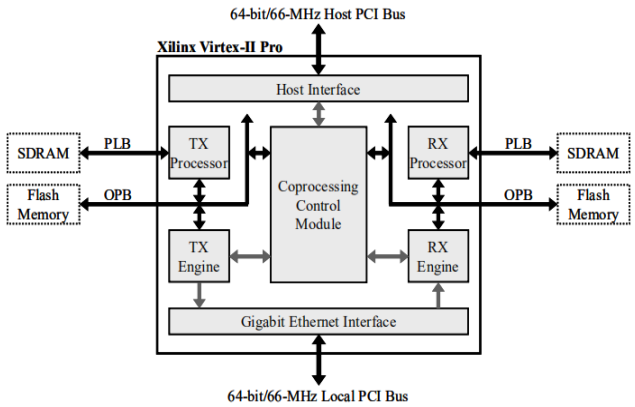


图3　TCP/IP卸载引擎的混合架构

通过实验证明，这个软硬件协同的架构可以克服单一嵌入式处理器在同时双向数据传输时性能远低于主机CPU的性能限制。这是因为在这个架构中使用了两个通用处理器独立地控制TX路径和RX路径，并且这种机制允许并行处理数据传输，并行处理结构如图4所示。除此之外，该体系结构通过使用通用嵌入式处理器和嵌入式Linux来保证可扩展性，这是因为软件很容易接受新特性，如TCP/IP的演进、安全功能、规范更新和上层协议的卸载机制。

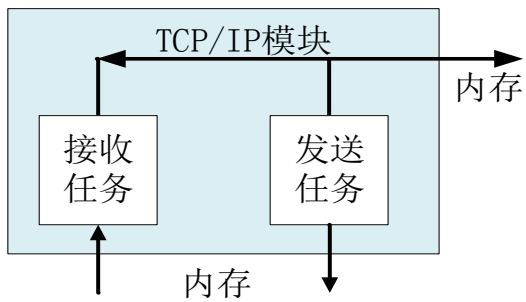


图4　TOE并行（发送、接收）结构图

随着网络速度与需求的不断增涨，将混合TOE架构应用于万兆以太网已成为不可阻挡的趋势，因为与以前的以太网相比，万兆以太网具有快速、简单及高性价比等优势。但是文献［19］

架构中的一些操作不能满足万兆以太网的时间限制，若将原架构中的PCI接口改为带x4链路的PCI express，则可以满足要求。通过接受这样的增强，用于千兆以太网的硬件实现和处理序列可以应用于万兆以太网。但是万兆以太网采用了更宽的数据宽度，这使得硬件在处理帧内数据时面临数据注册的困难。为此，文献［20］设计了FIFO形式的循环队列处理TCP帧数据注册算法，解决了FPGA设备接收、存储和发送TCP帧数据时的数据注册问题，该算法也可用于万兆以太网环境下其他协议的数据注册。

3 TOE的应用

TCP/IP卸载不仅在IP存储和企业数据业务领域发挥优势，还在文件服务、量子密钥分配、硅像素探测器读出和交互[21]、网络附加存储器（NAS）和视频编辑等方面发挥了巨大作用。其应用领域如图5所示。

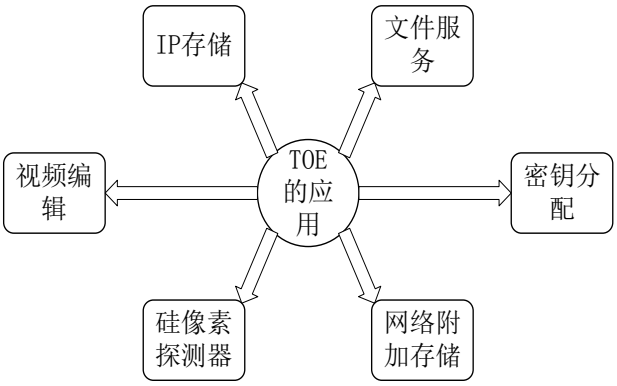


图5 TOE的应用

3.1 TOE技术在IP存储领域的应用

iSCSI（Internet Small Computer System Interface）协议是一种正处在发展中的网络存储协议，它通过IP协议传输SCSI指令和数据包。iSCSI可以用来组建IPSAN（IP Storage Area Network）[22]，IPSAN技术是存储服务器系统当中最主要也是最安全可靠的外置存储设备，外置设备最大的优势就是不会受到容量的限制，解决了磁盘本身存在的局限性。但是大量的CPU资源被占用会导致应用系统性能下降，存储系统性能低下，因此在IPSAN的协议处理过程中就明确采用了TOE技术来解决这个问题。

3.2 TOE在文件服务器的应用

TOE技术可以将协议栈卸载到硬件上，LRO（Large Receive Offload）技术可以把同一条TCP上接收到的多个小报文合并成一个大报文来传输，但因TOE技术的兼容性和安全性问题，以及LRO技术在软件上实现因而效果不明显的问题，李[6]等人基于TOE技术和LRO技术提出使用多核NPU（嵌入式神经网络处理器）来作为NIC，实现TCP接收数据路径中的校验和计算、报文乱序重组功能，并将合并之后的大报文经Linux网卡驱动程序交由协议栈处理，通过减少报文处理数量和中断数量来减轻CPU的负担，提升系统性能。被重组报文数量越多，对于TCP处理的性能越好。因此其能够在文件服务器中得到很好的应用，因为这种环境下具有大量的数据传输，而不适合像在线游戏这样延时敏感的应用场景。

3.3 TOE在量子密钥分配（QKD）系统的应用

由于量子密钥分配（QKD）系统产生的密钥可以加密通信内容，使信息传输更加安全，因此，QKD系统得到了广泛应用，但是随着信息交互需求的增加和密钥生成率的不断提高，QKD系统已经不能很好的满足需求。钟[23]等人为QKD系统定制了一个TOE将TCP/IP协议从中央处理器卸载到硬件上，硬件充当外围设备，通过直接存储器存取（DMA）和总线与CPU交互，降低了被病毒攻击的可能性。另外，在TOE中集成了身份认证算法和数据加密算法，防止第三方窃听和篡改数据，大大提高了QKD系统网络交互的安全性。与双核处理器实现的TCP/IP协议相比，优化后的交互速度提高了114%，达到560Mbps。这说明，TOE技术在QKD系统的网络交互部分有重要作用。

4 TOE问题

虽然TOE的应用提升了处理器的性能，但是仍然有许多问题急需解决：

（1）目前不同厂商部署的TCP／IP协议栈略有不同，这也意味着一个操作系统无法同时支持系统中两家厂商的网卡，因此需要规定TOE的标准接口。

（2）软件方案无法解决延时和吞吐率的瓶颈。

（3）TOE可以将数据直接放在存储器中，但是需要上层协议的协同处理。

（4）TCP/IP协议栈卸载后其数据安全性难以保证，防止第三方窃取和篡改数据是目前的一个重要问题。

5 结论和展望

TCP/IP卸载引擎技术是提升服务器性能的重要方法之一，本文讨论了TOE技术的实现方法以及不同方法的优缺点并分析了其适用场景，思考了接下来的研究方向。随着网络的不断发展以及技术的不断进步，数据安全不断遭到威胁，提高卸载的安全性刻不容缓。因此，可以通过将TOE技术与加密技术进行结合、在NIC上附加Firewall（防火墙）、IDS（入侵检测）等安全性功能来安全高效的进行数据传输。除此之外，在我目前参加的项目拟态调度器IP设计中，利用TOE实现TCP协议卸载等功能缓解了CPU的压力，拟态调度模块拟可以通过拟态调度的方法动态的选择异构体，为攻击者的攻击环节增加难度，提高了系统的安全性，因此，将拟态与TOE技术结合也是接下来的一个重要研究方向之一。

参考文献:

[1] 杭彦希,徐金甫,南龙梅,等. 基于PCIE接口的IPSec加速SoC设计[J]. 计算机工程与设计,2017,38(5):1212-1215,1257.
Hang Yanxi, Xu Jinpu, LongmeiNan, et al. IPSec accelerated SoC design based on PCIE interface[J]. Computer Engineering and Design, 2017,38(5):1212-1215,1257.

[2] 刘天华, 朱宏峰, 谭振华. 一种新的TOE体系结构的研究与原型实现[J]. 计算机工程，2007，33(10)：111-113
Liu Tianhua, Zhu Hongfeng, Tan Zhenhua. Research and prototype realization of a new TOE architecture[J]. computer engineering , 2007，33(10)：111-113

[3] 孙连超. 基于SoC的千兆以太网TCP_IP卸载引擎的研究与验证[D]. 西安电子科技大学,2018.
Sun Lianchao. Research and Verification of SoC-based Gigabit Ethernet TCP/IP Offload Engine[D]. Xidian University，2018.

[4] Feng W, Balaji P, Baron C, et al. Performance characterization of a 10-Gigabit Ethernet TOE [C]. // High-Performance Interconnects Proceedings 13th Symposium. Palo Alto：IEEE，2005：58-63.

[5] 朱宏峰, 刘天华, 刘杰,等. TCP/IP协议卸载技术性能与实现的研究[J]. 小型微型计算机系统，2007，28(004)：609-614.
Zhu Hongfeng, Liu Tianhua, Liu Jie, et al. Research on Performance and Implementation of TCP/IP Protocol Offloading Technolog[J]. Small microcomputer system，2007，28(004)：609-614.

[6] 李杰, 陈曙晖. 基于多核NPU的TCP数据接收卸载[J]. 计算机工程与科学，2016，38(7)：1344-1349.
Li Jie, Chenshujun. TCP data receiving and offloading based on multi-core NPU[J]. Computer Engineering and Science，2016，38(7)：1344-1349

[7] 李成龙,李韬,韩玉浩,等. DNS权威服务器FPGA加速技术研究[J]. 中国科学:信息科学，2020，50(04)：576-587.
Li Chenglong, Li Tao, Han Yuhao,et al. Research on FPGA Acceleration Technology of DNS Authoritative Server[J]. Science in China: Information Science，2020，50(04)：576-587.

[8] 薛镭,贺亚龙. 基于Offload和FPGA的网络传输设计与实现[J]. 机电设备，2018，35(4)：54-57,65.
Xue Lei, He Yalong. Design and Implementation of Network Transmission Based on Offload and FPGA[J]. Electrical Equipment，2018，35(4)：54-57,65.

[9] 张志宏, 吴庆波, 邵立松, et al. 基于飞腾平台TOE协议栈的设计与实现[J]. 计算机技术与发展，2014，000(007)：1-4.
Zhang Zhihong, Wu Qingbo, Shao Lisong, et al. Design and implementation of TOE protocol stack based on Feiteng platform[J]. Computer Technology and Development，2014，000(007)：1-4.

[10] Tianhua L, Hongfeng Z , Chuansheng Z, et al. Research and Prototype [C]// International Conference on Information & Communication Technologies. IEEE，2006：15-19

[11] 刘宇芳. 多核处理器平台资源管理的若干问题研究[J]. 计算机科学，2012，39(S1)：441-443.
Liu Yufang. Research on Several Issues of Multi-core Processor Platform Resource Management[J]. computer science，2012，39(S1)：441-443.

[12] Yoon I S, Chung S H , Kwon Y G. Implementation of a Software-Based TCP/IP Offload Engine Using Standalone TCP/IP without an Embedded OS [J]. Journal of Information ence & Engineering，2011，27(6)：1871-1883.

[13] Jang J, Jung J, Cho Y K, et al. Design of a Lightweight TCP/IP Protocol Stack with an Event-Driven Scheduler [J]. Journal of Information ence & Engineering，2012，28(6)：1059-1071.

[14] Hoskote Y, Bloechel B. A, Dermer G E,
et al. A TCP offload accelerator for 10 Gb/s Ethernet in 90-nm CMOS [J]. Solid-State Circuits，IEEE Journal of，2013，38(11)：1866-1875.

[15] Ding L, Kang P, Yin W, et al. Design and implementation of hardware-based low latency TCP offload engine for 10 Gbps Ethernet [C]// IEEE International Conference on Solid-state & Integrated Circuit Technology. IEEE，2016：287-299.

[16] Wu Huo, Liu Yiqing. FPGA-based TCP/IP protocol processing architecture for 10 Gigabit Ethernet [J]. Electronic Design Engineering，2020，28(9)：81-87 (in Chinese)
吴 惑,刘一清. 基于FPGA的万兆以太网TCP/IP协议处理架构[J]. 电子设计工程，2020，28(9)：81-87

[17] Mannem, Venugopal, Sommers, et al. Increasing Network Performance With A TCP/IP Offload ASIC. [J]. Computer Technology Review，2002，004(009)：485-493

[18] 夏杨. 基于FPGA的万兆以太网数据分发平台设计[D]. 北京理工大学,2016.
Xia Yang. Design of 10 Gigabit Ethernet Data Distribution Platform

Based on FPGA[D]. Beijing Institute of Technology，2016.

[19] Jang H K, Chung S H , Kim D K , et al. An Efficient Architecture for a TCP Offload Engine Based on Hardware/Software Co-design[J]. Journal of Information ence & Engineering，2011，27(2)：493-509.

[20] Yang X, Yueyang C. 10 Gigabit Ethernet TCP Frame Data Registration Algorithm Based on FPGA [C]// International Conference on Mechanical. 2016：224-243

[21] Li H, Zhang J, Gu J, et al. A readout method based on 10 Gigabit Ethernet for Sipixel detector[J]. 2018，24(5)：368-379

[22] 程红军, 陈洪, 张激. 应用于iSCSI的TOE技术分析[J]. 计算机工程，2004，30(009)：126-128.
Cheng Hongjun, Chen Hong, Zhang Ji. Analysis of TOE Technology Applied in iSCSI[J]. computer engineering，2004，30(009)：126-128 .

[23] Zhong Xiaodong, Chen Lian, Li Yinjie, Jin Ge. Design and verification of TCP/IP offload engine in quantum key distribution system.[J]. The Review of scientific instruments，2019，90(11)：1-10

[作者简介]

任芸莉（1996-），女，信息工程大学硕士研究生，主要研究方向为SoC芯片设计与验证。

刘冬培（1985-），男，博士，信息工程大学信息技术研究所助理研究员，主要研究方向为SoC芯片测试与验证。

韩国栋（1964-）男，博士，信息工程大学信息技术研究所研究员，主要研究方向为网络安全，信号处理，宽带网络和深度学习。

面向进程多变体软件系统的攻击面定性建模分析

邢福康，张铮，李秉政，曲晟，季新生
信息工程大学，郑州 450001

摘　要：在当前信息安全威胁日益严重的背景下，如何保护软件系统的安全性已经引起了广泛的关注。多变体执行技术能够通过对异构、冗余的执行体进行系统出口点的表决增加软件系统的安全性。在系统安全性度量方法中，攻击面度量是一种重要指标，传统的攻击面模型未考虑到多变体执行技术能够动态改变攻击面的特性，无法准确地衡量采用了多变体架构的多变体系统的攻击面。因此我们在传统的攻击面模型基础上增加了对于系统出口点表决部分，以使其能更加准确地衡量采用了多变体架构的软件系统安全性。本文还进行了实例分析，验证了多变体系统相较传统软件系统具有攻击面动态缩小的优点，能够更好地防御攻击。

关键词：多变体、攻击面、攻击面度量

Qualitative Modeling and Analysis of Attack Surface for Process Multi-Variant Software System

XING Fukang, ZHANG Zheng, LI Bingzheng, QU Sheng, JI Xinsheng
Information Engineering University, Zhengzhou 450001, China

Abstract: Under the background of information security threat becoming more and more serious, how to protect the security of software system has attracted extensive attention. Multi variant execution technology can increase the security of software system by voting the exit point of heterogeneous and redundant executors. In the system security measurement methods, attack surface measurement is an important indicator. The traditional attack surface model does not consider that the multi variant execution technology can dynamically change the characteristics of the attack surface, so it can not accurately measure the attack surface of the multi variant system with multi variant architecture. Therefore, based on the traditional attack surface model, we add the system exit point voting part to make it more accurate to measure the security of software system with multi variant architecture. In addition, the system has the advantages of multi aspect defense compared with the traditional system.

Key words: Multi-Variant Execution; attack surface; attack surface metric

1　引言

随着信息技术的高速发展，越来越多的软件系统出现在我们的身边，如今软件系统服务于社会的各个层面，为我们的日常生活工作提供了极大的便利。然而，随着软件系统深入我们的生活，软件系统的功能和保存的数据信息也越来越重要，针对软件系统进行的攻击也日益增多。早在1988年，Morries就利用软件系统的缓冲区溢出漏洞破坏了大量的计算机[1]。近年来，随着软件系统的发展，软件漏洞也层出不穷，如Bletsch等[2]提出了面向跳转的攻击（JOP），Levy等[3]提出了代码注入攻击，而目前最常见、最主流的攻击方式为面向返回的攻击（ROP），由Shacham等人于2007年提出[4]。

为了应对这些层出不穷的攻击行为，有很多相应的防御措施已被提出。这些防御措施包括运行前静态防御以及运行时的动态防御。如Stack-

基金项目：国家自然科学基金资助项目（No. 61521003），国家重点研发计划资助项目（Grant No. 2018YF0804003，Grant No. 2017YFB0803204）。

通讯作者：张铮（ponyzhang@163.com）

Guard [5] 通过静态地在gcc编译时增加随机化机制防止攻击。COWAN等[6]通过在编译时替换易受攻击的库进行软件的静态防护。但是由于Andrea Bittau等提出的BROP攻击[7]可以动态地观测软件的运行过程从而进行控制流劫持，静态防御措施的作用越来越小。在动态防御的措施中，控制流完整性技术（CFI）[8]通过检测程序的执行流程是否被篡改来防御攻击，然而由于其性能较差，目前阶段较难部署于实际生产用软件系统。多变体执行技术（MVX）最早由Cox[9]提出，其根据两个功能相同进程在内存分布上的随机化以及监控器保证软件运行的安全性。在此基础上，Salamat[10]提出了监控器独立运行的多变体架构，而Koning等[11]提出了MvArmor多变体系统架构，该架构基于对系统调用进行表决，采用该架构的用户可以在安全性和性能两个维度上进行选择，以实现在保护软件安全性的前提下保证其性能。

攻击面是衡量软件系统安全性的一个重要指标，采用攻击面描述可以通过集合的方式描述软件系统的安全性并对其进行度量。一般的攻击面模型基于I/O自动机模型对软件系统进行建模，其一般采用非冗余的架构，难以应用于类似多变体系统这类异构冗余的系统架构。文献［12］提出了一种在非相似余度系统中进行攻击面度量的方式，但是其采用的系统架构表决粒度和表决方式与多变体系统不同，无法准确度量多变体系统的攻击面。因此，本文在传统攻击面模型基础上，提出一种能够描述多变体系统攻击面的攻击面建模方式，说明采用多变体架构的多变体系统在运行过程中攻击面的变化，并通过实际的攻击展示多变体系统攻击面的变化。

2 相关工作

传统攻击面模型由Manadhata等[13]提出，在传统攻击面模型中，攻击面被定义为可被攻击者利用的系统资源的总和。Manadhata将系统攻击面的度量划分为三个维度，即系统的方法、系统的通道以及系统的不可信数据项。其中系统的方法包含系统入口点和系统出口点，系统的入口点即是系统从环境中接收输入数据的方法，系统的出口点是系统向其环境进行输出和发送数据的方法。在一般的软件系统中，操作系统即为其环境，软件系统向操作系统进行输出或数据发送一般需要调用系统调用，这些系统调用由操作系统提供给运行于操作系统上的软件系统使用。分别使用M，C，I表示一套软件系统S的方法、通道和不可信数据项，则该软件系统的攻击面可有以下三元组表示

$$surf_S = \langle M, C, I \rangle \#(1)$$

此处将系统的方法部分进一步细分为系统入口点M_{in}和系统的出口点M_{out}，则其攻击面可有以下四元组表示

$$surf_S = \langle M_{in}, M_{out}, C, I \rangle \#(2)$$

文献［14］根据采用特殊架构时系统攻击面会产生转移的特点提出了攻击面转移。文献［12］提出攻击面模型的构建是攻击面理论研究的重点内容，是准确评估系统安全性的关键，并提出了一种对如下图1非相似余度信息系统进行建模的方法。

图1中为一个典型的非相似余度架构图，包括输入模块，进行输入分发的输入代理模块，虚线框中功能相同的异构执行体部分，进行结果裁决的裁决模块以及输出模块。假设有一非相似余度系统s，其由s_1，s_2，$s_3 \cdots s_n$等异构执行体组成，若按照传统攻击面的建模方式进行建模，则系统攻击面R_S为

$$R_s = R_{S_1} \cup R_{S_2} \cup \cdots \cup R_{S_n}, n \in N^* \#(3)$$

其中的R_{si}为异构执行体s_i的攻击面，由此可见采用非相似余度架构的系统攻击面反而变大，与实际情况不相符，因此作者提出了一种加入了裁决机制的新攻击面建模方式，将裁决机制抽象为多个执行体攻击面取交集的过程，即引入裁决机制后系统s的攻击面如下

$$R_s = R_{S_1} \cap R_{S_2} \cap \cdots \cap R_{S_n}, n \in N^* \#(4)$$

可以看出，改进后的模型很好地表现了进行攻击面裁决后，系统攻击面显著减小，与事实相符合。然而进程多变体架构与传统的相似余度架构在裁决方式上有所不同，不能使用文献［14］中提出的非相似余度系统的建模方式。

3 攻击面模型

3.1 传统攻击面模型

对软件系统进行攻击面建模是度量软件系统安全性的重要步骤，在文献［13］中已经给出了

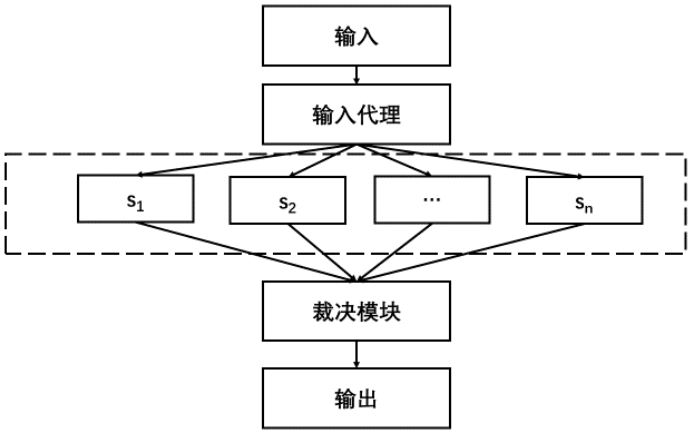


图1　非相似余度架构图

传统攻击面建模方式如下

对于系统集S_i，其其中一个系统s_i的环境E_{si}为一个三元组

$$E_{si}=\left\langle U_i,D_i,T_i\right\rangle \#(5)$$

其中U_i为攻击者集合，T_i为不含s_i的系统集合，D_i为数据集集合，如下图1所示

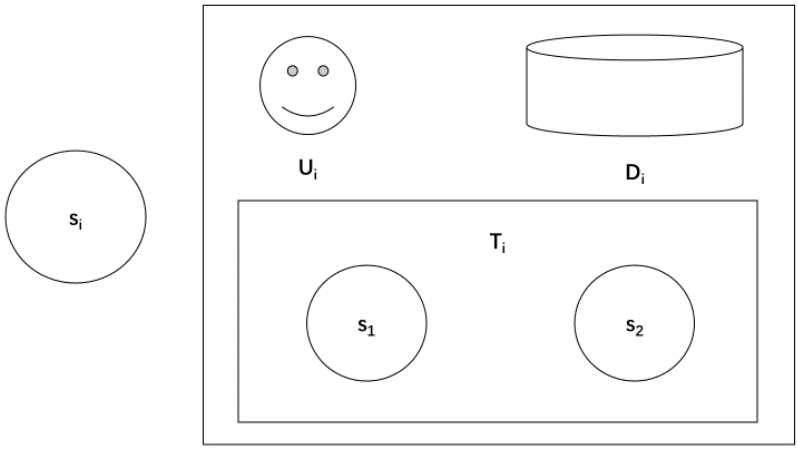


图2　系统环境图

则系统s_i的攻击面可表示为

$$surf_{s_i}=\left\langle M_{in}^{E_{si}},M_{out}^{E_{si}},C^{E_{si}},I^{E_{si}}\right\rangle \#(6)$$

定义系统s_i的攻击面资源集合R_{si}为

$$R_{si}=M_{in}^{E_{si}}\cup M_{out}^{E_{si}}\cup C^{E_{si}}\cup I^{E_{si}} \#(7)$$

3.2　多变体系统攻击面模型

本节主要工作是构建多变体系统的攻击面模型，先是对于多变体系统的介绍与分析，然后对传统攻击面模型进行扩展。本节使用形式化方式表示多变体系统的攻击面，又根据多变体系统在系统出口点处的表决机制对传统攻击面模型进行改进，以使其能解释多变体系统攻击面缩小的情况。

多变体系统的设计主要依据了异构冗余技术，其模型如图2所示

多变体系统的核心模块主要包括输入模块，多变体模块和监控表决模块。各模块的作用如下

输入模块：将系统输入进行分发，交给各个多变体进程进行执行。

多变体模块：含有多个功能等价的进程，进程均采取ASLR、PIE等地址随机化技术，使得各个进程的内存分布不相同。

监控表决模块：监控每个进程的执行，在进

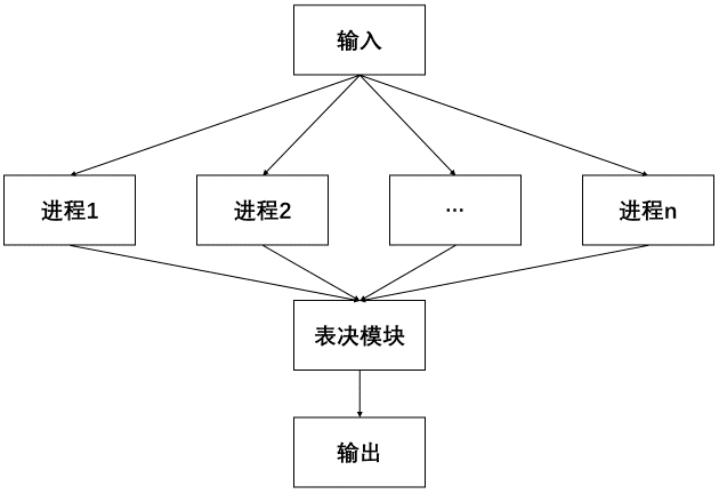


图3　多变体系统架构模型图

程需要调用系统调用时进行拦截，比较各个多变体进程的内存内容和系统调用是否一致，若一致则继续进行执行，否则产生错误，结束多变体系统的执行。

在多变体系统中，进程多变体的异构性体现了异构冗余的思想，由于多个进程多变体之间采用了地址随机化技术，攻击者难以同时采用缓冲区溢出攻击对所有进程多变体进行控制流劫持，因此当攻击者通过控制流劫持某一进程多变体进行恶意系统调用时，其他未被攻击成功的进程多变体仍维持正常的系统调用，表决模块拦截到各个进程多变体的系统调用并进行对比，从而阻止攻击。在多变体系统中存在一致表决、近似表决等多种表决策略，在本文中，多变体系统表决策略采取一致性表决，即所有多变体进程的系统调用一致时才表决通过，否则表决不通过。

4　多变体系统攻击面建模形式化表示

根据多变体系统特点，定义系统集S，多变体进程功能相同，且均能够独立完成系统功能，假设系统集S中存在n个等价的多变体进程，则记多变体进程i为s_i，$s_i \in S$，i<=n。由于各个多变体进程独立存在，不存在相互通信，故可视为多个独立的攻击目标，当系统集S受到攻击时，各个多变体进程同时成为被攻击的目标。若假设每个多变体进程的攻击面为R_{si}，则在进行表决之前，多变体系统的攻击面为R_s。根据传统攻击面理论，由于各个多变体进程功能相同且共用系统资源，此时多变体系统的攻击面R_s相较于每一个多变体进程的攻击面R_{si}并没有显著变化，这与多变体系统实际上提高系统安全性的结论相悖，产生这种结果是因为传统攻击面理论中并没有针对多变体系统的架构引入系统出口表决机制。

传统攻击面理论中并未引入异构冗余的概念，而多变体系统安全性的提升主要来自于其采取了异构冗余的架构，因此其于系统出口点可以进行一致性表决。因此，我们在传统攻击面理论的基础上引入系统出口表决机制，一些定义如下。

定义1

给定环境E = <U，D，T>，存在系统x和y，其攻击面分别为R_x和R_y，若系统x与y在方法、通道、数据三个方面完全一致，则称x与y同构，两者攻击面相同，记为$|R_x|=|R_y|$。若系统x与系统y在方法、通道、数据三个方面的任何一方面不一致，则称x与y不同构，两者攻击面不相同，若$R_x \subseteq R_y$，则称$|R_y|>|R_x|$，系统y的攻击面比系统x的攻击面大。

定义2

给定一个进程s，在其出口点存在对于操作系统方法的调用，即系统调用（syscall）。若设在操作系统中系统调用共有n种，则将操作系统调用抽象为n个两两垂直的单位向量$\alpha_{s1}\cdots\alpha_{sn}$。当两个系统调用不相同时，代表其的单位向量互相垂直，两者点积结果为0。在进程s执行过程中使用的系统调用向量按照顺序组成一个集合T_s，称为系统s的系统调用向量集合T_s。

定义3

给定一个多变体系统S中存在两个进程s_1，s_2，其系统调用向量集合分别为T_{s1}和T_{s2}，集合中元素个数分别为n_1和n_2，对Ts1和Ts2中对应位置的向量进行点积运算，记第一组运算结果不为零的向量之前的向量为集合A

$$A=\left\{\beta|\beta\in\left\{\alpha_1\cdots\alpha_k\right\},1\leqslant k\leqslant\min\left(n1,n2\right),\alpha_{k+1}^{T_{s1}}\cdot\alpha_{k+1}^{T_{s2}}=0\right\}\#(8)$$

在多变体系统中，如果进程s1和s2均未受到攻击，则应该拥有相同的系统调用，即T_{s1}与T_{s2}相同。记两者不同的系统调用集合为K则

$$K=\left(T_{s1}-A\right)\cup\left(T_{s2}-A\right)\#(9)$$

故该系统S出口点经过表决之后的攻击面集合实际为

$$M_{out\ 表决后}^{S}=M_{out\ 表决前}^{S}-K\#(10)$$

则系统S的攻击面可由下式进行描述

$$R_S=\left\langle M_{in}^{S},M_{out\ 表决前}^{S}-K,C^S,I^S\right\rangle\#(11)$$

可见经过系统出口点表决后多变体系统的攻击面实际上缩小了。

5 实例分析

采用了两组多变体执行架构的软件系统进行实例分析，通过分别与未采用多变体架构的功能相同的软件系统进行攻击面的对比分析，体现多变体系统在攻击面上的变化。

实例采用的两种软件系统，其中均使用了危险函数gets（），攻击者能够通过缓冲区溢出漏洞对程序的控制流进行篡改，从而威胁系统的安全。

对正常运行时的多变体系统和普通系统进行建模比较，由于未受到攻击，多变体系统的各个进程在裁决时系统调用序列相同，故此时多变体系统和普通系统的攻击面保持一致，可由下式进行表示

$$K=\left(T_{s1}-A\right)\cup\left(T_{s2}-A\right)=\varnothing\#(12)$$

$$M_{out\ 表决后}^{S多变体}=M_{out\ 表决前}^{S多变体}-\varnothing=M_{out\ 表决前}^{S多变体}\#(13)$$

$$R_{S多变体}=\left\langle M_{in}^{S多变体},M_{out\ 表决前}^{S多变体},C^{S多变体},I^{S多变体}\right\rangle=R_{S普通}\#(14)$$

在正常运行时，多变体系统和普通系统产生的系统调用表如下表所示

可以看出，当未受到攻击时，多变体系统和普通软件系统的系统调用一致，两者在系统出口点的攻击面相同，这与前面的建模分析结果一致。

表1 第一组软件系统未受攻击的系统调用表

系统调用序号	普通系统	多变体系统进程1	多变体系统进程2
1	Brk	Brk	Brk
2	Mmap	Mmap	Mmap
3	Access	Access	Access
4	Open	Open	Open
5	Fstat	Fstat	Fstat
6	Mmap	Mmap	Mmap
7	Close	Close	Close
8	Open	Open	Open
9	Read	Read	Read
10	Fstat	Fstat	Fstat
11	Mmap	Mmap	Mmap
12	Mprotect	Mprotect	Mprotect
13	Write	Write	Write
14	exit	exit	exit

表2 第二组软件系统未受攻击的系统调用表

系统调用序号	普通系统	多变体系统进程1	多变体系统进程2
1	Brk	Brk	Brk
2	Mmap	Mmap	Mmap
3	Access	Access	Access
4	Open	Open	Open
5	Fstat	Fstat	Fstat
6	Mmap	Mmap	Mmap
7	Close	Close	Close
8	Open	Open	Open
9	Fstat	Fstat	Fstat
10	Mmap	Mmap	Mmap
11	Close	Close	Close
12	Open	Open	Open
13	Read	Read	Read
14	Fstat	Fstat	Fstat
15	Mmap	Mmap	Mmap
16	Mprotect	Mprotect	Mprotect
17	Write	Write	Write
18	exit	exit	exit

当软件系统受到攻击时，普通软件系统会遭到控制流劫持，而多变体系统由于系统出口点表决的存在，会中断软件的运行，防止攻击者的攻击行为。在这种情况下，多变体系统会产生不同的系统调用，导致裁决生效，使得采用多变体架

构的软件系统攻击面变小，对多变体系统和普通系统进行建模分析对比如下式

$$K = \left(T_{s1} - A\right) \cup \left(T_{s2} - A\right) \neq \varnothing \#(15)$$

$$M_{out}^{S多变体}{}_{表决后} = M_{out}^{S多变体}{}_{表决前} - K < M_{out}^{S多变体}{}_{表决前} \#(16)$$

$$R_{S多变体} = \left\langle M_{in}^{S多变体}, M_{out}^{S多变体}{}_{表决后}, C^{S多变体}, I^{S多变体} \right\rangle < R_{S普通} \#(17)$$

当软件系统受到攻击时，两者的系统调用表如下表所示

表3　第一组软件系统受到攻击后的系统调用表

系统调用序号	普通系统	多变体系统进程1	多变体系统进程2
1	Brk	Brk	Brk
2	Mmap	Mmap	Mmap
3	Access	Access	Access
4	Open	Open	Open
5	Fstat	Fstat	Fstat
6	Mmap	Mmap	Mmap
7	Close	Close	Close
8	Open	Open	Open
9	Read		
10	Write		
11	Mmap		
12	Write		
13	Write		
…	fstat		

分析两组实例可知，攻击者进行攻击后，普通软件系统由于没有系统出口点的表决，在read系统调用处被攻击，攻击者之后可以进行控制流劫持操作。而多变体系统在read系统调用处的表决结果不一致，从而强制中断程序的运行，避免了攻击者的劫持。多变体系统由于受到攻击，只执行了部分系统调用，系统的攻击面在遭受攻击时减小了，这与前面的建模分析结果一致。

6　结束语

本文在传统攻击面模型的基础上提出了多变体系统的攻击面模型，形式化描述了在多变体系统中攻击面的变化方式。阐明了多变体系统在系统出口点的表决方式，解释了多变体系统受到攻击时系统攻击面变小的原理，并进行了实例验证分析。但是本文的研究仍不充分，接下来会继续在以下几个方面进行深入研究。

表4　第二组软件系统受攻击的系统调用表

系统调用序号	普通系统	多变体系统进程1	多变体系统进程2
1	Brk	Brk	Brk
2	Mmap	Mmap	Mmap
3	Access	Access	Access
4	Open	Open	Open
5	Fstat	Fstat	Fstat
6	Mmap	Mmap	Mmap
7	Close	Close	Close
8	Open	Open	Open
9	Fstat	Fstat	Fstat
10	Mmap	Mmap	Mmap
11	Close	Close	Close
12	Open	Open	Open
13	Read		
14	Write		
15	Mmap		
16	Write		
17	Write		
…	fstat		

对于多变体系统攻击面变化的量化研究。

对于不同架构的多变体系统进行建模分析。

对于采取不同表决方式多变体系统的攻击面分析。

参考文献:

[1] Hao S. S., Qing G., Sen M. A., et al. Progress in Research on Buffer Overflow Vulnerability Analysis Technologies[J]. Journal of Software，2018.

[2] BLETSCH T., JIANG X., FREEH V. W., et al. Jump-oriented programming: a new class of code-reuse attack[C]//The 6th ACM Symposium on Information, Computer and Communications Security. 2011：30-40.

[3] Levy E. Smashing the Stack for Fun and Profit[J]. Phrack Magazine，1996，8(49)：1-25.

[4] BletschT., JiangX., FreehV.. Mitigating code-reuse attacks with control-flow locking [C]. The 27th Annual Computer Security Applications Conference (ACSAC'11)，2011：353-362.

[5] Cowan C., Pu C., Maier D., et al. StackGuard : Automatic Adaptive Detection and Prevention of Buffer-Overflow Attacks[C]// Proc. 7th USENIX Security Conference，1998，98：63-78.

[6] COWAN C., BARRINGER M., BEATTIE S., et al. FormatGuard: automatic protection from printf format string vulnerabilities[C]// USENIX Security Symposium. 2001，91.

[7] Bittau A., Belay A., Mashtizadeh A., et al. Hacking Blind[C]// 2014 IEEE Symposium on Security and Privacy (SP). IEEE，2014.

[8] Wang Z. , Jiang X. . HyperSafe: A Lightweight Approach to Provide Lifetime Hypervisor Control-Flow Integrity[C]// Security & Privacy. IEEE，2010.

[9] CoxB. , EvansD. , FilipiA. , et al. N-Variant Systems: A Secretless Framework for Security through Diversity[C]. USENIX Security Symposium (SEC'06)，2006：105-120.

[10] Salamat B. , Jackson T. , Gal A. , et al. Orchestra: Intrusion Detection Using Parallel Execution and Monitoring of Program Variants in User-space[C]. Proceedings of the fourth ACM european confer-ence on Computer systems - EuroSys '09，2009：33-49.

[11] Koning K. , Bos H. , Giuffrida C. . Secure and Efficient Multi-Variant Execution Using Hardware-Assisted Process Virtualization [C]// Proc. of the Int'l Conf. on Dependable Systems and Networks. IEEE，2016.

[12] 张铮, 王立群, 李卫超. 面向非相似余度信息系统的攻击面模型[J]. 通信学报，2018，39(S2)：227-234.
ZHANG Z. , WANG L. Q. , LI W. C. . Research on formal model for an information system's attack surface with dissimilar redundant architecture [J]. Journal of Communications，2018，39（S2）：227-234.

[13] Manadhata P K. Game theoretic approaches to attack surface shifting [J]. Moving Target Defense II. Springer New York，2013：1-13.

[14] Manadhata P K . Game Theoretic Approaches to Attack Surface Shifting[J]. 2013.

[作者简介]

邢福康（1997—），男，学士，博士生，主要研究方向为网络空间安全，Web应用安全。

张铮（1976—），性别，博士，副教授，主要研究方向为网络空间安全，主动防御技术。

李秉政（1996—），男，学士，博士生，主要研究方向为网络空间安全，主动防御技术。

曲晟（1996—），男，学士，博士生，主要研究方向为网络空间安全，主动防御技术。

季新生（1968—）男，博士，教授，主要研究方向为网络空间安全，无线通信。

Sound Classification Method Based on the Novel Direct Convolutional Neural Network Model

WANG Gengrun, LI Haitao, ZHU Yuhang

Information Engineering University, 450002, China

Key words: sound classification; convolutional neural network; false alarm rate; MFCC feature; recording sounds

Abstract. It is widely used of the automatic sound classification method in the applied engineering， and the automatic sound classification method has made great progress with the development of the Convolutional Neural Network （CNN） model. In this paper， a new direct-CNN model was proposed to utilize the self-learning of CNN to classify the artificial voice from recording conversations. The proposed model had the advantage of simple architecture with comparative classification accuracy compared with the state-of-art Two-Stream CNN model in common datasets. When applied to the real recording conversation dataset， performance of the proposed model was proved to be satisfactory.

1 Introduction

Nowadays more and more recording sounds are stored in bigdata clusters. It includes video surveillance sounds， telephone recording sounds， sounds recorded by Internet of Things （IoT） devices and so on. How to classify these sounds efficiently to satisfy different demands has drawn much attention. One of the popular methods is to apply artificial intelligent methods to solve this problem. Chachada et al. gave a survey of Environmental Sound Recognition （ESC） ［1］. Karol utilized neural network to the ESC problem and achieved dramatic classify accuracy ［2］. Li et al. proposed a new CNN model and took advantage of both Mel Frequency Cepstral Coeffi cient （MFCC） features and raw data to achieve a higher accuracy rate ［3］. One of the state-of-the-art research was completed by Yu et al.， which utilized two combined features to give a more comprehensive representation of environment sounds ［4］. Considering high accuracy rate achieved by recent research， these methods must pay the price of computation complexity and training time， which were very valuable in practical applications such as handhold devices and IoT devices.

Neural network was first proposed by Lecun in the last century ［5］. Limited by its complexity and low computing power in those days， neural network was not applied to solve practical problems for a long time. With the development of computing power， especially which of CPU and GPU grew exponentially according to Moore’s law， large-scale neural network was feasible for more and more applications. Besides CNN models for solving identification problems of 2-D static data （such as pictures）， Recurrent Neural Network （RNN） was proposed to dispose the sequential data which utilized former and following input to predict current data ［6］. Besides， Long Short-Term Memory （LSTM） was an advanced popular network which overcome the shortage

of long dependence problem of RNN [7].

Dealing SEC problems with neural network has achieved much progress at present [1-4, 8]. However, there are some reality problems in utilizing those complicate methods. On the one hand, many state-of-the-art methods achieved high accuracy at the cost of great amount of labelled training data, which was unreachable in many application scenarios; on the other hand, many application devices were computing power limited, which needed simple network architectures. In this paper, a simple direct-CNN model was proposed to overcome these problems. It achieved comparable accuracy rate in common datasets compared with existing methods. Besides, it was easy to be utilized in many computing limited devices. When applied this model to the classification problem of the real recording conversation dataset, its effect was proved satisfactory.

The following part of this paper was organized as follows: in part 2, the architecture of the proposed model was described in detail. Then in part 3, the experiment results were present, under two common open datasets and one real-world dataset. Finally, conclusions were drawn in part 4.

2 Model

2.1 Input features

There are some features to describe the characteristic of sounds, such as Log-Mel spectrogram (LM), MFCC, chroma, spectral contrast, tonnetz and so on. In recent research, LM and MFCC were two common features which were derived from the Mel filters. In the work of Yu et al. [4], they obtained a good result with hybrid utilizing of different features. However, their model increased the burden of computation. In this paper, the MFCC feature was used to reduce the complexity of the model. The MFCC feature is a frequency domain parameter and has been adjusted for the hearing characteristics of human. It changes the 1-D raw sound data to 2-D data by Mel-filters, which could be used as the input of CNN models directly. As a result, it is very suitable for the training of the SEC models. Fig. 1 showed MFCC features (together with first and second derivatives) of a 4-seconds sound file which was taken from the UrbanSound8k dataset.

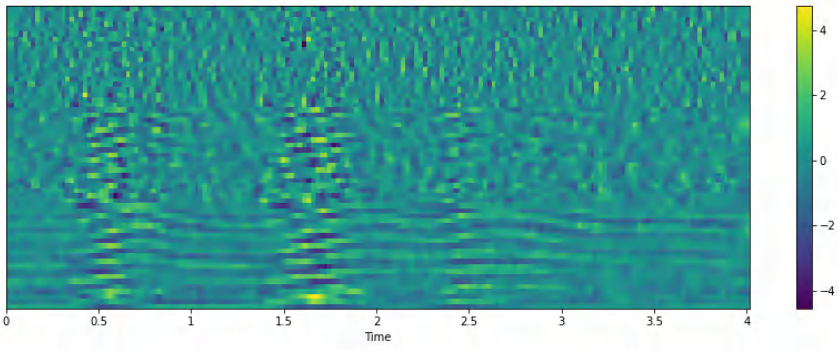


Fig. 1 MFCC features of a 4s sound file

2.2 Architecture

In this paper, a novel direct-CNN model was proposed. Its advantage was connecting the input data directly to the following convolutional layers, which to extract the information of original input data thoroughly. Its main architecture was demonstrated as Fig. 2.

First, the MFCC features together with first and second order derivatives were calculated, and then the MFCC features were divided into fixed length fragments and forming the training dataset with their labels. Second, the above training data was input to the direct-CNN. In this model, five convolutional layers were adopted with the activation of Rectified Linear Unit (ReLU). The innovation of this model was that the input layer was also added to

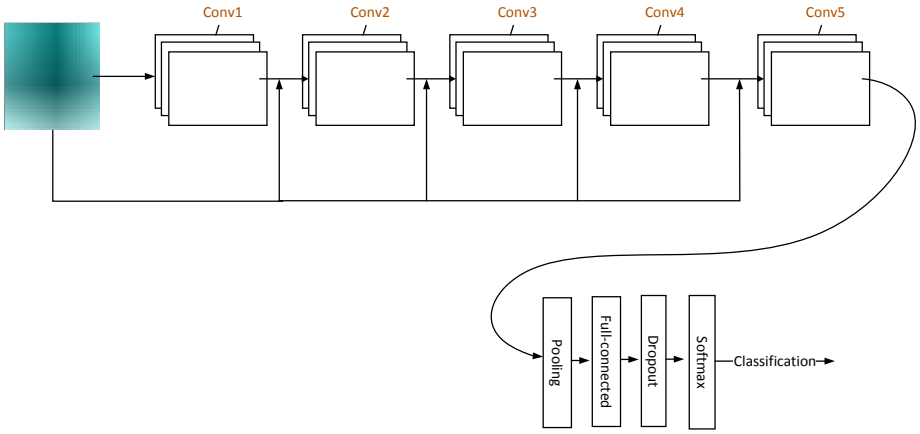


Fig. 2 The architecture of the direct-CNN

the input of all the following convolutional layers. Therefore, the original input data could enhance the feature extraction ability of every convolutional layers. Third, the convolutional layers were followed by a pooling layer, a full-connect layer, a dropout layer, and a softmax-layer, then to give the prediction. When these configurations were completed, this model could be trained. In the direct-CNN, the categorical_crossentropy loss function was used, and the optimizer was Adam.

2.3 Reference standards

Accuracy rate is commonly used to evaluate the performance of a new model. In this paper, two aspect accuracy rates were considered. One was the accuracy for the whole test dataset, which was the baseline of the performance of the proposed model. It indicated the rate of samples which were classified correctly. The other was the error rate of one particular class of the dataset, which was also known as the probability of false alarm. The second performance indicator was used in some special application scenarios, which was strictly require a lower error prediction rate of one class while not particularly depended on the classifications accuracy of other classes.

In binary classification, the first accuracy rate was calculated as follows,

$$P_{acc} = \frac{N_{tp} + N_{tn}}{N_{tp} + N_{tn} + N_{fp} + N_{fn}} \tag{1}$$

where N_{tp} was the number of positive samples which were classified correctly, N_{tn} was the number of negative samples which were classified correctly, N_{fp} was the number of negative samples which were classified as positive, N_{fn} was the number of positive samples which were classified as negative.

And the false alarm rate was calculated as:

$$P_{fa} = \frac{N_{fp}}{N_{tn} + N_{fp}} \tag{2}$$

The recall rate was calculated as:

$$P_{rec} = \frac{N_{tp}}{N_{tp} + N_{fn}} \tag{3}$$

3 Experiments

To evaluate the performance of the proposed method, two open datasets and a real-world dataset were adopted in the experiments: the TIMIT dataset and the UrbanSound8k dataset. The TIMIT dataset was provided by Massachusetts Institute of Technology (MIT), Stanford Research Institute (SRI) and Texas Instruments (TI). It contained a total of 6300 sentences (10 sentences spoken by 630 speakers from 8 major dialect regions of the United States) [9]. The UrbanSound8k was provided by Justin et al. It contained 27 hours of audio with 18.5 hours of annotated sound event occurrences across 10 sound classes and was the widely used open dataset for environmental sound classi fi cation as a real marked data [10]. Besides these two datasets, the proposed method was also evaluated on a real-world telephone recording sound dataset. This dataset was collected by our team. Under the authorization of tele-

phone users， this dataset was retrieved from recorded telephone calls of daily life. It included 8355 conversations. Furthermore， these conversations were marked as Real Calls （RCs， 4745 conversations）， calls initiated by Machine Recording （MRCs， 2506 conversations）， and Mute Calls （MCs， 1104 conversations）. From this dataset， the MRCs were wanted to be found automatically， which were always advertisement calls and junk calls. Besides， for this dataset the false alarm rate is very important for practical applications. That is because most telephone users do not want their RCs to be classified as MRCs， which would be filtered by the following applications. In the experiment on this dataset， the models were retrained， and the false alarm rates were tested particularly.

3.1 UrbanSound8k

As for the UrbanSound8k dataset， Folder 1-8 were used for training and Folder 10 was used for testing. The batch size was set as 32. The experiment results were presented in Fig. 3. Three models were evaluated： traditional CNN （marked as CNN） [2]， Two-Stream CNN （marked as TSCNN） [4] and this new model （marked as direct-CNN）. The traditional CNN was compared as a baseline model， and the TSCNN was the state-of-the-art model which was proved to have gained excellent accuracy on the UrbanSound8k dataset [4]. In the result， both TSCNN and our model achieved comparable accuracy rate under the same setting （TSCNN-83.1%， ours-81.0%）. Besides， these two models exhibited great progress over the traditional CNN （71%）. In Tab. 2， the parameter numbers and the time consumption over each step of these models were presented. It can be found that our model had the advantage of lesser parameters and could be trained faster compared with TSCNN.

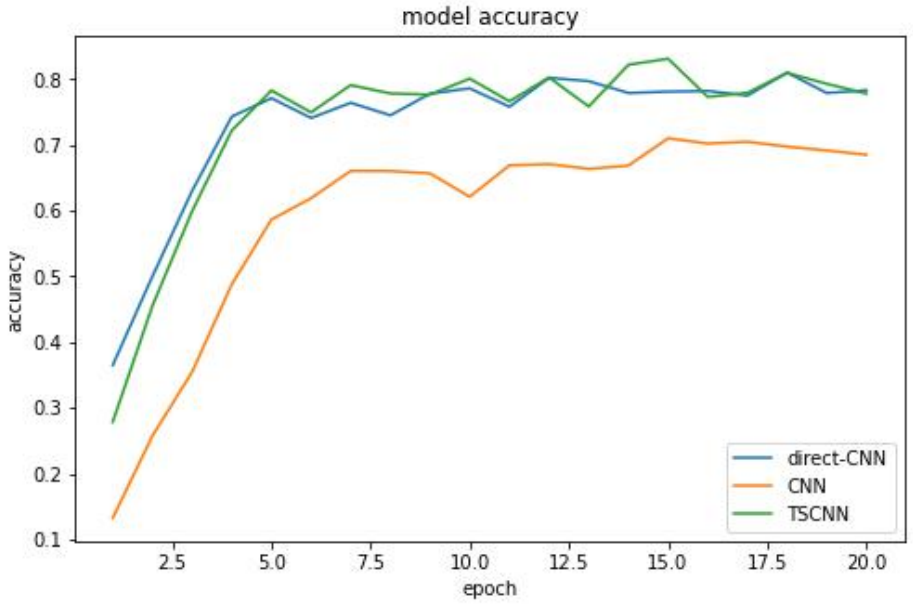


Fig. 3 Accuracy of models over each epoch on the UrbanSound8k dataset

Tab. 1 Accuracy of models on the UrbanSound8k dataset

Model names	Accuracy (%)
direct-CNN	81.0
CNN	71
TSCNN	83.1

3.2 TIMIT

The TIMIT dataset was a simple dataset to evaluate the model performance. We tested the generalization ability of the direct-CNN model on this dataset before applying it to real applications. In this experiment， data files of the TIMIT dataset were divided

Tab. 2 Number of parameters and average time of every training step of these models

Model names	Total params	Trainable params	Non-trainable params	Average Time of every training step (ms/step)
direct-CNN	173290	172906	384	24
CNN	52778	52778	0	12
TSCNN	431530	430653	877	75

into two categories: male talks and female talks, to evaluate the binary classification ability of the new model. Results were presented in Tab. 3. These results were like that on the urbanSound8k dataset, which demonstrated the validation of the proposed model.

Tab. 3 Accuracy of models on the TIMIT dataset

Model names	Accuracy (%)
direct-CNN	99.1
CNN	95.3
TSCNN	99.6

3.3 Telephone recording sounds

For the real-world recording sound dataset, test results of these models were showed in Tab. 4. Like the results on the common datasets, the accuracy rate performance of the new model was satisfactory as expected. The recall rates and the false alarm rates of these models were also evaluated. By unbalanced dataset, the false alarm rate could be decreased effectively. It was showed that the new model got even better performance on this target, which was important to real applications.

Tab. 4 Experiment results of these models on the telephone recording sounds

Model names	Accuracy (%)	False alarm (%)	Recall (%)
direct-CNN	81.0	7.5	76.3
CNN	75.1	21.8	70.2
TSCNN	87.3	9.0	79.7

4 Conclusions and future work

In this paper, a novel direct-CNN model was proposed to improve the availability of sound classification model under the real-world sound dataset. This model exhibited high accuracy rate and lower complexity, which was acceptable for computing resource limited devices. As for the real-world telephone recording sound dataset, the accuracy and false alarm rates of the new model were evaluated, and the results showed that the performance was excellent. In the following research, this model could be utilized to the auto-classification voice apps to shield telephone users from the recording calls such as advertisement and telemarketing, even fraud calls.

References:

[1] Chachada S.; Kuo C. -C. J. Environmental sound recognition: A survey. APSIPA Trans. Signal Inf. Process. 2014, 3.

[2] Piczak K J. Environmental sound classification with convolutional neural networks[C]// 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP). IEEE, 2015.

[3] Li S, Yao Y, Hu J, et al. An Ensemble Stacked Convolutional Neural Network Model for Environmental Event Sound Recognition[J]. Applied Sciences, 2018, 8(7): 1152.

[4] Su Y, Zhang K, Wang J, et al. Environment Sound Classi fi cation Using a Two-Stream CNN Based on Decision-Level Fusion[J]. Sensors, 2019, 19(7): 1733.

[5] Lecun Y, Boser B, Denker J, et al. Backpropagation Applied to Handwritten Zip Code Recognition[J]. Neural Computation, 2014, 1(4):541-551.

[6] Elman, J. L. Finding structure in time. CRL Technical Report 8801, Center for Research in Language, University of California, San Diego, 1988.

[7] HochreiterS, SchmidhuberJ. Long short-term memory, Neural computation 9(8): 1735-1780, 1997.

[8] Graves A, Mohamed A R, Hinton G. Speech Recognition with Deep Recurrent Neural Networks[C]// IEEE International Conference on Acoustics. IEEE, 2013.

[9] Garofolo, John S., Lamel, Lori F., Fisher, William M., Fiscus, Jonathon G., and Pallett, David S. DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. NIST speech disc 1-1. 1. NASA STI/Recon technical report, 93, 1993

[10] SalamonJustin, JacobyChristopher, Juan Pablo Bello. A Dataset and Taxonomy for Urban Sound Research[C]// ACM, 2014.

About the authors

WANG Gengrun was born in 1987. He received his Ph. D. in communication engineering from PLA University of Science

and Technology. He is now an assistant research fellow. His research interests include network security， data processing.（Email： wanggengrun@gmail. com）

LI Haitao was born in Tai'an， Shandong Province. He received the B. E. degree in information engineering from Information Engineering University of PLA， Zhengzhou， China， in 2004， and the M. S. degree from the National Digital Switching System Engineering and Technological Research Center （NDSC）， Zhengzhou， in 2007. He is currently an associate researcher with NDSC. His research interests include communication network security， data processing， and embedded system design. （Email： CA_LHT@126. com）

ZHU Yuhang was born in 1982. He received his B. E. degree in communication and information system from Information Engineering University. He is now a Ph. D. candidate of software engineering. His research interests include Network Science and Telecommunication Network Security. （Email： sharkin_zhu@163. com）

支持节点分割与异构备份的服务功能链部署方法

丁绍虎，谢记超，陈博，胡涛，刘迪洋
中国人民解放军战略支援部队信息工程大学，河南，郑州，450000

摘　要：针对服务功能链中虚拟网络功能面临的恶意攻击和随机失效风险，提出一种支持节点分割与异构备份的服务功能链部署方法。该方法引入虚拟网络功能节点拆分和异构冗余思想，在满足服务功能链部署约束条件下，以最小化链路资源开销为目标建立了优化模型，并设计了基于宽容分层序列思想与贪婪选择的服务功能链部署算法。实验结果表明，相比于传统冗余备份部署方法，该方法在较大幅度提高服务功能链抗攻击性能的同时，降低了17%的计算资源开销和10%的链路带宽资源开销，提升了12%的服务功能链请求接受率。

关键词：服务功能链、虚拟网络功能、可靠性、备份、节点分割

Service function chain deployment method supporting node splitting and heterogeneous backup

DING Shaohu, XIE Jichao, CHEN Bo, HU Tao, LIU Diyang
PLA Strategic Support Force Information Engineering University, Zhengzhou 450000, China

Abstract: Aiming at the malicious attacks and random failure risks faced by virtual network functions in the service function chain, a service function chain deployment method supporting node splitting and heterogeneous backup is proposed. This method introduced the idea of virtual network function node splitting and heterogeneous redundancy. Under the condition of meeting the deployment constraints of the service function chain, an optimization model is established with the goal of minimizing bandwidth resource overhead. And designed a service function chain deployment algorithm based on the forbearing stratified sequencing and greedy selection. The experimental results show that compared with the traditional redundant backup deployment method, the proposed method greatly improves the anti-attack performance of the service function chain while reducing 17% of the computing resource overhead and 10% of the bandwidth resource overhead. In addition, the request acceptance rate of the service function chain has been increased by 12%.

Key words: service function chain; virtual network function; reliability; backup; node splitting

1　引言

企业的流量通常需要经过一系列网络设备的处理，如防火墙、负载均衡器、网络地址转换器等，而这一组有序的专用网络功能序列被称为服务功能链［1］（Service Function Chain， SFC）。传统的服务功能链部署方式需要在网路中部署大量的专用网络功能硬件，在面对飞速发展的新兴网路服务局面时存在的问题日益明显［2］，如：新服务上线周期长、弹性扩展能力差、升级成本高、资源利用率低等。网络功能虚拟化基础设施即服务模式（Network Function Virtualization Infrastructure as a Service， NFVIaaS）的出现为解决传统服务功能链部署方式所面临的困境打开了新的局面［3］，通过将传统的专用网络功能硬件软件与硬件解耦，借助虚拟化技术将所需的网络功能软件运行在云环境下的通用硬件设备中，即可完成企业租户所需网络功能的按需、灵活部署，实现企业租户所需服务功能链的灵活、高效部署。

网络功能虚拟化基础设施即服务模式是一项

基金项目：国家重点研发计划(2017YFB0803201, 2017YFB0803204)，国家自然科学基金(61802429, 61872382, 61521003)
Foundation item: The National Key R&D Program of China (2017YFB0803201, 2017YFB0803204), The National Science Foundation of China (61802429, 61872382, 61521003)

很有前景的技术，使得企业租户外包服务功能链的部署与运维工作成为可能。然而处于新兴阶段的网络功能虚拟化基础设施即服务模式若想实现广泛的推广与应用还有许多的问题需要解决，其中保障服务功能链的可靠性便是其中之一［4］。相比于高集成度的专用网络功能硬件，借助虚拟化技术基于通用硬件设备实现的虚拟网络功能（Virtual Network Function， VNF）引入了许多新的层面，通用硬件层、通用操作系统层、通用虚拟化管理层等，经济全球化下第三方软、硬件解决方案也使得相关软、硬件面临着难以彻底消除和管控的缺陷、漏洞、甚至是后门等问题，这一系列原因使得“软化”的虚拟网络功能可靠性相对较低，面临着巨大的节点失效风险。

导致虚拟网络功能节点失效的原因复杂多样，如：硬盘、内存、网络接口等硬件故障，操作系统、虚拟机管理程序、虚拟网络功能软件等软件故障［5］，此外还面临着潜在攻击者利用缺陷、漏洞、后门等发起的恶意攻击导致的故障。任何一个虚拟网络功能节点失效都将极大的威胁着服务功能链的可靠性，影响着相关网络服务的可用性。

为了应对虚拟网络功能节点面临的失效风险，现有的解决方法主要通过可生存性的服务功能链部署来应对虚拟网络功能的可靠性问题，以此提高服务功能链的弹性能力，保证服务功能链所提供服务的可靠性。现有的解决方法主要分为两类：（1）迁移式［6-11］：当虚拟网络功能节点失效时，通过将相关虚拟网络功能节点以及虚拟链路进行迁移实现故障恢复，从而保证服务功能链可以为用户继续正常提供服务。（2）冗余备份式［12-15］：在服务功能链部署阶段提前在底层网络提供冗余备份资源，若出现虚拟网络功能节点失效，则迅速切换到备份的节点资源上，从而保证服务的连续性。总的来说，冗余备份式方法服务质量好，但会造成较大的资源消耗，影响租户服务的请求接受率；而迁移式方法虽然能节约资源消耗，但会造成较长的服务中断时间。

总结分析发现，针对虚拟网络功能节点面临的失效风险，当前的可生存性的服务功能链部署方案在应对虚拟网络功能节点随机失效方面取得了较好的效果，但是在应对潜在攻击者利用漏洞后门而发起的恶意攻击时仍存在一定不足。现有的冗余备份方法，在进行备份时未考虑主、备节点的异构性，导致主、备节点存在相同、相似缺陷。当主节点因遭受恶意攻击而无法提供服务时将会切换到备份节点，此时由于备份节点与主节点缺陷一致，攻击者无需再次付出大量的时间和精力代价，即可轻易的造成备份节点的再次失效，引起服务的长时间中断。文献［16-17］研究表明，在进行冗余备份时考虑节点间的异构性，对系统的抗攻击性有较好的效果。

针对服务功能链面临的恶意攻击和随机失效问题，本文在相关研究的基础上，引入虚拟网络功能节点拆分和拟态防御的异构冗余思想，提出了一种支持节点分割与异构备份的服务功能链部署方法，并以最小化链路资源开销为目标建立了优化模型，最后设计并实现了基于宽容分层序列思想与贪婪选择的服务功能链部署算法。

2 网络模型与问题描述

2.1 网络模型

如图1所示，该图简要描述了云环境下服务功能链部署示意图。租户的服务功能链请求通常包含一组有序的虚拟网络功能节点，云服务提供商接收到租户的服务功能链请求后，根据当前底层物理网络的资源状态，按照一定规则和策略，将租户的服务功能链请求映射到底层物理网络拓扑之上，进而租户利用已部署的服务功能链为用户提供相应的服务。为了便于描述和建立模型，接下来对本文所使用的关键符号进行描述，所使用的相关符号及其定义与先前的研究一致［18］。

1）物理网路。底层物理网络可用无向图 $\overline{G}=(\overline{N},\overline{S},\overline{L})$ 表示，其中 $\overline{N}$ 表示服务器集合，$\overline{S}$ 表示交换机集合，$\overline{L}$ 表示物理链路集合。使用 n,s 分别表示底层服务器节点和交换机节点的最大数量，底层交换机的连接关系可用矩阵 $\mathbf{B}_{s\times s}$ 表示，其中 $B_{i,j}\in R^{+}$ 表示交换机 i 与交换机 j 间的可用通信带宽，与交换机 u 直连的交换机集合可用 $V=\Lambda(u)=\{v|B_{u,v}>0\}\subseteq\overline{S},u\in\overline{S}$ 表示。服务器与交换机的连接关系可用矩阵 $\mathbf{H}_{n\times s}$ 表示，其中 $H_{i,j}\in\{0,1\}$ 表示服务器 i 与交换机 j 是否连接。服务器节点的资源类型（CPU、内存、存储空间、可编程硬件等）集合可用 K 表示，底层服务器节点

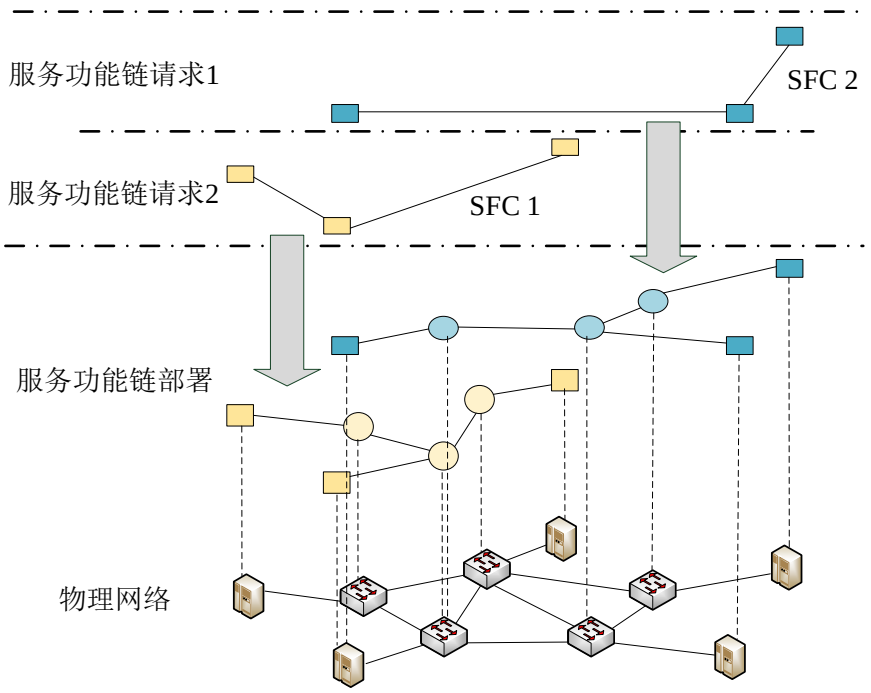


图1　基于云环境的服务功能链部署

的可用资源容量矩阵用$\mathbf{C}_{n\times k}$表示，其中$C_{i,j}\in R^{+}$表示服务器节点i第j类资源的可用数量。使用矩阵$\mathbf{C}^{rem}_{n\times k}$和$\mathbf{B}^{rem}_{s\times s}$分别表示当前物理网络中服务器和物理链路可用资源数量。

2）虚拟网络功能。底层物理网络支持的VNF类型集合可用P表示，虚拟网络功能资源需求系数矩阵可用$\mathbf{Q}_{p\times k}$表示，其中$Q_{i,j}$表示第i类虚拟网络功能处理单位带宽流量所需的j类资源数量。

3）服务功能链请求。网络中的服务功能链请求集合可用R表示，每一个服务功能链请求可用一个五元组表示$r=<\bar{u}^{r},\bar{v}^{r},\beta^{r},\tau^{r},\psi^{r}>$，其中$\bar{u}^{r}$表示接入交换机，$\bar{v}^{r}$表示出口交换机，$\beta^{r}$表示所需处理的流量的大小，$\tau^{r}$表示该服务功能链的生存周期，$\psi^{r}$则表示该服务功能链所包含的虚拟网络功能序列，请求$r$中第$i$个虚拟网络功能的类型可用$VNF^{r}_{i}\in P=\{1,2,\cdots,p\}$表示，请求$r$中虚拟网络功能的总数量用$m$表示。每一个服务功能链请求$r$都可以用一个有向图$G^{r}=(N^{r},L^{r})$表示，其中$N^{r}$表示由接入交换机、虚拟网络功能、和出口交换机构成的节点集合，L^{r}则表示连接这些节点的虚拟链路集合。

4）服务功能链部署。服务功能链的部署可表示为$M:G^{r}\rightarrow\overline{G}$，表示将服务功能链请求$G^{r}$映射到底层物理网络拓扑$\overline{G}$之上。云服务提供商接收到租户的服务功能链请求$r=<\bar{u}^{r},\bar{v}^{r},\beta^{r},\tau^{r},\psi^{r}>$后，根据当前底层物理网络的带宽资源$\mathbf{B}^{rem}_{s\times s}$和服务器资源$\mathbf{C}^{rem}_{n\times k}$状态，按照一定部署规则和策略，完成该服务功能链所需虚拟网络功能节点和虚拟链路的预部署，以确定物理网络是否可满足租户的需求，进而接受或拒绝租户的服务功能链请求。若接受租户请求，则随后完成所需虚拟网络功能的实例化和转发路径的下发。

2.2　问题描述

1）攻击实施过程。

尽管攻击者实施攻击的方式和具体实施方法多种多样，但是攻击的实施过程通常具有阶段性特征［19-20］，攻击者对目标系统实施攻击的过程主要可分为三个阶段：目标系统侦察阶段、系统漏洞挖掘和缺陷探测阶段、执行攻击阶段，如图2所示。

攻击者实施攻击的具体过程为：（1）通过各种方式和手段对目标系统的特征进行扫描和侦察，以了解和掌握目标系统的详细信息，如硬件设备型号、操作系统版本号、软件系统版本号等。（2）

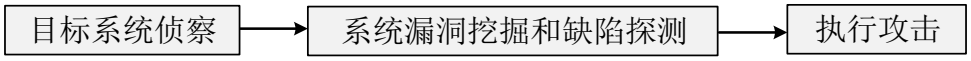


图2　攻击实施过程

在获得目标系统的信息特征后，对目标系统所采用的具体硬件设备、操作系统版本、应用软件版本等进行漏洞挖掘和缺陷探测，以获取目标系统可被利用的攻击途径和攻击靶点，同时针对攻击靶点开发或寻找可利用的恶意代码和攻击工具。（3）最后，攻击者针对目标系统的漏洞或缺陷，利用已掌握的恶意代码对目标系统执行攻击，以实现对目标系统敏感信息的窃取、篡改和破坏，致瘫或致乱目标系统，甚至造成目标硬件设备损坏。

对于攻击者来说，从确定对目标系统的攻击意图，到最终成功实施对目标系统的攻击，这期间往往需要花费大量的时间、精力和财力，特别是在系统漏洞挖掘和缺陷探测、以及攻击代码的寻找或开发阶段，需要进行大量的准备工作，而且还存在攻击者无法利用现有系统漏洞和缺陷的可能性。可是一旦攻击者掌握了针对某一特定特征目标系统的攻击方法，再对具有相同特征的攻击目标进行攻击时，可立刻对目标系统发动攻击，迅速实现攻击意图。

因此若系统间同构（具有相同的特征）时，则系统面临着共性缺陷问题，即：由于系统环境特征相同，存在着相同的可利用漏洞与缺陷，在防御恶意攻击时存在巨大的缺陷，攻击者一旦掌握了针对该特征目标系统的攻击方法和手段，即可轻易的实现对该类该特征目标系统的重复攻击，以较低的时间、精力和财力代价轻易地实现攻击意图。

2） *现有SFC冗余备份部署方法存在的问题*

如图3所示，描述了服务功能链的冗余部署的典型场景，图中服务器节点的灰度代表着服务器节点的信息特征，若两个服务器节点同构，则在图中灰度是一致的，如服务器节点N1和N8两者即为同构的服务器节点。图3的上半部分描述了一条包含3个虚拟网络功能的服务功能链请求，图3下半部分则描述了该服务功能链请求在物理网络的实际部署情况，该请求所包含的3个虚拟网络功能 VNF_1^r、VNF_2^r、VNF_3^r 分别部署于服务器节点N4、N1和N5，其中第二个虚拟网络功能 VNF_2^r 有备份需求，而现有的SFC冗余备份部署方法往往不考虑主节点与备份节点的异构性，如图3所示，其备份节点（backup node， BN）$VNF_{2,BN}^r$ 部署的服务器节点N8与主节点（primary node， PN）VNF_2^r 部署的服务器节点N1是同构的，导致主节点与备份节点存在共性缺陷。当主节点因遭受攻击而无法提供服务时将会切换到备份节点，此时由于主备节点缺陷一致，攻击者无需再次付出大量的时间、精力等代价，即可轻易的造成备份节点的失效与服务中断，引起服务的长时间中断，造成用户服务不可用，违反云服务提供商与租户签订的服务等级协议。因此在为租户关键虚拟网络功能提供备份服务时，应考虑底层服务节点的异构性，以避免主备节点面临的共性缺陷问题，提高该服务功能链冗余备份节点抵抗恶意攻击的有效性，提升潜在恶意攻击者的攻击难度和攻击代价。

此外当前的SFC部署方法往往未考虑SFC请求中虚拟网络功能节点的可拆分属性（即：可将大的虚拟网络功能节点拆分成多个小的同等功能的虚拟网络功能节点，以降低部署单个虚拟网络功能节点所需的资源量），因此无法更精细、有效的利用底层物资资源［21-22］。特别是在冗余备份场景下，当主虚拟网络功能节点与备份虚拟网络功能节点需要的资源较多而底层物理网络资源又不太充足时，将难以找到能够提供足够资源的服务器节点，造成虚拟网络功能节点映射失败，进而使得租户的SFC请求遭受拒绝。

若在服务功能链部署时，可将虚拟网络功能节点拆分为两个较小的同等功能虚拟网络功能节点，则可在备份时同样用较小的虚拟网络功能节点为两者提供共享的冗余备份，可在一定程度上降低备份资源开销。总的来说，在进行服务功能链部署时未考虑虚拟网络功能节点的可拆分属性，将会导致云服务提供商碎片化资源无法充分利用，致使底层网络备份资源开销过大，影响租户服务功能链请求接受率，进而影响云服务提供商的长期收益。

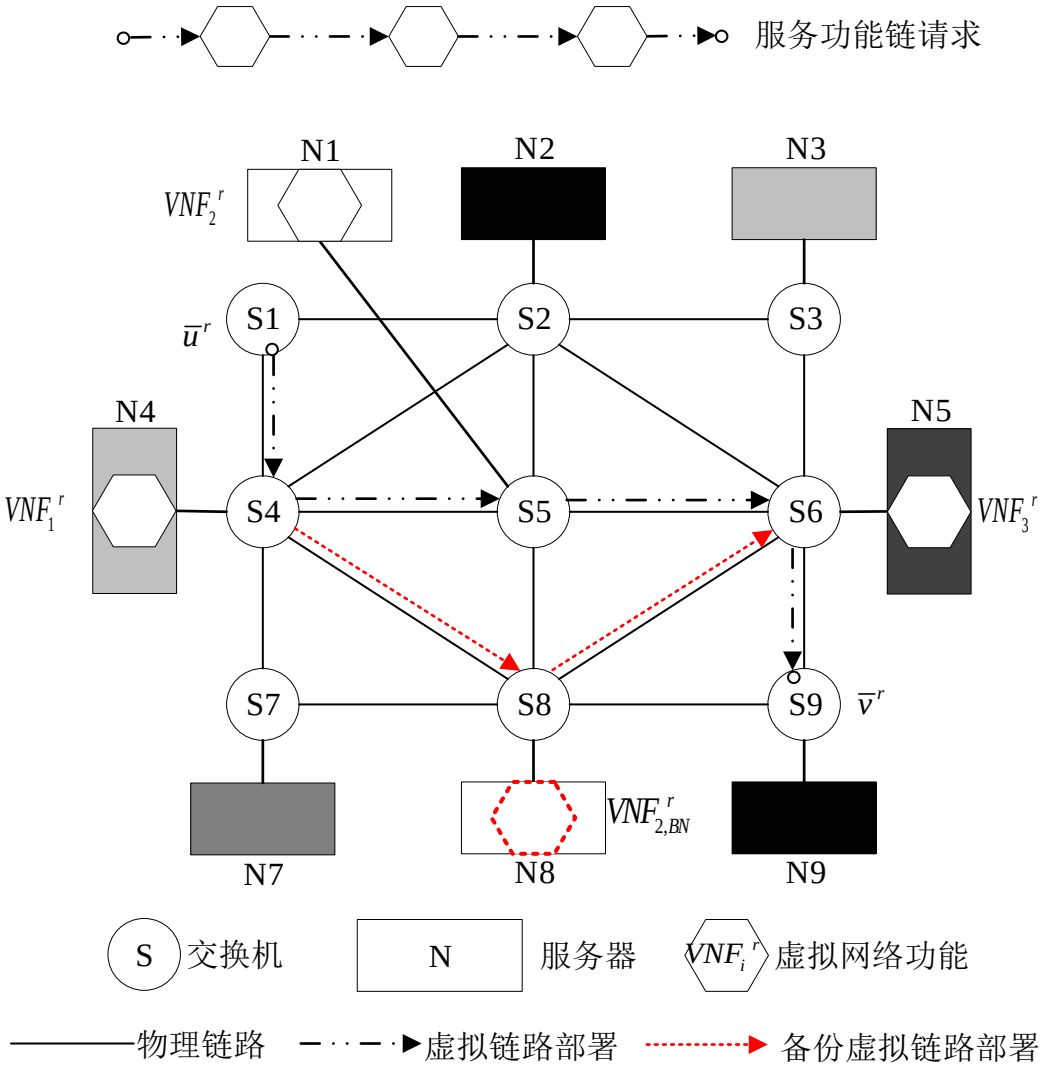


图3　支持冗余备份的服务功能链部署

3　支持节点分割与异构备份的SFC部署方法

为了增强服务功能链冗余备份节点所提供冗余保护的有效性，提高潜在恶意攻击者对服务功能链实施攻击的难度和代价，同时实现底层物理资源的充分利用，本文充分利用了云环境下虚拟网络功能可灵活部署的优势，提出了一种支持节点分割与异构备份的服务功能链部署方法，该方法主要包含以下两个步骤：（1）构建增强型SFC拓扑；（2）增强型SFC拓扑的异构冗余部署。

3.1　构建增强型SFC拓扑

当租户的SFC请求到达时，依据租户SFC中虚拟网络功能节点的可拆分属性和冗余备份需求属性对租户SFC请求初始拓扑进行扩展，通过节点分割与冗余备份方式，为租户构建增强型SFC拓扑，进而依据增强型拓扑对SFC进行部署。

如图4所示，本节以一个包含3个虚拟网络功能节点的SFC请求为例对所提方法进行介绍，图中$\overline{u}^r$和$\overline{v}^r$分别表示接入交换机和出口交换机。租户的SFC请求中明确，仅第二个虚拟网络功能节点VNF_2^r可进行拆分，而第二、三个虚拟网络功能节点VNF_2^r和VNF_3^r需要进行冗余备份。

按照租户的SFC请求需求，首先根据租户SFC请求中虚拟网络功能节点的可拆分属性对相关节点进行拆分。如图5所示，对允许节点分割的VNF_2^r进行拆分，将VNF_2^r拆分为两个资源需求较少

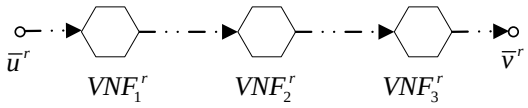


图4　租户服务功能链请求示例

的主节点（primary node， PN）$VNF^r_{2,PN,1}$和副节点$VNF^r_{2,PN,2}$，此时$VNF^r_{2,PN,1}$和$VNF^r_{2,PN,2}$在部署时所需的资源均为部署VNF^r_2所需资源的二分之一，所需要处理流量的大小也为原VNF^r_2流量处理任务的二分之一。对于不允许分割的VNF^r_i节点，为了后续便于模型的描述和建立，我们将VNF^r_i用$VNF^r_{i,PN,1}$表示，而不允许拆分的节点不存在副主节点$VNF^r_{i,PN,2}$。

在完成节点拆分后，需要完成对有备份需求虚拟网络功能节点的冗余备份，如图5所示，分别为有备份需求的VNF^r_2和VNF^r_3提供冗余备份节点（backup node， BN）$VNF^r_{2,BN}$和$VNF^r_{3,BN}$。此时备份节点$VNF^r_{2,BN}$是主节点$VNF^r_{2,PN,1}$和$VNF^r_{2,PN,2}$的共享备份节点，同时为两者提供冗余备份服务，后续在对$VNF^r_{2,BN}$进行部署时，所需的资源与主节点$VNF^r_{2,PN,1}$和$VNF^r_{2,PN,2}$一致，也为原VNF^r_2节点所需资源的二分之一。相比于不进行节点拆分的冗余备份方式，拆分后的节点所需资源降低，即使在物理网络负载较高的情况下，依然更容易找到满足需求的承载服务器，此外拆分后进行冗余备份时节约了二分之一的备份资源消耗。而不支持节点拆分的VNF^r_3，在部署备份节点$VNF^r_{3,BN}$时所需的资源与VNF^r_3一致。此外在进行节点拆分与冗余备份的同时，需要同时完成扩展节点与前置、后置虚拟网络功能节点间虚拟链路的连接，以保证增强型SFC拓扑应有的连通性，至此基于租户SFC请求的增强型SFC拓扑构建完成，在下一小节将描述增强型SFC拓扑的异构冗余部署方法。

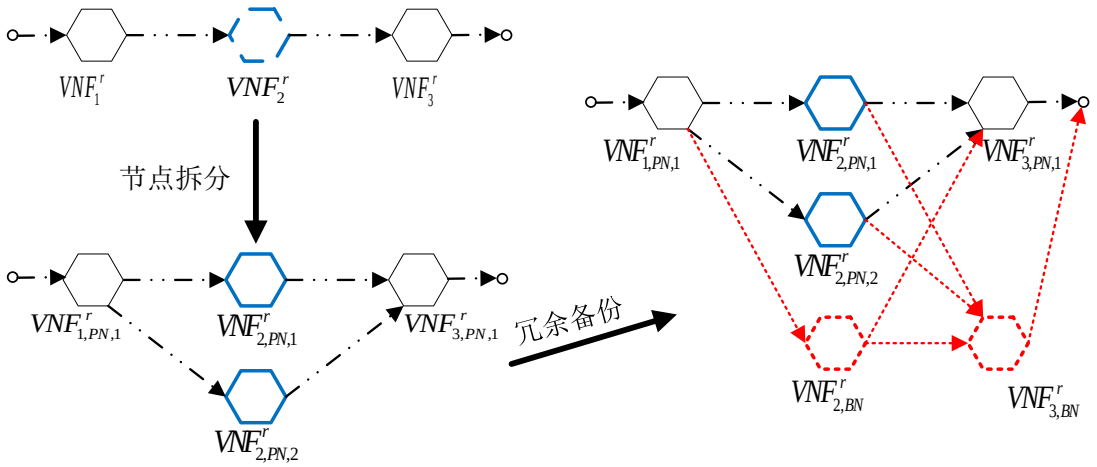


图5　构建增强型服务功能链拓扑

3.2　增强型SFC拓扑的异构冗余部署

在进行增强型SFC拓扑部署时，相关虚拟网络功能节点无论部署在哪里所需消耗的服务器资源是一样的，然而部署虚拟网络功能节点选择的位置却极大的影响着节点间虚拟链路的部署情况。若进行部署时，网络功能节点选择的位置距离前置、后置虚拟网络功能节点过远，则会导致所需部署的虚拟链路路径过长（节点间跳数过大），造成服务功能链部署带宽成本加大，甚至会由于路径过长，导致服务功能链传输延迟过大，影响用户服务体验。为此我们在进行了增强型SFC拓扑部署时，采用分阶段部署的策略，目标是试图最小化带宽资源消耗。

首先我们对主链路的所有主节点进行部署，如图6所示，在该阶段我们对主链路的三个主节点$VNF^r_{1,PN,1}$、$VNF^r_{2,PN,1}$和$VNF^r_{3,PN,1}$进行部署，本实例中$VNF^r_{1,PN,1}$、$VNF^r_{2,PN,1}$和$VNF^r_{3,PN,1}$分别部署于N4、N1、N5服务器节点，最后采用最短路径法完成相关节点间虚拟链路的部署。

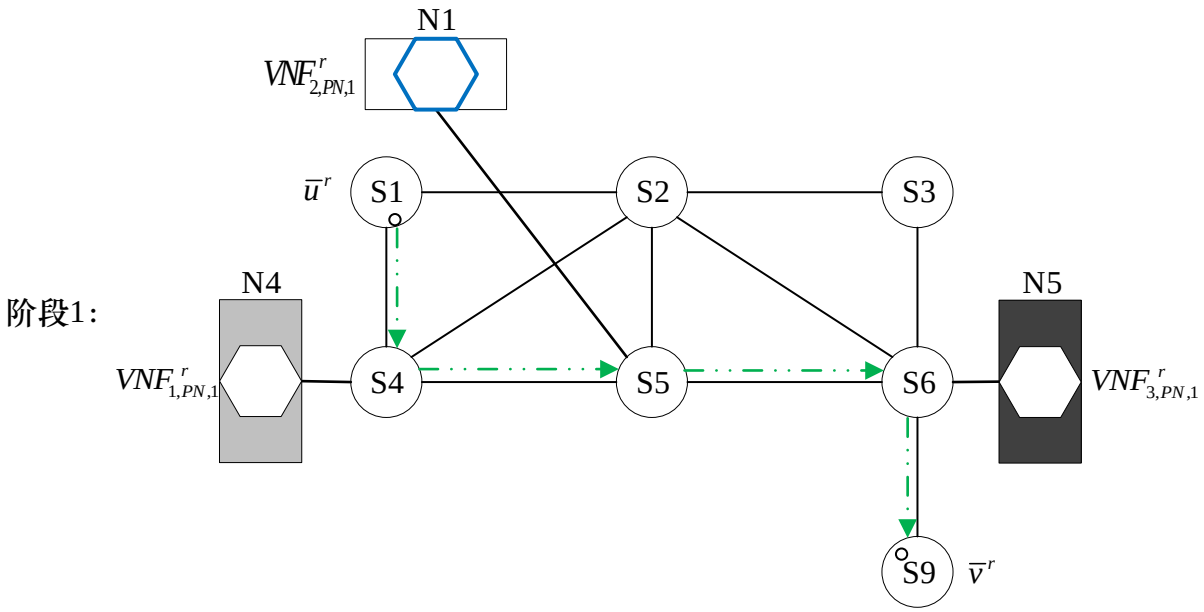


图6　主链路主节点部署

接下来进入阶段2，对主链路的所有副节点进行部署，在该阶段完成对所有已拆分节点副节点的部署。本实例中涉及 $VNF^{r}_{2,PN,2}$ 副节点的部署，在选择部署 $VNF^{r}_{2,PN,2}$ 的服务器节点时，应考虑承载 $VNF^{r}_{2,PN,1}$ 和 $VNF^{r}_{2,PN,2}$ 服务器节点间的异构性，同时也需要考虑用于承载 $VNF^{r}_{2,PN,2}$ 服务器节点距离前置节点 $VNF^{r}_{1,PN,1}$ 和后置节点 $VNF^{r}_{3,PN,1}$ 的距离，以避免虚拟链路过长导致带宽资源成本提高以及延迟过大。如图7所示，本实例将 $VNF^{r}_{2,PN,2}$ 部署于与N1异构的服务器节点N2，并采用最短路径法完成相关虚拟链路的部署。

接下来进入阶段3，对备份链路所有的备份节点进行部署，本实例中涉及的节点有 $VNF^{r}_{2,BN}$ 和 $VNF^{r}_{3,BN}$，如图8所示。首先对备份节点 $VNF^{r}_{2,BN}$ 进行部署，在部署时同样需要考虑承载服务器节点间的异构性，同时也需要合理的选择承载服务器节点，以避免连接前置节点 $VNF^{r}_{1,PN,1}$ 和后置节点 $VNF^{r}_{3,PN,1}$ 链路过长。为此本实例将 $VNF^{r}_{2,BN}$ 部署于服务器N7，并利用最短路径法完成相关虚拟链路的部署，此时用于承载 $VNF^{r}_{2,PN,1}$、$VNF^{r}_{2,PN,2}$ 和 $VNF^{r}_{2,BN}$ 的服务器N1、N2、N7相互间均是异构的。接下来对备份节点 $VNF^{r}_{3,BN}$ 进行部署，首先依然是在选择服务器节点时保证承载服务器节点间的异构性，同时尽量减少部署虚拟链路时的带宽资源消耗。本实例中部署 $VNF^{r}_{3,BN}$ 备份节点时需要部署的备份虚拟链路较多，需要构建与前置 $VNF^{r}_{2,PN,1}$、$VNF^{r}_{2,PN,2}$ 和 $VNF^{r}_{2,BN}$ 节点间的虚拟链路，与后置节点出口交换机 $\overline{v}^{r}$ 间的虚拟链路。经过以上3个阶段，即可完成增强型SFC拓扑的异构冗余部署。在下一节，我们将具体介绍支持节点分割与异构备份的服务功能链部署模型。

4　支持节点分割与异构备份的SFC部署模型

首先对本文所使用的关键符号及其定义进行总结，如表1所示。

定义二值矩阵 $SA^{r}_{1\times m}$ 表示请求 r 的VNF可拆分属性矩阵，元素 $SA^{r}_{1,i}$ 表示请求 r 中第 i 个VNF的可拆分属性，若 $SA^{r}_{1,i}=1$，则表示 VNF^{r}_{i} 可进行拆分，则请求 r 的增强型拓扑中将会存在节点 $VNF^{r}_{i,PN,1}$ 和 $VNF^{r}_{i,PN,2}$。此外，另 $SA^{r}_{1,0}=0, SA^{r}_{1,m+1}=0$，分别表示接入交换机节点 $\overline{u}^{r}$ 和出口交换机 $\overline{v}^{r}$ 的不可进行拆分。

定义二值矩阵 $BR^{r}_{1\times m}$ 表示请求 r 的VNF备份需求矩阵，元素 $BR^{r}_{1,i}$ 表示请求 r 中第 i 个VNF的备份需求，若 $BR^{r}_{1,i}=1$，则表示在部署时需要为 VNF^{r}_{i} 提供冗余备份，则请求 r 的增强型拓扑中将会存在节点 $VNF^{r}_{i,BN}$。此外，另 $BR^{r}_{1,0}=0, BR^{r}_{1,m+1}=0$，分

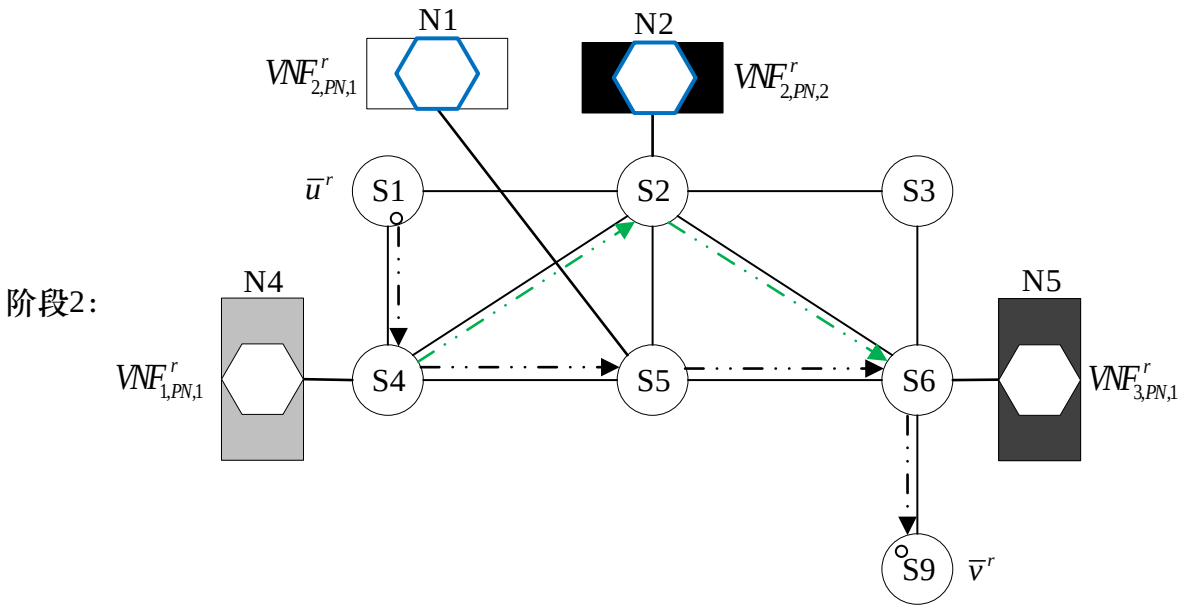


图7　主链路副节点部署

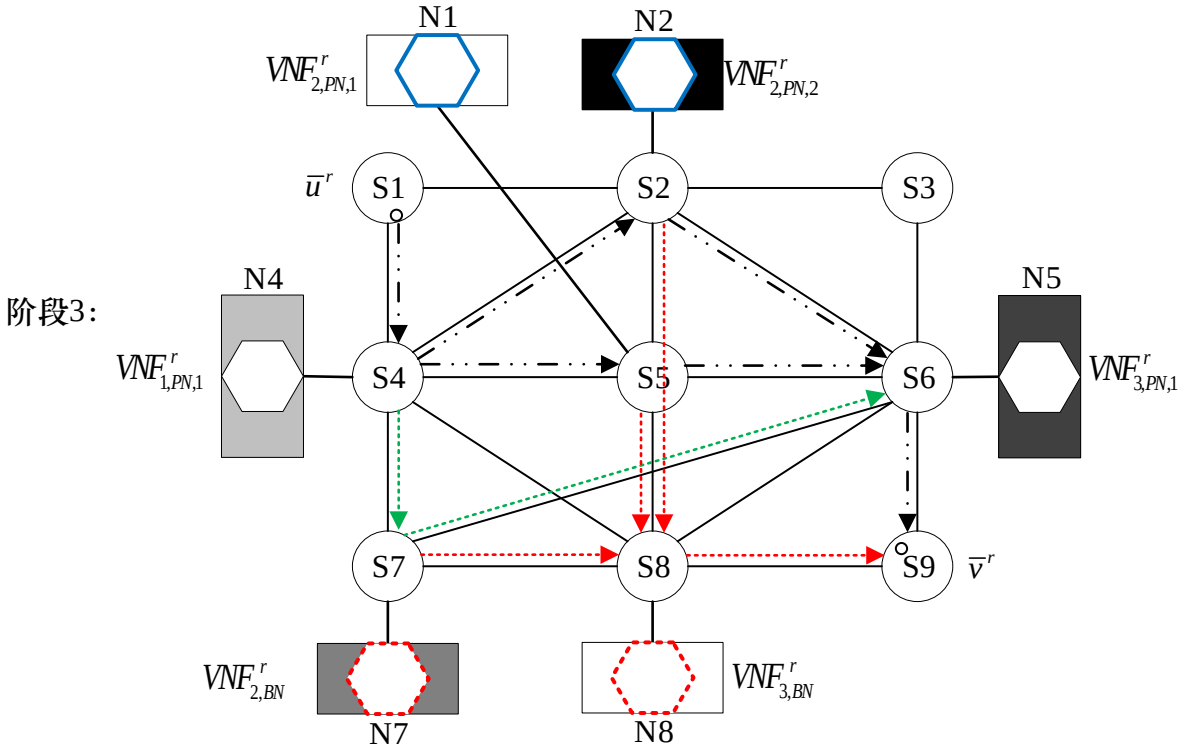


图8　备份链路备份节点部署

别表示接入交换机节点$\overline{u}^r$和出口交换机$\overline{v}^r$无需进行冗余备份。

为了便于描述以及模型建立，我们将增强型

表1　本文所使用的关键符号定义

符号	描述
$\overline{G}=(\overline{N},\overline{S},\overline{L})$	底层物理网络
$\overline{N},\overline{S},\overline{L}$	分别表示通用服务器集合、交换机集合、物理链路集合
n,s	分别表示底层网络中服务器、交换机的总数量
$\mathbf{B}_{S\times S}$	交换机连接矩阵
$B_{i,j}\in R^{+}$	表示交换机节点i到j的通信链路容量
$V=\Lambda(u)=\{v\|B_{u,v}>0\}$	表示与交换机u直连的交换机集合
$\mathbf{H}_{n\times s}$	服务器与交换机连接矩阵
$H_{i,j}\in\{0,1\}$	表示服务器节点i是否连接在交换机j上
K,k	表示服务器资源类型集合、类型总数量
$\mathbf{C}_{n\times k}$	底层服务器资源容量矩阵
$C_{i,j}\in R^{+}$	表示服务器节点i上可提供的第j类资源的数量
$\mathbf{C}^{rem}_{n\times k},\mathbf{B}^{rem}_{s\times s}$	表示当前网络服务器和物理链路的资源余量
$\mathbf{C}^{rem}_{n\times k},\mathbf{B}^{rem}_{s\times s}$	表示当前网络服务器和物理链路的资源余量
P	表示VNF类型集合
$\mathbf{Q}_{p\times k}$	VNF资源需求系数矩阵
$Q_{i,j}$	i类型VNF处理单位带宽流量所占用的j类资源数量
R	表示SFC的请求集合
$r=<\overline{u}^{r},\overline{v}^{r},\beta^{r},\tau^{r},\psi^{r}>$	租户SFC请求信息
$\overline{u}^{r},\overline{v}^{r}$	表示接入交换机和出口交换机
β^{r}	表示该服务链所需处理的流量大小
τ^{r}	表示该请求的生命周期
ψ^{r}	表示处理流量所需的VNF序列
m	该请求中VNF的总数量
$VNF^{r}_{i}\in P=\{1,2,\cdots,p\}$	表示请求r中第$i\in\{1,2,\cdots,m\}$个VNF的类型
$G^{r}=(N^{r},L^{r})$	SFC请求有向图
N^{r}	节点（接入交换机，VNFs，出口交换机）集合
L^{r}	连接节点的虚拟链路集合
$M:G^{r}\rightarrow\overline{G}$	SFC请求拓扑G^{r}映射到物理网络拓扑$\overline{G}$之上
$VNF^{r}_{i,PN,1},VNF^{r}_{i,PN,2},VNF^{r}_{i,BN}$	分别表示VNF^{r}_{i}的主节点、副节点和备份节点
$SA^{r}_{1\times m}$	虚拟网络功能可拆分属性矩阵
$BR^{r}_{1\times m}$	虚拟网络功备份需求矩阵
$VN^{r}_{3\times m}$	增强型拓扑节点矩阵
$VN^{r}_{1,i},VN^{r}_{2,i},VN^{r}_{3,i}$	分别表示VNF^{r}_{i}的主节点、副节点和备份节点
$VL^{r,i-1,i}_{3\times 3}$	虚拟链路存在性矩阵
$n^{r}_{1,i},n^{r}_{2,i},n^{r}_{3,i}$	分别表示$VN^{r}_{1,i},VN^{r}_{2,i},VN^{r}_{3,i}{}^{r}$所部署的服务器节点
$(X_1,X_2,\cdots,X_Z)$	服务器节点的特征信息
$(Y_1,Y_2,\cdots,Y_z)$	两个服务器节点间的异构度向量
HLT	两个服务器节点间的异构度值
$F^{r}_{3\times m\times n}$	虚拟节点与服务器节点间的映射关系矩阵
$E^{r,(j',i-1),(j,i)}_{s\times s}$	虚拟链路与底层物理链路的映射关系矩阵
$B^{r,(j',i-1),(j,i)}_{\mathrm{cost}}$	一条虚拟链路的带宽资源消耗

SFC拓扑中的拆分节点（主、副节点）和备份节点$VNF^{r}_{i,PN,1}$、$VNF^{r}_{i,PN,2}$、$VNF^{r}_{i,BN}$分别用虚拟节点（virtual node，VN）$VN^{r}_{1,i}$、$VN^{r}_{2,i}$、$VN^{r}_{3,i}$表示，如图9所示。此外另$VN^{r}_{1,0}$表示接入交换机节点，$VN^{r}_{1,m+1}$表示出口交换机节点。定义矩阵$VN^{r}_{3\times m}$用于表示请求r的增强型拓扑节点矩阵，元素

$VN_{j,i}^{r} \in \{0, 0.5, 1\}$，其中 $j \in \{1, 2, 3\}$，$i \in \{1, 2, \cdots, m\}$。若 $VN_{j,i}^{r} = 0$，则表示不存在该节点；若 $VN_{j,i}^{r} = 0.5$ 则表示存在节点拆分，部署 $VN_{j,i}^{r}$ 时所需的服务器资源为部署 VNF_{i}^{r} 所需服务器资源的二分之一；若 $VN_{j,i}^{r} = 1$，则表示部署 $VN_{j,i}^{r}$ 时所需的服务器资源等于部署 VNF_{i}^{r} 所需的服务器资源。

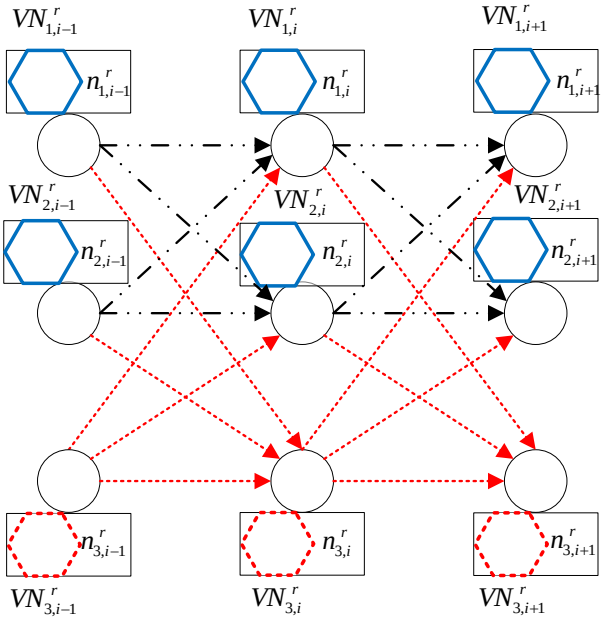


图9　增强型拓扑节点

由VNF可拆分属性矩阵 $SA_{1\times m}^{r}$ 和备份需求矩阵 $BR_{1\times m}^{r}$ 可求得服务功能链请求 r 的增强型拓扑节点矩阵 $VN_{3\times m}^{r}$，矩阵各元素的计算公式如下所示：

$$VN_{1,i}^{r} = \frac{1}{1 + SA_{1,i}^{r}} \tag{1}$$

$$VN_{2,i}^{r} = \frac{SA_{1,i}^{r}}{2} \tag{2}$$

$$VN_{3,i}^{r} = \frac{BR_{1,i}^{r} \times SA_{1,i}^{r}}{2} \tag{3}$$

定义矩阵 $VL_{3\times 3}^{r,i-1,i}$ 表示请求 r 的增强型拓扑节点间虚拟链路存在性矩阵，元素 $VL_{j',j}^{r,i-1,i} \in \{0, 0.5, 1\}$ 表示节点 $VN_{j',i-1}^{r}$ 与节点 $VN_{j,i}^{r}$ 间虚拟链路存在性以及所需带宽相比于 β^{r} 的比例，其中另 $VN_{j',0}^{r}$ 表示接入交换机节点，另 $VN_{j,m+1}^{r}$ 表示流出交换机节点。若 $VL_{j',j}^{r,i-1,i} = 0$，则表示节点 $VN_{j',i-1}^{r}$ 与节点 $VN_{j,i}^{r}$ 间不存在虚拟链路连接。若 $VL_{j',j}^{r,i-1,i} = 0.5$，则表示节点 $VN_{j',i-1}^{r}$ 与节点 $VN_{j,i}^{r}$ 间存在虚拟链路连接，且虚拟链路的带宽为原 VNF_{i-1}^{r} 与 VNF_{i}^{r} 间虚拟链路单宽 β^{r} 的二分之一；若 $VL_{j',j}^{r,i-1,i} = 1$，则表示节点 $VN_{j',i-1}^{r}$ 与节点 $VN_{j,i}^{r}$ 间存在的虚拟链路带宽为 β^{r}。矩阵 $VL_{3\times 3}^{r,i,i-1}$ 可由备份需求矩阵 $BR_{1\times m}^{r}$ 和可拆分属性矩阵 $SA_{1\times m}^{r}$ 求得，矩阵各元素计算公式如下所示：

$$VL_{1,1}^{r,i-1,i} = \frac{1 + \overline{SA_{1,i-1}^{r}} \times \overline{SA_{1,i}^{r}}}{2} \tag{4}$$

$$VL_{1,2}^{r,i-1,i} = SA_{1,i}^{r} \times \frac{\overline{SA_{1,i-1}^{r}}}{2} \tag{5}$$

$$VL_{1,3}^{r,i-1,i} = BR_{1,i}^{r} \times \frac{1 + \overline{SA_{1,i-1}^{r}} \times \overline{SA_{1,i}^{r}}}{2} \tag{6}$$

$$VL_{2,1}^{r,i-1,i} = SA_{1,i-1}^{r} \times \frac{\overline{SA_{1,i}^{r}}}{2} \tag{7}$$

$$VL_{2,2}^{r,i-1,i} = \frac{SA_{1,i-1}^{r} \times SA_{1,i}^{r}}{2} \tag{8}$$

$$VL_{2,3}^{r,i-1,i} = \frac{BR_{1,i}^{r} \times SA_{1,i-1}^{r}}{2} \tag{9}$$

$$VL_{3,1}^{r,i-1,i} = BR_{1,i-1}^{r} \times \frac{1 + \overline{SA_{1,i-1}^{r}} \times \overline{SA_{1,i}^{r}}}{2} \tag{10}$$

$$VL_{3,2}^{r,i-1,i} = \frac{BR_{1,i-1}^{r} \times SA_{1,i}^{r}}{2} \tag{11}$$

$$VL_{3,3}^{r,i-1,i}=BR_{1,i-1}^{r}\times BR_{1,i}^{r}\times\frac{1+\overline{SA_{1,i-1}^{r}}\times\overline{SA_{1,i}^{r}}}{2}\tag{12}$$

在对增强型SFC拓扑进行冗余部署时需要保证承载主节点$VN_{1,i}^{r}$、副节点$VN_{2,i}^{r}$和冗余备份节点$VN_{3,i}^{r}$的服务器间的异构性，接下来对承载服务器的异构性进行建模。选择z个属性（如：CPU类型、操作系统类型、虚拟主机管理程序类型等）对服务器节点的信息特征进行抽象，则物理服务器节点的信息特征可用z元组$(X_1,X_2,\cdots,X_Z)$表示。定义异或运算规则$\oplus$，用于将服务器节点n_1和n_2的第i个信息特征进行异或操作。若服务器节点n_1和n_2的第i个信息特征相同，即$X_i=X_i'$，则$X_i\oplus X_i'=0$；若服务器节点n_1和n_2的第i个信息特征不相同，即$X_i\neq X_i'$，则$X_i\oplus X_i'=1$。

定义向量$(Y_1,Y_2,\cdots,Y_z)$表示两服务器间n_1和n_2的异构度向量。定义函数$f(n_1,n_2)$用于计算两服务器间的异构度向量$(Y_1,Y_2,\cdots,Y_z)$，计算公式如下：

$$(Y_1,Y_2,\cdots,Y_z)=f(n_1,n_2)=(X_1\oplus X_1',X_2\oplus X_2',\cdots,X_z\oplus X_z')\tag{13}$$

定义函数$g(n_1,n_2)$用于计算两服务器间的异构度值，计算公式如下：

$$g(n_1,n_2)=\left|f(n_1,n_2)\right|=\sqrt{Y_1^2+Y_2^2+\cdots+Y_z^2}\tag{14}$$

$$\sum_{n\in\overline{N}}F_{j,i,n}^{r}=\begin{cases}0 & if\ VN_{j,i}^{r}=0\\ 1 & f\ VN_{j,i}^{r}>0\end{cases}\quad\forall r\in R,\forall i\in\{1,\cdots,m\},\forall j\in\{1,2,3\}\tag{17}$$

定义矩阵$E_{s\times s}^{r,(j',i-1),(j,i)}$表示请求$r$虚拟节点$VN_{j',i-1}^{r}$与虚拟节点$VN_{j,i}^{r}$间虚拟链路与底层物理链路的映射关系矩阵，元素$E_{u,v}^{r,(j',i-1),(j,i)}\in\{0,1\}$表示$VN_{j',i-1}^{r}$与$VN_{j,i}^{r}$间的虚拟链路是否部署在了交换机$u$与交换机$v$之间的物理链路之上。在进行虚拟链路部署时应满足以下物理链路带宽资源约束条件：

$$E_{u,v}^{r,(j',i-1),(j,i)}\times VL_{j'j}^{r,i-1,i}\times\beta^{r}\leqslant B_{u,v}^{rem}\quad\forall r\in R,\forall i\in\{1,\cdots,m+1\},\forall j,j'\in\{1,2,3\},\forall u,v\in\overline{S}\tag{18}$$

定义$B_{\cos t}^{r,(j',i-1),(j,i)}$表示部署$VN_{j,i}^{r}$与$VN_{j',i-1}^{r}$间的虚拟链路所消耗的带宽资源，$B_{\cos t}^{r,(j',i-1),(j,i)}$的计算公式如下所示：

$$B_{\cos t}^{r,(j',i-1),(j,i)}=\sum_{u\in\overline{S}}\sum_{v\in\Lambda(u)\subseteq\overline{S}}E_{u,v}^{r,(j',i-1),(j,i)}\times VL_{j'j}^{r,i-1,i}\times\beta^{r}\quad\forall r\in R,\forall i\in\{1,\cdots,m+1\},\forall j,j'\in\{1,2,3\}\tag{19}$$

由于虚拟节点$VN_{j,i}^{r}$部署于任何服务器节点所消耗的服务器资源是相同的，但是$VN_{j,i}^{r}$选择的部署服务器节点位置却极大的影响着后续部署虚拟链路所消耗的带宽资源，因此为了节约部署租户SFC请求的资源消耗，本文以最小化链路资源开销作为增强型SFC拓扑异构冗余部署部署的优化目标，最小化链路资源开销的目标函数表达式如下：

$$\min\left[\sum_{i\in\{1,\cdots,m+1\}}\sum_{j\in\{1,2,3\}}\sum_{j'\in\{1,2,3\}}\sum_{u\in\overline{S}}\sum_{v\in\Lambda(u)\subseteq\overline{S}}E_{u,v}^{r,(j',i-1),(j,i)}\times VL_{j'j}^{r,i-1,i}\times\beta^{r}\right]\tag{20}$$

5 算法设计

本文以式（20）最小链路资源开销为目标，用$n_{1,i}^{r}$、$n_{2,i}^{r}$和$n_{3,i}^{r}$分别表示$VN_{1,i}^{r}$、$VN_{2,i}^{r}$和$VN_{3,i}^{r}$所部署的服务器节点，此外另$n_{1,0}^{r}$、$n_{1,m+1}^{r}$分别等于接入交换机节点和出口交换机节点。为了保证承载主节点$VN_{1,i}^{r}$、副节点$VN_{2,i}^{r}$和冗余备份节点$VN_{3,i}^{r}$的服务器间满足一定的异构度，定义异构度阈值（heterogeneous level threshold， HLT）用于保证相关服务器间的异构度值，则服务器节点$n_{1,i}^{r}$、$n_{2,i}^{r}$和$n_{3,i}^{r}$间的异构度需要满足以下约束条件：

$$g(n_{1,i}^{r},n_{2,i}^{r})\geqslant HLT,g(n_{1,i}^{r},n_{3,i}^{r})\geqslant HLT,g(n_{2,i}^{r},n_{3,i}^{r})\geqslant HLT\quad n_{1,i}^{r},n_{2,i}^{r},n_{3,i}^{r}\in\overline{N}\tag{15}$$

此外，用于承载相关虚拟$VN_{1,i}^{r}$、$VN_{2,i}^{r}$和$VN_{3,i}^{r}$节点的服务器还需要满足资源容量限制。定义三维矩阵$F_{3\times m\times n}^{r}$用于表示请求$r$虚拟节点与服务器节点间的映射关系矩阵，元素$F_{j,i,n}^{r}\in\{0,1\}$表示虚拟节点$VN_{j,i}^{r}$是否部署在服务器节点$n$上，若$VN_{j,i}^{r}$部署于服务器节点$n$，则$F_{j,i,n}^{r}=1$。则为了保证底层物理服务器的资源容量限制，则需要满足以下服务器资源约束条件：

$$F_{j,i,n}^{r}\times VN_{j,i}^{r}\times Q_{VNF_i^r,k}\times\beta^{r}\leqslant\mathbf{C}_{n,k}^{rem}\quad\forall r\in R,\forall i\in\{1,\cdots,m\},\forall j\in\{1,2,3\},\forall n\in\overline{N},\forall k\in K\tag{16}$$

为了保证每个$VN_{j,i}^{r}$节点能且仅能部署在一个服务器节点，需要满足节点映射唯一性约束条件：

以式（1）~式（19）为约束条件，设计了一种基于宽容分层序列思想与贪婪选择的服务功能链部署算法，算法流程如算法1所示。

算法1　基于宽容分层序列思想与贪婪选择的服务功能链部署算法

输入： 服务功能链请求信息r

输出： 请求r的增强型异构冗余部署部署方案

依据请求r的可拆分属性矩阵$SA^r_{1\times m}$和备份需求矩阵$BR^r_{1\times m}$，计算请求r的增强型拓扑节点矩阵$VN^r_{3\times m}$以及节点间链路存在性矩阵$VL^{r,i-1,i}_{3\times 3}\ i\in\{1,2,\cdots,m\}$。

$\overline{N}^{occopy}$=None

###主链路主节点部署阶段。

For $VN^r_{1,i}$ in $VN^r_{3\times m}$#遍历增强型拓扑节点矩阵$VN^r_{3\times m}$中待部署主链路主节点$VN^r_{1,i}\ i\in\{1,2,\cdots,m\}$。

从$\overline{N}-\overline{N}^{occopy}$中筛选出剩余资源数量满足$VN^r_{1,i}$部署资源需求的服务器节点构成服务器集合$\overline{N}'$。

For n' in $\overline{N}'$#遍历$\overline{N}'$中的服务器节点。

若$VN^r_{1,i}$部署与n'，计算此时连接n'与$n^r_{1,i-1}$的虚拟链路$VL^{r,i-1,i}_{1,1}$采用最短路径部署时的带宽资源消耗。

从$\overline{N}'$中筛选出虚拟链路部署带宽消耗最小的服务器构成集合$\overline{N}''$。

从$\overline{N}''$中选出节点资源最少的服务器节点作为$VN^r_{1,i}$的预部署节点$n^r_{1,i}$。

对服务器节点$n^r_{1,i}$的资源余量进行预更新，并将$n^r_{1,i}$加入$\overline{N}^{occopy}$。

对相关虚拟链路所映射的物理链路资源余量进行预更新。

###主链路副节点部署阶段。

For $VN^r_{2,i}$ in $VN^r_{3\times m}$#遍历增强型拓扑节点矩阵$VN^r_{3\times m}$中待部署主链路副节点$VN^r_{2,i}\ i\in\{1,2,\cdots,m\}$。

从$\overline{N}-\overline{N}^{occopy}$中筛选出剩余资源数量满足$VN^r_{2,i}$部署资源需求的服务器节点构成服务器集合$\overline{N}'$。

For n' in $\overline{N}'$#遍历$\overline{N}'$中的服务器节点。

计算n'与$n^r_{1,i}$节点间的异构度$g(n',n^r_{1,i})$。

若$g(n',n^r_{1,i})\geqslant HLT$，将$n'$添加到服务器集合$\overline{N}''$。

筛选出$VL^{r,i-1,i}_{3\times 3}$、$VL^{r,i,i+1}_{3\times 3}$中与$VN^r_{2,i}$相关且起、止虚拟节点所部署服务器节点均已确定的虚拟链路，构成虚拟链路集合L''。

For n''in $\overline{N}''$#遍历$\overline{N}''$中的服务器节点。

For l''in L''#遍历L''中的虚拟链路。

若$VN^r_{2,i}$部署于n''时，计算此时虚拟链路l''采用最短路径部署时的带宽资源消耗，并计算l''带宽资源消耗累加和。

将L''中虚拟链路带宽资源消耗累加和作为$VN^r_{2,i}$部署于n''的带宽资源消耗。

从$\overline{N}''$中筛选出部署带宽消耗最小的服务器构成集合$\overline{N}$。

从$\overline{N}$中选出节点资源最少的服务器作为$VN^r_{2,i}$的预部署节点$n^r_{2,i}$。

对服务器节点$n^r_{2,i}$的资源余量进行预更新，并将$n^r_{2,i}$加入$\overline{N}^{occopy}$。

当$VN^r_{2,i}$部署于$n^r_{2,i}$时，对相关虚拟链路所映射的物理链路资源余量进行预更新。

###备份链路备份节点部署阶段

For $VN^r_{3,i}$ in $VN^r_{3\times m}$#遍历增强型拓扑节点矩阵$VN^r_{3\times m}$中待部署备份链路备份节点$VN^r_{3,i}\ i\in\{1,2,\cdots,m\}$。

从$\overline{N}-\overline{N}^{occopy}$中筛选出剩余资源数量满足$VN^r_{3,i}$部署资源需求的服务器节点构成服务器集合$\overline{N}'$。

For n'in $\overline{N}'$#遍历$\overline{N}'$中的服务器节点。

计算n'与$n^r_{1,i}$、$n^r_{2,i}$节点间的异构度$g(n',n^r_{1,i})$、$g(n',n^r_{2,i})$。

若$g(n',n^r_{1,i})\geqslant HLT$，$g(n',n^r_{2,i})\geqslant HLT$，则将$n'$添加到服务器集合$\overline{N}''$。

筛选出$VL^{r,i-1,i}_{3\times 3}$、$VL^{r,i,i+1}_{3\times 3}$中与$VN^r_{3,i}$相关且起、止虚拟节点所部署服务器节点均已确定的虚拟链路，构成虚拟链路集合L''。

For n''in $\overline{N}''$#遍历$\overline{N}''$中的服务器节点。

For l''in L''#遍历L''中的虚拟链路。

若$VN^r_{3,i}$部署于n''时，计算此时虚拟链路l''采用最短路径部署时的带宽资源消耗，并计算l''带宽资源消耗累加和。

将L''中虚拟链路带宽资源消耗累加和作为$VN^r_{3,i}$部署于n''的带宽资源消耗。

从$\overline{N}''$中筛选出部署带宽消耗最小的服务器构成集合$\overline{N}$。

从$\overline{N}$中选出节点资源最少的服务器作为$VN^r_{3,i}$的预部署节点$n^r_{3,i}$。

对服务器节点$n^r_{3,i}$的资源余量进行预更新，并将$n^r_{3,i}$加入$\overline{N}^{occopy}$。

当$VN^r_{3,i}$部署于$n^r_{3,i}$时，对相关虚拟链路所映射的物理链路资源余量进行预更新。

若$VN^r_{3\times m}$中的虚拟节点和$VL^{r,i-1,i}_{3\times 3}\ i\in\{1,2,\cdots,m\}$中的虚拟链路均部署成功，则接受该服务功能链请求$r$，否则拒绝该服务功能链请求。

若接受请求r，则按照预部署方案完成相关虚拟节点实例化和虚拟链路的映射，并完成相关服务器节点和物理链路资源余量的更新。

对上述算法的计算复杂度进行分析：计算增强型拓扑节点矩阵的计算复杂度为m，其中m为请求r中虚拟网络功能的最大数量；计算增强型拓扑节点间虚拟链路存在性矩阵的计算复杂度也为m。主节点、副节点、备份节点部署的三个阶段，计算过程基本相同，其中以备份节点部署阶段的计算复杂度最高，三个部署阶段的最大计算复杂度

可放宽为备份节点计算复杂度的三倍。在备份节点部署阶段，待部署备份节点的最大数量为m；筛选满足资源需求服务器集合的计算复杂度为$|\overline{N}||K|$，其中$|\overline{N}|$为底层物理服务器总数量，$|K|$为服务器资源类型总数量；筛选满足异构度需求服务器集合的最大计算复杂度也为$|\overline{N}|$；遍历待选服务器节点以确定带宽资源消耗最小的服务器节点的最大计算复杂度为$|\overline{N}||\overline{L}|$，其中$|\overline{L}|$为物理链路总数量；因此备份节点部署阶段的最大计算复杂度为$m|\overline{N}|(|K|+|\overline{L}|)$。综上所述，算法1总的计算复杂度为$m|\overline{N}|(|K|+|\overline{L}|)$。

6 实验仿真

6.1 实验设置

算法使用Python实现，运行在Intel Core i7-8700 3.20Ghz、内存为32GB的台式机上。实验拓扑采用数据中心Fat-tree网络拓扑结构［23］，包含128个服务器节点、80个交换机节点以及256条链路（已忽略128条服务器节点到交换机节点唯一的直连链路），每个服务器节点的计算资源为32，每条链路的带宽为32。选取CPU、操作系统和虚拟机管理程序3个维度的特征作为服务器节点的信息特征，每维信息特征均包含2种可选类型，每个服务器节点3个维度的信息特征分别从各自2种类型中随机选取，实验中令$HLT=1$，即服务器间异构的条件为满足至少一维信息特征不相同。实验中设置了8种类型的VNF，分别是4种常用VNF类型代表以及4种租户自定义VNF，相关VNF资源需求系数如表2所示。每个服务功能链请求所需处理的流量从｛1，2，3｝中随机选取，每个请求包含4个VNF，4个VNF的类型从表2中随机选取4种。为了更加明显的展示出支持节点分割与异构冗余备份属性对相关部署方法的重要影响，实验中令所有服务功能链请求中的VNF均支持节点拆分、均需要进行冗余备份。服务功能链请求的到达服从参数为0.04泊松过程，服务功能链的生命周期服从均值为5000个单位时间的指数分布。

表2 VNF资源需求系数

VNF类型	NAT	Firewall	Proxy	IDS	UD_1	UD_2	UD_3	UD_4
计算资源需求/单位带宽	1	2	2	6	1	2	3	4

为了评估本文方法的性能，实验将本文所提的支持节点分割与异构冗余备份的部署方法（deployment method supporting node split and heterogeneous backup， DMNSHB）与以下方法进行了对比实验。用于对比的方法有：不支持节点分割与冗余备份的基准部署方法（baseline deployment method， BDM）；支持节点分割的部署方法（deployment method supporting node split， DMNS ）；不考虑节点异构性的传统冗余备份部署方法（deployment method supporting traditional redundancy backup， DMTB）；以及支持异构备份的异构冗余部署方法（deployment method supporting heterogeneous backup， DMHB）。

在对相关部署方法的抗攻击性能进行测试时进行了如下实验假设：攻击者再对目标服务功能链进行攻击时，以承载服务器节点的某一维特征信息作为攻击目标，持续的进行漏洞挖掘和缺陷探测，成功利用该维度信息特征的漏洞或缺陷所需花费的时间服从均值为2000的指数分布。当攻击者掌握某维度特征的漏洞和缺陷后，再次对同类特征的备份节点进行攻击时所花费的时间可忽略不计，即攻击时间主要花费在漏洞挖掘和缺陷探测阶段。攻击者极具耐心，会持续攻击目标服务功能链的节点，直至服务功能链服务中断。每种部署方法下实验重复多次，并对实验结果进行统计分析。

主要从以下六个方面对比相关部署方法的性能。（1）服务功能链请求接受率；（2）服务器计算资源利用率；（3）服务功能链请求的平均计算资源开销；（4）服务功能链请求的平均带宽资源开销；（5）攻击时间成本与服务功能链失效率；（6）平均攻击时间成本。

6.2 实验结果分析

（1）服务功能链请求接受率。

在本文所设置的实验环境下，相关部署方法的服务功能链请求接受率情况图10所示。从图中可以看出，相比于不进行冗余备份的BDM、DMNS部署方法，进行冗余备份的DMTB、

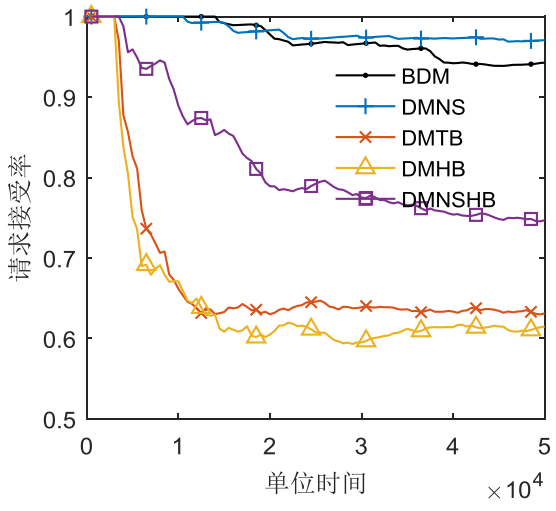


图10　服务功能链请求接受率

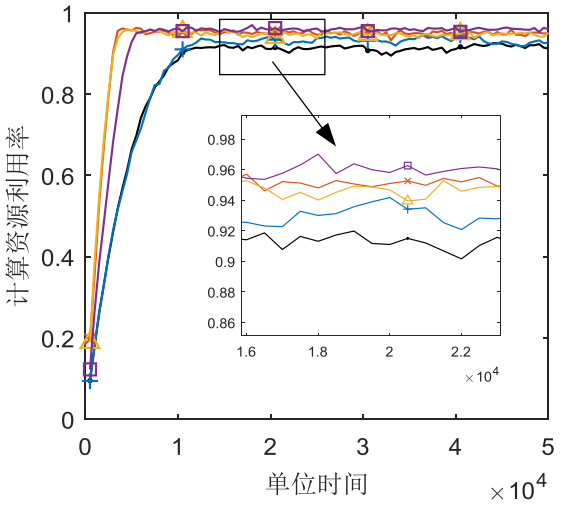


图11　服务器计算资源利用率

DMHB、DMNSHB 部署方法由于在提供冗余备份时需要消耗大量底层物理资源，导致请求接受率急剧下降。在支持冗余备份的部署方法中，本文所提的支持节点拆分 DMNSHB 方法，相比于不支持节点拆分的 DMTB、DMHB 在提高了约 12% 的请求接受率，非常明显的展示出了支持节点拆分属性的重要性。此外在不支持冗余备份的部署方法中，支持节点拆分的 DMNS 相比于不支持节点拆分的基准部署方法 BDM 在请求接受率方面也有一定的提高。这是由于若部署方法支持节点拆分，则可将需要资源较多的虚拟网络功能节点拆分为资源需求小的虚拟网络功能节点，在底层物理网络中、特别是在负载较高的情况下更容易找到满足条件的可部署服务器节点，降低租户服务功能链请求无法部署而被拒绝的概率。

（2）服务器计算资源利用率。

从图 11 可以看出，在本文所设置的实验条件下，相比于不进行冗余备份的 BDM、DMNS 部署方法，进行冗余备份的 DMTB、DMHB、DMNSHB 相对具有稍高一些的服务器计算资源利用率。而本文所提的支持节点拆分 DMNSHB 冗余备份方法，相比于不支持节点拆分的 DMTB、DMHB 冗余备份部署方法提高了约 2~3% 的服务器计算资源使用率，这也是支持节点拆分属性的作用。

（3）服务功能链请求的平均计算资源开销。

相关部署方法的服务功能链请求的平均计算资源开销如图 12 所示，从图中可以看出提供冗余备份的 DMTB、DMHB、DMNSHB 部署方法相比

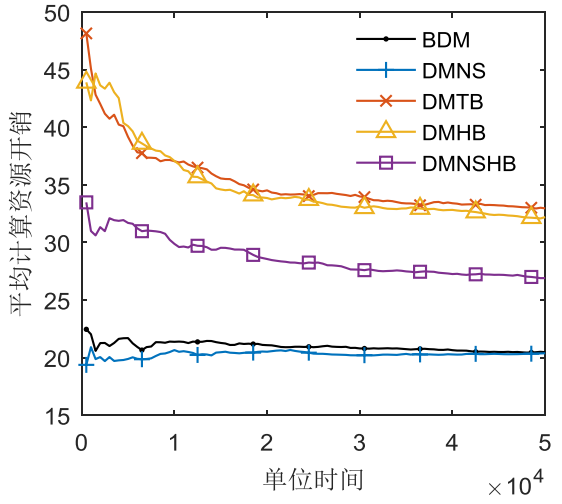


图12　服务功能链请求的平均计算资源开销

于不进行冗余备份的 DMTB、DMHB 部署方法在计算资源开销上有明显的升高，而 BDM、DMNS 提高了近一倍的计算资源开销，而本文所提的支持节点拆分的 DMNSHB 冗余部署方法，相比于 DMTB、DMHB 冗余部署方法降低了约 17% 的计算资源开销。将虚拟网络功能节点拆分为较小的虚拟网络功能节点后，可以用较少的资源为拆分后的主、副节点提供共享的冗余备份。

（4）服务功能链请求的平均带宽资源开销。

从图 13 服务功能链请求的平均带宽资源开销图可以看出，提供冗余备份的 DMTB、DMHB、DMNSHB 部署方法相比于不进行冗余备份的 DMTB、DMHB 部署方法在带宽资源开销上同样

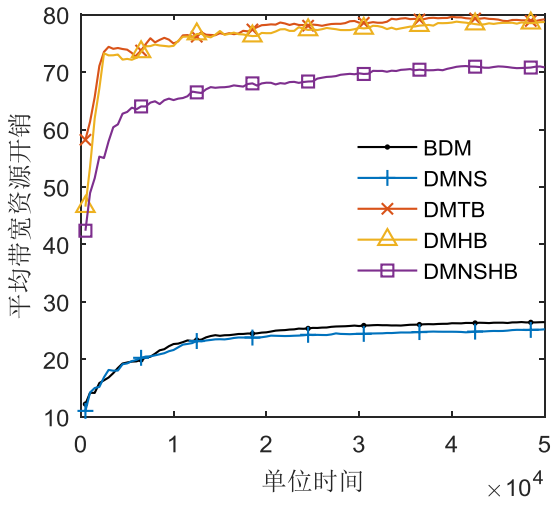


图13　服务功能链请求的平均带宽资源开销

有明显的升高现象。本文所提的支持节点拆分的DMNSHB部署方法，相比于DMTB、DMHB部署方法降低了约10%链路带宽资源开销。

此外从图13还可以看出，随着时间的推移，所有部署方法下的服务功能链请求平均带宽资源开销均出现了升高现象，这是由于随着底层网络负载的升高，伴随着服务功能链请求的随机离开与到达，网络中的可用资源分布比较散乱，不再能将新到达的请求进行较优的部署，需要对现有的已部署的服务功能链进行动态的迁移与调整，才能进一步实现资源的优化使用，该问题的解决方案值得做进一步的深入研究。

（5）攻击时间成本与服务功能链失效率累积分布函数。

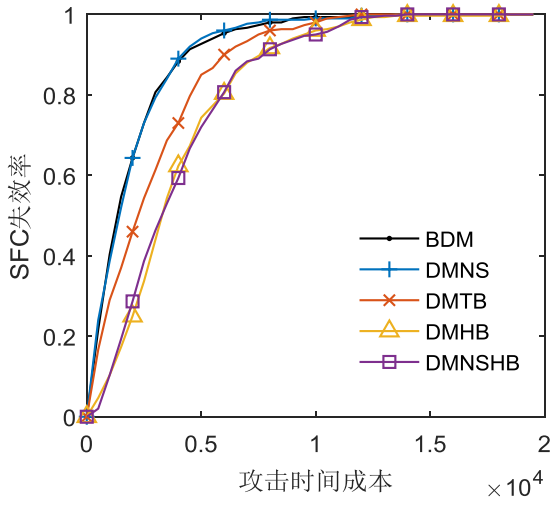


图14　攻击时间成本与服务功能链失效率累积分布函数

本文对相关部署方法下服务功能链的抗攻击性能也行了对比，从图14的攻击时间成本与服务功能链失效率累积分布函数图可以看出，相比于不进行冗余备份的DMTB、DMHB部署方法，提供冗余备份的DMTB、DMHB、DMNSHB部署方法下的服务功能链抗攻击性有明显的提升，而在部署时考虑主、备节点间异构性的DMHB、DMNSHB部署方法抗攻击性能提升最为明显。

（6）平均攻击时间成本。

从图15可以看出，本文所提的考虑节点异构性的DMNSHB部署方法相比于传统冗余备份部署方法DMTB使攻击者的攻击成本提高了约38%，这是由于DMNSHB在为虚拟网络功能节点提供冗余备份时，保证了原节点与备份节点间承载服务器的异构性，避免了主、备节点存在共性缺陷的问题，提高了攻击者的攻击难度和攻击代价。而DMTB部署方法在提供冗余备份时未考虑承载服务器间的异构性，不能保证主、备承载服务器的异构性。

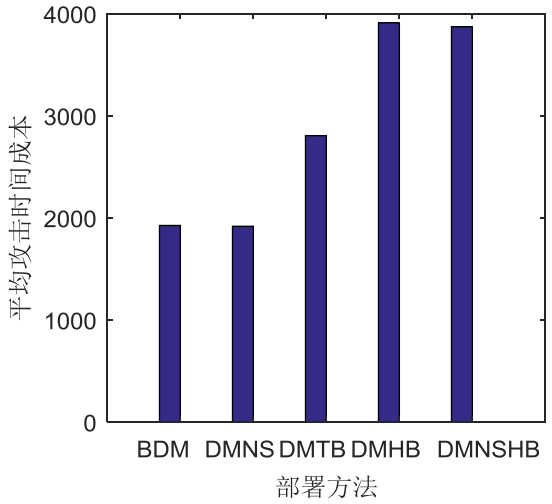


图15　各部署方法下平均攻击时间成本

7　本文小结

本文分析了服务功能链传统冗余备份部署方法存在的问题，由于未考虑承载服务器节点异构性而导致节点间面临共性缺陷，未支持节点可拆分属性而无法更加有效的利用底层资源，由此提出了一种支持节点分割与异构备份的服务功能链部署方法。该方法引入虚拟网络功能节点拆分和

异构冗余备份的思想，在进行服务功能链部署时，对虚拟网络功能节点进行拆分，同时在冗余备份时保证主、备节点承载服务器间的异构性。最后以最小化链路带宽开销为优化目标，建立了支持节点分割与异构备份的SFC部署模型，并设计了基于宽容分层序列思想与贪婪选择的服务功能链部署算法。实验表明，本文所提方法在较大幅度提高服务功能链抗攻击性，提高了12%的服务功能链请求接受和3%的服务器计算资源使用率，降低了17%的计算资源开销和10%的链路带宽资源开销。

参考文献:

[1] Bhamare D, Jain R, Samaka M, et al. A survey on service function chaining[J]. Journal of Network and Computer Applications，2016，75:138-155.

[2] Yi B, Wang X, Li K, et al. A comprehensive survey of Network Function Virtualization [J]. Computer Networks，2018，133：212-262.

[3] Network Functions Virtualisation (NFV)；Use Cases［EB/OL]. https: //www. etsi. org/deliver/etsi_gr/NFV/001_099/001/01. 02. 01_60/gr_NFV001v010201p. pdf. 2017
.

[4] Cotroneo D, De Simone L, Iannillo A K, et al. Network Function Virtualization: Challenges and Directions for Reliability Assurance［C]. IEEE International Symposium on Software Reliability Engineering Workshops. IEEE Computer Society，2014：37-42.

[5] Network Functions Virtualisation (NFV)；Resiliency Requirements［EB/OL]. https://www. etsi. org/deliver/etsi_gs/NFV-REL/001_099/001/01. 01. 01_60/gs_NFV-REL001v010101p. pdf. 2015
.

[6] Joseph D A, Tavakoli A, Stoica I. A policy-aware switching layer for data centers[C]. ACM SIGCOMM 2008
Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Seattle, Wa, Usa, August. DBLP，2008:51-62.

[7] Qazi Z A, Tu C C, Chiang L, et al. SIMPLE-fying middlebox policy enforcement using SDN[C]. ACM SIGCOMM 2013 Conference on SIGCOMM. ACM，2013:27-38.

[8] Barham P, PRAgovic B, Fraser K, et al. Xen and the art of virtualization［C]. ACM SIGOPS operating systems review. ACM，2003，37(5)：164-177.

[9] Open H. Checkpoint/Restore In Userspace: Linux-Prozesse archivieren[J]. 2013.

[10] Rajagopalan S, Williams D, Jamjoom H, et al. Split/Merge: System Support for Elastic Execution in Virtual Middleboxes［C]. NSDI. 2013，13：227-240.

[11] Rajagopalan S, Williams D, Jamjoom H. Pico Replication: A high availability framework for middleboxes［C]. Proceedings of the 4th annual Symposium on Cloud Computing. ACM，2013：1-5.

[12] Casazza M, Fouilhoux P, Bouet M, et al. Securing Virtual Network Function Placement with High Availability Guarantees[J]. 2017:1-9.

[13] Beck M T, Botero J F, Kai S. Resilient allocation of service Function chains［C]. Network Function Virtualization and Software Defined Networks. IEEE，2017:1-6.

[14] Kang J, Simeone O, Kang J. On the Trade-Off between Computational Load and Reliability for Network Function Virtualization［J]. IEEE Communications Letters，2017，PP(99)：1-1.

[15] Jajodia S, Ghosh A K, Swarup V, et al. Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats［M]. Springer Publishing Company，Incorporated，2011.

[16] 季新生, 赵硕, 艾健健, 等. 异构备份式的虚拟网映射方法研究[J]. 电子与信息学报，2018，40(05):1087-1093.

[17] 网络空间拟态防御原理——广义鲁棒控制与内生安全.［M]. 科学出版社，2018.

[18] 丁绍虎, 谢记超, 张鹏, 等. 基于风险感知的关键虚拟网络功能动态迁移方法[J]. 通信学报，

[19] SEXTON J, STORLIE C, and NEIL J. Attack chain detection［J]. Statistical Analysis and Data Mining: The ASA Data Science Journal，2015，8(5-6)：353－363.

[20] 仝青, 张铮, 张为华, et al. 拟态防御Web服务器设计与实现[J]. 软件学报，2017(4).

[21] Sahhaf S S , Tavernier W , Rost M , et al. Network service chaining with optimized network function embedding supporting service decompositions[J]. Computer Networks，2015，93(P3):492-505.

[22] 汤红波, 袁泉, 卢干强, 等. 一种支持节点分割的vEPC虚拟网络功能部署模型[J]. 电子与信息学报，2017，039(003):546-553.

[23] Li D, Hong P, Xue K, et al. Virtual Network Function Placement Considering Resource Optimization and SFC Requests in Cloud Datacenter［J]. IEEE Transactions on Parallel and Distributed Systems，2018，29(7):1664-1677.

[作者简介]

丁绍虎（1979—），男，博士，副研究员，主要研究方向为网络安全。

谢记超（1993—），男，硕士，助理研究员，主要研究方向为网络安全、网络功能虚拟化。

陈 博（1989—），男，硕士，助理研究员，主要研究方向为网络安全、网络体系结构。

胡 涛（1993—），男，硕士，博士生，主要研究方向为软件定义网络。

刘迪洋（1995—），男，学士，硕士生，主要研究方向为网络鲁棒性优化。

On Distributed Object Storage Architecture Based on Mimic Defense

YU Haiyang, LI Hui, YANG Xin, MA Huajun

Shenzhen Graduate School, Peking University, Shenzhen 518055, ChinaShenzhen Key Lab of Information Theory & Future Network Arch., Shenzhen 518055, China

Key words: distributed object storage system; mimic defense; data security

Abstract. With the advent of the era of big data， cloud computing， Internet of things and other information industries continue to develop. There is an increas- ing amount of unstructured data such as pictures， audio and video on the Internet. And distributed object storage system has become the mainstream cloud storage solution. With the increasing number of distributed applications， data security in distributed object storage system has become the focus. For distributed object storage system， traditional defenses are means that fix discovered system vulner- abilities and backdoors by patching， or means to modify the corresponding struc- ture and upgrade. However， these two kinds of means are hysteretic and hardly deal with unknown security threats. Based on mimic defense theory， this paper constructs the principle framework of distributed object storage system， and in- troduces the dynamic redundancy and heterogeneous function in the distributed object storage system architecture， which increases the attack cost， and greatly improves the security and availability of data.

1 Introduction

With the rapid development of the global economy and continuous technological innovation， cloud computing， big data， artificial intelligence， etc. are booming. Among the growing Internet data， the growth of unstructured data is particularly significant， such as audio， video， and pictures. More and more applications and enterprises use ob- ject storage， and "object storage" has become very popular.

Object Storage Service （OSS）， also called object-based storage， is a method of solv- ing and processing discrete units， and can provide data storage services based on objects in distributed systems. Object storage supports REST （Representational State Transfer） or SOAP （Simple Object Access Protocal） style storage interface to provide external storage service ［1］， which can be directly used for application.

With the widespread use of distributed object systems， the security of the system becomes increasingly important. Accidents such as data leakage， data tampering， and data loss frequently occur in object storage systems， causing incalculable losses to in- dividuals and system service providers ［2］.

Mimic Defense ［3］ is an active defense behavior， because its field of action is the cyberspace security field， so it is also called Cyber Mimic Defense （CMD）. The mimic defense is a theory put

forward by Wu Jiangxing， an academician of the Chinese Acad- emy of Engineering， to prevent serious threats to the system caused by "unknown loopholes" by imitating the "mimic phenomenon" in the biological world［3］.

Dynamic Heterogeneous Redundancy ［3］（DHR） is a principle method for mimicry defense. It is based on the Dissimilar Redundancy Structure（DRS） in the reliability field and introduces multi-dimensional dynamic redundancy. It has the high reliability of DRS and high security of mimicry at the same time.

The foundation of DHR architecture is heterogeneity. The greater the difference in attributes between the executor sets， the less likely to have the same loopholes and the stronger the defense capabilities. Redundancy technology is a technology that uses component parallel models to improve system reliability. Through the combination of redundancy and heterogeneity， attackers who rely on a specific environment and plat- form cannot easily break the system. The dynamic transformation mechanism adds dynamics on the basis of redundancy and heterogeneity. When one or more isomers are breached， after receiving the warning message， the dynamic transformation mechanism will regenerate new isomers to replace the currently attacked isomers. The information cannot be reproduced due to previous attacks and the environment. Then all the work the attacker did before will be wiped out.

Erasure Coding （EC）， a data redundancy mechanism in distributed systems， has been widely researched and applied in data processing and other fields due to its high storage utilization and strong data availability ［4］. A （k， n）-threshold erasure code splits the original data into k parts， then generates n （$n > k$） slices using a complex encoding algorithm， and finally stores them in different nodes. The original data can be recovered by using any k'（$k' \geqslant k$） slices. When $k' = k$， this erasure code is a maximum distance

separable（MDS） code， which has the property of optimal storage utilization. In partic- ular， Reed-Solomon code is the most widely studied coding scheme that satisfies the characteristics of MDS in history. It has been used in actual large-scale distributed stor- age systems， such as Microsoft Windows Azur ［6］， Ceph ［7］， HDFS ［8］ etc.

This paper proposes a distributed object storage framework based on mimic defense and multiple ECs. In the implementation process， the log information of the blockchain storage system is used to achieve the purpose of log immutability and traceability， and the consistent hashing is used ［10］ Efficiently distribute and route data.

The remaining part of this paper is structured as follows. The second part introduces the distributed object storage system model based on mimic defense. The third part introduces the system architecture design. The fourth part introduces the functional module design of the system， and the fifth part analyzes the system through experi- mental simulation. The sixth part gives the final conclusion and future work.

2 System model

Distributed object storage systems usually store object data and object metadata sep- arately， thereby separating the control flow and data flow of the system， so that the system has high throughput and high scalability ［9］.

2.1 Mimic distributed object storage model

Distributed object storage architecture generally consists of three parts：Client， Ob- ject-based Storage Device， and Metadata Server. Among them， the client provides users with a simple and easy-to-use storage service platform， and interacts with metadata servers and object storage devices. The metadata server is responsible for the storage and management of object metadata， and provides functions such as object location service and permission access control for clients. The object storage device is the core of the object storage and is responsible for managing persistent objects. When the object is stored， the device provides an object access inter-

face for the client and completes data reading and writing for the client.

The mimic distributed object storage model adds multiple erasure codes， dynamic random transformations and other functions， and uses multiple redundancy and active defense mechanisms to increase the redundancy and the uncertainty of data storage， so as to respond to attacks and improve the system security.

In the system， data management and metadata management are separated. The mon- itor is added to the system. And data management nodes and metadata management nodes are dynamically configured. The system provides users with mimic object service interface. The system model is shown in Fig 1. Among them， the mimic object service interface realizes redundancy and heterogeneity， and the monitor realizes the dynamic configuration.

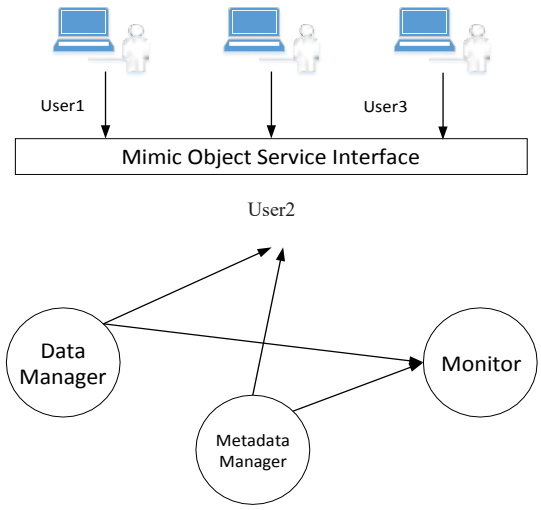


Fig 1 mimic distributed object storage model

2.2 Storage model characteristics

The distributed object storage system that incorporates mimic features has the fol- lowing characteristics：

1 Redundancy

The system adopts a data redundancy strategy that combines erasure codes and mul- tiple copies. For object data， the system adopts a redundancy strategy of multiple eras- ure codes， which can improve hardware utilization and data reliability. For object metadata， the system adopts a multi-copy redundancy strategy.

2 Heterogeneity

At the software level， the system realizes the heterogeneous characteristics of mim- icry through data redundancy. When the system stores data， it uses multiple erasure codes to encode the data. At the hardware level， data nodes and metadata nodes use hardware devices with different architectures， different operating system platforms， and different operating system versions. Use physical dissimilarity to increase the hetero- geneity of the system and prevent rapid attacks on a single vulnerability.

3 Dynamic configuration

The system mainly realizes the dynamics of mimic defense through the monitor module. The monitor module is responsible for monitoring the data nodes and metadata nodes， updating the status of the data nodes and metadata nodes according to the traces of the attacker， and can actively take them offline when necessary， and send data mi- gration instructions. After the node is successfully offline， the administrator needs to repair the node to eliminate security risks. Nodes in safe can be online at any time and added to the running system. In addition， when using erasure code to encode data， the parameters of erasure code are also dynamically changed， which increases the uncer- tainty of the system and further improves the security of the system.

4 Low storage cost

Compared with the multi-copy strategy， erasure code has the characteristics of smaller memory usage， higher fault tolerance， and smaller repair bandwidth. The sys- tem adopts multiple erasure codes as the realization of redundancy and heterogeneity， which has high space utilization and strong fault tolerance. Uploaded files are encoded by erasure

code， and data is stored in units of code blocks. The fine-grained data dedu- plication effect is provided， which further improves the space utilization rate.

5 Log tamper resistance

The log information is stored in the blockchain in the system to ensure that the log information cannot be tampered with. And the system can make corresponding strate- gies based on the log audit results to improve the security and performance of the sys- tem.

3 System architecture design

The system integrates mimic defense mechanism， its architecture is shown in Fig 2. The system is mainly divided into three parts： mimic interface service layer， metadata service layer， and data service layer. Each layer of service is composed of multiple service nodes， supports horizontal expansion， and fundamentally solves the problem of data capacity. Metadata service layer and data service layer are completely

transparent to users.

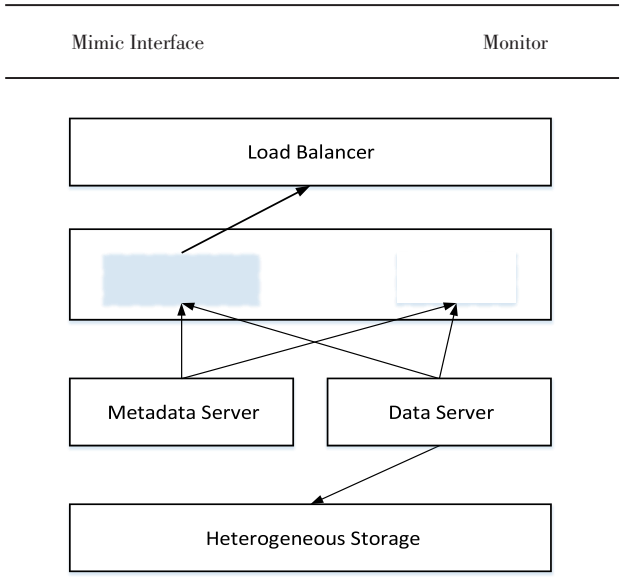


Fig 2 System architecture

3.1 Mimic interface service

There are three states of service nodes of mimic interface service layer： executor， monitor and candidate. The executor， monitor and candidate can all be multi-nodes. The executor provides REST-based HTTPS interface services and is responsible for processing client requests forwarded by the load balancing layer. The monitor is re- sponsible for monitoring the status information of the metadata service and data service nodes， and detecting whether they are attacked and whether there is data loss. And take the initiative to take the node offline and other functions. All data nodes in the system are distributed to monitors. Each monitor is responsible for managing the status of some nodes， and each data node is uniquely managed by one monitor. Each monitor not only needs to save the state of the data nodes that it needs to manage， but also needs to save the data nodes managed by other monitors. When a new monitor is added or an old monitor exits， the topology of the data node managed by the monitor in the system can be dynamically updated.

Every monitor needs to have at least one candidate. When a monitor’s service is unavailable， one of the monitor’s candidates can become a new monitor and continue to provide services. Candidates is to ensure high availability of monitors.

The mimic interface service perceives data nodes and metadata nodes through heart- beat information. The consistent hash is adopted to locate and route data information and metadata information through two hash rings respectively. The data transmission

between the mimic interface service， metadata service and data service is through HTTPS protocol to ensure the security.

3.2 Metadata service

The metadata service layer is composed of multiple metadata service nodes and is responsible for managing the metadata information of the file， such as the creation time， the permission information， and the erasure code information.

3.2 Data service

The data service layer is composed of multiple data service nodes， and the bottom storage can be mounted with different storage clusters such as IPFS and CephFS， mak- ing the bottom heterogeneous

and replaceable. What the data service stores is not the entire file， but the coded block information after the erasure code encode.

4 Functional module design

4.1 Mimic feature function module

1 Dynamic configuration

In the mimic interface service layer， the monitor internally maintains all the states of the data service layer and the metadata service layer. The status parameters include the number of times the node has been attacked， the number of data loss， and the number of data tampering. And according to these states to replace the data service node and metadata service node. The monitor will also randomly select the object file in the sys- tem to check whether its data block is lost or tampered with. If it exists， it will be rec- orded in the corresponding data service node status， and the object will be restored. At the same time， the monitor will also record the attack information of the data node or metadata node. This attack information is an important indicator of the offline of data service nodes and metadata service nodes. The system also uses erasure coding as data redundancy strategy. In addition， each monitor has at least one candidate. When the monitor receives an attack or fails， one candidate will be selected as the new monitor to provide services to ensure high availability. Among them， the process of selecting a new monitor from the candidates is the same as the process of selecting the master in Raft［11］.

2 Redundant, heterogeneous function module

When constructing a distributed object system based on mimic defense， the problem that needs to be solved is redundancy and heterogeneity. The heterogeneity of the sys- tem is mainly used in fault-tolerant mechanisms. In distributed systems， the commonly used fault-tolerant mechanisms include multiple copy and erasure codes. In the system， for object metadata， a three-copy redundancy strategy is used， that is， three copies of metadata information are stored. And for object data， an erasure code is used. In the system， a variety of erasure codes are constructed into an erasure code pool. When the object file is stored， three erasure codes are dynamically and randomly selected in the erasure code pool to encode them to realize the heterogeneous storage. In addition， the bottom layer of the data service node can store data through local file system or mount- ing a cluster.

In addition， physical heterogeneity can also be used to increase heterogeneity. Phys- ical heterogeneity refers to the use of hardware devices with different architectures for data storage nodes and metadata storage nodes， different operating system platforms， and different operating system versions. Use physical dissimilarity to increase the het- erogeneity of the system and prevent rapid attacks on a single vulnerability.

3 Dynamic selection algorithm

When storing objects， the system dynamically selects three erasure codes in the eras- ure code pool to encode the content of the object. And the parameters of erasure codes， such as the number of blocks， the number of coded blocks， and weights also change dynamically. But when size of object is too small， it uses three-copy redundancy strat- egy to store the object. As the object size increases， the number of code blocks and data blocks will also increase.

Among all erasure codes， the decoding efficiency of different erasure codes is dif- ferent. So different erasure codes have different weights. In the process of selecting erasure codes， the weight of erasure codes with higher encoding and decoding efficiency is set to be larger， so as to make them more likely to be selected.

4.2 Blockchain log module

The main characteristic of the blockchain is that it is open and non-tamperable. The blockchain that records logs in the system is a private chain， which is only open to users and system modules， and is invis-

ible to other applications. The log collection module collects and converts the log generated in the system， and then stores it in the log storage unit and the blockchain storage unit. The log module structure is shown in Fig 3. The

log information includes service type， log level， operation time， source operation object， target operation object， and operation details.

When the system is running， the log collection module is responsible for collecting the log information of the log source， and then convert the log format and store it in the log storage module. At the same time， the log collection module will store batch logs in the blockchain storage unit. The log query and analysis unit queries the log information from the log storage unit， puts the result information into the result set， randomly selects part of the data from the result set to verify the operation in the blockchain stor- age unit， and marks the failed log as untrusted logs.

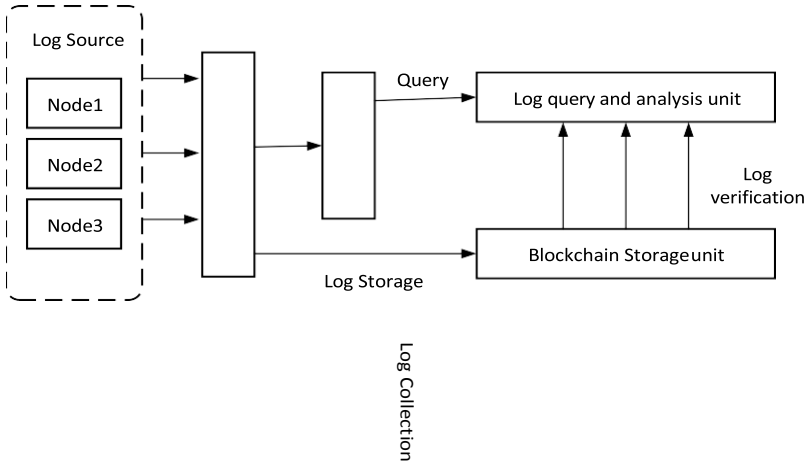


Fig 3 Log module structure

The system can make corresponding strategies based on the results of the log query and analysis unit to improve system security. If the attacker illegally tampered with the data or deleted the data， it would be recorded in the log and sent corresponding instructions to the monitor to restore and re-store the data. If the security of some data nodes is lower than a certain threshold， the monitor will take measures such as data migration and reconstruction of isomers. In addition， through log statistics， the system can also find data with a higher frequency of access， and add these data to the cache， which can improve the performance of the entire system.

4.3 Data positioning

In distributed systems， the consistent hashing［10］ solves the problem of data distri- bution and routing in dynamic network topology. The system uses the consistent hash- ing to achieve efficient data location operations.

Two hash rings are used in the system to manage data information and metadata information. The system will encode the uploaded object， store all the encoding blocks in the data node， and store the whole metadata information of the object in the metadata node in three copies.

1 Upload

When receiving an object upload request sent by the client， the load balancing layer distributes the client request to the interface service layer， and the executor is responsi- ble for processing the request. First， three erasure codes are selected through a dynamic selection algorithm. And erasure codes selected encode the uploaded object， and the encoded code block is stored to the corresponding data node through data location， and then the metadata infor-

mation of the object is stored in the corresponding metadata node in the form of three copies， and finally returns the result information of the object upload request to the client. The object upload process is shown in Fig 4.

Data positioning is divided into two parts： data and metadata. 1） Data. The system stores the coded blocks encoded by the erasure code to the data node respectively. When storing， each code block does the following operations： first calculate the hash value of each code block， then find the smallest data node larger than the hash value according to this hash value， and finally store the code block to the data node. 2） Metadata. The system stores the metadata in three metadata nodes. When storing， at first the system calculates hash value according to the object ID， and then maps its hash value to the hash ring according to the pseudo-random algorithm， selects three metadata nodes of hash ring. The metadata information is respectively stored in the three metadata nodes.

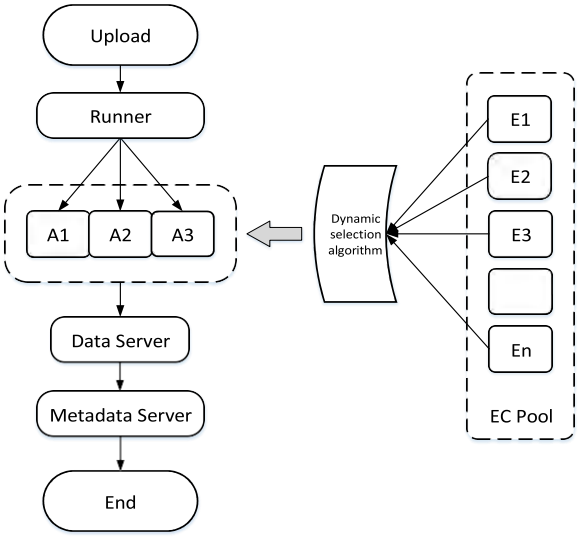


Fig 4 Upload Object

2. Download

When receiving the object download request， the load balancing layer distributes the request to the interface service layer. The executor is responsible for processing the request. First， the system calculates the hash value according to object ID， maps the hash value to the hash ring of metadata through a pseudo-random algorithm， and obtain the metadata information from the first metadata node. Then obtain the hash values of all the coded blocks according to the metadata information， maps the hash values to the hash ring， select the smallest data node greater than the hash value as the target node， and obtain the coded block information from the target node. Three pieces of object information are decoded according to the corresponding erasure code and parameter information， and finally the object information is returned to the client according to the principle of multi-value judgment.

4.4 Data migration

When a data node performs online and offline operations， the data needs to be mi- grated.

1 Offline

Data migration will occur in the following scenarios： the security of a data node is reduced due to multiple attacks， or the data node is unavailable due to hardware and other reasons. For this scenario， it is necessary to do some offline operation on the data node， but the data on the data node is not lost. The system only needs to do data migra- tion.

During data migration， you need to find the next data node of this data node clock- wise in the hash ring. You may as well set the node to be migrated as A and the target node to be found as node B. In this case， you only need to migrate all the data in node A to node B.

2 Online

When a new data node is added to the system， data migration is also required.

When a new node is added， it is necessary to map the node to the hash ring， and then take the mapping point as the starting point， and the first node found in clockwise di- rection is the target node. We may as well set the target node as A and the newly added node as B. At this time， we need to migrate part of the data in node A （the mapping on the hash

ring falls on node B） to node B.

4.5 Data recovery

Data recovery can be divided into the following two scenarios：

1 Pieces of data loss

At this point， only one data block is lost. In view of this situation， the system has two kinds of data repair strategies. 1） When the user downloads an object， the system finds that a data block is lost. At this time， the system can use erasure code to recover the data， and store the recovered data block back to the data node； 2） In addition， the monitor in the system will regularly and randomly select some objects to check whether there is data loss or tampering with the object. If so， it will recover data with the char- acteristics of erasure code， and store the recovered data on the data node again.

2 Partial or total data loss

At this time， some or even all of the data blocks is lost. The data node in the system not only stores the data of this node， but also stores the metadata information corre- sponding to the data of this node and the metadata information of the previous node. The metadata information of the data stored in this node can be obtained from the next node. According to the metadata information， the whole metadata information can be queried in the metadata service node， and the data can be recovered by erasure code， and the recovered data will be stored in the node again. If the data node needs to be offline， the subsequent operations are the same as data migration.

5 Experiment

5.1 Experimental platform

All codes are implemented in the C++ programming language. An experiment plat- form is four machines with Intel（R） Xeon（R） Silver 4116 CPU running at 2.10GHZ with 16GB RAM， and the operating system is Ubuntu Server 18.04.

5.2 Performance analysis

The system uses the LoadRunner to upload and download on system performance， and calculates the upload and download rates of files under different client numbers and data sizes. As shown in Fig 5， Fig 6.

After analysis of the data in Fig 5 and Fig 6， the following conclusions can be drawn：

1 When the number of threads downloading files on the client side is constant, the upload and download rates of files increase as the file size increases. When

the file size is larger than 128MB， the file download rate growth rate tends to be flat.

2. When the file size is constant， the file upload and download rate increases as the number of threads downloading files on the client side increases.

3. The client can increase the file upload and download rate by reasonably seg- menting the uploaded data. At the same time， the file upload and download rate can also be improved by expanding the number of server nodes and improving the performance of server hardware.

When files are uploaded and downloaded， there is an erasure code encoding and decoding operation. This part will have a certain performance loss. When the file is

downloaded， the executor will download multiple data for multi-value judgment， and this part also has some performance loss.

5.3 Erasure Code

In addition to the erasure code of jerasure， BRS， BMBR， BMSR is also used in the system. When uploading and downloading files， the system selects three erasure codes to encode the file. Different erasure codes have different weight. Fig 7 shows rate in different erasure codes to encode file. And the size of test file is 300MB.

The system set different weight according to the encoding time shown in Fig 7. In the process of selecting erasure codes， the weight of erasure codes with

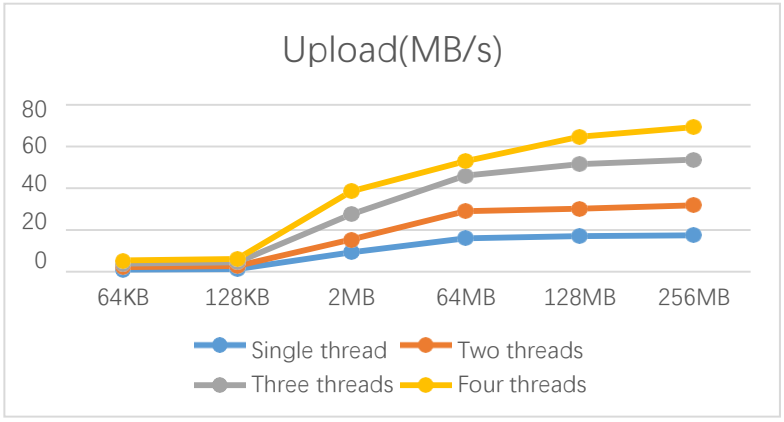


Fig 5 Upload rate under different number of threads and file size

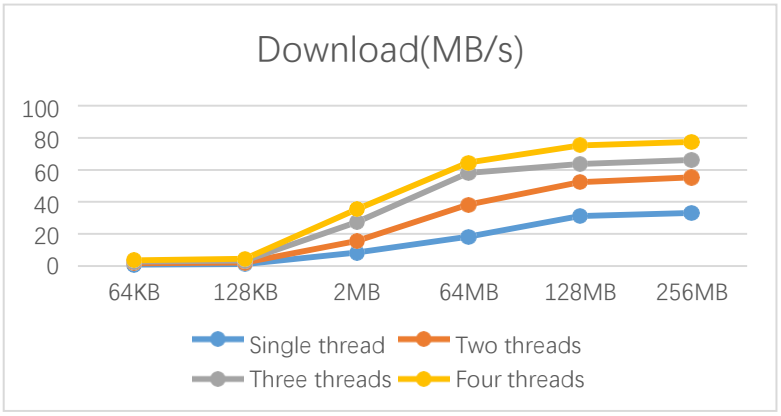


Fig 6 Download rate under different number of threads and file size

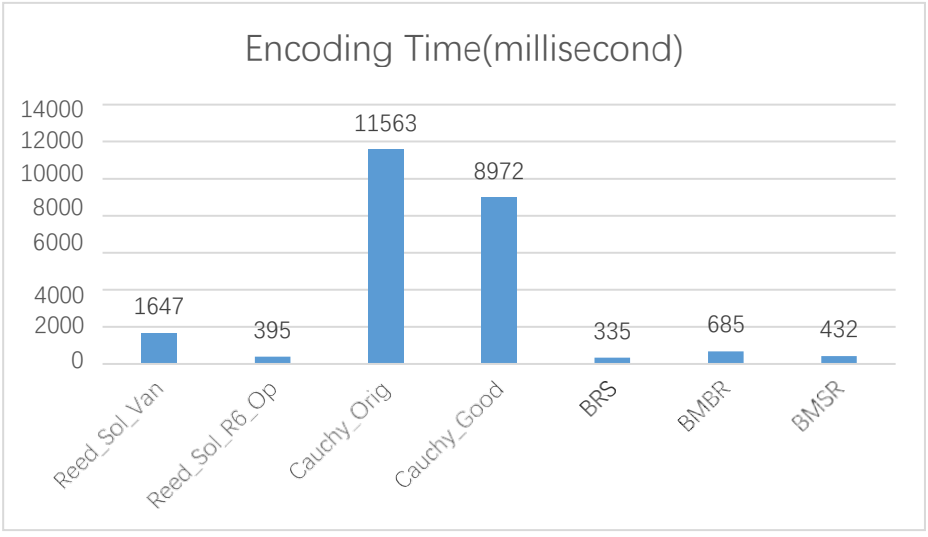


Fig 7 Encoding time of （3，3）-threshold erasure code.

higher encoding and decoding efficiency is set to be larger， so as to make them more likely to be selected.

5.4 Data reliability analysis

Assume that the average availability of each node is u， the probability that the node can work normally. For the system model proposed in this arti-

cle， when the object is uploaded， the system will select a variety of erasure codes for encoding operation， as shown in Fig 4. And the erasure code in the system are all erasure code with MDS characteristics. Assuming that the uploaded object is encoded by erasure code， a total of *n* blocks are generated， of which there are *m* data blocks and *k* coding blocks， that is， $n=m+k$. Then if and only if the number of unattacked nodes in the system is less than

k， that is， at most *k-1* nodes in the system are available， the attacker can attack success- fully， and the original data will be lost. Through calculation， it can be concluded that the probability of the attacker's success is：

$$\sum_{=0}^{-1} (1-)^{-} \quad (1)$$

$$1-\sum_{=0}^{-1} (1-)^{-} \quad (2)$$

5.5 Failure repair analysis

When a node in the system fails due to an attack， in order to maintain the redundancy of the system， it is necessary to download data from the node that has never failed， restore the lost data and store it in the corresponding node. This process is called repair process. Replication technology can support fast data recovery. When a data block is lost， the system can directly copy the lost data from other nodes and store it on the corresponding node. For erasure code， the repair process requires certain network resources and computing resources. In the repair process， firstly download data from *k* storage nodes， reconstruct the original data， re-encode and store the missing data block on the corresponding node. Therefore， the system needs to take into account the following points： 1） high node repair efficiency and support for fast and reliable data repair to shorten the time of the system being attacked； 2） node repair cost is small， less net- work resources are consumed in the repair process， and system repair costs are reduced.

6 Conclusions and future work

Based on the distributed object system， this paper incorporates the DHR mechanism， and proposes a mimic distributed object storage system architecture. By increasing the uncertainty of the system， the supply cost of the attacker's attack on the system is in- creased， so that system vulnerabilities and backdoors are difficult to be exploited and triggered， effectively improving the security of the system. In the system， the dynamic update of the status monitoring module， erasure code encoding and decoding calcula- tion， data node and metadata node status， etc.， increases the system's request response time and has a certain impact on performance. The future work is to establish an attack model for system verification， and conduct further research on various functional mod- ules and mechanisms to improve the system architecture， so as to gradually change the status quo of asymmetric attack and defense costs and passive defensive methods.

References:

[1] Halili F, Ramadani E. Web services: a comparison of soap and rest services[J]. Modern Applied Science，2018，12(3)：175.

[2] Bouleghlimat I, Hacini S. Big Data Processing Security Issues in Cloud Environment: Pro- ceedings of the 3rd Conference on Computing Systems and Applications[M]. Advances in Computing Systems and Applications. 2019

[3] WU Jiangxing "Research on Cyber Mimic Defense，" Journal of Cyber Security，Vol. 1，No. 4，Oct. 2016.

[4] Balaji S B, Krishnan M N, Vajha M, et al. Erasure coding for distributed storage: An over- view[J]. Science China Information Sciences，2018，61(10)：100301.

[5] Li J, Li B. Erasure coding for cloud storage systems: a survey[J]. Tsinghua Science and Technology，2013，18(3)：259-272.

[6] Huang C, Simitci H, Xu Y, et al. Erasure coding in windows azure storage [C]//Presented as part of the 2012 {USENIX} Annual Technical Conference ({USENIX}{ATC} 12). 2012：15-26.

[7] Weil S A, Brandt S A, Miller E L, et al. Ceph: A scalable, high-performance distributed file system [C]//Proceedings of the 7th symposium on Operating systems design and imple- mentation.

2006：307-320.

[8] "What Is Apache Hadoop?" http://hadoop. apache. org/,Aug. 2017.

[9] Mesnier M, Ganger G R, Riedel E. Object-based storage[J]. IEEE Communications Mag- azine，2003，41(8)：84-90.

[10] Karger D, Lehman E, Leighton T, et al. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web [C]//Proceedings of the twenty-ninth annual ACM symposium on Theory of computing. 1997：654-663.

[11] Ongaro D, Ousterhout J. In search of an understandable consensus algorithm [C]//2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14). 2014：305-319.

[12] Stoica I, Morris R, Karger D, et al. Chord: A scalable peer-to-peer lookup service for in- ternet applications [J]. ACM SIGCOMM Computer Communication Review，2001，31(4)：149-160.

About the authors

YU Haiyang is now studying for a master's degree in Computer Application Technology at Peking University Shenzhen Graduate School. The research field is big data application and dis- tributed storage. Research interests include： cloud computing， cloud storage， blockchain. （Email： 1801213410@pku. edu. cn）

LI Hui is currently a Full Professor with Peking University， Shenzhen Graduate School. He is also the Double hired Professor of the Peng Cheng Laboratory， the Director of the Shenzhen Key Lab of Information Theory & Future Internet Architecture， PKU Lab of CENI （CENI： China Environment for Network Innovations）， National Major Research Infrastructure， Shenzhen En- gineering Lab of Converged Networks. His research interests include future network architecture， cyberspace security， distributed storage， and blockchain. （Email： huilihuge@163. com）

YANG Xin received the B. Eng. degree from the Department of Computer Science and Engi- neering， South China University of Technology， in 2016. She is currently pursuing the Ph. D. degree with the School of Information Science， Peking University. She is also the student of the Peng Cheng Laboratory. Her research interests include cyber security， future network architec- ture， and distributed storage systems. （Email： yangxin2016@pku. edu. cn）

Ma Huajun was born in 1982. He is a postgraduate student of Peking University Shenzhen Graduate School. His research interests include network security and distributed system technol- ogy. （Email： mahj@sz. pku. edu. cn）

数据包分类算法可优化研究综述

——(王兆辉 国家数字交换系统工程技术研究中心,郑州 450001)

摘　要：数据包分类算法种类繁多且设计复杂，各类方案需要根据实际网络环境需要具体分析，通过对传统数据包分类算法的不断优化总结和新体系架构的性能分析，目前学界已经从多种角度对数据包分类算法的性能有了大幅改进，P4的提出更是通过利用自身的可定制属性为高速低功耗处理交换机的设计带来可能，本文综合目前的成熟研究方案分别从硬件、软件和P4高级协议语言等方面对于现在的数据包分类算法和结构设计做了研究分析，并通过对比总结展望下一步包分类算法发展前景。

关键词：TCAM、包分类、P4、前缀树、分解、深度学习

Overview of Research on Optimization of Packet Classification Algorithms

Wang Zhaohui

National Digital Switching System Engineering Technology Research Center, Zhengzhou 450001

Abstract: There are many types of data packet classification algorithms and their designs are complex. Various solutions need to be analyzed according to the actual network environment. Through continuous optimization of traditional packet classification algorithms, summary and new Performance analysis of the architecture, the current academic community has greatly improved the performance of the packet classification algorithm from various angles, and the proposal of P4 makes it possible to design high-speed and low-power processing switches by using its own customizable properties. This paper integrates the current mature research schemes and analyzes the data packet classification algorithm and structure design from three aspects: hardware, software and P4 high-level protocol language. And looking forward to the next step in the development of packet classification algorithms through comparison and summary.

Key words: TCAM; Packet classification; P4; prefix tree; decomposition; deep learning

1　引言

数据包分类在很多网络管理系统中有着十分重要的应用，例如用户访问控制、防火墙设置、服务质量供应和流量实时监管，数据包分类算法的性能通常决定了网络路由器的性能瓶颈。尤其是当前高性能的多协议支持的交换数据网络以及基于可编程数据交换平台的高速交换机都需要高效的数据包分类算法支持，数据包分类算法经过长期的发展已经有了成熟的设计体系，但是随着交换网络体系规模的不断扩大和通信协议的加速更新，不断优化包分类算法以满足多流量多架构的数据交换网络是目前网络体系进步发展必不可少的一步。

通过在旧算法体系的基础上，针对交换网络具体的性能需求和功耗分析，设计具有针对性的包分类算法是对于改进高性能交换机的重要指标之一，本文结合目前最新的和业界使用最广泛包分类算法体系及其衍生架构，整体从硬件、软件和P4高级编程语言编译器对现有数据包分类算法优化和架构方案方进行分析，并结合各个算法体系的优劣势来分析提出个人建议的下一步技术发展的研究方向。

2 包分类算法优化方案

2.1 基于硬件TCAM的包分类优化方案

2.1.1 TCAM架构

基于硬件的TCAM（三态内容可寻址关联存储器）是用于高速数据包分类的标准设备之一[1]，由于其处理速度快和规则管理简单而被广泛应用于数据包分类设计方案。TCAM的包分类查找结构如图1所示。基于TCAM的包分类关键在于规则检查和编码算法，规则有两种类型：分别是为标头的每个字段指定固定值（或是特定前缀范围）的简单规则和范围规则。通常，当源端口或目标端口处于特定间隔时，将应用范围规则。编码范围规则需要几个TCAM条目（被称为范围扩展），因此尽管大多数规则是简单规则[2]，但仍需要大量条目用于编码范围规则，而且随着包转发需求的不断增加使用范围规则的百分比正在不断增长[3]，所以如何有效地执行操作（使用尽可能少的TCAM条目）、如何对单个规则进行编码和寻找一种实用的算法可以为任意范围规则提供最佳编码是可优化研究方向的重点，尽管TCAM可以在一个时钟中比较所有条目，但它仍具有三个主要缺点，即硬件成本高，功耗高以及存储范围数据效率低下，针对基于TCAM的包分类方案的主要问题[4]。下文将从TCAM编码算法和结构优化的角度分析目前的主要优化设计思路方案。

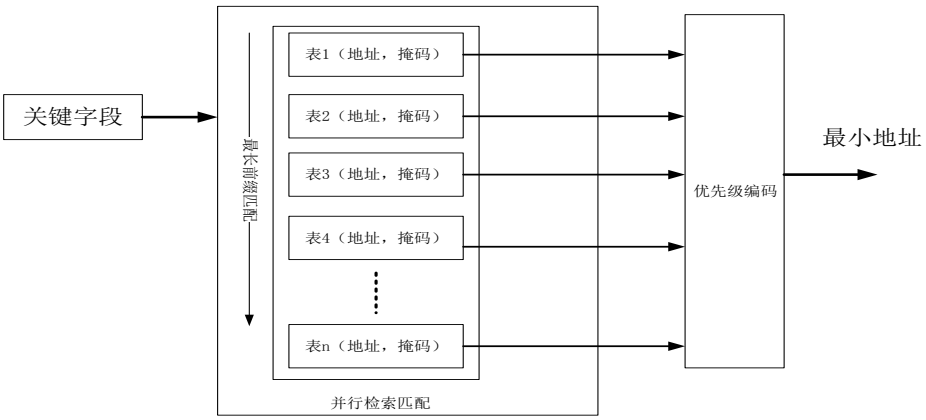


图1 TCAM的包分类查找结构

2.1.2 TCAM编码算法

TCAM编码算法优化是解决TCAM问题方案的核心研究方向，由Weiwen Yu[5]等提出的伪TCAM（Pseudo-TCAM）设计了一种静态随机存取存储器的数据包分类架构。该架构包含一个静态随机存取存储器阵列、一组用于从输入键中提取选定位以形成存储器地址的多路复用器和一个将选定规则与输入键进行比较的比较器组成。Pseudo-TCAM提出了一种新颖的规则映射策略，通过设置一个特殊的比赛单位，来处理部分规则集中那些位于低优先级列表底部附近的且具有短的服务协议/数据协议前缀的规则，然后其余规则映射到常规匹配单元。该映射算法通过使用更动态的2级位选择策略，从而在比特选择过程中有更大的灵活性。Hsin-T sung Lin[6]等提出了一种使用鉴别器和伪规则的TCAM的快速多匹配包分类算法，该方案使用MUD存储原始规则，并选择性地为一些规则生成伪规则。通过检查从伪规则产生的TCAM条目的数量是否小于可用TCAM条目的数量来判定和存储存储伪规则，然后生成搜索关键字对组内的TCAM条目执行多维匹配数据包分类，该算法旨在在有限数量的可用TCAM条目下合并尽可能多的组。

Wang Kai等[7]提出了一种基于空间高效的基

于TCAM的数据包分类方案，称为TCAM-PC，TCAM-PC机可以自适应地选择包含相同源/目的IP地址的不同规则的压缩程度，具体实现方法为首先设计一个选择算法并将其引入TCAM规则匹配过程，利用TCAM的掩码寄存器，禁用不用于执行匹配操作的块，通过压缩嵌入在不同规则中的相同信息部分的空间来减少TCAM内存消耗，并确保基于数据包的预定转发率TCAM中全局掩码寄存器和块掩码寄存器之间的协作。

由Hayotjon Aliev [8]等提出的基于IPsec（网络协议安全）数据包的包分类算法，该算法主要针对IPsec数据包中的数据包分类和加密过程所做的一个硬件加速器体系结构，因为是一个确切的目标网络应用，该架构采用了两个IPsec协议的VPN路由器构建的VPN隧道网络，每个IPsec路由器由出站、入站分类器和过滤引擎组成。通过采用共享的TCAM，该方案允许一个共享TCAM在所有分类规则数据库之间共享单个数据和控制总线。Hui Li [9]等提出的TCAM-SRAM算法是结合了范围编码方案和静态随机存取存储器结构的优点，将TCAM范围存储问题描述为如何找到一组合适的地址长度数量来使得TCAM的使用更加有效。文章中给出了其先进的静态随机存取存储器架构，该架构主要包含两个部分，一个是静态随机存取存储器结构下存储范围的设置，还有一个是并行搜索TCAM以找到最佳匹配规则的算法组成。该方案可以有效提高TCAM的分组分类引擎的效率。

在针对TCAM高发热高功耗的问题上，Eric Torng等[10]提出的解决方案是一种三值统一框架（TUF），并设计了其分类器压缩算法，该算法的核心有两点，第一个如何是表示中间分类器的三进制数据结构，还有一个是组合中间分类器的过程，简单来说就是描述了使用哪种三进制数据结构以及如何将它们组合在一起的问题，该算法具有很大的灵活性，随着目前分类器大小和复杂度的不断增加，其优势相比较其他类型方案[11][12]更具有前沿性。

2.2 基于软件方案的数据包分类算法优化

基于软件的数据包分类算法相对于硬件实现方法具有良好的程序灵活性，可以以较低的实现成本实现。目前的包分类算法研究方向主要集中在流表结构、大小和数量上进行优化和改进[13]，基于决策树和分解以及TSS的技术是比较主流的几种设计思路。

2.2.1 基于决策树的方案

决策树也可以称为前缀树（即trie）是一种的树形数据结构，用于存储动态集或关联数组，其中的树枝键通常由字符串组成。基于决策树的方法多采用启发式方法将空间递归地切成较小的子空间[14]。可以通过压缩数据结构（从而允许使用更小和更快的内存）或减少内存访问次数来加快此类过程[15]。

在决策树结构设计思路上由S.Zheng [16]等提出的基于总前缀长度的高效聚类分组分类算法（TPLBC）。该方案通过将具有相似总前缀长度的元组空间进行聚类，根据前缀总长度首先构建一个优先剪枝二叉树，然后将具有最高优先级的规则被存储到相应优先级AQT的根节点中。该空间被分解成四个大小相等的空间，分解后的空间中包含的规则中优先级最高的规则存储在该空间对应的节点中。重复空间分解，直到每个规则都被存储。按照这种方式，TPLBC算法可以减少所需的访问元组空间的数量并在这些群集中构建基于优先区域的四叉树结构。G.Antichi [17]等提出的JA-trie算法是基于熵的预处理步骤应用和分类规则集，在这里熵与树的数据结构中创建的节点数严格相关。该算法本身是对多比特树算法的一种改进，通过使用一种新的机制来合并通配符条目。核心思想是查找时不直接包含通配符跨度，而是设定一个固定的步幅。如图3所示的JA-trie结构，在此示例中，有四个简单规则，每个规则基于一个IP地址，通过将规则分为不同的8位长步幅来进行合并通配符条目。实验表明该算法可以在最佳的内存占用和数深度最小两种性能指标上达到平衡。

通过用深度学习（DL）构建决策树。H.Jami [18]等提出了一种基于DL的多位查找方案，这种多位查找方案由一个RL系统由一个代理和一个环境组成，代理与环境反复交互。该环境由一组规则和一个决策树组成，如图4所示，查找算法是在数据包头中找到一定数量的位位置（例如SrcIP的第i和j位或DstIP的第m和n位等）以生成位图，并将这些位称为有效位（EB）。通过构造有效位可以将内部节点和叶节点结合在一起。所有的这些

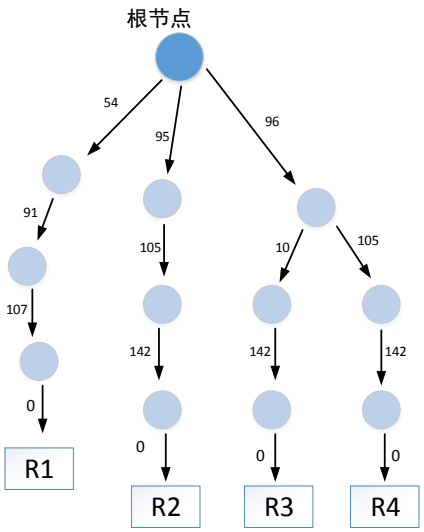


图3　JA-trie结构

有效位级联以形成查找表的索引。该查找表已预先计算并存储在内存中，因此对于传入的数据包，将计算有效位位置中的值，并使用该值作为索引在查找表中进行搜索。该方案可以过滤掉大多数不相关的规则，仅对高度相关的匹配规则进行匹配。

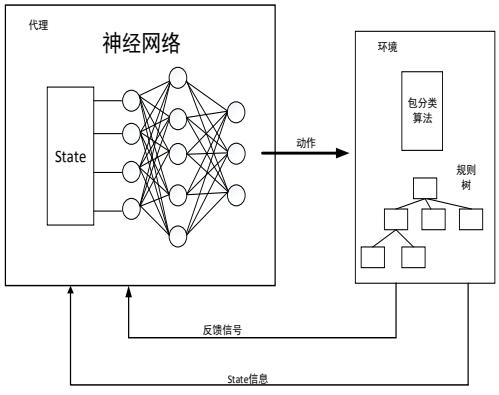


图4　RL系统

2.2.2　基于分解和TSS的优化方案

在基于分解识别的方法中，数据包头分为多个字段，并且对所有单个字段独立执行查找操作。所有字段的部分结果被合并以产生最终的匹配结果[19]。通常，基于分解的方法具有三个阶段：预处理阶段，搜索阶段和合并阶段。基于分解方法的主要优势是可以在搜索阶段探索并行算法和使用各种数据结构。

C. Hsieh[20]等提出的多维切割可伸缩多字段包分类算法MC-SBC，是使用查找表来避免计算差异问题，通过利用大型计算平台中的高效指令来提高系统吞吐量。该方案表明由于规则集中的稀疏性和有规则的分布，MC-SBC旨在使用完全匹配的有效位快速找出一些候选规则，通过利用规则集中的有效位位置，设计具有可行性和灵活性且易于实施的两阶段体系结构。

随着虚拟化技术的发展，数据的云端存储可以在意外情况发生时更好的保证数据的安全性，Open vSwitch是与大多数虚拟机管理程序和容器系统一起使用的一种基于OpenFlow的交换机，是在Linux基金会的Open vSwitch项目[21]中作为开源发布的多层软件交换机，它是专门为解决软件中的OpenFlow分类问题而设计的。

Open vSwitch中使用元组空间搜索分类器（TSS）[22]作为包分类算法对其内核和用户空间进行所有分组分类。TSS可以将流表实现为单个的哈希表。如果控制器随后添加具有不同匹配形式的新流，则分类器将创建第二个哈希表，该哈希表会在这些流中匹配的字段上散列。当Open vSwitch用户空间通过其OpenFlow表处理数据包时，它会跟踪作为转发决策一部分而被查询的数据包字段位。数据包头字段的按位跟踪在使用简单的OpenFlow流表构造兆流条目时非常有效。TSS考虑每个数据包的TCP目标端口。并且每个兆流也将在TCP目标端口上匹配，这将导致端口扫描性能的下降。如何设置有效的在线算法来生成最优的兆流是一个研究的重点。B. Pfaff[23]等将注意力集中在生成更加准确的近似值上。由于未能匹配必须包含的字段会导致不正确的数据包转发，从而使此类错误变得不可接受，因此其研究核心在于寻找近似值在更多必要字段上进行匹配。J. Daly[24]等提出的TupleMerge（TM）分类器是对TSS分类算法的一种优化，通过将来自多个TSS表的规则放到一个哈希表中，从而减少了表的数量，并且不再需要预分类器。

Luyang Xu[25]等结合结合决策树和TSS的优势设计了一种分割排序算法，通过分区排序对数据包进行分类，在这里分区排序的核心是规则集排序，即将一个规则集划分成少量可排序的规则集，然后再将每个可排序规则集存储在一个多键二叉查找树中，来将初始规则集划分为支持高速分类和快速更新的较小规则集。与决策树方法类

似，分区排序算法通过使用平衡搜索树来实现每个分区的数据包快速分类。

2.3 基于P4的数据包分类算法设计

2.3.1 P4协议编程语言

P4作为一种独立于编程协议的特定语言，能够定义自己的标头解决方案，作为一种控制器和网络设备之间的协议处理，其接口提高了对网络进行编程的抽象级别，P4是设计用于对转发平面进行编程的高级语言，它为网络设备所有者提供了以编程方式修改转发平面的能力。此外，它还促进了协议独立性，与OpenFlow不同，协议独立性使得可以根据需要创建标头字段，从而允许在该字段中对设备进行编程[26]。最终用户无需制造商输入即可创建自定义协议和算法的能力，也使他们能够保持对其知识产权的完全控制。

P4有三个主要特性：

可重新配置性：可以在部署交换机之后对分组的解析和处理进行精细定义。

协议独立性：能够使用具有特定名称和类型的新字段定义新的报头，因此可以使用匹配动作表和动作指定分组过程。

目标无关性：用P4编写的程序是与硬件无关的描述，它需要特定的编译器，这些编译器可以根据这些与硬件无关的命令来编译能够在特定设备上运行的硬件相关二进制文件[27]。

2.3.2 基于FPGA的P4定义数据包解析器

P4可以有效地解决OpenFlow交换机中包头字段相关的问题[28]，其数据包处理方式可以总结为匹配表加动作表。使用上层控制程序定义并排列“匹配+动作”表，并利用编译器设定这些表之间的流程。P4程序的其他部分描述了操作，流水线表，标头字段以及数据包应发送到的端口，P4语言被用于解决要求多个OpenFlow版本支持新功能的问题，并最终在FPGA上实现其RTL。该体系结构全面支持数据平面中的任何标头数据包。Abbas Yazdinejad等[29]提出一种基于FPGA的P4数据包处理解析器架构，成功在FPGA平台上使用P4语言中完成了高速数据包解析和处理，相对于传统的基于OpenFlow的流表该方案提高了传输包吞吐率和处理素的，该解析器架构可分为两个部分：

包头定义，P4中标头的定义与C相似。要定义标头，需要先定义数据包头中的字段列表和位的宽度。根据不同的协议类型，需要的参数和字段长度可通过设置变量来量化需要处理的数据包包头信息。以以太网包头为例，在P4描述的包头字段定义中需要定义目的地址（48位），源地址（48位）和协议类型（16位）。

解析器，解析器分为匹配和指令两个阶段，匹配阶段是在包头定义结束后，将提取的字段列表送往状态选择位，对每个字域段提取变量有效值通过使用P4编写的字段匹配通道进行包头匹配，整个字段匹配通道如图5所示。指令阶段是指通过所有字段匹配的包头对应流表动作指令进行包的转发处理操作，该阶段处理与OpenFlow交换机处理流程相近。

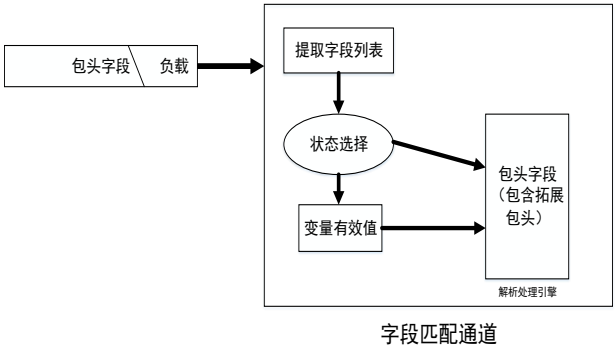


图5 P4字段匹配通道

根据协议无关解析器的设计原理，需要在基于P4的解析器架构的基础上进行编程及代码生成，这里可以使用专用于P4进行编程的编译参考器如P4-hlir[30]，该编译器可以提供目标协议无关的P4解析器。在使用P4-hlir编译P4描述协议无关的数据包解析器时该编译器可以将使用P4语言进行编写后的代码转换为python源码来生成，再送交到高层设计环境中管理数据平面的应用程序。还有一种是P4-RTL[31]编译器，即将P4源码生成RTL级硬件描述语言，使得编译好的抽象化描述语言例化为实际的硬件结构，还有包括p4pktgen[32]等为P4源代码生成测试数据的工具方案，将P4描述语言应用到数据包分类算法的设计是未来设计基于可编程数据平面架构的一个重要方向。

3 优化方案比较及分析

3.1 基于硬件TCAM方案算法性能比较分析

基于TCAM方案所存在的不足，相关方案具

体主要是从如何设计进行范围编码、降低功耗、减少存储和条目更新几个方面本节中我们对前文中提到TCAM硬件包分类方案做比较分析，如表1所示，分析发现目前多数研究机构在基于TCAM的包算法分类优化方案上集中在设计多范围有效编码算法来减少存储降低功耗，目前的分类算法还存在算法本身较为复杂，架构改变较大无法应用在实际工业设计中等问题。

3.2 基于软件方案算法性能比较分析

基于软件的包分类算法设计方案性能分析如表2所示，软件方案主要是构建树结构和分解的思想，基于树结构主要缺陷在于构建树枝节点而产生的资源占用量增加，基于分解的方案主要在于算法结构复杂需要使用多种数据结构，采用深度学习的方案是目前比较新型的包分类优化方案，具有很好的性能表现，但是准确率无法保证100%是目前比较致命的一个不足之处。

表1 基于硬件方案算法性能分析

	优势	不足
Pseudo-TCAM	整体硬件成本更低,并且由于处理逻辑更简单,处理速的更快,支持动态更新,并且在更新过程中不需要暂停分类器。	由于其映射规则较为复杂,软件设计难度较大,更新成本高。
TCAM的快速多匹配包分类算法	消除了冗余的TCAM访问,提高了搜索性能并降低了功耗,由于可以有效地控制被占用的TCAM条目的数量,所以避免了支持具有大量重叠规则的规则集的可行性问题	在内存需求方面占用上较高
TCAM-PC	可以满足较小的存储需求,并且具有较高的存储利用率,可以保证高速数据包转发速率	/
IPsec数据包硬件加速器	有效减少硬件成本和数据延迟	仅针对IPsec协议数据包,具有较强针对性
TCAM-SRAM	通过减少表示所有范围所需的前缀数量来减少TCAM条目的数量	/
三值统一框架(TUF)	压缩算法效果更优,对于多条目处理速度更快	在较少条目的小规则分类器上压缩算法优化效果较差

4 结论和展望

数据包分类算法是设计高速SDN交换机的核心技术之一，本文讨论了数据包分类在算法和结构的优化方案选择，分析了目前各种分类算法的不足并探索了可优化的方向。

本文分别从基于硬件TCAM方案、软件方案和P4方案三个方面去对可优化数据包分类算法进行研究分析，基于硬件方案设计的主要优势在于其高速率解析性能和流表查询速度，但是昂贵的成本和维护费用是该方案的主要缺点，基于软件的设计方案在架构设计灵活性和升级成本上有着一定优势，该方案设计思路集中在于对流量预测、压缩聚合、树优化等，但是在处理速度上不如硬件方案，如何结合软硬件设计是未来对包分类算法结构的可优化设计方向之一。P4是一种可以根据用户需要自定义的协议无关包处理语言，其自身的高度自由性和可定制性可以解决OpenFlow流表空间占用过多的问题，如何在OpenFlow交换机上结合P4的可定制优势去优化流表空间存储结构是提高流表空间利用率和查表效率的研究方向之一。

表2　基于软件方案算法性能分析

	优势	不足
分层智能切割算法(HiCuts)	减小的存储和查询空间,缩短更新数据结构中的规则所需要的时间	大量增加构建决策树的时间,尤其对于一些多规则树算法优化效果有限
JA-trie算法	减少内存大小的占用空间,同时保持很小的树深,产生快速的关键查找时间。	该算法在内存减小占用量上效果不如TSS算法
高效聚类分组分类算法(TPLBC)	减少所需的访问元组空间的数量。并在这些集群中构建有效的优先级AQT,获得良好的空间性能和搜索速度	在规则集较小,所需的内存较小的情况下,TPLBC算法的性能优势并不突出
多维切割可伸缩多字段(MC-SBC)	可实现数据结构以便快速处理和轻松更新,所提出的系统能够随着规则集大小的增加和规则集中的增长字段数量而保持可伸缩性。	在数据包吞吐量上性能表现不佳
元组空间搜索分类器(TSS)	相较于其他软硬件算法,TSS具有网络虚拟化所需的可编程性,具有高度的灵活性和通用性,兼容多平台差异。	分区/表的数量很大,降低包分类速度
TupleMerge(TM)分类器	对TSS分类算法的一种优化,减少需要搜索的表,节省搜索时间,提高数据包分类速度	相比TSS算法会增加冲突数
DL的多位查找方案	提高匹配规则效率,减少前缀树的遍历级数,减小内存占用	转发准确率不能达到100%
分割排序算法	可以对数据包进行快速分类的同时还支持规则的快速更新	/

参考文献:

[1] YangT.,LiuA. X.,ShenY.,FuQ.,LiD. and LiX.，Fast OpenFlow Table Lookup with Fast Update [C]//2018 Proceedings IEEE INFOCOM 2018，Honolulu，HI,2018:2636-2644.

[2] Liu A X, Meiners C R, Torng E. TCAM Razor: A Systematic Approach Towards Minimizing Packet Classifiers in TCAMs [J]. Networking，IEEE/ACM Transactions on，2010，18(2):490-500.

[3] NorigeE.,LiuA. X. and TorngE.. A Ternary Unification Framework for Optimizing TCAM-Based Packet Classification Systems [J]. IEEE/ACM Transactions on Networking，2018,26(2):657-670.

[4] 刘中金,李勇,苏厉,等. TCAM存储高效的OpenFlow多级流表映射机制[J]. 清华大学学报(自然科学版),2014,54(4):437-442.
Liu Zhongjin, Li Yong, Su Li, etc. TCAM storage efficient OpenFlow multi-level flow table mapping mechanism [J]. Journal of Tsinghua University (Natural Science Edition),2014,54(4):437-442.

[5] YuW.，SivakumarS. and PaoD.. Pseudo-TCAM: SRAM-Based Architecture for Packet Classification in One Memory Access [J]. IEEE Networking Letters,2019,2(1):89-92.

[6] LinH. and WangP.. Fast TCAM-Based Multi-Match Packet Classification Using Discriminators[J]. IEEE Transactions on Multi-Scale Computing Systems,2018,4(4):686-697.

[7] WangK. and Wu Hengkui. TCAM-PC: Space-efficient TCAM-based packet classification with packet-forwarding-rate constraints [C]// 2015 12th IEEE International Conference on Electronic Measurement & Instruments (ICEMI),Qingdao,2015:260-264.

[8] . AlievH,S. ChaeW. and KimH.. Cost-efficient architecture of IPsec classification engine with TCAM [C]//The 13th International Computer Engineering Conference,Cairo,2017:20-25.

[9] Q. Dai and LiH.. An Advanced TCAM-SRAM Architecture for Ranges Towards Minimizing Packet Classifiers[C]//2018 IEEE 20th International Conference on High PerformanceComputing and Communications,United Kingdom,2018:158-163.

[10] NorigeE.,LiuA. X. and TorngE.. A Ternary Unification Framework for Optimizing TCAM-Based Packet Classification Systems [J]. IEEE/ACM Transactions on Networking,2018,26(2):657-670.

[11] RottenstreichO.，KeslassyI.，HassidimA.，KaplanH. and PoratE.. Optimal In/Out TCAM Encodings of Ranges [J]. In IEEE/ACM Transactions on Networking,2016,24(1):555-568.

[12] CaiY.，YuF. R.，LiangC.，SunB. and YanQ.，. Software-Defined Device-to-Device (D2D) Communications in Virtual Wireless Networks With Imperfect Network State Information (NSI) [J]. In IEEE Transactions on Vehicular Technology，2016，65(9):7349-7360.

[13] 唐菀,冯伟,杨喜敏,田野. 软件定义网络中OpenFlow流表空间优化技术研究进展[J]. 中南民族大学学报(自然科学版),2019,38(03):459-465.
Tang Wan,Feng Wei,Yang Ximin,Tian Ye. Research progress of OpenFlow flow table space optimization technology in software-defined networks [J]. Journal of South-Central University for Nationalities (Natural Science Edition),2019,38(03):459-465.

[14] LiX. and ShaoY.. Memory compression for Recursive Flow

Classification Algorithm in Network Packet Processing Devices[C]// 2018 IEEE 3rd Advanced Information Technology，Electronic and Automation Control Conference（IAEAC），Chongqing，2018：1502-1505.

[15] 100-Gigabit NPU with Integrated TM and CPUs, EZchip TechnologiesLtd.［OL］[2020-08-06]. Available:http://www. ezchip. com/pnp4. htm.

[16] ZhengS. , BiX. and LuoJ. . An Efficient Total Prefix Length-Based Clustering Packet Classification Algorithm［C］//2016 International Conference on Network and Information Systems for Computers (ICNISC)，Wuhan，2016：46-49.

[17] AntichiG. , CallegariC. , MooreA. W. , GiordanoS. and AnastasiE. . JA-trie: Entropy-based packet classification［C］//2014 IEEE 15th International Conference on High Performance Switching and Routing (HPSR)，Vancouver，BC，2014：32-37.

[18] JamilH. and WengN. . Multibit Tries Packet Classification with Deep Reinforcement Learning［C］//2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR)，Newark，NJ，USA，2020：1-6.

[19] HungS. , IlievN. , VamananB. and TrivediA. R. . Self-Organizing Maps-Based Flexible and High-Speed Packet Classification in Software Defined Networking［C］//2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems(VLSID)，Delhi，NCR，India，2019：545-546.

[20] HsiehC. and WengN. ,Many-field packet classification for software-defined networking switches［J］. 2016 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)，Santa Clara，CA，2016：13-24.

[21] FarhadiH. and NakaoA. , . Rethinking Flow Classification in SDN［C］//2014 IEEE International Conference on Cloud Engineering，Boston，MA，2014：598-603.

[22] JeongS. , LeeD. ,LiJ. and HongJ. W. . OpenFlow-based virtual TAP using open vSwitch and DPDK［C］//NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium，Taipei，2018：1-9.

[23] PfaffB. et al. The design and implementation of open vSwitch［C］// 12th USENIX Symposium on Networked Systems Design and Implementation，Oakland，CA，2015：117-130.

[24] Daly J ,Bruschi V , Linguaglossa L ,et al. TupleMerge: Fast Software Packet Processing for Online Packet Classification［J］. IEEE/ACM Transactions on Networking，2019，27（4）：1417-1431.

[25] YingchareonthawornchaiS. ,DalyJ. ,LiuA. X. and E. Torng. A Sorted-Partitioning Approachto Fast and Scalable Dynamic Packet Classification［J］. IEEE/ACM Transactions on Networking，2018，26（4）：1907-1920.

[26] LewisB. , L. Fawcett, BroadbentM. and N. Race. Using P4 to Enable Scalable Intents in Software Defined Networks［C］//2018 IEEE 26th International Conference on Network Protocols (ICNP)，Cambridge，2018：442-443.

[27] ZannaP. , RadcliffeP. and ChavezK. G. . A Method for Comparing OpenFlow and P4［C］. //2019 29th International Telecommunication Networks and Applications Conference (ITNAC)，Auckland，New Zealand，2019：1-3.

[28] VrsPéter, Kiss A. Security Middleware Programming Using P4［C］// International Conference on Human Aspects of Information Security. 2016：277-287.

[29] YazdinejadA. , BohlooliA. and JamshidiK. . P4 to SDNet: Automatic Generation of an Efficient Protocol-Independent Packet Parser on Reconfigurable Hardware［C］//2018 8th International Conference on Computer and Knowledge Engineering (ICCKE)，Mashhad，2018：159-164.

[30] P4 NLanguage Consortium. p4c-hlir［2020-08-06］［OL]. https://github. com/p4lang/p4-hlir.

[31] IšaR. , Benáček and VP. . Puš. Verification of Generated RTL from P4 Source Code［C］//2018 IEEE 26th International Conference on Network Protocols (ICNP)，Cambridge，2018：444-445.

[32] NötzliA. , KhanJ. , FingerhutA. , BarrettC. , and . AthanasP. P4pktgen: Automated test case generation for p4 programs［C］// Proceedings of the Symposium on SDN Research，USA，2018：51-27.

一种针对拟态工业控制器的裁决及调度方法

张奕，刘星宇，张兴明
之江实验室，杭州 311100

摘　要：本文针对工业系统的安全性问题，根据拟态防御理论中的动态异构冗余模型，并结合工业系统的实际应用场景，首先提出了一种针对工业现场协议的四异构执行体拟态混合裁决方法，再依据执行体间的异构度及共模防御系数提出了一种在有限的异构资源下的执行体调度方法，最后将本文调度算法与随机调度算法进行仿真分析。结果表明，这种新的裁决和调度算法，不但可以有效支撑数字和模拟信号混合的复杂拟态工控应用场景，同时还能快速识别共模攻击、减小系统的共模逃逸时间，进一步提升工控系统的防御能力。
关键词：拟态防御、工业控制器、裁决调度

A Method for Arbitration and Scheduling ofMimicry Industrial Controllers

ZHANG Yi, LIU Xingyu, ZHANG Xingming
Zhejiang Lab, Hangzhou 311100, China

Abstract: Aiming at the security issues of industrial systems, according to the dynamic heterogeneous redundancy (DHR) model in the mimic defense theory, and combined with the actual application scenarios of industrial systems. Firstly, a hybrid adjudication method of four heterogeneous executants for industrial field protocols is proposed, and then based on the heterogeneity of the executive body set and the common-mode defense coefficient, an executive scheduling method under limited heterogeneous resources is proposed, Finally, the algorithm in this paper and the random scheduling algorithm will be simulated and analyzed. The results show that the new algorithm can not only effectively support complex simulated industrial control application scenarios with mixed digital and analog signals, but also can quickly identify common mode attack, reduce the common mode escape time of the system, and further improve the defense capability of industrial control system.
Key words: Mimic Defense; Industrial Controllers; Arbitration and Scheduling

1　引言

近年来，随着我国强国制造战略地全面推进，工业领域数字化、网络化、智能化的改造加快发展，逐渐成为我国实施制造强国和网络强国战略的重要保障。在我国，工业控制系统包含大量的关键性基础设施，涉及国计民生的方方面面，因此工业控制系统信息安全是关系我国生产安全和和经济发展的重大问题［1］。目前我国工控安全面临越来越严峻的安全形势，虽然近几年国产工控系统技术，尤其是在安全可靠性上有了长足的进步，但长期以来我国工控领域的嵌入式软件、总线协议、工控软件等核心技术仍未实现自主可控，特别是针对未知漏洞、未知后门的攻击基本

没有安全防护能力，同时也面临着安全威胁加速渗透、攻击手段复杂多样等新挑战。目前，工业控制系统的安全防护还是基于传统的被动式防御理念，采用的防火墙、入侵检测、入侵防护系统等安全手段已经无法起到有效的防护作用。

邬江兴教授提出了网络空间拟态防御（CMD， Cyberspace Mimic Defense）［2-5］理论，将不可控的网络空间安全威胁问题转化为自主可控的网络空间服务鲁棒性控制问题，从内部构造机制出发将网络空间的被动防御转化为主动的内生安全［6］。拟态防御技术以动态异构冗余（DHR， Dynamic Heterogeneous Redundancy）为核心架构，该架构由输入代理、异构功能等价执行体、裁决器、输出代理等组成。裁决器作为各执行体的输入代理，将输入数据向各执行体进行发送，同时对各执行体的处理结果进行判决产生唯一输出［7］，另外裁决器完成对各执行体的调度，并根据裁决结果对处于异常状态的执行体进行清洗恢复。拟态防御的核心思想是通过功能等价体的动态切换，使异构冗余架构的执行体具有动态化、随机化的属性，使攻击者的攻击难度和成本大幅提高。

2 相关研究

目前的工控系统中的硬件控制器通常采用了冗余机制，但几乎都是同构的冗余，从硬件芯片型号到执行体中的执行程序都是相同结构、相同的代码。从芯片、驱动程序、操作系统、执行程序中一旦有一个后门被利用或者漏洞被攻击成功，则整个控制器就被攻击成功了。针对以上问题，文献［8］提出了一种在工业控制领域应用的拟态安全处理机架构，该架构硬件部分由三套异构冗余处理机及其外围电路以及一个拟态调度器组成，软件部分包括设备驱动、中间件以及异构操作系统，但该系统在执行体调度时不存在随机性和动态性，且当某一执行体遭受攻击进入清洗恢复状态时，系统安全性能降级，此阶段该系统被攻击的风险大大提升。文献［9］提出了一种四异构冗余执行体的拟态架构，与文献［8］相比系统的随机性和动态性有所加强，但并没有考虑当遭受到共模攻击时，如何提升系统的安全增益。文献［10］提出了一种基于历史表现的执行体选择方法，文献［11］提出了一种基于拟态防御的差异化反馈调度判决算法，根据判决算法的可靠度系数及多数判决算法选出可信的执行体。但上述调度策略基于系统存在大量可利用的异构的执行体，而拟态工业系统由于成本和开发难度的限制，很难做到支持大量可利用的异构的执行体，因此上述的调度策略这这种条件限制下不能取得理想的结果。另外工业系统中执行体输出的数据成分复杂，目前的基于一致性的择多判决算法［12］并不能准确裁决。

因此，本文首先提出了针对工业现场协议的裁决方法，然后提出了一种在有限的异构资源下的基于共模防御系数的执行体调度方法，最后通过仿真实验将该方法与现有的随机调度算法进行对比，分析算法在安全性，降低共模攻击成功率，以及在发生共模攻击后减少逃逸时间方面的性能。

3 基于混合数据类型工业协议的拟态裁决方法

在不同的工业应用场景中，尤其是一些复杂的工业现场，工业控制系统需要接入大量不同类型的IO设备，包括开关量IO设备和模拟量IO设备，因此在工业控制系统对各类IO设备进行实时控制的工业协议中会混合有不同的数据类型。目前针对混合数据类型工业协议还没有特别高效和准确的拟态裁决方法，主要采用的是基于内容一致性的择多裁决方法。由于不同的模拟量IO设备具有不同的数据量程，且不同的异构主控制器之间存在运算精度差异，在对模拟量IO的控制数据进行计算时，会产生一定的计算误差，因此在使用基于内容一致性的择多裁决方法对混合数据类型工业协议进行裁决时，对于开关量IO设备的控制数据有良好的裁决效果，但是对模拟量IO设备的控制数据无法进行准确的裁决，存在对主控制器的攻击误判甚至出现攻击逃逸等问题。本文针对上述问题，提出了一种应用于异构多执行体架构的拟态工业控制系统的基于混合数据类型工业协议的拟态裁决方法。

3.1 模拟量数据归一化

要实现对模拟量数据的裁决，首先需要对模拟量数据进行归一化处理，在每个控制周期主控制器单元需要对不同数据类型的IO实时控制数据进行归类，按照数字量和模拟量分别进行计算并

组包，并在协议中的特定位置进行标记。在每个控制周期各执行体在进行IO设备模拟量计算时，需要将不同IO设备的模拟量统一到一个固定的取值范围，具体公式如下：

$$R_i = \frac{r_i}{F_I} \times d \tag{1}$$

其中，r_i为实际IO设备模拟量数值，F_i为该IO设备的量程值，d为精确有效数字对应的放大系数。

3.2 裁决过程

结合各执行体当前状态和调度机制，从中选择出3个执行体进入裁决池。将裁决池中执行体发送的IO控制数据进行解析，按照不同的数据类型，将数据分别放入不同的裁决区中进行裁决，数字量放入裁决区A，模拟量放入裁决区B；分别对裁决区A中的数据和裁决区B中的数据进行择多裁决，数字量和模拟量采取不同的判决方式，具体判决规则如下：

进行数字量类型数据判决时，判断3个执行体相同位号对应的数字量值是否一致，如果某1个与其他2个不同，则认为特殊的这个位号出错，若三者都不一致，则三者都出错；

进行模拟量类型数据判决时，先取3个执行体相同位号对应的模拟量的中值，假设$\{Q_1, Q_2, Q_3\}$为3个执行体相同位号对应的模拟量，其中$Q_1 \leqslant Q_2 \leqslant Q_3$，则中值为$Q_2$；再计算其他两个值与中值的偏差，如果偏差大于误差配置值则认为该执行体位号对应的值出错，假设为δ为误差配置值，若$Q_2 - \delta \leqslant Q_1 \leqslant Q_2 + \delta$，则$Q_1$对应执行体的该位号正确，若$Q_2 - Q_1 < Q_2 - \delta$或$Q_1 > Q_2 + \delta$，则$Q_1$对应执行体的该位号出错；同理，若$Q_2 - \delta \leqslant Q_3 \leqslant Q_2 + \delta$，则$Q_3$对应执行体的该位号正确，若$Q_3 < Q_2 - \delta$或$Q_3 > Q_2 + \delta$，则$Q_3$对应执行体的该位号出错；

裁决时遍历所有裁决区数据，判断为出错的执行体，累加1个出错数，所有位号轮询后累积出错数最多的执行体判定为异常，出错最少的执行体判定为可信的执行体，出错数在中间的执行体判定为中间状态，若三个执行体出错数一致，则都判定为出错，取判定状态为正常的执行体的结果作为裁决结果。裁决结束后，将裁决区A的裁决结果和裁决区B的裁决结果相加作为最终的裁决结果，将裁决结果错误数最少的执行体所对应的IO设备控制数据标记为可信数据，将裁决结果错误数最多的执行体标记为异常；若裁决结果错误数最少的主控制器有多个，则随机选取其中的某个执行体，将其所对应的IO数据标记为可信数据。

4 基于共模防御系数的调度方法

根据拟态理论，在拟态系统中采用的异构执行体越多，获得的安全增益越大，但同时也伴随着系统的复杂度变大，成本变高，可维护性变差等问题［13］。因此在实际的工程设计中需要对系统的复杂度和安全性进行折中。在本文的实际设计方案中，采用的是具有四个异构执行体的工控处理系统。在文献［9］中对四异构执行体处理机系统的安全增益进行了详细的分析，经过择多判决后，系统的错误率降低的非常明显，超过了两个数量级。本文在此基础上针对四个异构执行体的工控系统提出了一种基于共模防御系数的调度方法，旨在保证系统安全增益的同时能够减小共模攻击的成功率并缩短逃逸时间。

4.1 前提假设

假设1. 由于系统内含有多异构执行体，系统的输入输出设计复杂，为便于分析，在本文中假设输入和输出部件不会对系统的安全增益成影响。

假设2. 为便于分析，假设对执行体中任一漏洞攻击成功则导致执行体失效，且这种失效在时间维度上可以看作是均匀的随机事件。

假设3. 为便于分析，在简化模型中，假设对系统中不同执行体的攻击致失效率是相同的，且两个执行体间的失效是相互独立的。

假设4.

为便于分析，任何攻击成功则导致失效的执行体，可在对执行体进行清洗后重新恢复。

4.2 相关定义

令$\mathrm{Set}_E = \{E_1, E_2, \cdots, E_m\}$是所有异构构件之集［14］，令构件$E_i(1 \leqslant i \leqslant m)$上漏洞之集记为$VUL_I$，所有构件漏洞之集记为$VUL = \bigcup_{i=1}^{m} VUL_I$。在此基础上，给出如下若干定义。

定义1. 共模攻击.

VUL_I与VUL_j相互之间的重合部分称为VUL_I与VUL_j的共生漏洞。对于两个执行体间的共生漏洞，将其称为2阶共生漏洞。以此类推，将k个执行体间的漏洞称为k阶共生漏洞。利用共生漏洞发起的

攻击为共模攻击。

定义2. 拟态逃逸.

对于n个执行体的处理结果进行择多比较，如果有$k(\frac{n}{2} \leqslant k \leqslant n)$个执行体的输出结果一致，则按照k个执行体的处理结果，作为相对正确结果进行输出。可以看出，如果系统中包含的最大共生漏洞阶数是小于k的，那么采用k阶一致裁决方法能够感知到漏洞威胁：如果系统中存在k阶或k阶以上共生漏洞，那么将不再能感知到该漏洞威胁，产生拟态逃逸现象。

定义3. 恢复时间T_H.

恢复时间为失效的执行体被裁决器清洗，到其恢复进入工作状态并与其他执行体实现状态同步所需要的时间T_H。

定义4. 拟态逃逸时间T_C.

从当系统因受到针对共生漏洞的攻击导致系统出现拟态逃逸，到系统恢复到正常工作状态所需要的时间T_C。

定义5. 异构度σ_{ij}.

在拟态防御中，执行体对象结构差异越大，越能增加系统内部结构的复杂性，执行体之间存在共生漏洞的概率就越小。异构度包括许多方面，如编程语言、运行的操作系统平台、硬件架构等[15]。异构度是用于描述异构程度的一种量化参数，数值越小代表结构差异越大，且$\sigma \in (0,1]$。σ_{ij}表示第i个执行体与第j个执行体之间的异构度。

定义6. 执行体失效概率P_i.

对于执行体而言，当攻击者通过某种攻击手段对执行体中的漏洞攻击成功时，则认为该执行体失效。执行体失效的概率密度用$p_i(t)(1 \leqslant i \leqslant m)$表示，因此某一时刻的失效概率为$P_i = \int_0^t p_i(t)dt$。

定义7. 共模防御系数μ_{ij}.

μ_{ij}表示第i个执行体与第j个执行体之间的共模防御系数，数值越小表示发生的共模攻击的概率越低，系统初始化状态时，每个执行体间的共模防御系数就是各执行体间的异构度，正常运行时，裁决器根据执行体每次的输出情况，判断执行体是否受到共模攻击并更新共模防御系数。式（2）为共模防御系数的更新公式

$$\mu_{ij} = \sigma_{ij} \times a^{\left(1+\frac{w_{ij}}{T}\right)} \tag{2}$$

其中，w_{ij}为表示第i个执行体与第j个执行体之间发生共模攻击的次数，当裁决器判定第i个执行体与第j个执行体发生共模攻击时，会将其加1。T为系统运行的周期数。a为常数，用来调节共模防御系数受异构度与共模攻击概率的影响程度。

共模防御系数的可行性说明：由上述的共模防御系数定义及更新机制可知，执行体的共模防御系数表示执行体间的异构程度以及一段时间内各异构体间受到共模攻击的频率，共模防御系数越小表明该执行体之间发生共模攻击的概率越低。根据公式（3）可以得到当发生共模攻击时，选择共模防御系数较小的一组执行体比随机选择一组执行体发生共模攻击逃逸的概率要小。

$$\frac{\min(\mu_{ij},\mu_{i'j'})}{\mu_{ij}+\mu_{i'j'}} \leqslant \frac{1}{2} \tag{3}$$

4.3 调度策略

4.3.1 调度方式

本文设计的拟态工业控制系统含有四个异构的执行体，在正常工作时，策略控制与调度模块随机选取四个异构的执行体中的三个进入裁决池，裁决器按照上节所述的裁决方法进行裁决，对裁决结果异常的执行体进行标记，当执行体被标记的异常数达到所设定的清洗阈值后，裁决器对该执行体进行清洗操作。系统正常工作时的调度方式图1所示。

在每个裁决周期，裁决模块会监控各个执行体的运行状态，并将各执行体的工作状态反馈给策略控制与调度模块。在系统的运行过程中各执行体会产生以下四种状态：工作状态、清洗状态、考察状态、挂起状态。

工作状态：默认执行体的状态为“工作状态”，处于该状态时执行体参与系统的裁决调度。对于异常数达到清洗等级的执行体，裁决模块下发相应的清洗命令，同时将该执行体的状态记录为“清洗状态”。如果记录的某个执行体清洗次数达到“挂起”等级，裁决模块下发挂起命令，同时将该执行体的状态记录为“挂起状态”。挂起状态时需要告警通知维护人员干预。

清洗状态：当某一执行体进入到“清洗状态”后，该执行体不参与裁决调度，当裁决模块收到“清洗状态”的执行体发送的数据后，将该执行体

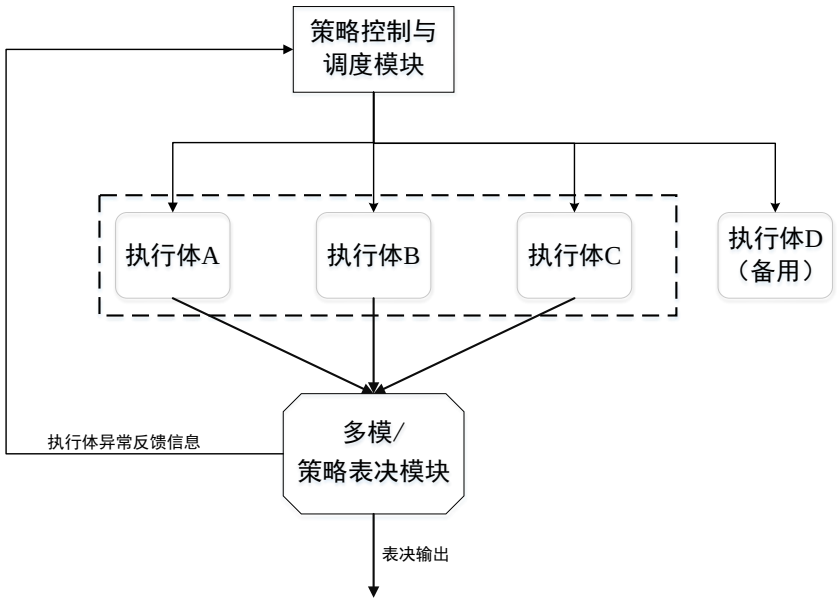


图1　系统正常工作时的调度方式

的状态记录为“考察状态”。

考察状态：为了规避执行体恢复的震荡期而设置考察状态，标记为“考察状态”的执行体不参与裁决调度，考察周期数根据实际配置确定，当考察周期达到后将该执行体标记为“工作状态”。

挂起状态：被标记为“挂起状态”的执行体不参与裁决调度，同时该执行体也应该停止发送实时数据。如果该执行体被人工干预后重新工作，则裁决模块应将其标记为“考察状态”。

执行体各个状态间的切换示意图如图2所示：

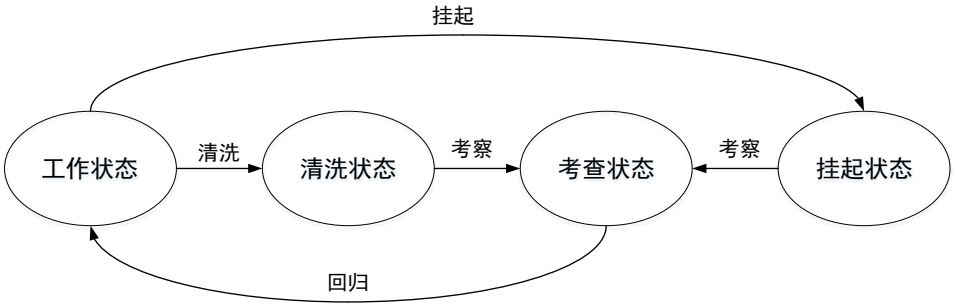


图2　执行体状态切换示意图

当有执行体不处于“工作状态”，系统会因缺少可调度和裁决的执行体时，系统的安全等级会进行降级，不同的安全等级下系统的调度策略如下表1所示：

表1　不同的安全等级下系统的调度策略

安全等级	可调度执行体个数	调度策略
四级	4	从四个可调度执行体中选取三个进行裁决,裁决出可信执行体
三级	3	将三个可调度执行体全部进行裁决,裁决出可信执行体
二级	2	选取共模防御系数较小的执行体作为可信执行体
一级	1	将该执行体作为可信执行体

4.3.2 共模攻击识别及处置

当裁决模块在进行裁决时，若出现裁决结果为2：1（即某两个执行体裁决结果一致与另一个不一致），则裁决模块会将备用的执行体加入重新进行裁决，具体调度如下图3所示：

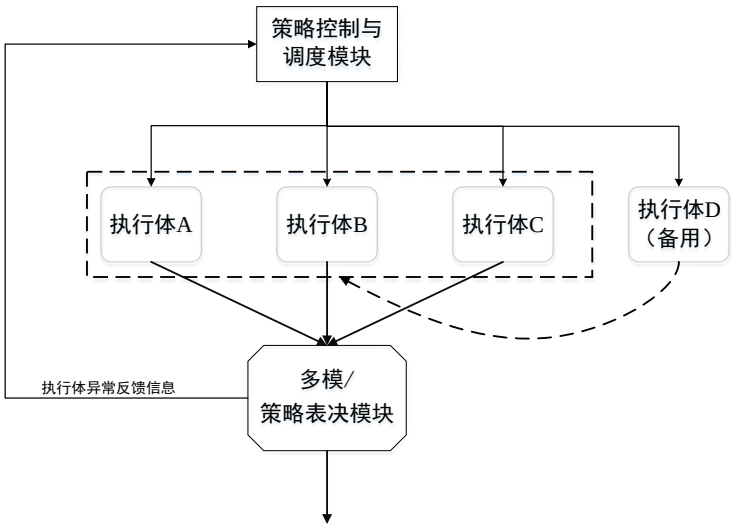


图3 加入备用执行体裁决示意图

若此时裁决结果为3：1（即三个执行体裁决结果一致且与另一个不一致），则判定不一致的执行体状态异常，裁决模块对其进行清洗。

若裁决结果为2：2（即四个执行体裁决结果两两一致），则裁决模块判定系统受到共模攻击，此时选取共模防御系数较小的一组执行体作为可信执行体，并从另一组执行体中随机选择一个执行体进行清洗。

待清洗的执行体恢复正常工作模式后，裁决模块重新对四个执行体进行裁决，若裁决结果为3：1，则认为清洗的这一组执行体发生了共模攻击，且清洗选择正确，裁决模块更新这一组执行体的共模防御系数，同时将该组另一个执行体清洗。

若重新裁决后，裁决结果仍为2：2，则判定未清洗的这一组执行体发生了共模攻击，且裁决模块未正确选择清洗对象，系统发生共模攻击逃逸，此时将发生共模攻击的这一组执行体全部清洗并更新这一组执行体的共模防御系数。

若裁决模块对某一执行体连续清洗次数超过一定的敏感度阈值γ后，则裁决模块判定未清洗的这三个执行体遭受了3阶共模攻击，且发生共模攻击逃逸，此时将发生共模攻击的这三个执行体全部清洗。当裁决模块判定系统受到3阶共模攻击超过敏感度阈值threshold后，系统在正常工作时会采用所有执行体都进行裁决的4余度调度策略，如图4所示：

4.3.3 算法描述

综合上面的共模攻击的识别及处置并结合拟态随机调度的策略，具体算法描述如下：

INPUT： 调度集E（E_1, E_2, E_3, E_4），敏感度threshold；

OUTPUT：裁决结果F；

1）初始化超阈值执行池列表，OverList = null；

2）for i = 1； i <k； i++

3）调度集E（E_1, E_2, E_3, E_4），从调度集中各执行体的输出结果随机选择3个结果构建出结果池PoolList（x，y，m），剩下的一个执行体的输出结果为备份结果BackUpList（n）；

4）计算PoolList（x，y，m）.δ_i；//择多判决出各执行体输出结果

5）if PoolList（x，y，m）.δ_i == 3：0；return PoolList（x，y，m）；

6）else PoolList（x，y，m）.δ_i == 2：1；PoolList（x，y，m）.add（BackUpList（n））；//将备份结果添加到结果池中

7）计算PoolList（x，y，m，n）.δ_i'；

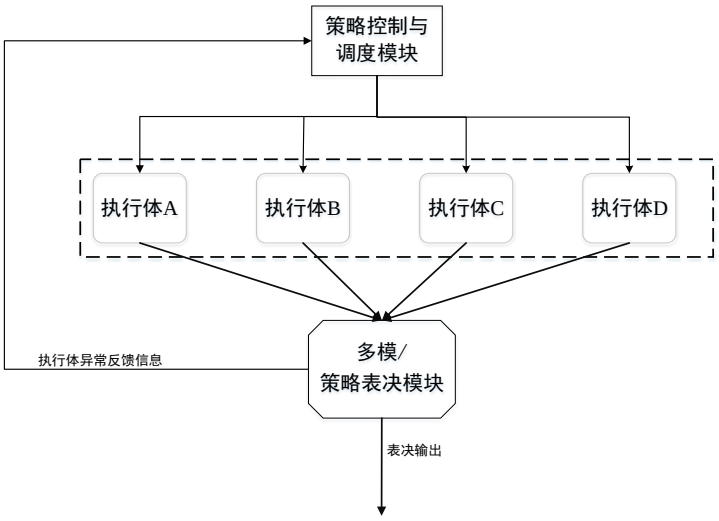


图4　执行体4余度调度策略示意图

8）if PoolList（x，y，m，n）.δ_i'== 2：2；return Move（PoolList［x，y，m，n］.μ_{max}，OverList）；//将其中共模防御系数μ较大的执行体移除结果池并对其进行清洗

9）PoolList［x，y，m］.add（new（m））；//将清洗过并重新上线的执行体n的结果数据加入到结果池中；

10）计算PoolList（x，y，n）.new（m）.δ_i''；

11）if PoolList（x，y，m，n）.δ_i'' == 3：1；return PoolList［x，y，m，n］.remove（n），//将占少数数据结果的执行体n移除结果池并对其进行清洗操作；

12）else if PoolList（x，y，m′，n）.new（m）.δ_i'' == 2：2；return PoolList［x，y］.P（σ）= PoolList［x，y］.P（σ）+ p_2；//执行体x，y被共模攻击概率增加p_2；

13）return PoolList［x，y，m，n］.remove（x，y）；//将可能发生共模攻击的执行体x，y移出结果池并对其进行清洗操作；

14）PoolList［x，y，m］.add（new（m））；//将清洗过并重新上线的执行体m的结果数据加入到结果池中；

15）end for

16）if PoolList［x，y，m］.p（σ）> PoolList［x，y，m］.p（σ）.threshold；

17）return PoolList［x，y，m］.cnt ++；////发生共模攻击概率超过阈值次数加1

18）if PoolList［x，y，m］.cnt > PoolList［x，y，m］.cnt.threshold；

19）return PoolList（x，y，m，n）// 调度集E（E_1, E_2, E_3, E_4），将调度集中各执行体的输出结果构建出结果池PoolList（x，y，m，n）

结合基于混合数据类型工业协议的拟态裁决方法以及基于共模防御系数的调度算法，本文设计的针对拟态工业系统的调度裁决方法可以用图5表示。

5　仿真实验

本文提出了一种面向拟态工业控制系统的调度方法与判决方法，为了验证该方法的有效性，本节对其进行了仿真分析（在仿真时命名为HPF Scheduling算法），并与文献［9］提出的基于四个执行体的随机调度算法（在仿真时命名为Stochastic Scheduling算法）进行了性能对比。

5.1　仿真环境

本文结合工程实际设计了一个具有四个异构执行体的工业控制系统模型，在仿真验证环节中仅保留最基本的控制节点要素，即控制器单元、仲裁器和模拟IO单元。为模拟拟态工业控制系统的异构特征，其中四个控制器单元采用不同架构的处理器。

仿真环境的硬件架构图由下图6所示：

其中四个主控单元（执行体）与裁决器之间通过以太网进行连接，裁决器与模拟IO之间通过CAN总线连接，裁决器作为主控单元的输入输出代理，将主控单元的IO控制命令经过裁决后下发

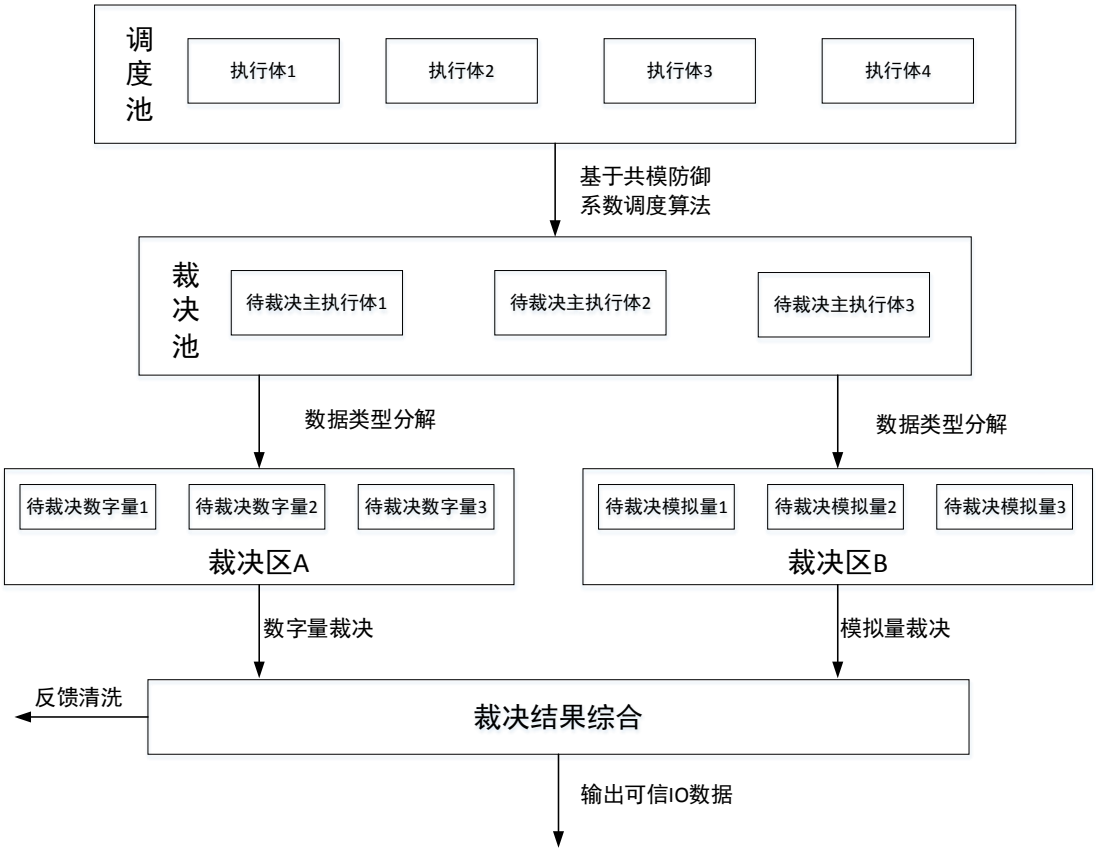


图5　针对拟态工业系统的调度裁决方法

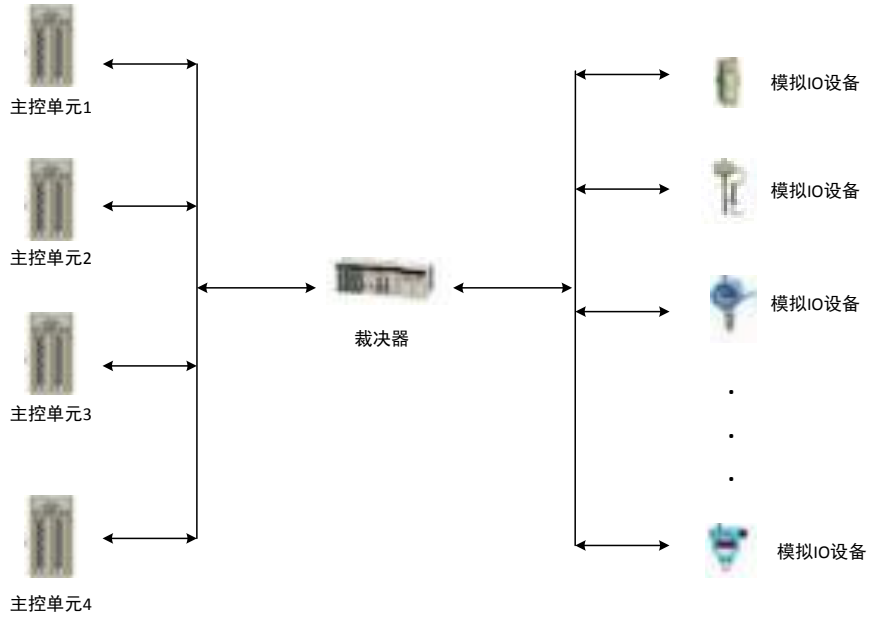


图6　仿真环境的硬件架构示意图

给IO设备，同时将IO上送的采集数据发送给主控单元。仲裁器由FPGA实现，完成上文介绍的调度、判决和清洗策略，四个主控单元分别采用A7、M7、H7和龙芯架构。

5.2　仿真测试方法

工控系统往往需要处理许多敏感信息，包括

温度信息、阀门开关、运行控制等，若工控系统中存在后门程序，则攻击者可以利用这些后门程序实现快速、精确定位与攻击。在本文仿真实验情景中，我们通过白盒测试的方法，向执行体中植入后门程序来进行测试验证。

我们通过简单的代码实现通过模拟温度数据触发后门攻击的程序，程序实现为当输入的温度数据为预先设置的值时即达到触发条件，其攻击触发结果为导致IO设备关机。其中分别在各个执行体的程序中注入了可以造成共模攻击的后门程序，若输入的是某些特殊的温度数据则会触发相应的共模攻击。

仿真时假设攻击者攻击成功这一事件可视为服从失效率为λ的指数分布［16］，其中λ代表攻击致失效的难度，概率密度用$f(t)=\lambda e^{-\lambda t}$表示，则单个执行体的失效概率为：

$$p(t)=\int f(t)dt=1-e^{-\lambda t} \tag{4}$$

通过上式可以说明攻击者在开始的一段时间内能够迅速了解系统状况，攻击成功的概率迅速增加，后随着时间的增长，对于系统的了解程度逐渐变缓，攻击成功的成功率慢慢趋近于1。为了量化执行体结构的差异性，本文利用文献［17-18］中相似度度量方法，计算了各执行体之间在芯片架构间的异构度M：

$$M=\begin{bmatrix}1.0,0.34,0.28,0.07\\0.34,1.0,0.36,0.05\\0.28,0.36,1.0,0.08\\0.07,0.05,0.08,1.0\end{bmatrix}$$

表2为各执行体间的异构度。

表2　四个执行体间异构度

芯片类型	A7	M7	H7	龙芯
A7	1.0	0.34	0.28	0.07
M7	0.34	1.0	0.36	0.05
H7	0.28	0.36	1.0	0.08
龙芯	0.07	0.05	0.08	1.0

另外假设执行体的恢复时间T_H为100个控制周期，四个执行体的攻击致失效难度$\lambda_1=\lambda_2=\lambda_3=\lambda_4=0.01$。

5.3　仿真结果

首先对两种算法在差模攻击下的安全性能进行仿真，由于调度算法的设计，当某一执行体被攻击后会被清洗，因此在该执行体从清洗到恢复的过程中系统处于安全降级状态，此时执行体集被成功攻击概率的关系函数为：

$$P_s=C_m^k\prod_i^k p_i\prod_j^{m-k}(1-p_j) \tag{5}$$

其中p_i为

$$p_i=\int_t^{t+T_H}f(t)dt \tag{6}$$

而HPF Scheduling算法在差模攻击下将异构度作为调度依据，此时执行体集被成功攻击概率的关系函数为：

$$P_H=C_m^k\sigma_{ij}\prod_i^k p_i\prod_j^{m-k}(1-p_j) \tag{7}$$

图7为安全性能的仿真结果，在实验中分别对两个算法进行了100000个控制周期的仿真，从图7中可以看出HPF Scheduling算法和Stochastic Scheduling算法在控制周期的开始阶段，系统攻击成功的概率增加，后随着时间的增长，攻击成功的成功率趋近于0。在使用HPF Scheduling算法时系统的攻击成功率，要低于使用Stochastic Scheduling算法时系统的攻击成功率，说明HPF Scheduling算法的安全性能较Stochastic Scheduling算法有所提高。

图8为当系统遭受到攻击者发起的二阶共模攻击时，系统分别使用HPF Scheduling算法和Stochastic Scheduling算法的攻击成功率，实验中分别对两个算法进行了100000个控制周期的仿真。

从图8中可以看出，在使用HPF Scheduling算法时系统的二阶共模攻击成功率要低于使用Stochastic Scheduling算法时攻击成功率。由于HPF Scheduling算法在识别系统遭受到二阶共模攻击后，利用共模防御系数对执行体进行判定，选取共模防御系数小的执行体组合，相比于Stochastic Scheduling算法的随机选取，共模攻击成功率较低。

在运行10000个周期后，HPF Scheduling算法的共模防御系数的基于共模攻击概率统计的机制生效，此后相对于Stochastic Scheduling算法，HPF Scheduling算法使得系统的共模攻击成功率进一步降低。

图9为当系统遭受二阶共模攻击后，分别使用HPF Scheduling算法和Stochastic Scheduling算法的下逃逸时间，分别对两个算法进行了100次的二阶

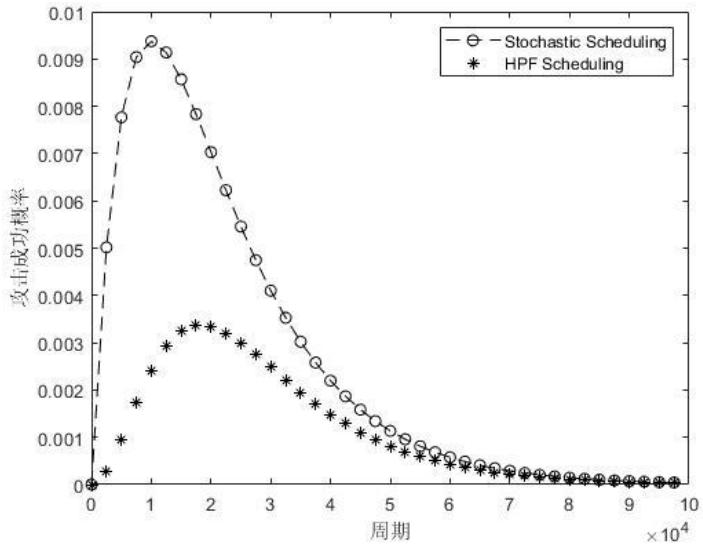


图7　差模攻击下的两种算法的安全性能

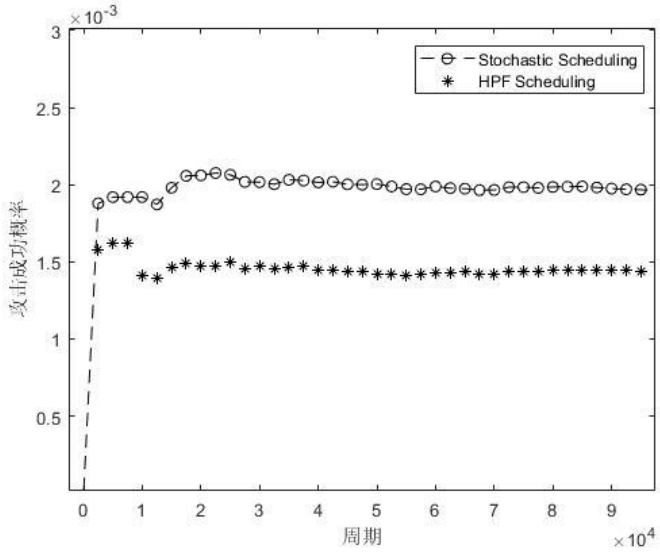


图8　二阶共模攻击下两种算法的安全性能

共模攻击模拟。其中系统逃逸时间的坐标轴一个单位代表执行体的恢复时间。

从图9中可以看出，由于HPF Scheduling算法能够识别二阶共模攻击，因此在两个执行体的恢复时间的内可以消除系统的被攻击状态，而Stochastic Scheduling算法由于不能识别二阶共模攻击，采用随机调度的策略，因此系统逃逸时间存在波动且大于使用HPF Scheduling算法下的逃逸时间。

图10为当系统使用HPF Scheduling算法遭受三阶共模攻击后的逃逸时间，实验进行了20次的三阶共模攻击模拟。其中系统逃逸时间的坐标轴一个单位时间为一个控制周期。仿真时三阶共模攻击的敏感度设为10。

从图10中可以看出HPF Scheduling算法在三阶共模攻击的次数到达敏感度以前采用的是四取三随机调度的策略，因此在识别三阶共模攻击时取决于随机调度的结果，系统的逃逸时间存在波动，但当HPF Scheduling算法检测三阶共模攻击的次数到达敏感度以后，采用四余度裁决，此时能够快速识别三阶共模攻击，因此之后的系统逃逸时间稳定在最小的逃逸时间。

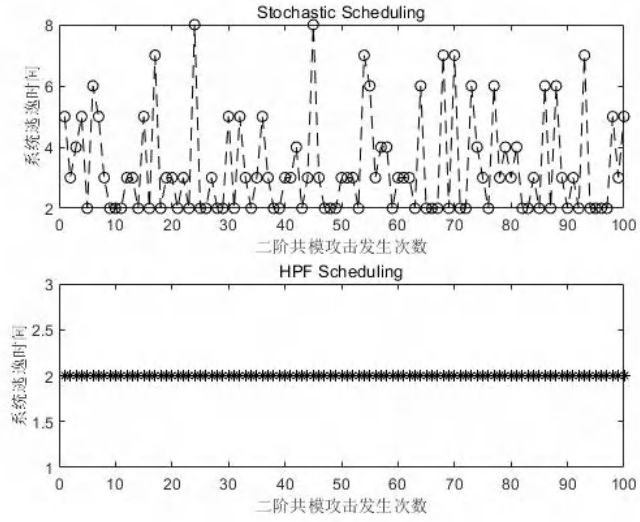


图9　两种算法二阶共模攻击的逃逸时间

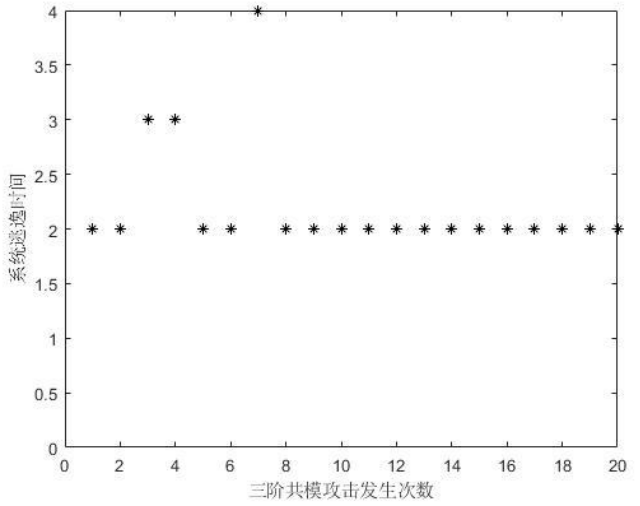


图10　HPF Scheduling算法三阶共模攻击逃逸时间

6　结束语

拟态的核心思想是通过拟态控制层的调度判决，使得系统呈现动态化、随机化的属性，来进行主动防御。但是目前的调度算法和判决算法并不能很好地适应拟态防御工业系统的应用场景。本文设计了一种面向拟态工业控制系统的调度裁决算法，有效主动地防御攻击者的入侵。仿真结果表明，该方法在保证安全性能的同时，能够快速识别共模攻击，并能有效减小系统共模逃逸的时间。在后续研究中，将对该算法做相关优化，使之具有更好的实用性。

参考文献:

[1] 匡恩．工业控制网络安全态势报告．http://www．kuangn. com/2016/04/041，2015.

[2] 邬江兴．网络空间拟态防御研究[J]. 信息安全学报，2016.

[3] 邬江兴．网络空间拟态安全防御[J]. 保密科学与技术，2014.

[4] 邬江兴．拟态计算与拟态安全防御的原意和愿景[J]. 电信科学，2014.

[5] 邬江兴,张帆,罗兴国．拟态计算与拟态安全防御[J]. 中国计算机学会通讯，2015.

[6] Liqun Wang. The Attack Surface Shifting in the Mimic Defense System［M］. IEEE 4th International Conference on Computer and Communications. 2018

[7] Bolin Ma. Security Research of Redundancy in Mimic Defense System［M］. IEEE 3th International Conference on Computer and Communications. 2017

[8] 魏帅,于洪,顾泽宇等. 面向工控领域的拟态安全处理机架构[J]. 信息安全学报,2017.
[9] Ling OuYang. Analysis of Mimic Defense and Defense Capabilities based on Four-Executor. Proceedings of the 2018 International Conference on Advanced Mechatronic Systems，Zhengzhou，China. 2018
[10] 普黎明,刘树新,丁瑞浩等. 面向拟态云服务的异构执行体调度算法[J]. 通信学报. 2020.
[11] 沈丛麒,陈双喜,吴春明. 基于信誉度与相异度的自适应拟态控制器研究[[J]. 通信学报2018
[12] 欧阳城添,王曦,郑剑. 自适应一致表决算法[J] 计算机工程. 2011
[13] 武兆琪,张帆,郭威等. 一种基于执行体异构度的拟态裁决优化方法[J]. 计算机工程. 2020
[14] 李卫超,张铮,王立群等. 基于拟态防御架构的多余度裁决建模与风险分析[J]. 信息安全学报,2018.
[15] 高明,罗锦,周慧敏等. 一种基于拟态防御的差异化反馈调度判决算法[J]. 研究与开发. 2020
[16] Guo Wei. Scheduling Sequence Control Method Based on Sliding Window in Cyberspace Mimic Defense[J]. IEEE Access. 2019
[17] QIU D H. Measuring software similarity based on structure and property of class diagram [C]. Proceedings of Sixth International Conference on Advanced Computational Intelligence. 2013
[18] WANG Z P. A DNS framework design based on mimic security defense[J]. Electronic Journal. 2017

[作者简介]

张奕（1990—），男，硕士，之江实验室工业互联网研究中心工程师，主要研究方向为工业控制系统安全。

刘星宇（1991—），男，硕士，之江实验室工业互联网研究中心工程师，主要研究方向为工业控制系统安全。

张兴明（1963—），男，之江实验室工业互联网研究中心研究员，主要研究方向为拟态计算及拟态安全。

基于联合历史置信度的拟态权重策略研究

应飞[1,2]，罗论涵[2]，解维[2)]

[1]同济大学；

[2]中国电子科技集团公司第三十二研究所

摘　要：在经典的拟态IPO模型中，输出裁决器作为拟态安全防御系统的最终输出，其裁决结果直接决定着整个系统对安全威胁的防御效果，拟态裁决结果通常由多数一致性策略决定。在实际应用中，常对多数一致性表决算法进行改进，加入各执行体的置信权重，并根据裁决器的裁决结果通过反馈控制器实时更新各拟态执行体的权重。传统的基于历史置信度的权重策略会导致整个拟态系统在长时间经历单一类型攻击后，高置信权重的拟态执行体类型逐渐趋同，使得当网络空间攻击类型变换时，整个拟态系统无法快速正确防御甚至失效。本文提出了一种基于联合置信权重的拟态裁决策略，通过实际攻击检测，并应用于拟态构造容器云上，实验结果显示不仅提高了拟态构造容器云上应用服务的内生安全防御能力，而且提高了整个拟态构造容器云的资源利用率。

关键词：拟态构造、容器云、拟态裁决、历史置信度

Research on Mimic Arbitration Strategy Based on Associate Historical Confidence Degree

Ying Fei[1,2], Luo Lunhan[2], Xie Wei[2)]

1.Tongji University;

2.The 32nd Research Institute of China Electronics Technology Group Corporation

Abstract: In classic mimicry IPO model, voting output unit is the final output of a mimic defense system, the voting result determines the defense effect to the cyberspace attack threats, the mimic voting result is usually determined by the majority consistency strategy. We often imporve the algorithm by adding the executors' confidence weights in practice, and updating the executors' weight according by the feedback controller in real-time. Traditional strategy based on historical confidence degree will lead to the executors with high confidence degree weight point to greater similarity after suffering a single type of attack during a long time, while the cyberspace attack type changes, the mimic defense system cannot work well. In this paper, a mimic arbitration strategy based on associate historical confidence degree is proposed. Applied to container cloud based on mimic structure(CCBMS) and verified with actual attack, this strategy can not only improves the endogenous security defense capability of application services running on the CCBMS, but also increase the efficiency as the experimental results show.

Key words: mimic structure; container cloud; mimic arbitration; historical confidence degree

1　引言

随着“新基建”的推进，云计算将加快应用落地进程，在互联网、政务、金融、交通、物流、教育等不同领域实现快速发展[1]。网络空间已经成为继“陆、海、空、天”之后的第五维战略空间[2]。容器技术作为新兴的网络空间云计算解决方案，以其灵活轻便的特点，得到了广泛的推广使用，2017年就有超过9成的企业使用了容器虚拟化技术[3]。随着大量云端计算的使用，在为应用服务提供便捷高效的访问同时，也带来的众多安全隐患，其中API和接口漏洞、未知风险配置文件等被列为了云计算的十大高危威胁[4]。

为了增加虚拟化容器云环境的安全性，研究

基金项目：上海市科学技术委员会科研计划项目（18511104402）

者们做了大量的研究，其中大部分聚焦于虚拟化容器云的安全调度，使用优化粒子群算法和蚁群算法提高容器云环境下运行微服务应用的安全可靠性[5-7]。以及其他方法来保护部署运行在虚拟化容器云平台上的应用服务的安全稳定运行，如使用面向攻击者的博弈模型等[8]。其共同点都是根据攻击者的攻击行为选择不同的调度算法来保护容器云的安全。

与基于攻击经验的后验防御方式不同，通过在传统容器云平台上叠加基于动态异构冗余的拟态防御机制，可以有效增强原生容器云平台在不额外添加安全防护机制的前提下的安全性，提出了根据拟态裁决结果进行调度的方法[9][10]。并推广到可以适用于通用运行环境的拟态资源调度算法[11]。但这些研究都基于拟态裁决策略相对简单时的情况下的拟态调度算法的改进，在这种简单的拟态裁决策略下，应用以微服务的形式进行部署，在系统正常工作时，拟态裁决器通过对异构微服务执行体的HTTP响应结果进行拟态裁决，选取多数一致性的裁决结果作为拟态微服务的对外响应结果。

我们在实际生产应用中，对于随机选取的异构执行体，拟态裁决器在裁决时并不对各异构执行体的响应结果一视同仁，而是为各执行体设置了权重，权重的大小由反馈控制器根据拟态裁决器的裁决结果不断更新，当执行体的响应结果与拟态一致性输出一致时，权重保持不变，反之调减。拟态裁决器在进行拟态裁决时，将各执行体与其归一化的权重值（通常使用历史置信度作为权重）相乘作为多数一致性裁决的输入集。例如文献［12］通过使用以历史置信度作为权重的多数一致性裁决策略，调低多次被裁决器识别为异常的执行体的置信度权重，有效规避了共模逃逸的风险[13]，但当整个基于拟态构造的容器云环境长期暴露在单一类型攻击的场景下时，会使得具有某一特定特征值的一类执行体的历史置信度权重持续降低，此时如果突然遭受另一维度类型的攻击时，这一部分异构执行体将无法为拟态构造的容器云上的应用服务提供有效的一致性表决支持，我们称之为拟态构造容器云环境下面对单一类型攻击的防御退化。攻击者常通过单一类型攻击的方式持续训练拟态防御系统，使其降低甚至失去对其他威胁的防御能力。

为解决此类问题，本文提出了一种基于联合置信权重的拟态裁决策略，综合考虑裁决时各执行体的的多数一致性裁决结果和当前时刻进入拟态输出裁决器的执行体组信息，使异常执行体只在特定的异构执行体群组中表现低置信度，使其在其他场景下仍能发挥异构裁决的支撑作用。提高了拟态构造容器云环境下的资源利用率。

2 相关工作

2.1 拟态构造容器云平台

基于拟态构造的容器云使用容器虚拟化技术为云化服务构建异构执行个体，由于虚拟化容器本身依赖的物理资源来自多元化的软硬件实体，具有天然的异构性，因此很容易为容器化的拟态应用服务构建异构执行体，结合输入代理器、输出裁决器和反馈控制器，构成了具有拟态应用内生安全防御特性的容器云[14]。

拟态构造的容器云具有以下几个基本特征：

1. 整体架构符合拟态I【P】O模型；

2. 在输入端具有可配置的一对多输入分配器；

3. 包含具有多元性或多样性的异构容器镜像池；

4. 具有独立的输出裁决器，能够汇聚多路响应并输出多数一致性结果；

5. 具有基于裁决结果的容器调度器；

6. 具有拟态负反馈控制器。

在实际应用当中，拟态应用服务在拟态构造的容器云的各计算节点根据操作系统，Web服务，文件系统、数据库等层次构建异构镜像，形成异构执行体池，请求通过输入代理分发到异构执行体池中的3个执行体上，异构执行体在分别进行独立计算后，将应用服务的响应结果发送给输出裁决器进行多数一致性裁决，形成的多数一致性裁决结果作为拟态应用服务的最终响应结果，同时裁决出的异常执行体的信息将通过反馈控制器为执行体的调度和权重调整提供依据。

正是基于容器虚拟化天然的多元异构性，加之一些策略性的动态调度机制，能够以极小的软硬件成本，构建一个面向云空间应用服务的高度普适的拟态构造容器云平台，为避免应用服务中的漏洞所带来的安全隐患提供了架构层面的解决

方案。

在一般情况下，拟态应用服务的请求和响应是基于HTTP协议的报文，其中响应报文由状态行、消息报头、空行和响应体构成。

面对各异构执行体对统一请求报文做出的响应，输出裁决器先比较响应报文的状态行，如果状态行一致且为有效输出响应（即状态码为2XX），则直接对各执行体响应报文的响应体进行一致性表决。响应体的多数一致性表决结果，连同一致性的状态行和随机消息报头将作为异构执行体响应的响应输出。

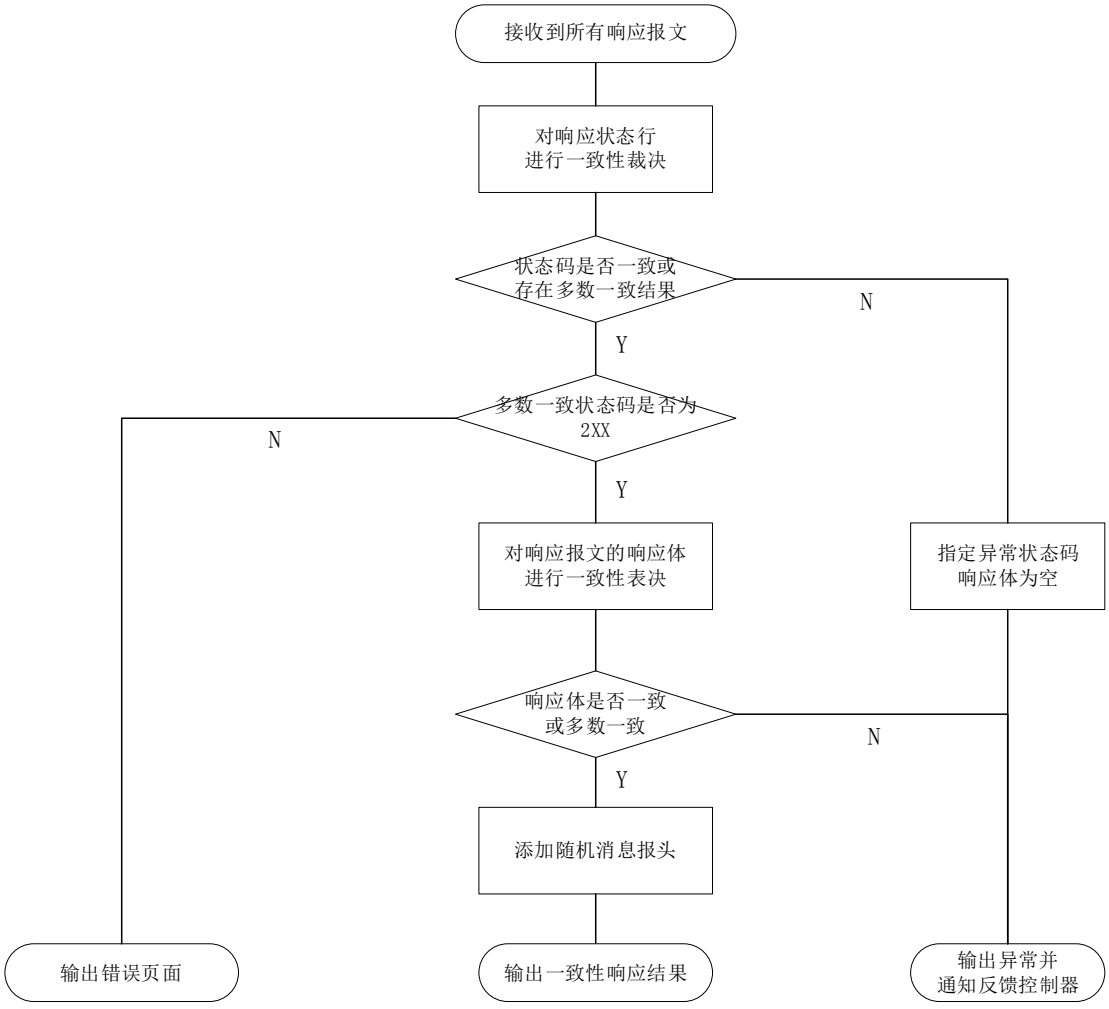


图1　执行体响应裁决流程示意图

由于应用服务的漏洞只在特定的环境配置下才会被触发和利用，因此大部分基于漏洞的攻击请求报文在经由异构执行体处理时，特定环境配置以外的执行体无法对攻击请求做出响应，在这种情况下，只有异常执行体的响应状态码为2XX，其余执行体均会输出4XX。

通过对拟态Web[15]、拟态路由器[16]、拟态DNS[17]等拟态框架的研究发现，典型的拟态构造裁决主要采用多数一致的表决算法：

按响应输出结果一致的执行体个数作为输出裁决的基本择多依据，进一步地，当所有异构执行体的结果存在不一致时，成为拟态系统产生一次扰动，此时结合反馈控制器实时更新的各执形体的历史置信度作为异构执行体的加权参数[12]，将其归一化之后，选择权重和最大的执行体结果作为输出，响应数据不同的执行体视为异常执行体。反馈控制器会调减异常执行体的历史置信度。

2.2　单一类型攻击引起的防御退化

从传统的基于历史置信度的裁决策略中我们可以看出，每次历经一次扰动，反馈控制器就会

调整异常执行体的历史置信度，一般的，为了简化实施和计算，默认所有异构执行体的初始历史置信度为1，每次被标识为异常的执行体，其历史置信度降为当前值的d倍，即第i个执行体的历史置信权重W_i满足：

$$W_i = d^n, d \in (0.5,1) \#(1)$$

其中n为第i个执行体被裁决为异常的次数，d为置信度衰减系数，简单起见不妨取$d = 0.9$，并讨论如下情形：

对于采用基于历史置信度的裁决策略的拟态架构容器云环境长时间历经单一类型攻击，例如针对Tomcat7.0的漏洞攻击时，在持续一段时间后，拟态架构容器云的异构执行体池中Web服务层为Tomcat7.0的所有执行体的历史置信度都将远低于其他异构执行体（历史置信度为1），不难看出，当所有Web服务层为Tomcat7.0的异构执行体的历史置信度都降为0.5以下，即历经7次被裁决为异常执行体，此时随机选取的3个异构执行体中，如果包含2个web服务层为Tomcat7.0的异构执行体E_1、E_2，另一个是web服务层非tomcat7.0的异构执行体E_3，则根据基于历史置信度的裁决算法，E_3的响应始终会作为多数一致性响应结果输出，在这种情况下，任何针对E_3的攻击行为都会生效，从而形成单一类型攻击的防御退化，导致防御失效。

一般的，当系统连续历经单一类型攻击超过N次，N满足下式时，在使用简单反馈控制器的拟态架构容器云环境中突然遭受另一类型攻击时有极大概率会导致防御失效。

$$N > -\frac{log_d(M - 1)}{p} \#(2)$$

式中N为系统连续历经单一类型攻击的次数，M为同时参与裁决的异构执行体个数，p为具有单一类型攻击漏洞的执行体在异构执行体池中的所占比例。当异构执行体池是分n层冗余构建时：

$$\frac{1}{p} = \prod_{n \neq i} R_n \#(3)$$

不难看出，持续单一类型攻击容忍次数N_{min}和异构执行体池的异构层数n呈指数关系。

可以看出，在使用传统历史置信度权重的裁决策略下，可以通过增加拟态裁决的执行体数目和增加拟态异构资源池的异构层数等方法延缓对单一类型攻击带来的防御退化出现的时间，但无法根本性的避免。

3 基于联合置信权重的拟态裁决策略

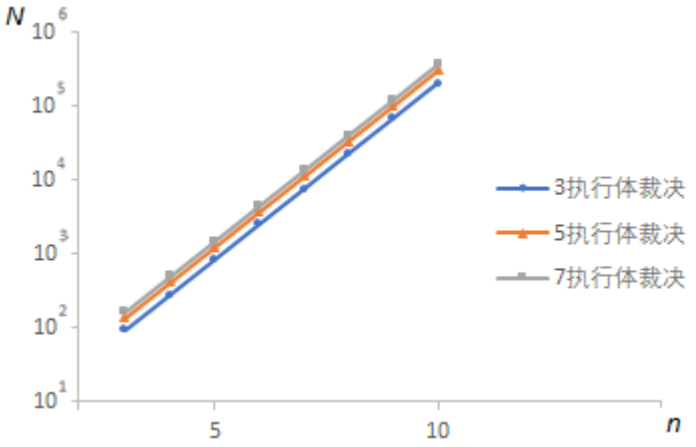
3.1 模型原理

针对使用简单反馈控制器环境下单一类型攻击带来的防御退化，在拟态构造容器云环境下使用基于联合置信权重的拟态裁决策略作为输出裁决器的一致性裁决策略。即每个异构执行体的历史置信度并不是唯一的，而是由和其进行联合裁决的异构体息息相关的，换句话说，每个执行体在与不同执行体一起进行多数一致裁决时具有不同的历史置信度，称为联合历史置信度。每次执行输出裁决时，使用该执行体与其他各执行体的联合历史置信度的和（或算术平均值）作为该执行体的响应输出权重，替代传统的历史置信度进行权重归一化并选择权重和最大的执行体结果作为输出。而反馈控制器可以保持不变，只是对应的将原先的执行体历史置信度的更新修改为对执行体联合历史置信度的更新，方便对已有的拟态容器云平台进行改造。

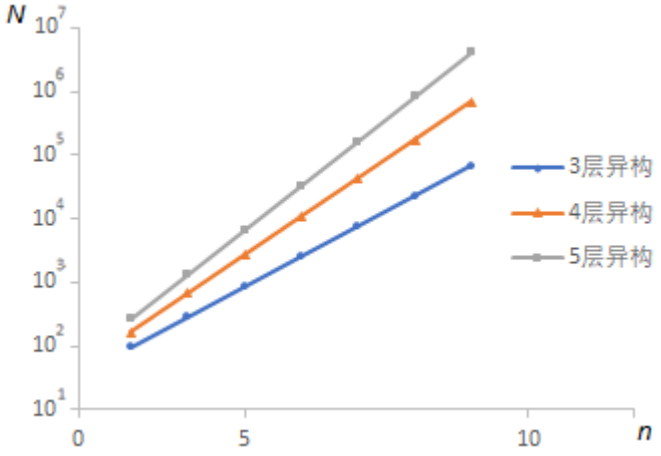
3.2 算法实现

考虑在具有m冗余异构执行体的拟态基础设施环境中，在传统的历史置信度裁决算法中，每一个执行体拥有各自独立的历史置信度，即全局维护m个历史置信度值，初始是一个$m \times 1$的矩阵$C_0 = (1,1,1,\cdots,1)^T$，是一个m个元素均为1的列向量，在每次裁决执行完成后，根据裁决结果左乘裁决结果矩阵$diag(r_1, r_2, r_3, \cdots, r_m)$，当执行体$i$裁决异常时，$r_i$为置信度衰减系数$d$，其他$r$均为1。当系统持续经历足够长时间的单一类型攻击时，则除所有包含攻击漏洞的执行体历史置信度<0.5以外，其余执行体的历史置信度均为1，此时任意选取的3个执行体中若包含2个单一攻击类型漏洞执行体，而另一个不含该单一攻击类型漏洞，此时按照基于历史置信权重的多数一致性表决策略，历史置信度为1的那一个执行体的响应结果即为多数一致输出，显然无法对该执行体其他层次隐含的漏洞做出主动防御。

本算法将全局维护的$m \times 1$历史置信度矩阵推广至$m \times m$，即全局维护$m \times m$的联合历史置信度矩阵，记作$A_{m \times m} = (C_i)$，C_i为$m \times 1$的列向量，表



（a） 不同裁决执行体数目



（b） 执行体异构层数不同

图 2 持续单一类型攻击容忍次数 N_{min} 和异构执行体池的异构层数 n 的关系示意图

示执行体 i 作为参与裁决的异构执行体之一时 m 个异构执行体的历史置信度，同样的，初始状态下 $C_i = (1, 1, 1, \cdots, 1)^T$，是一个 m 个元素均为 1 的列向量，每次根据裁决结果进行如下处理：

1. 如果所有执行体裁决结果均一致，联合历史置信度矩阵不变；

2. 如果执行体裁决出现多数一致性结果，对于输出响应和多数一致性结果相同的执行体 E_{same}，将 $C_{E_{same}}$ 左乘裁决结果矩阵 $diag(r_1, r_2, r_3, \cdots, r_m)$，而对于与多数一致性结果不同的执行体 E_{diff}，对应的 $C_{E_{diff}}$ 保持不变。

在下一次输出裁决时，各异构执行体的置信度权重由联合历史置信度矩阵和当前参与输出裁决的执行体共同决定，此时 $C' = A \times (e_1, e_2, e_3, \cdots, e_m)^T$，当异构执行体 i 参与拟态裁决时，$e_i = 1$，否则，$e_i = 0$。

使用 C' 代替传统的历史置信度矩阵作为置信度权重进行多数一致性裁决，由于异构执行体的联合置信权重只会在特定执行体加入裁决器中时才会下降，因此任意时刻异构执行体所表现出的历史置信度权重与当前进入裁决器的其他异构执行体密切相关。使得在进行多数一致性裁决时，各执行体的权重设置能够有的放矢。

4 实验验证

以部署在拟态构造容器云上的金融理财服务为例，通过构建异构镜像，形成异构执行体资源池，为验证方便此处通过 2 层，每层 3 个冗余度的方式，共构建 9 个异构镜像。

各异构镜像的具体参数如下：

拟态构造容器云随机使用异构镜像组成 5 个异构执行体，每个异构执行体的初始历史置信度为

```
[root@mimic ~]# docker images
REPOSITORY              TAG             IMAGE ID        CREATED         SIZE
jrlc/centos/tomcat      lastest         d89ee7f82794    2 hours ago     558.8 MB
jrlc/fedora/tomcat      lastest         c1232db06c06    2 hours ago     638 MB
jrlc/ubuntu/tomcat      lastest         561a64ad0dd5    2 hours ago     578.9 MB
jrlc/centos/jboss       lastest         230e34968ad4    2 hours ago     577.8 MB
jrlc/fedora/jboss       lastest         689c31675699    2 hours ago     599.8 MB
jrlc/ubuntu/jboss       lastest         c1232db06c06    2 hours ago     585.2 MB
jrlc/centos/resin       lastest         08393e824c32    2 hours ago     600.9 MB
jrlc/fedora/resin       lastest         d05612441714    2 hours ago     639.1 MB
jrlc/ubuntu/resin       lastest         8cf1bfb43ff5    2 hours ago     611.4 MB
```

图 3　异构镜像池

表 1　异构执行体冗余度

编号	镜像名称	操作系统	Web服务器
1.	jrlc/centos/tomcat	CentOS 7.6	Tomcat 7.0.81
2.	jrlc/fedora/tomcat	Fedora 31	Tomcat 7.0.81
3.	jrlc/ubuntu/tomcat	Ubuntu 16.04	Tomcat 7.0.81
4.	jrlc/centos/jboss	CentOS 7.6	Jboss 4.0.4
5.	jrlc/fedora /jboss	Fedora 31	Jboss 4.0.4
6.	jrlc/ubuntu /jboss	Ubuntu 16.04	Jboss 4.0.4
7.	jrlc/centos/resin	CentOS 7.6	Resin 4.0.29
8.	jrlc/fedora /resin	Fedora 31	Resin 4.0.29
9.	jrlc/ubuntu /resin	Ubuntu 16.04	Resin 4.0.29

1，使用相同异构镜像的执行体的历史置信度相同。输出裁决器从5个异构执行体的响应结果中选择加权多数一致的响应结果作为最终的拟态输出。对裁决响应不一致执行体进行替换，同时加入随机扰动机制，对长期未裁决出异常的执行体同样进行替换。

在异构镜像池中，Tomcat 7.0.81镜像中包含CVE-2017-12615漏洞，通过PUT方法可以实现上传文件功能[18]。Resin镜像具有CVE-2012-2969的漏洞，远程攻击者在URL中可以通过任意扩展名的DOS设备文件名读取系统上的任意COM或LPT设备连续的数据流、进而通过目录遍历获取金融理财服务的WEB-INF目录中的文件内容[19]。

分别使用传统历史置信度权重和联合历史置信度作为裁决策略进行多数一致性裁决，使部署在拟态构造容器云上的金融理财服务长时间历经单一类型的针对Tomcat 7.0漏洞的攻击，每次出现非一致性输出记为一次扰动，各异构执行体的历史置信度将会随扰动次数的增加而变化，通过反馈控制器输出每次扰动后的各执行体的历史置信度。

通过实验可以看出，在使用同样的裁决异常置信度衰减率的情况下，同样历经15次针对Tomcat 7.0服务器的攻击，此时更换使用针对Resin服务器的攻击，执行体1，3，5，7，9进入裁决器，此时执行体1，3，5为没有漏洞的返回结果，执行体7，9为一致的异常响应，使用传统历史置信度权重时，执行体1，3，5响应的权重和为1.98，执行体7，9响应的权重和为2。所以此时执行体7，9的响应结果会作为多数一致结果输出，产生逃逸现象。而使用基于联合历史置信度的异构执行体权重时，执行体1，3，5响应的权重和为2.52，执行体7，9响应的权重和为2。因此执行体1，3，5的响应结果会作为多数一致结果输出，达到拟态防御的目的。

表 2　异构执行体历史置信度随扰动次数变化情况

执行体编号 扰动次数	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1
1	0.9	1	0.9	1	1	1	1	1	1
2	0.9	0.9	0.9	1	1	1	1	1	1
3	0.9	0.9	0.81	1	1	1	1	1	1
4	0.81	0.9	0.81	1	1	1	1	1	1
5	0.81	0.81	0.81	1	1	1	1	1	1
6	0.73	0.81	0.73	1	1	1	1	1	1
7	0.73	0.73	0.73	1	1	1	1	1	1
8	0.66	0.73	0.66	1	1	1	1	1	1
9	0.66	0.66	0.66	1	1	1	1	1	1
10	0.66	0.66	0.59	1	1	1	1	1	1
11	0.59	0.66	0.59	1	1	1	1	1	1
12	0.59	0.59	0.59	1	1	1	1	1	1
13	0.53	0.59	0.53	1	1	1	1	1	1
14	0.53	0.53	0.53	1	1	1	1	1	1
15	0.48	0.53	0.48	1	1	1	1	1	1

经过多次实验，在不同的异常置信度衰减率情况下，使用基于联合历史置信度的拟态权重策略在经过500次单一类型攻击后仍能对其他类型的攻击做出主动防御，相比使用传统历史置信度的拟态权重策略可以明显延长引起拟态容器云出现防御退化的单一类型攻击次数。

表 3　基于联合历史置信度的异构执行体权重随扰动次数变化情况

执行体编号 扰动次数	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1
1	0.94	1	0.94	1	1	1	1	1	1
2	0.94	0.92	0.94	1	1	1	1	1	1
3	0.96	0.96	0.90	1	1	1	1	1	1
4	0.88	0.96	0.90	1	1	1	1	1	1
5	0.90	0.90	0.92	1	1	1	1	1	1
6	0.85	0.88	0.85	1	1	1	1	1	1
7	0.84	0.80	0.84	1	1	1	1	1	1
8	0.84	0.79	0.85	1	1	1	1	1	1
9	0.80	0.75	0.77	1	1	1	1	1	1
10	0.74	0.75	0.77	1	1	1	1	1	1
11	0.76	0.75	0.77	1	1	1	1	1	1
12	0.81	0.75	0.75	1	1	1	1	1	1
13	0.75	0.75	0.75	1	1	1	1	1	1
14	0.84	0.79	0.84	1	1	1	1	1	1
15	0.76	0.72	0.76	1	1	1	1	1	1

表 3　不同异常置信度衰减率情况下两种策略的持续单一类型攻击容忍次数

置信度衰减率	**0.7**	**0.8**	**0.9**	**0.92**	**0.95**	**0.98**	**0.99**
传统策略	28	44	92	119	193	490	>500
本文策略	>500	>500	>500	>500	>500	>500	>500

根据实验数据可以看出，基于联合历史置信度的裁决策略下，异常执行体的衰减较慢，而且并不是持续递减的，这是由于每个异构执行体的历史置信度权重并不是仅由扰动异常决定的，而是结合了产生扰动时的所有参与裁决的执行体信息，是根据裁决结果有条件的降低异常执行体的历史置信度，避免了异构执行体持续历经单一类型攻击后的防御退化，在降低异构执行体的置信度的同时，保持了异构执行体对其他攻击行为的冗余防御效果。通过使用联合历史置信度，不仅提高了拟态构造容器云上应用服务的内生安全防御能力，而且提高了整个拟态构造容器云的资源利用率。

5　结论

本文基于传统的拟态异构执行体历史置信度，提出了基于联合历史置信度的拟态权重策略。将拟态系统全局维护的 $m \times 1$ 历史置信度矩阵推广至 $m \times m$ 矩阵，使每次参与裁决的异构执行体的置信度权重能够反映出其相对于当前进入裁决器的其他异构执行体的历史置信度情况，使裁决加权能够有的放矢，而不是使用一概而论的异构执行体权重，通过将基于联合历史置信度的拟态权重策略应用于拟态构造容器云环境中，相较于应用传统的历史置信度权重策略，在长期历经单一类型攻击后，表现出了更好的拟态防御效果，并没有受到单一类型攻击训练的影响，同时保持了拟态构造容器云平台的异构执行体资源池的活性，提高了整个拟态构造容器云的资源利用率。

参考文献：（参考文献格式参照2015年新标准GBT 7714-2015）

［1］ 梅雅鑫.“新基建”时代，5G势不可挡云网融合正当时［J］.通信世界，2020（21）：11-12.

［2］ 邱达超，王海燕，李振华.谈移动互联网时代的手机信息安全［J］.电信工程技术与标准化，2020，33（08）：21-26.

［3］ 2019年全球及中国云计算行业发展现状及2019-2020年云计算行业发展趋势预测［EB/OL］.［2020-08-25］.http：//www. chyxx. com/industry/201907/765109.html.

［4］ Cloud Computing Top Threats ［EB/OL］.［2020-08-25］. https：//cloudsecurityalliance.org/ topthreats.

［5］ 谭文安，查安民，陈森博.优化粒子群的云计算任务调度算法［J］.计算机技术与发展，2016，26（07）：6-10.

［6］ 王鹏.容器云环境中基于信任的调度算法的研究［D］.桂林理工大学，2018.

［7］ 张琼.云环境中安全容器动态迁移的研究［D］.北京邮电大学，2019.

［8］ 牛侃.基于博弈模型的云服务资源调度技术研究［D］.解放军信息工程大学，2017.

［9］ 季新生，徐水灵，刘文彦，仝青，李凌书.一种面向安全的虚拟网络功能动态异构调度方法［J］.电子与信息学报，2019，41（10）：2435-2441.

［10］ 张杰鑫，庞建民，张铮，郜铭，张浩，聂广来.面向拟态构造Web服务器的执行体调度算法［J］.计算机工程，2019，45（08）：14-21.

［11］霍立田，邵培南，徐李定，徐骏.拟态通用运行环境的资源管理与调度技术［J］.计算机工程，2020，46（02）：159-169.

［12］武兆琪，张帆，郭威，卫今，谢光伟.一种基于执行体异构度的拟态裁决优化方法［J］.计算机工程，2020，46（05）：12-18.

［13］张淼，季新生，刘文彦，扈红超，霍树民.基于放置策略动态化的共存攻击主动防御方法［J］.信息工程大学学报，2018，19（03）：257-263.

［14］邬江兴.网络空间拟态防御原理：下册［M］.2版.

［15］仝青，张铮，张为华，等.拟态防御Web服务器设计与实现［J］.软件学报，2017，28（4）：883-897.

［16］马海龙，伊鹏，江逸茗，等.基于动态异构冗余机制的路由器拟态防御体系结构［J］.信息安全学报，2017，2（1）：29-42.

［17］王禛鹏，扈红超，程国振.一种基于拟态安全防御的DNS框架设计［J］.电子学报，2017，45（11）：2705-2714.

［18］CVE - CVE-2017-12615［EB/OL］.［2020-08-25］. http：//cve. mitre. org/cgi-bin/cvename.cgi？ name=CVE-2017-12615.

［19］CVE - CVE-2012-2969［EB/OL］.［2020-08-25］. http：//cve. mitre. org/cgi-bin/cvename.cgi？ name=CVE-2012-2969.

基于UVM的拟态调度器芯片SOC验证平台设计

钟丹[1]，徐庆阳[1]，杜侃[2]，王家林[1]，朱婧瑀[1]

[1]天津市滨海新区信息技术创新中心，天津 300457；

[2]西安紫光国芯半导体有限公司，西安 710075

摘　要：在新的网络空间安全形势下，针对当前网络空间外部威胁和内部漏洞相互交织问题，以拟态调度器安全处理架构为核心，搭建了基于UVM的拟态调度器芯片SOC验证平台。对验证平台的各个组件进行了阐述，并对调度器系统进行了功能验证和安全测试。采用黑盒验证和白盒验证，模拟网络环境发送各种协议的报文，构造正常报文和攻击报文交织的场景，通过自动化的检查，证明了拟态调度器芯片能有效防御攻击报文。

关键词：网络空间安全、拟态调度器、SOC验证、自动化检查、UVM

The Design of Mimics Scheduler SOC Verification Platform Based on UVM

Zhong Dan

Information Technology Innovation Center of Tianjin BINhai New Area

Abstract: In the new cyberspace security situation, aiming at the interweaving of external threats and internal vulnerabilities in cyberspace, a SoC verification platform for mimicry scheduler chip based on UVM was built with the security processing architecture of mimicry scheduler as the core. Each component of the verification platform was introduced and the function of the system was tested. It has simulated the network environment by sending packets of various protocols to construct the scenario of interweaving normal message and attack message through black box verification and white box verification. It is proved that the mimicry scheduler chip can effectively defend attack messages and protect network security through automatic inspection.

Key words: netwrok security; mimicry scheduler; SoC verification; automatic inspection; UVM

1　概述

网络空间在蓬勃发展的同时，正面临着严峻的安全形势，存在大量针对网络空间的恶意攻击事件，另外网络系统复杂，不可避免的存在漏洞，因此网络空间既有外部威胁，又有内部安全漏洞问题，两者相互交织，安全风险严峻复杂。在新的网络空间安全形势下，基于先验知识的传统防御手段难以应对各种攻击，需要转变防御思路，定义新的防御边界，巩固防线纵深，从被动迈向内生安全的主动防御[1]。

拟态调度器芯片采用了拟态安全防御技术，能从主动性、变化性和随机性中获得有利的内生防御态势，通过拟态环境进行动态变化，对攻击者则表现为难以观察和预测，从而大幅度增加包括未知的可利用的漏洞和后门在内的攻击难度和

成本[2-3]。其主要原理为：输入代理器收到外部服务请求后，根据冗余控制器的代理策略，将外部服务请求发送给选定的一个或者多个异构功能等价体；异构功能等价体在接收到服务请求后工作运行，输出服务响应发送至输出代理器，并将拟态裁决参数发送给冗余控制器；输出代理器接收到服务响应后，根据冗余控制器的输出仲裁策略，选择其中一个异构功能等价体的输出作为外部服务响应进行发送。具体的拟态裁决参数为多个异构体的输出报文。冗余控制器[4]对多个异构体之间的输出报文进行比较时，需要对同一外部输入报文产生的输出报文进行比较，才能从中裁决出正常的异构功能等价体[5]。

本文针对拟态调度器芯片的架构，搭建了基于UVM的SOC验证平台。在控制平面，通过验证CPU BOOT，配置各表项和寄存器，验证了CPU的功能。在数据平面，为了验证协议一致性，采用以太网VERIFICATION IP和PCIE VERIFICATION IP模拟了异构执行体的行为，用户侧的行为，交换芯片的行为，同时采用参考模型和SCOREBOARD实现了数据平面报文的自动比对。同时，本文还构建了正常流量和攻击流量交织的场景，拟态调度器芯片对攻击流量进行了识别，最终输出了可信度高的正常报文[6-7]，证明了拟态调度器芯片能有效的防御攻击报文，捍卫网络安全。

2 拟态调度器芯片架构说明

在异构多核处理的背景下，单个处理器可能由于内部故障或者受外部攻击输出错误结果，此时需要调度器对各处理器的输出结果进行判决并输出调度器认为正确的结果。本芯片在一些应用场景下可以通过以太网接口与拟态异构处理器及交换芯片互联。系统支持底层8路Serdes接口，上层通过NOC配置选择上层PCIE控制器或者GE MAC或者XGE MAC，再实现底层Serdes资源共享的情况下实现上层Mac灵活调配。

其结构如图1所示，图中灰色线框内描述了芯片的主要结构。

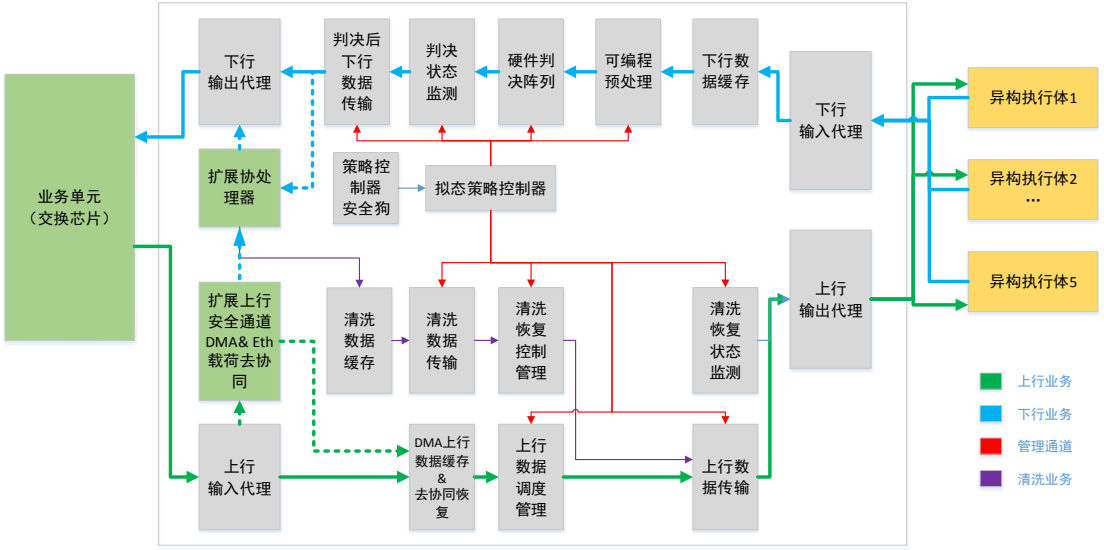


图1 拟态调度器芯片架构图

系统采用标准的SoC架构，包括双核CPU、内存、内部总线矩阵、低速外设等。考虑到业务处理、拟态策略计算的处理需求以及拟态安全隔离的需要，系统拟采用双CPU架构。其中处理SDK的CPU性能要求比较高，并且需要跑linux操作系统，所以考虑选用性能比较强的CPU。用来处理拟态策略的CPU相对来说性能要求不高。但是考虑到设计验证复杂度、时间风险等，考虑采用同构设计，两个CPU采用同样的CPU核。

内部总线采用AMBA AHB总线，由于存在多个Master及Slave，可以采用通用的Bus Matrix将他们互联，实现总线互通，另外实现异步桥接。

系统中存在多个master及slave设备，这些设备通过Bus Matrix互联，并且进行全局地址分配。

系统包含UART、I2C、Timer、Watchdog、GPIO、SPI等多个低速外设，通过AHB Lite挂接到Bus Matrix上，暂定总线主频50MHz。

另外，由于SDK CPU及Strategy CPU都有访问低速外设的需求，并且需要将两组CPU做好严格物理隔离，所以需要在系统中例化两套低速外设。

拟态调度器作为异构引擎与外部唯一的数据传输接口，需要完成异构引擎输入/输出信息的处理，协议解析，数据分发与数据判决。还需要进行数据传输流量进行监测，拟态策略控制上行及下行数据分发及判决策略，看门狗实现系统日志记录及系统错误恢复，系统还设有清洗恢复机制，用来对异构引擎进行清洗恢复。

主要完成工作分为以下五部分：一，异构引擎输出信息处理（下行）；二，异构引擎输入信息处理（上行）；三，拟态策略管理；四，看门狗；五，清洗管理。

其中异构引擎输入/输出信息处理还涉及到数据接口数据传输的设计与实现，拟态调度器主要数据传输接口有以太网接口，PCIE接口，每个接口数据传输的方式不同，需要针对不同接口进行设计实现。

3 拟态调度器芯片验证平台设计

3.1 UVM验证方法学简介

IC行业按照摩尔定律快速发展，晶体管的数量越来越多，整个系统越来越复杂。此时，单纯的Verilog验证已经难以满足条件。1999年，OSCI成立，致力于System C的开发。但是System C存在内存泄露的问题，并且在构建异常的测试用例时显得力不从心，而SystemVerilog是一个Verilog的扩展集，可以完全兼容Verilog，无论是对算法类或者非算法类的设计，都能轻松应对。UVM验证方法学为SystemVerilog搭建验证平台提供了丰富的库函数，构建了基本的平台框架，方便验证工程师将更多的精力致力于功能验证，而不是平台构建。UVM验证是一种覆盖率驱动的带约束的随机测试，能将验证的空间缩小到重点的范围内，在时间和效率上为项目的快速收敛提供了重要的依据。同时验证人员通过编写参考模型，将参考模型的输出和待测对象的输出进行自动化的比对，提高了验证的效率和准确度。本章节将讲述本文设计的验证平台的具体结构。

3.2 验证平台整体结构

全系统验证平台对于数据平面，采用ETH VERIFICATION IP或者PCIE VERIFICATION IP发送激励，SCOREBOARD端到端比对报文的形式，对于控制平面，采用Tests，即用例，将CPU

控制程序生成BIN文件，加载到flash，CPU再执行BIN文件的形式实现。如图2拟态调度器soc验证平台，验证平台南向用户侧挂接一个UART VERIFICATION IP，北向异构执行体挂接五个UART VERIFICATION IP。提供5个以太网VERIFICATION IP和5个PCIE RC VERIFICATION IP模拟5个异构执行体。提供1个1G以太网VERIFICATION IP和1个10G以太网VERIFICATION IP模拟用户侧发送数据流。提供1个PCIE EP VERIFICATION IP模拟对接的交换侧芯片。提供挂接的DDR模型。提供挂接的FLASH模型。南北向交互的数据流能实现自动比对。

平台中包括如下组件：

（1） protocol_env： 主要实现对各底层env的封装以及各个组件之间的连接。

（2） protocol_env_cfg：主要用于对 验证平台进行全局配置。

a，包括对eth_env和eth_env_cfg的实例化开关，对eth_env_cfg中部分Verification IP属性的配置，以及对端口属性的配置。

b，包括对pcie_env和pcie_env_cfg的实例化开关，对pcie_env_cfg中部分Verification IP属性的配置，以及对端口属性的配置。

（3） eth_env：eth VERIFICATION IP的顶层，包含了eth VERIFICATION IP的所有组件，用于向DUT中发送标准的eth协议数据包。

（4） eth_env_cfg：eth VERIFICATION IP的配套配置文件，用于VERIFICATION IP的详细属性配置。

（5） pcie_env：pcie VERIFICATION IP的顶层，包含了pcie VERIFICATION IP的所有组件，用于向DUT中发送标准的pcie协议数据包。

（6） pcie_env_cfg：pcie VERIFICATION IP的配套配置文件，用于VERIFICATION IP的详细属性配置。

（7） i2c_env：完成i2c接口的激励发送功能。

（8） jtag_env：完成jtag接口的激励发送功能。

（9） uart_env：完成uart接口的激励发送功能。

（10） scoreboard：主要实现：

a，上行ETH VERIFICATION IP 1G激励和上行输出包的比对，如图2中的Igress_eth_chk。

b，上行ETH VERIFICATION IP 10G激励和上行输出包的比对，如图2中的Igress_eth_chk。

c，下行ETH VERIFICATION IP X5或者PCIE VERIFICATION IP激励和下行输出数据包的比对，如图2中的Egress_pkt_chk。

d，uart收发数据包的比对，如图2中的uart_pkt_pre&chk。

（11） DDR0_model，DDR1_model：挂接的DDR模型，模拟DDR的行为。

（12） EEPROM：挂接的eeprom模型，模拟eeprom的行为。

（13） SD_CARD：挂接的SD_card模型，模拟SD_card的行为。

（14） Tests是加载到flash或者SD_card的CPU程序，由软件，如gcc编译生成BIN文件。

（15） BOOTLOADER是加载到flash的CPU boot程序，由软件，如gcc编译生成BIN文件。

（16） RAL_MODEL：寄存器模型，用于在验证平台中进行寄存器访问。

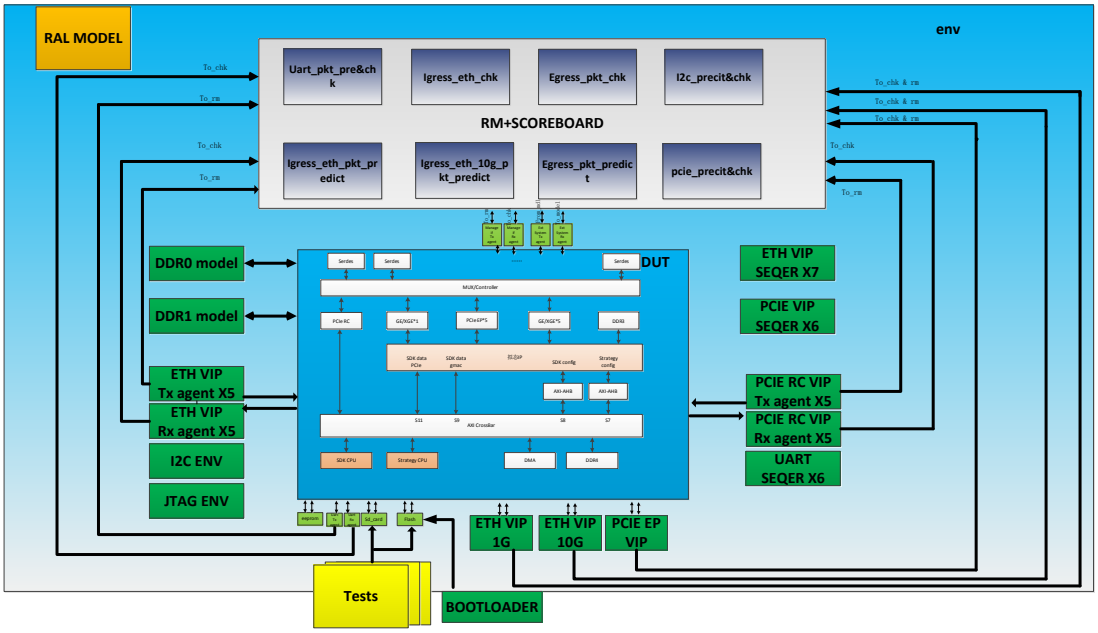


图2 拟态调度器soc验证平台图

3.3 主要数据流分析以及验证流程

如图3，拟态交换机应用场景下主体数据处理流程如下：

（1） 拟态调度器芯片通过PCIE/GE/XGE接口（VERIFICATION IP模拟）从交换芯片接收上送的数据报文；

（2） 根据连接接口是PCIE还是GE/XGE选择不同的数据处理流程，数据最终进入协议无关拟态调度器IP；

（3） 拟态调度器IP根据异构执行体相关的配置进行数据复制、分发；

（4） 复制、分发之后的报文通过PCIE/GE/XGE接口上送至异构执行体CPU（PCIE VERIFICATION IP和以太网VERIFICATION IP模拟），至此上行业务处理完毕；上行激励和上行输出包的比对，如图2中的Igress_eth_chk。

（5） 各个异构执行体处理完毕的数据报文通过PCIe/GE/XGE接口送至拟态调度器芯片；

（6） 拟态调度器IP对各处理器的输出结果进行判决并输出调度器认为正确的结果；

（7） 获取上行业务处理是缓存在DDR中的数据结合调度结果进行数据报文组装；

（8） 将组装好的数据报文通过PCIE接口下发至交换芯片，通过GE/XGE接口下发至用户侧。至

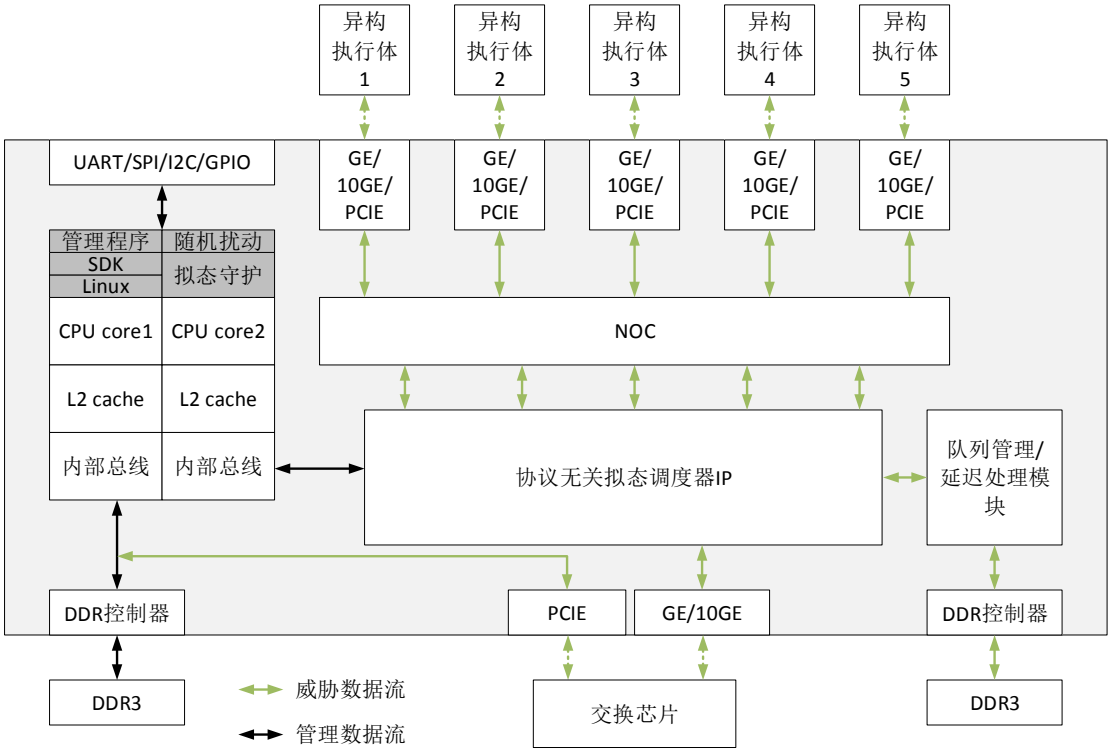


图 3 拟态调度器数据流向图

此下行业务处理完毕，全部的拟态调度器芯片业务也处理完毕。下行 ETH VERIFICATION IP X5 或者 PCIE VERIFICATION IP 激励和下行输出数据包的比对，如图 2 中的 Egress_pkt_chk。

3.4 soc CPU 相关验证

对于 soc CPU 的验证，包括 CPU 的 boot，以及对各表项和寄存器的读写访问。加载 C 程序，进行读写测试，完成对于芯片所有 Register 和 Memory 空间的配置与管理。

3.5 CPU 外设验证

对于 uart 的验证，UART ENV 对接 uart 接口，通过 uart driver 发送满足 uart 接口的时序，发送字符，一份发给 dut，一份发给 rm，用于预期。Tests 控制 CPU 将 CPU 收到的来自 uart 的字符再发送出来，uart rx agent 中的 monitor 采集到字符后，发送给 scorboard，通过比较发送和收到的字符，检测 uart 收发包的正确性。

对于 flash 的验证，加载 CPU 控制程序到 flash，CPU 先从 flash 读取程序到内部 ram，再执行。通过这个过程验证 flash 加载的正确性。

对于 i2c 的验证，eeprom 设备模型作为 i2c 从设备，挂载在 i2c 总线上，让 i2c dut 在主模式下或者 boot 模式时访问。I2c 为从模式时，采用 I2c 时序的 driver，通过 sequence 发包，读写全芯片寄存器，集成 RAL 模型，在图 2 的 I2c_predict&chk 完成读写比对。

对于 JTAG 的验证，采用 JTAG 时序的 driver，通过 sequence 发包，读写全芯片寄存器。集成 RAL 模型，在图 2 的 Jtag_predict&chk 完成读写比对。

4 测试结果分析

本文基于图 2 所示的验证平台，对拟态调度器芯片进行了白盒和黑盒测试。测试结果分析见图 4-图 7。

4.1 CPU 测试分析

图 4 中 sck，si，so 总线波形显示的是 CPU 加载 boot，apb 总线接口波形显示的是 CPU 读写 uart 外设。

4.2 拟态调度器下行数据流分析

图 5 是下行数据流波形图，以 i 打头的信号是异构执行体 CPU 的输出，以 o 打开的信号是南向 3 个以太网口的报文输出。下行 ETH VERIFICATION IP X5 或者 PCIE VERIFICATION IP 激励和下行输出数据包的比对，在图 2 中的 Egress_pkt_chk 比对通过。

4.3 判决攻击报文分析

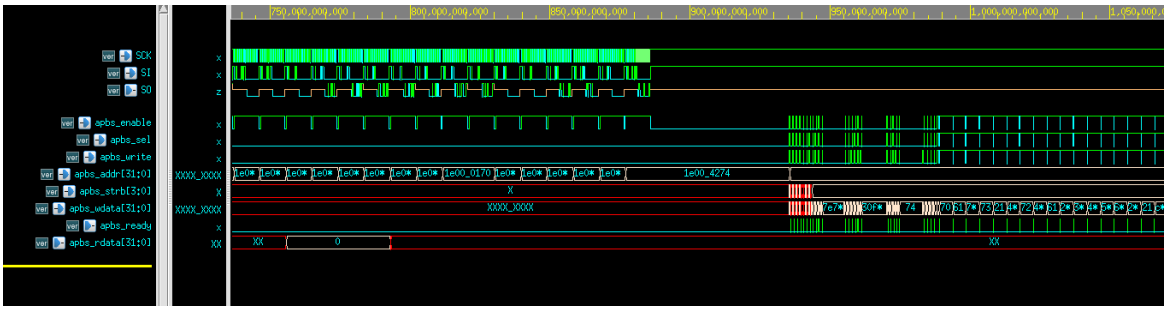


图 4　CPU BOOT 和访问外设波形图

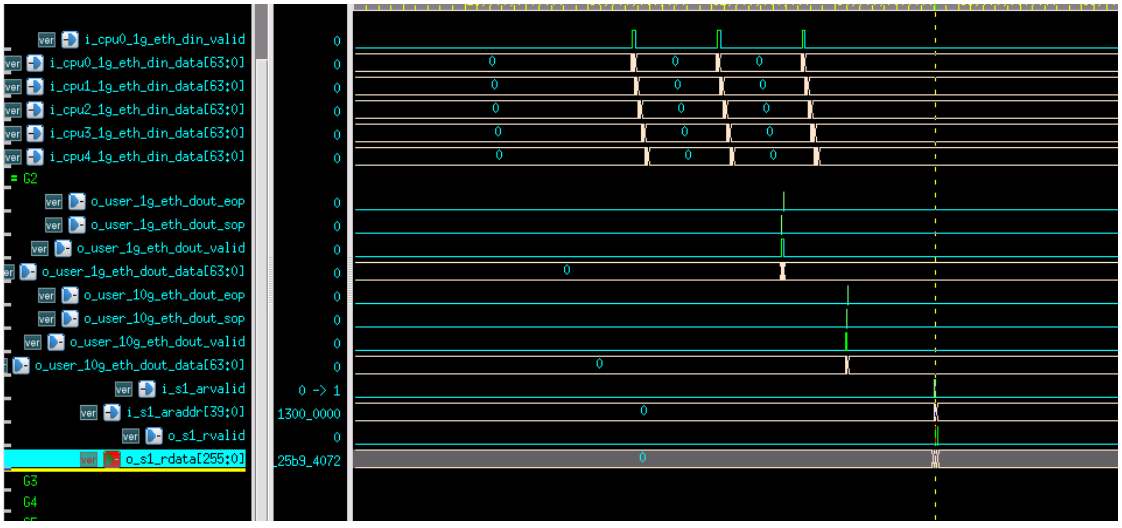


图 5　下行数据流波形图

图 6 示意了 3 个异构执行体情况下芯片识别攻击报文，输出可信度报文的过程。

图 6 中共有 3 个 CPU 通道，选择 CPU0 为主通道，其中代号 p 表示一种协议类型，例如 tcp，udp，ipv4，ipv6 等，n 表示该种协议的第几个包，例如 p1n1 代表协议 1 的第 1 个包，p2n2 代表第 2 种协议的第 2 个包。

南向用户侧发送各种协议的混合报文，其中包含攻击报文，到达北向 3 个 CPU 后，3 个 CPU 分别发出响应报文。

该 3 路混合协议交织攻击报文的流量到达判决模块后，判决模块根据规则进行可信度计算，最终判决后的输出如图 6 所示，最终输出的报文有效过滤了攻击报文（p2n2）`，证明了拟态判决的正确性。

本文设计的验证平台在图 2 的 Egress_pkt_chk 中根据拟态判决的规则编写了参考模型，在激励中模拟了 5 路 CPU 受到攻击的各种随机情况，拟态判决芯片的输出都和参考模型输出的结果比对通过，证明了拟态判决芯片能有效抵抗攻击报文。

4.4 覆盖率分析

UVM 验证是一种覆盖率驱动的带约束的随机测试，需要通过代码覆盖率分析来确保测试的完备性。图 7 挑选了其中的 uart 模块作为例子。通过图 7 可以看到，uart 整体覆盖率达到 90.03%。其中行覆盖率 100%，翻转覆盖率 82.62%，状态机覆盖率 100%，条件覆盖率 75.61。我们对未覆盖部分进行了分析和解释，撰写了分析报告，最终覆盖率达到了可解释的 100%。

其他模块的代码覆盖率处理与此类似。

5 结束语

本文设计的针对拟态调度器芯片的架构的 SOC 验证平台，经过控制平面和数据平面对芯片各个功能的验证，保证了芯片各项功能的快速收敛。同时，本文构建了正常流量和攻击流量交织

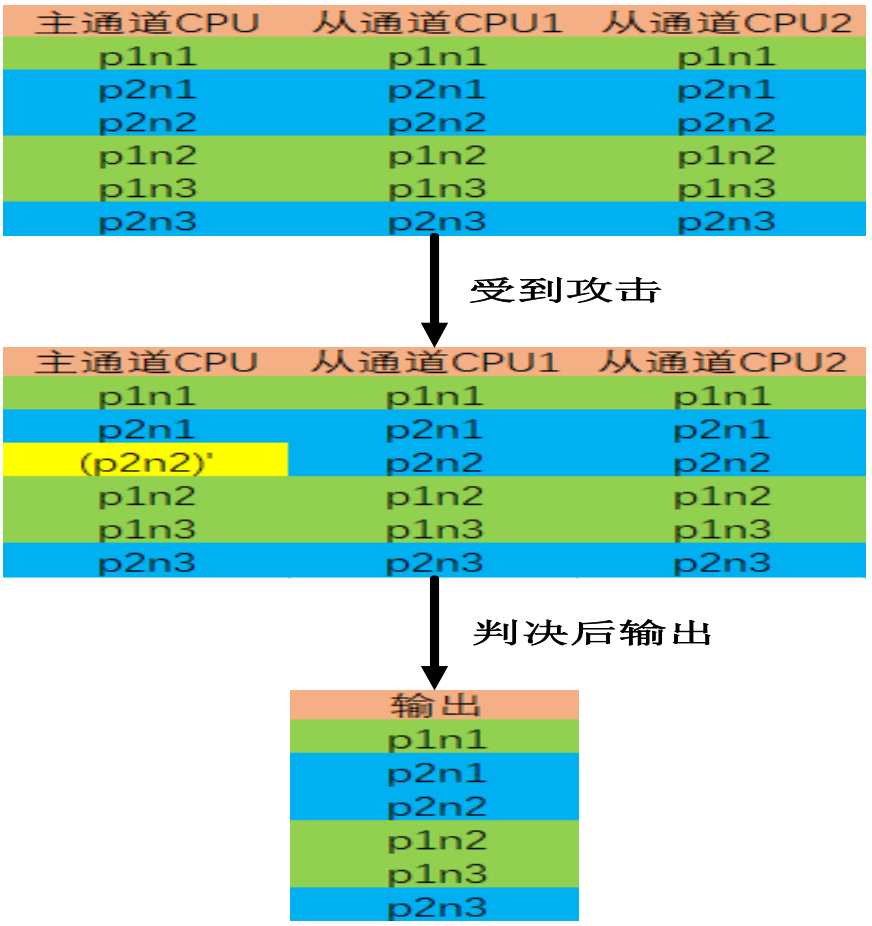
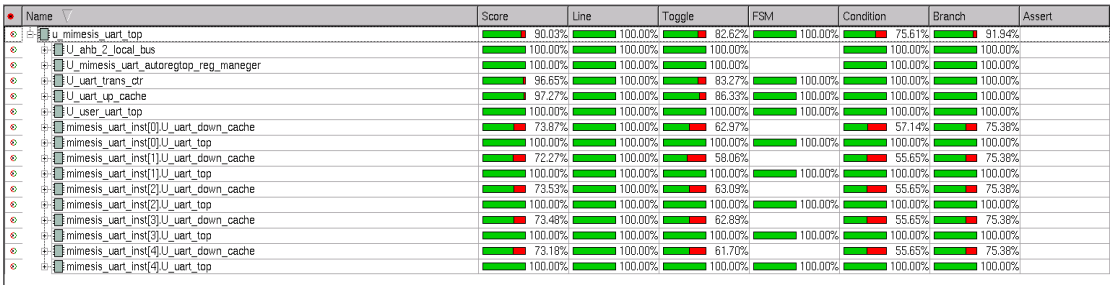


图 6　判决攻击报文示意图



Name	Score	Line	Toggle	FSM	Condition	Branch	Assert
u_mimesis_uart_top	90.03%	100.00%	82.62%	100.00%	75.61%	91.94%	
U_ahb_2_local_bus	100.00%	100.00%	100.00%		100.00%	100.00%	
U_mimesis_uart_autoregtop_reg_maneger	100.00%	100.00%	100.00%		100.00%	100.00%	
U_uart_trans_ctr	96.65%	100.00%	83.27%	100.00%	100.00%	100.00%	
U_uart_up_cache	97.27%	100.00%	86.33%	100.00%	100.00%	100.00%	
U_user_uart_top	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	
mimesis_uart_inst[0].U_uart_down_cache	73.87%	100.00%	62.97%		57.14%	75.38%	
mimesis_uart_inst[0].U_uart_top	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	
mimesis_uart_inst[1].U_uart_down_cache	72.27%	100.00%	58.06%		55.65%	75.38%	
mimesis_uart_inst[1].U_uart_top	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	
mimesis_uart_inst[2].U_uart_down_cache	73.53%	100.00%	63.09%		55.65%	75.38%	
mimesis_uart_inst[2].U_uart_top	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	
mimesis_uart_inst[3].U_uart_down_cache	73.48%	100.00%	62.89%		55.65%	75.38%	
mimesis_uart_inst[3].U_uart_top	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	
mimesis_uart_inst[4].U_uart_down_cache	73.18%	100.00%	61.70%		55.65%	75.38%	
mimesis_uart_inst[4].U_uart_top	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	

图 7　uart 代码覆盖率图

的场景，通过波形和判决结果分析，拟态调度器芯片对攻击流量进行了识别，最终输出了可信度高的正常报文，证明了拟态调度器芯片能有效的防御攻击报文。

6 参考文献:

[1] 邬江兴．网络空间拟态防御导论[M]．北京:科学出版社，2017.
WU J X. Introduction to cyberspace mimic defense[M]. Beijing: Science Press，2017.

[2] Knight, Hyneman F. "Risk, uncertainty and profit. "Houghton Mifflin Company，1921.

[3] octopus""Mimic. Wikipedia, the Free Encyclopedia. https://en.wikipedia. org/wiki/Mimic_octopus.

[4] Voas J, Ghosh A, Charron F, et al. "Reducing uncertainty about common-mode failures. "In Proc. IEEE Symp. Software Reliability Engineering (SRE' 97)，pp. 308-319，1997.

[5] 魏帅, 于洪, 顾泽宇, 等．面向工控领域的拟态安全处理机架构[J]．信息安全学报，2017(1)：54-73.
WEI S, YU H, GU Z Y, et al. Architecture of mimic security processor for industry control system[J]. Journal of Cyber Security，2017(1):54-73.

[6] 张铮, 马博林, 邬江兴．web 服务器拟态防御原理验证系统测试与分析[J]．信息安全学报，2017(1)：13-28.

ZHANG Z, MA B L, WU J X. The test and analysis of prototype of mimic defense in web servers[J]. Journal of Cyber Security，2017（1）：13-28.

[7] 上海市科学技术委员会给国家科技部高新技术发展及产业化司的专题报告，"拟态防御原理验证系统测试评估工作情况汇总"，2016. 8

一种交换芯片延迟测试方法

李庆龙，汪欣，王盼
天津市滨海新区信息技术创新中心，天津 300450

摘　要：交换芯片支持FC-AE-ASM、RapidIO和以太网三种协议，以及协议转换功能，依托协议转换的功能，可以在没有RapidIO 3.0协议的标准测试仪和异构协议的标准测试仪的情况下，只使用一个以太网测试仪与一颗交换芯片相连接，对交换芯片进行协议转换，输出要测试的协议，作为被测交换芯片的输入，能够正常完成交换芯片RapidIO和异构协议的延迟测试，对交换芯片的延迟性能指标做出有力的依据。

关键词：交换芯片、RapidIO、协议转换、异构协议

Method for testing delay of exchange chip

LI Qinglong, WANG Xin, WANG Pan
Tianjin Binhai New Area Information Technology Innovation Center,Tianjin 300450,China

Abstract: The switch chip supports FC-AE-ASM, RapidIO and Ethernet three protocols, as well as the protocol conversion function. Relying on the protocol conversion function, it can be used in the absence of the standard tester of the RapidIO 3. 0 protocol and the standard tester of the heterogeneous protocol Next, only use an Ethernet tester to connect with a switch chip, perform protocol conversion on the switch chip, and output the protocol to be tested as the input of the tested switch chip, which can normally complete the delay of the switch chip RapidIO and heterogeneous protocols The test provides a strong basis for the delay performance index of the switching chip.

Key words: switch chip; RapidIO; protocol conversion; heterogeneous protocol

1　引言

近几年国家非常重视国产芯片的发展，电子技术得到了快速的发展，使得国内的芯片行业有了很快的进步，芯片的功能及性能也有了很大的提升。交换芯片在通信行业中起着非常重要的角色，交换芯片的功能及性能对通信质量的影响也至关重要，特别是交换芯片的延迟性能。

随着科技和科技的发展，对交换芯片功能及性能要求也越来越严格，芯片功能强大、性能优越对市场的竞争力会越大。交换芯片的延迟是芯片性能关键指标，交换芯片延迟能够准确性测试就显得至关重要，本文主要说明对国内最新一款交换芯片延迟测试的方法。

2　交换芯片延迟测试原理

交换芯片延迟测试方法有很多种，但是测试的原理都是相同的，数据包开始的第1bit到达交换芯片入口到该bit离开交换芯片的时间间隔，即为交换芯片的延迟量。交换芯片的延迟路径框图如图1所示。

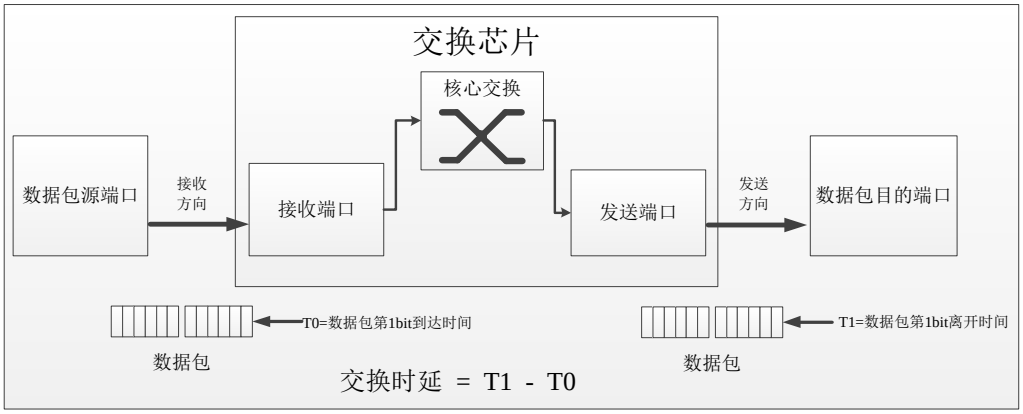


图1　交换芯片延迟路径框图

3　交换芯片延迟测试方法

3.1　RapidIO交换延迟测试方法

当前市场上没有RapidIO 3.0协议的标准测试仪，因此无法利用RapidIO协议测试仪进行RapidIO 3.0交换延迟测试。根据交换芯片的协议转换功能特性和以太网测试仪的时间戳功能，构建基于以太网测试仪的交换芯片延迟测试，在这种测试方法中需要两颗交换芯片。其中，第一颗交换芯片将以太网报文转换成RapidIO协议包，向第二颗交换芯片提供RapidIO数据包；第二颗交换芯片为最终被测试对象。

RapidIO的交换延迟采用差值法进行计算，用于消除交换芯片外部物理介质的传输延迟。图2为测试RapidIO交换延迟T1的原理框图，图3为测试RapidIO交换延迟T2的原理框图。思博伦以太网测试仪发出一个数据包到交换芯片A的Port0，配置组播路由转发到Port1和Port24，数据包从Port0转发到Port24并传输到测试仪，测试仪收到此数据包就会标记时间戳为T1-2（T2-2）；数据包从交换芯片A的Port0转发到Port1，经Port1传输到交换芯片B的Port1，该数据包在交换芯片B中按相应路由进行转发，经交换芯片B的Port1传回到交换芯片A的Port1，最后通过交换芯片A的Port24传输到测试仪，测试仪收到此数据包同样标记时间戳为T1-1（T2-1）；思博伦以太网测试仪对收到两个数据包的时间戳T1-1（T2-1）与T1-2（T2-2）做差值，即为T1（T2）时间；

T1-1测试路径是：思博伦以太网测试仪→交换芯片A的Port0（以太网报文）→经协议转换后变成RapidIO协议→交换芯片A的Port1（RapidIO数据包）→交换芯片B的Port1（RapidIO数据包）→交换芯片B的Port0（RapidIO数据包）→交换芯片B的Port0的Serdes接口内部自环（RapidIO数据包）→交换芯片B的Port0（RapidIO数据包）→交换芯片B的Port1（RapidIO数据）→交换芯片A的Port1（RapidIO数据包）→经协议转换后变成以太网协议→交换芯片A的Port24（以太网报文）→思博伦以太网测试仪，T1-1数据传输路径为图2中的标号□□□□□□□□□□□　　，在交换芯片B的路径上包含两次交换转发分别是标号□□□　和标号□□□　，标号□是Serdes接口数据内部自环（PMA环回）；T1-2测试路径是：思博伦以太网测试仪→交换芯片A的Port0（以太网报文）→交换芯片A的Port24（以太网报文）→思博伦以太网测试仪，T1-2数据传输路径为图2中的标号□□　；T1-1和T1-2两个时间做差值，即为T1时间。

T2-1测试路径是：思博伦以太网测试仪→交换芯片A的Port0（以太网报文）→经协议转换后变成RapidIO协议→交换芯片A的Port1（RapidIO数据包）→交换芯片B的Port1（RapidIO数据包）→交换芯片B的Port1（RapidIO数据包）→交换芯片A的Port1（RapidIO数据包）→经协议转换后变成以太网协议→交换芯片A的Port24（以太网报文）→思博伦以太网测试仪，T2-1数据传输路径为图I.3中的标号□□□□□□　，在交换芯片B的路径上只包含一次交换转发即标号□□□　；T2-2测试路径是：思博伦以太网测试仪→交换芯片A的Port0（以太网报文）→交换芯片A的Port24（以太网报文）→思博伦以太网测试仪，T2-2数据传输路径为图I.3中的标号□□　；T2-1和T2-2两个时间做差值，即为T2时间。

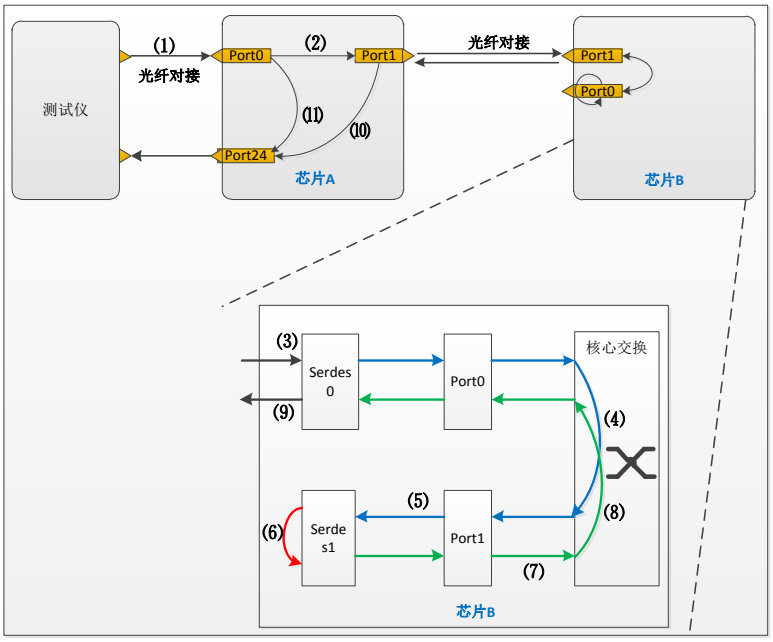


图2　RapidIO交换延迟T1路径测试原理框图

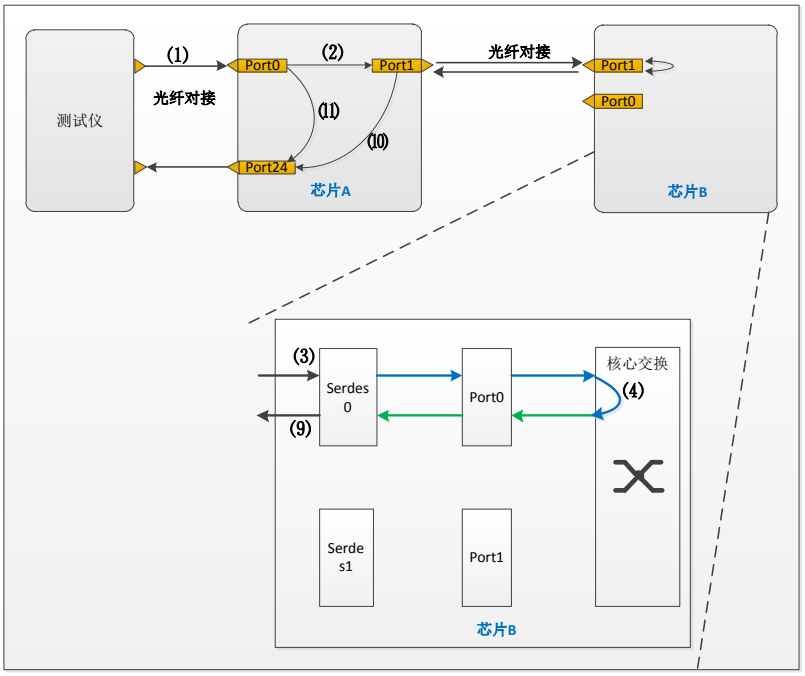


图3　RapidIO交换延迟T2路径测试原理框图

3.2　异构协议交换延迟测试原理

本文测试的交换芯片支持FC-AE-ASM、RapidIO和以太网三种协议，以及协议转换模块，三种协议可以通过协议转换模块任意进行协议的转换。对于数据报文输入交换芯片和输出交换芯片的协议不同，即在交换芯片进行了协议转换，为异构协议。

当前市场上没有支持异构协议的标准测试仪，无法直接利用标准测试仪进行异构协议交换延迟的测试。根据交换芯片的协议转换功能特性和以太网测试仪的时间戳功能，构建基于以太网测试仪的交换芯片异构协议交换延迟测试，该方法使用以太网测试仪一台和交换芯片两颗，其中，第一颗交换芯片作为协议转换使用，把测试仪发出

的以太网报文转换成RapidIO数据包或者FC-AE-ASM协议报文，第二颗交换芯片才是最终被测试的对象。

异构协议交换延迟同样采用差值法进行计算，用于消除交换芯片外部物理介质的传输延迟。FC-AE-ASM协议与RapidIO协议、以太网协议与RapidIO协议和以太网协议与FC-AE-ASM协议三种异构协议交换延迟的测试原理框图如图4和图5、图4为测试交换延迟T1的原理框图，图5为测试交换延迟T2的原理框图，下述为FC-AE-ASM协议与RapidIO协议的异构延迟交换测试方法，其与以太网协议与RapidIO协议和以太网协议与FC-AE-ASM协议的异构协议方法方法一致，只需改变协议类型即可。

思博伦以太网测试仪发出一个数据包到交换芯片A的Port0，配置组播路由转发到Port1和Port24，数据包从Port0转发到Port24并传输到测试仪，测试仪收到此数据包就会标记时间戳为T1-2；数据包从交换芯片A的Port0转发到Port1，经Port1传输到交换芯片B的Port1，该数据包在交换芯片B中按相应路由进行转发，经交换芯片B的Port2传回到交换芯片A的Port2，最后通过交换芯片A的Port24传输到测试仪，测试仪收到此数据包同样标记时间戳为T1-1；思博伦以太网测试仪对收到两个数据包的时间戳T1-1与T1-2做差值，即为T1时间；

T1-1测试路径是：思博伦以太网测试仪→交换芯片A的Port0（以太网报文）→经协议转换后变成RapidIO协议→交换芯片A的Port1（RapidIO数据包）→交换芯片B的Port1（RapidIO数据包）→经协议转换后变成FC-AE-ASM协议→交换芯片B的Port0（FC-AE-ASM报文）→交换芯片B的Port0外部光纤环回→交换芯片B的Port0（FC-AE-ASM报文）→经协议转换后变成RapidIO协议→交换芯片B的Port2（RapidIO数据包）→交换芯片A的Port2（RapidIO数据包）→经协议转换后变成以太网协议→交换芯片A的Port24（以太网数据包）→思博伦以太网测试仪，T1-1数据传输路径为图I.4中的标号□□□□□□□□□□□□，在交换芯片B的路径上包含两次协议转换及交换转发分别是标号□□□和标号□□□，标号□是外部光纤环回；T1-2测试路径是：思博伦以太网测试仪→交换芯片A的Port0（以太网报文）→交换芯片A的Port24（以太网报文）→思博伦以太网测试仪，T1-2数据传输路径为图I.4中的标号□□；T1-1和T1-2两个时间做差值，即为T1时间。

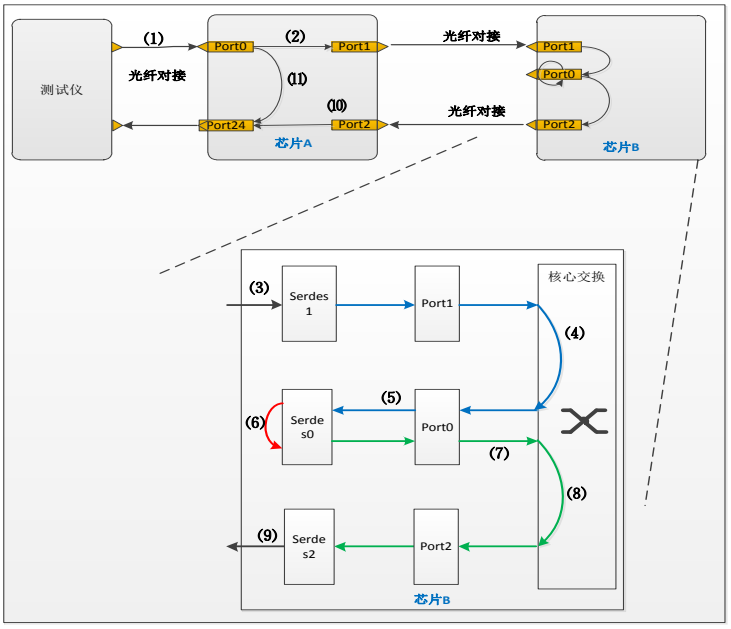


图4　异构协议交换延迟T1路径测试原理框图

思博伦以太网测试仪发出一个数据包到交换芯片A的Port0，配置组播路由转发到Port1和Port24，数据包从Port0转发到Port24并传输到测试仪，测试仪收到此数据包就会标记时间戳为T2-

2；数据包从交换芯片 A 的 Port0 转发到 Port1，经外部光纤连接数据包传输到交换芯片 A 的 Port2，最后通过交换芯片 A 的 Port24 传输到测试仪，测试仪收到此数据包同样标记时间戳为 T2-1；思博伦以太网测试仪对收到两个数据包的时间戳 T2-1 与 T2-2 做差值，即为 T2 时间；

T2-1 测试路径是：思博伦以太网测试仪→交换芯片 A 的 Port0（以太网报文）→经协议转换后变成 RapidIO 协议→交换芯片 A 的 Port1（RapidIO 数据包）→交换芯片 A 的 Port2（RapidIO 数据包）→经协议转换后变成以太网协议→交换芯片 A 的 Port24（以太网报文）→思博伦以太网测试仪，T2-1 数据传输路径为图 I.5 中的标号□□□ ；T2-2 测试路径是：思博伦以太网测试仪→交换芯片 A 的 Port0（以太网报文）→交换芯片 A 的 Port24（以太网报文）→思博伦以太网测试仪，T2-2 数据传输路径为图 I.5 中的标号□□ ；T2-1 和 T2-2 两个时间做差值，即为 T2 时间。

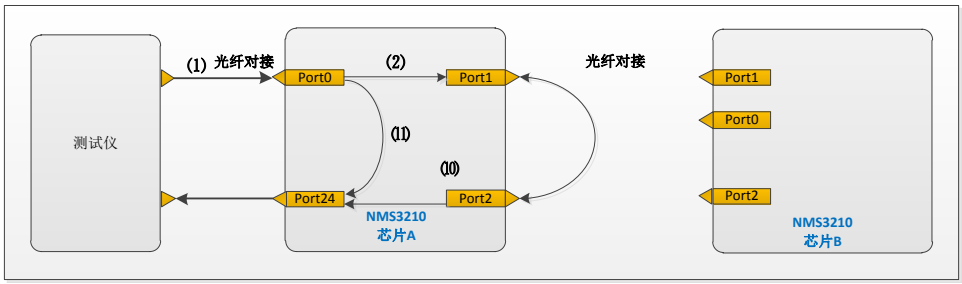


图 5　异构协议交换延迟 T2 路径测试原理框图

则 T1 与 T2 的差值是两个完整路径的异构协议交换延迟，分别是 RapidIO 协议转成 FC-AE-ASM 协议和 FC-AE-ASM 转换成 RapidIO 协议，对应标号分别是图 4 中的□□□ 和□□□ 。因此单次异构协议交换的延迟的值应该是（T1-T2）/2 ± 32ns。

4　结语

随着经济和科技的快速发展，对交换芯片延迟性能指标要求也越来越高，延迟的大小成为交换芯片性能的重要指标，直接影响着交换芯片的市场。依托此交换芯片协议转功能，能够在当前市场上没有 RapidIO 3.0 协议的标准测试仪和异构协议的标准测试仪的情况下，完成交换芯片 RapidIO 和异构协议的延迟测试，对交换芯片的延迟性能指标做出有力的依据。

参考文献:

[1] 郦伟,肖鹏. 基于 RapidIO 协议的网络路径分配策略[J]. 计算机工程与设计,2017.

[2] 俞大磊,邓豹,韩强. 基于 VPX 标准的以太网交换模块设计[J]. 工业控制计算机,2017.

[3] 张永泽. 一种高速 FC 交换机的关键技术研究[D]. 电子科技大学,2020.

一种具有内生安全特性的拟态MCU架构

陈德沅[1]，谭力波[1]，王盼[1]，于洪[2]

[1]天津市滨海新区信息技术创新中心，天津 300450；

[2]解放军战略支援部队信息工程大学，郑州 450002

摘　要：随着半导体产业的发展，MCU芯片应用场景愈加广泛，在如今大数据的背景下，以漏洞、后门和病毒攻击为主要威胁方式的MCU安全问题愈加受到重视。业界主流的被动防御式技术无法满足应对破坏者的可知攻击及未知的安全威胁，无法保障个人信息及国家安全公共服务系统的不可侵犯性。本文针对业界主流MCU攻击手段，提出了一种具有主动防御机制的内生安全拟态MCU架构，论述了其工作机制原理与验证方式。

关键词：拟态防御、内生安全、容错技术、MCU攻击手段

A mimicry MCU architecture with endogenous safety characteristics

CHEN Deyuan[1], TAN Libo[1], WANG Pan[1], YU Hong[2]

1.Tianjin Binhai New Area Information Technology Innovation Center , Tianjin 300450;

2.Information Engineering University, Zhengzhou 450002

Abstract: With the development of the semiconductor industry, MCU chip has been applied into more and more extensive scenarios. In the context of big data widespread, MCU security issues formed mainly as vulnerabilities, backdoors, and virus attacks are attracting more and more attention. The mainstream passive defense technology in the industry cannot cope with the security threat of the known attack from the saboteur and location, and cannot guarantee the inviolability of personal information and the national security public service system. Aiming at the mainstream MCU attack methods in the industry, this paper proposes an endogenous security mimic MCU architecture with active defense mechanism, and discusses its working mechanism and verification methods.

Key words: mimic defense; endogenous safety; MCU attack methods

1　引言

目前MCU应用广泛，并且在电路中起到关键的控制作用。国内及国际市场需求广泛，但由国内生产的MCU份额却寥寥无几。同时由于MCU的安全可靠性技术保障现状所致，单个MCU可能由于外部攻击或内部漏洞导致计算结果错误或失效，在基于未知漏洞、未知后门的攻击面前形同“裸奔”[1]。市场急需一种安全自主可控、安全可信的高安全MCU，满足在工业控制、汽车电子、通讯设备、导航系统等领域的高安全防护需求。拟态MCU采用拟态防御系统架构原理，由四个异构的MCU加一个调度器（FPGA）来实现动态异构冗余的拟态MCU架构，实现了安全本体构建拟

态动态化、异构性和多样性的特点，隐藏异构运算子系统内部的物理结构与漏洞，从外界或攻击者来看，拟态MCU与普通MCU功能结构相同，表现出透明性，但内部结构表征呈现不确定性，打造出高安全、高鲁棒的控制平面，建立内生安全防御机制，重新定义了高安全MCU芯片的防御体系。

2 业界主流MCU防御手段

随着产业技术革命不断升级，作为基础技术支撑的工业控制和物联网得到了空前的发展和应用，产品更新换代和新兴应用快速发展将推动MCU在该领域中的关键地位。但随着MCU的应用越加广泛，暴漏出当前MCU的高安全可靠性保护技术却没有紧跟发展，MCU保护措施的局限性通常以软件和硬件信息方面为主，导致了单个MCU可能由于外部攻击或内部漏洞导致计算结果错误或失效[2]。以近年来发生的加拿大的一个水利SCADA控制系统、波兰某城市地铁系统被黑客攻破为例，给人民及国家的生产安全造成巨大影响，典型工业控制系统和物联网入侵防护已迫在眉睫[3]。

目前针对MCU硬件防护的措施主要分为硬件保护和软件保护两个层面；

1） **硬件层面主要采用通过硬件接口方面，使用安全熔丝来控制接口的操**

作，如，阻止存储器中的数据发送到输出缓冲器里。

2） **软件层面为主要的保障措施，其典型安全保障方案为**

a．使用软件数据加密保护，对数据进行加密，可以有效阻止攻击者的大部分数据破解行为。

b．对工控机采用系统沙箱锁定机制来对操作系统进行固化，除操作系统正常运行的核心程序以及业务软件之外的程序都不可执行；

c．提供缓存溢出攻击保护和线程注入攻击防护的功能；

d．在网络层通过防火墙功能组织网络攻击，限制无关主机对工控机设备的网络访问。

市面上的安全方案多以软件信息方面为主，目前还没有针对硬件漏洞，后门防护的有效措施。

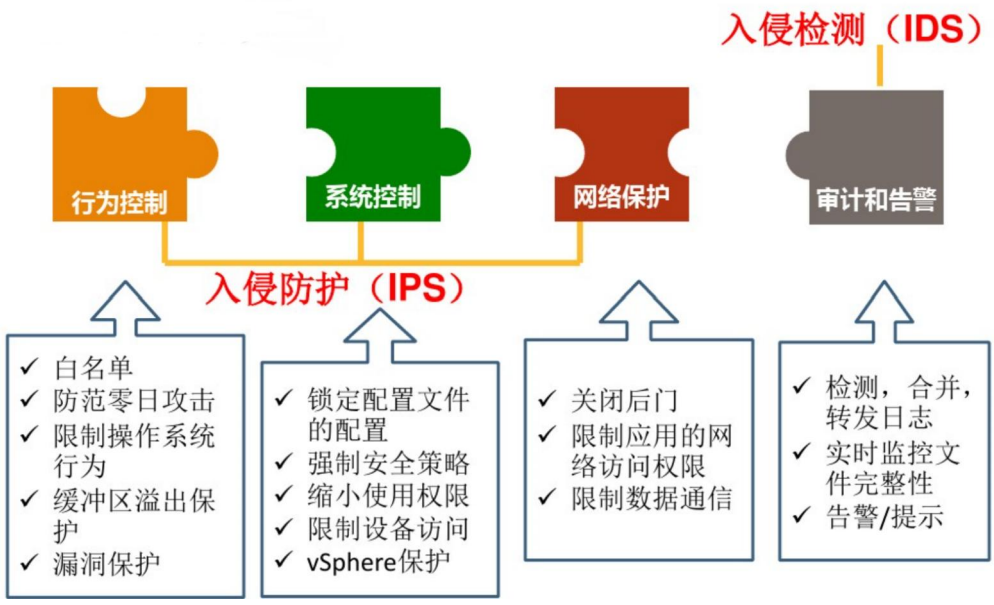


图1-1 MCU防护机制

MCU应用广泛，并且在电路中起到关键的控制作用，MCU保护措施的局限性通常以软件和硬件信息方面为主，导致了单个MCU可能由于外部攻击或内部漏洞导致计算结果错误或失效。

通过MCU的典型时序攻击、穷举攻击等攻击手段，暴漏出单个MCU可能由于外部攻击或内部漏洞导致计算结果错误或失效的情况，从而导致重大事故发生。

目前MCU安全可靠性保障局限性为：

1. 硬件保护层面单一，通过硬件熔丝来保护

电路源程序，平台本身的安全漏洞问题严重；

2. 软件层面虽然保护措施较多，但局限性依然有软件保护措施不可避免的天生缺陷：

a． 漏洞隐蔽性暴漏严重，不能有效的防止硬件设计上预留的后门与不可估计的漏洞；

b．不能防止大算力的穷举破解，只能延缓破解时间；

c．杀毒软件安装及升级更新问题；

d．专用平台和通用平台漏洞；

e．工控网络的脆弱性；

3 拟态化防御原理

常见MCU系统多数使用静态构架，无法有效抵御攻击者的持续探测与攻击，导致系统网络态势呈现易攻难守的局面。针对传统MCU存在的安全性问题，由我国自主创新的拟态防御策略可以很好的解决这一问题。拟态防御（Mimic Defense）理论是我国邬江兴院士提出的基于动态异构冗余（DHR， dynamic heterogeneous redundancy）的架构，不仅提高了系统的动态随机性，还为目标系统创建同功能异构的执行空间。拟态MCU防御系统利用异构性、多样性来改变系统的相似性和单一性，利用动态性、随机性改变系统的静态性、确定性，期望利用动态异构的构架使得隐藏漏洞不被利用[1]。在运行期间，动态调用多个异构执行体，在输出端进行一致性判决，并使用反馈控制模块对异构执行体进行重新调度和清洗等[4]。拟态防御思想已被应用于多种网络设备及软件设计中，为系统提供内生的安全防护性能。

拟态MCU架构的核心理论是动态异构冗余（dynamic heterogeneous redundancy， 简称DHR）架构，如图1所示。MCU系统的功能可以概括为“输入-处理-输出”，即，结构化设计中的IPO（input-process-output）。动态异构冗余结构在“处理”环节使用异构执行体集进行处理，将同一输入通过输入代理复制为n份，并分发给执行体集中的n个异构执行体进行处理，将处理结果收集至表决器进行表决，得到唯一的相对正确的输出。异构元素组成异构构件，由动态选择算法选择异构构件组成在线的执行体集。根据运行时的反馈信息，动态选择算法会产生新的执行体集以替换当前集合。

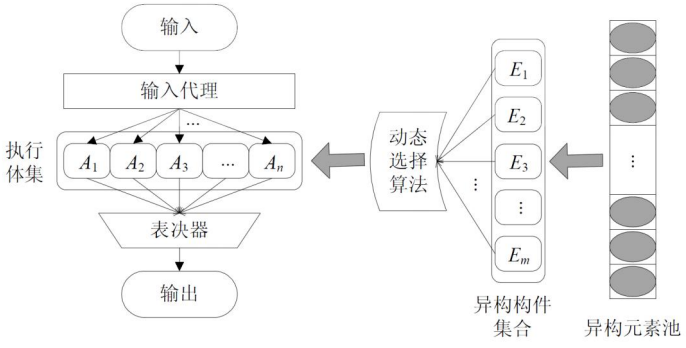


图2-1动态异构冗余（DHR）结构

动态异构冗余结构的基础是异构性。执行体应尽可能地保证在各种属性或特征上的异构，才能避免相同漏洞的同时出现。异构的层面越多，即执行体异构的属性越多，能够防御的漏洞越多，攻击难度就越高.异构性可以基于多样性实现，应用的多样性、操作系统的多样性、程序设计语言的多样性均能带来一定的异构性[5]。

动态性作为异构性在时间维度上的增益，能够补救“同源”带来的异构性不足的问题.动态选择算法使动态异构冗余结构具有动态性.如果当前执行体集被攻击突破，动态选择算法在得到系统反馈后，就会重新生成执行体集替换当前执行体集，改变攻击所依赖的环境，使相同攻击难以维持或再现.另一方面，动态性的存在使系统在不同的时间段表现出不一样的特征，对攻击者呈现出不确定性，进一步增大了攻击难度。

冗余是指多执行体处理同一请求，对比不同执行体的处理结果，通过表决得到相对正确的响应返回给用户。冗余性与异构性相互配合，实现对攻击所依赖的单一环境的改变，增大了攻击难度，提高了系统的安全性[6]。

表决器的分布可以采用集中式，如在最终输

出前进行一次表决；也可采用分布式，如对每一层的输出均进行表决，主要取决于输出的形式是否便于比较。

拟态安全防御技术能从主动性、变化性和随机性中获得有利的内生防御态势，通过芯片架构、操作系统、应用软件、编译选项、数据结构、逻辑算法等要素的主动跳变或快速迁移来实现拟态环境，以防御者可控的方式进行动态变化，对攻击者则表现为难以观察和预测目标变化，从而大幅度增加包括未知的可利用的漏洞和后门在内的攻击难度和成本。目前行业内主要MCU产品均缺少具有内生安全的主动防御功能，在基于未知漏洞、未知后门的攻击面前形同“裸奔”。结合各领域中MCU数据的高安全防护需求，改变当前MCU安全技术领域的局限性，亟需研制具有拟态内生安全防御技术的MCU产品，即拟态MCU产品。

4 拟态化MCU架构

基于拟态防御体系架构原理，动态异构冗余构造及运行机制是拟态防御体系有效性的核心，也是广义鲁棒控制的基础。通过在异构冗余中引入闭环负反馈控制，能在功能不变的条件下对目标对象进行重构、重组、重建，使对象具有多样化、动态化和随机化的场景，从而防御已知和未知的威胁。

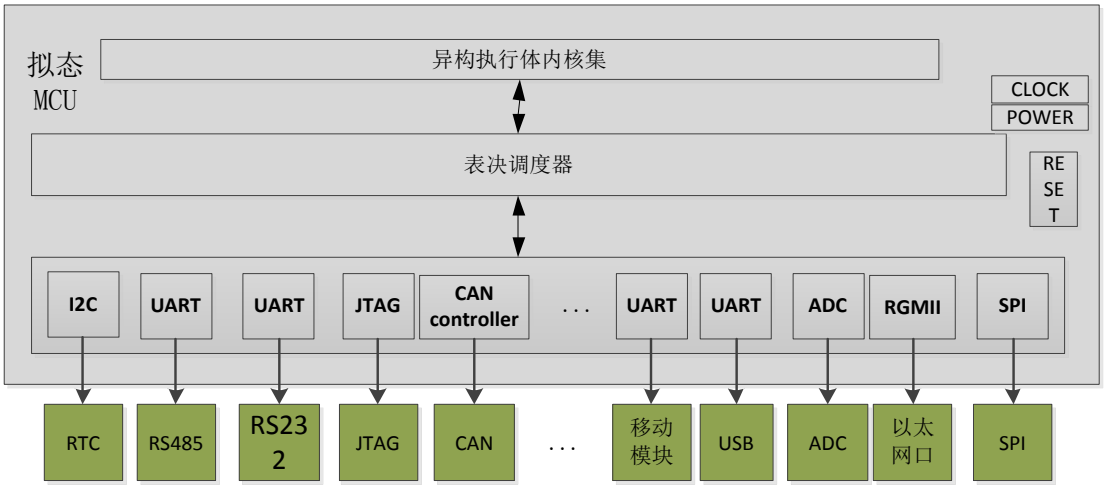


图3-拟态MCU硬件架构

拟态MCU硬件架构如上图所示，包括多个异构MCU内核，拟态调度器及接口部分。其中异构MCU内核作为异构功能体，架构分别使用MIPS、AVR、ARM和Coldfire等，用于执行具体的协议数据处理、未知报文处理、系统控制权限管理、系统日志管理等功能。四个MCU架构的差异性可以保证各执行体在功能等价的前提下，存在的缺陷和错误具有差异化，从而可以被调度器探测。

拟态MCU正常工作时选择至少三个MCU内核参与判决，其余多个MCU内核作为备份。拟态调度器将输入数据分发至执行参与判决工作的MCU内核，在MCU中执行具体的协议处理，未知报文处理、系统控制权限管理、系统日志管理等功能。参与判决的三个MCU将处理后的输出数据输入拟态调度器中进行判决，拟态调度器根据裁决结果和拟态策略输出唯一的判决结果并进行MCU内核的清洗与替换。

调度器部分包括上行分发逻辑，负责将数据复制分发至多个异构MCU内核；拟态判决模块提供多路数据输入，判决后输出判决结果和唯一数据。正常工作模式下，不少于三个MCU内核参与判决，至少一个MCU内核作为备份，因此拟态策略根据判决结果和当前模组状态切换工作MCU内核并控制清洗模块清洗异常MCU内核，实现反馈和动态重构。还具备密钥生成模块供MCU内核进行数据加密和身份认证使用，以及TCM模块用于作为可信计算的可信根管理。

对外接口

拟态MCU具有丰富的对外接口以满足各种不同的应用场景。这些接口主要包括：

（1） RS232；

（2） RS485；

（3） CAN总线；

（4） 以太网口；

（5） I²C接口；

（6） SPI及AD接口；

（7） JTAG；

（8） USB接口。

拟态MCU通过硬件拟态架构，构建拟态防御系统，利用异构性、多样性来改变系统的相似性和单一性，使用拟态多模裁决和动态调度机制，建立内生安全防御机制，识别和抵御安全威胁，有效阻断攻击路径；利用动态性、随机性改变系统的静态性、确定性，利用动态异构的构架使得隐藏漏洞不被利用。基于此，拟态MCU架构，实现了安全本体构建拟态动态化、异构性和多样性的特点，隐藏异构运算子系统内部的物理结构与漏洞，从外界或攻击者来看，拟态MCU与普通MCU功能结构相同，表现出透明性，但内部结构表征呈现不确定性。

MCU具备备份MCU内核替换、反向验证机制、随机扰动机制。具有拟态防御系统的拟态MCU相比以往普通MCU，具有以下优点：

1）具有可靠性，多个MCU运行，择多判决运算结果，运算结果更可靠；

2）具有动态性，系统四个MCU由拟态策略动态选择工作模式；

3）具有抗攻击性，实现了MCU动态异构冗余架构，使出现多数或完全相同的错误的概率降低，降低了由于外部攻击或内部漏洞导致计算结果错误的概率。

拟态MCU有效的弥补了现有安全技术的MCU在硬件保护和软件保护层面的局限性。同时拟态MCU支持可信计算，通过可信根一层层度量每一部件的安全性，有效的阻止了恶意软件非授权安装/运行、设备网络假冒、基于完整性篡改的攻击或异常等，对单一内核CPU的攻击行为。实现了在新的MCU现状场景和安全形势下，重新定义了高安全MCU芯片的防御体系。

5 拟态化MCU验证

为便于控制与反馈结果，针对拟态MCU开发了一套软件控制上位机，用以控制并验证拟态MCU内状态是否符合设计。设计实验了将MCU1注入错误，模拟未知方式攻击下导致的MCU内核1失效，拟态MCU系统能够主动防御，内生安全，将MCU内核1识别出异常，并对MCU内核1进行清洗，恢复至MCU内核1异常前状态。

图5-1 MCU内核1发生错误

总结

MCU市场潜力巨大，前景十分广阔，同时各应用领域对高安全、高可靠性的MCU芯片需求更是紧迫，我国急需由自主可控、高安全的MCU国产产品。拟态MCU芯片，在具有MCU功能的基础之上，打造出内生安全防御机制，具有动态性、

图5-2　MCU内核1被清洗

图5-3　MCU内核1恢复正常工作状态

随机性的高安全和高鲁棒控制平面。弥补了现有MCU安全保障局限性的现状，重新定义了防御手段，有望在信息系统软硬构件供应链可信性不能确保的全球化生态环境下，以创新的系统构造技术开辟出一条破解软硬构件“自主可控、安全可信”难题的新途径。

参考文献:

[1] 邬江兴．网络空间拟态防御研究[J]．信息安全学报，2016，1(4)：1-10.
WU J X. Research on cyber mimic defense[J]. Journal of Cyber Security，2016，1(4)：1-10.

[2] 刘庆．中国MCU市场观察[J]．电子产品世界，2018.

[3] Sergei P. Skorobogatov. Semi-invasive attacks－A new approach to hardware security analysis [J]. UNIVERSITY OF CAMBRIDGE Computer Laboratory，2005. 4.

[4] 仝青, 张铮, 张为华, 等．拟态防御 Web 服务器设计与实现[J]．软件学报，2017，28(4)：883-897.

[5] 庞建民, 张宇嘉, 张铮, 等．拟态防御技术结合软件多样化在软件安全产业中的应用[J]．中国工程科学，2016，18(6)：74-78.

[6] 常箫, 张保稳, 张莹．一种面向网络拟态防御系统的信息安全建模方法[J]．通信技术，2018，v. 51;No. 313(01)：171-176.

智慧交通拟态安全芯片系统原型设计

孙远航，李彧，李召召，成诚
网络通信与安全紫金山实验室，南京 211111

摘　要：拟态计算（mimic computing）是一种具备高安全属性的新型计算体系结构，其通过异构冗余构建了具备安全属性的计算空间，从理论上解决了信息系统安全的可信、可测问题。通过广义鲁棒性概念的扩展，将信息系统的安全性问题等价为系统缺陷漏洞，并通过异构冗余的方式解决各种威胁和后门漏洞，实现内生安全机制。智慧交通系统对于网络安全尤其重视，拟态安全架构可以大大降低安全风险，但是智慧交通系统对于整体系统的体积、功耗、成本比较敏感，而异构冗余处理器必然带来系统复杂度、成本、功耗的增加，如果使用分立器件搭建系统就会严重影响智慧交通系统规模应用。一种比较有效的解决方法是将整个拟态系统芯片化，能够大大降低系统成本、功耗。本文重点介绍了智慧交通拟态安全芯片系统基于FPGA的原型设计。实验结果证明，本文所提到的原型能够基于Xilinx的FPGA顺利实现，达到智慧交通领域拟态防御系统的要求。

关键词：智慧交通、拟态安全、芯片设计、FPGA

Smart transportation mimic security chip system prototype design

Sun Yuanhang, Li Yu, Li Zhaozhao, Cheng Cheng
Purple Mountain Laboratories, Nanjing 21100, China

Abstract: Mimic computing is a new type of computing architecture with high safety attribution. It builds the computation space with safety attribution using heterogeneous redundancy,and solves the reliability and testability problems of information systems theoretically. With the extension of robustness, making the safety problems of information system equaling to the vulnerability of system,and solving the kinds of threaten and loopholes using the method of heterogeneous redundancy. And all these make the endogenous security system. The smart transportation cares more about the network safety,and the mimic security architecture can reduce the security risk greatly. But smart transportation is sensitive about the volume,power and cost of the whole system,and the heterogeneous redundancy computation will bring the increase of complexity,cost and power necessarily. Using the discrete devices to build the system will astrict the large-scale application of the smart transportation system. A effective method is build the system on an ASIC,that will reduce the cost and power largely. In this paper,we build the smart transportation mimic security chip system base on the FPGA. Experiment result shows that the proposed system can work correctly on the Xilinx FPGA system,and can meet the requirements of smart transportation mimic security system.

Key words: smart transportation; mimic security; asic design; FPGA

1 引言

当前社会数字系统发展迅速，“智慧”概念已经逐步深入到人们的日常生活过程中，包括智慧购物、智慧支付、智慧信息管理、智慧资源管理、智慧环境管理、智慧安防管理、智慧园区服务管理、智慧交通等。通过智能终端、互联网络、中央处理系统、云服务器等，构建万物互联的智慧系统。其中智慧交通是“智慧”的理念在交通运输行业的具体体现，这里交通具体包括城市交通、高速公路、轨道交通、园区交通的等多个领域。智慧交通是以物联网为信息采集、交换与服务的基础支撑平台；以智慧的信息化决策和处理技术为基本手段，通过对海量的交通信息的梳理、过滤、挖掘和利用，从而集人与人类智慧、交通物理网络及相关信息为一体，构建一整套现代交通运输系统[1]。随着现在人们智能设备的普及化、交通出行工具的多元化、信息需求多样化，智慧交通系统也变得更加复杂，不仅包含传统意义上的地铁、公交、私家车等交通工具，还包括智慧候车亭、智能汽车充电桩、公共自行车设施、智能路灯、智能信号灯、智慧停车系统等。一种典型的智慧交通系统如图1所示，各类交通终端信息通过移动通信网络、以太网网络传输给云服务器，所有这些接入智慧交通系统的智能终端搜集的信息都需要经过中央处理系统进行分析、决策、调度，通过交通网络实现真正意义上的“智慧”。

图1 智慧交通系统示意图

但是科技让交通变得更加智慧的同时，也引入了一些隐患，如果这类智慧交通网络被恶意入侵、控制，可能带来比较大的信息泄露、交通控制等方面的巨大风险，轻者会给人们带来错误信息及其他不便，重者会造成用户个人隐私信息、个人支付账户信息泄露，从而带来财产损失。更为严重的，如果地铁、交通信号灯等被恶意入侵，就有可能造成严重的交通事故，从而给国家、社会及人民群众带来巨大的生命、财产安全风险。所以，智慧交通系统必须严格保证系统网络的安全性，这也是影响智慧交通大规模发展的关键因素。当前智慧交通系统安全领域相关研究比较少，可以在这个方向投入资源研究，增强系统的安全性。

拟态防御架构以不确定性应对未知威胁。网络空间防御不再追求建立一种无缺陷、无漏洞、无后门、完美无瑕的防御系统，而是构建一种动态的、异构的、冗余的体系架构。期望通过增加系统的动态性减少系统的可探测性，增加系统的随机性降低系统的可渗透性，增加系统的异构冗余性提高协同攻击的难度，增加系统的动态性、随机性和多样性提升攻击链诸环节的实施难度[2]。

在异构冗余架构内，任意异构功能体之间不存在同步或协同机制，任何单独功能体的设计缺陷导致的异常问题最多影响到本功能体输出结果，在拟态架构下通过一致性判决机制可以识别出这种异常，过滤异常结果，隔离异常影响，而系统整体功能不会受到影响。

如同拟态防御安全架构在其他系统中实际应用中体现出的效果，理论上这个架构在智慧交通系统中成功应用的话必将大大提升智慧交通系统的安全性。但是智慧交通系统对于整体系统的体积、功耗、成本比较敏感，而异构冗余处理器必然带来系统复杂度、成本、功耗的增加，如果使用分立器件搭建系统就会严重影响智慧交通系统的规模应用。一种比较有效的解决方法是将整个拟态系统芯片化，能够大大降低系统成本、功耗。本文重点介绍了智慧交通拟态安全芯片系统基于FPGA的原型设计。实验结果证明，本文所提到的原型能够基于Xilinx的FPGA顺利实现，达到智慧交通领域拟态防御系统的要求。

2　芯片系统原型设计

2.1　系统总体架构设计

传统意义上智慧交通网络处理系统如图2所示，由于中央处理单元单一，容易被攻破，造成安全漏洞风险。

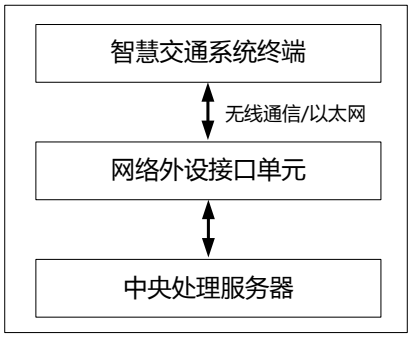


图2　传统智慧交通网络系统

增加拟态安全系统的智慧交通网络系统如图3所示，其中灰色部分是拟态安全系统结构。相比于传统系统，拟态安全系统增加了用来实现异构冗余的多个异构处理引擎。另外在异构多核处理的背景下，单个处理器可能由于内部故障或者受外部攻击输出错误结果，此时需要调度器对各处理器的输出结果进行判决并输出调度器认为正确的结果。

智慧交通拟态安全系统中，拟态安全系统需要完成交通数据输入/输出信息的处理，协议解析，数据分发与数据判决，异构引擎数据处理。还需要进行数据传输流量进行监测，拟态策略控制上行及下行数据分发及判决策略，看门狗实现系统

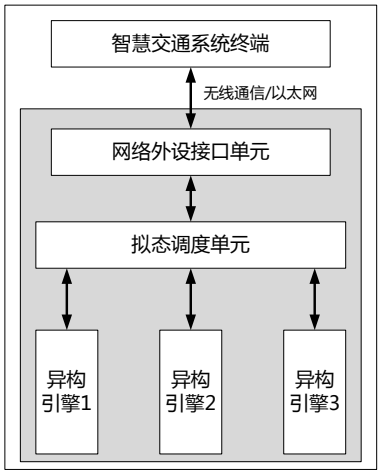


图3　拟态安全智慧交通系统

日志记录及系统错误恢复，系统还设有清洗恢复机制，用来对异构引擎进行清洗恢复等工作。

其中交通数据输入/输出一般采用以太网接口及串行UART接口，异构引擎一般至少采用三种以上不同指令架构的CPU核，内部配套DMA、中断处理、内部互联总线、时钟复位处理等组件。

上述智慧交通拟态安全系统可以采用传统的分立器件搭建，比如三个异构引擎采用三颗相应的CPU芯片，拟态调度单元及外设接口单元采用专用ASIC芯片或FPGA等实现，在板级实现系统。但是整体系统的体积、功耗、成本等必然会增加，这个在对这些因素比较敏感的智慧交通系统中很难大规模推广应用。一种比较有效的解决方法是将整个拟态系统芯片化，能够大大降低系统成本、功耗。为了验证整个系统的可行性，本文基于FPGA搭建了系统的芯片原型系统。

2.2　基于FPGA的智慧交通拟态安全系统设计

为了尽快搭建FPGA原型系统，本文调研了目前市面上可选的FPGA芯片资源情况、接口情况，结合对拟态系统资源的预估，本文选用了市面上常用的Xilinx公司的VC709开发板作为基础硬件平台，芯片内部选用Xilinx FPGA配套的以太网、DMA、内部互联总线、低速UART外设、中断信号处理模块、时钟复位模块等软硬核IP[3]。为了达到CPU异构的目标，需要选用不同CPU指令集的CPU核。另外考虑原型验证快速性、经济性等特点，本系统选用了市面上常见的几种不同指令集架构的开源CPU核。具体来说本系统CPU核选用了Xilinx FPGA自带了Microblaze CPU核，以及

开源的ARM指令集架构的Cortex M1、RISC-V指令集架构的蜂鸟E203软核。搭配自研的拟态调度器IP，实现整体智慧交通拟态安全系统。原型系统架构如图4所示：

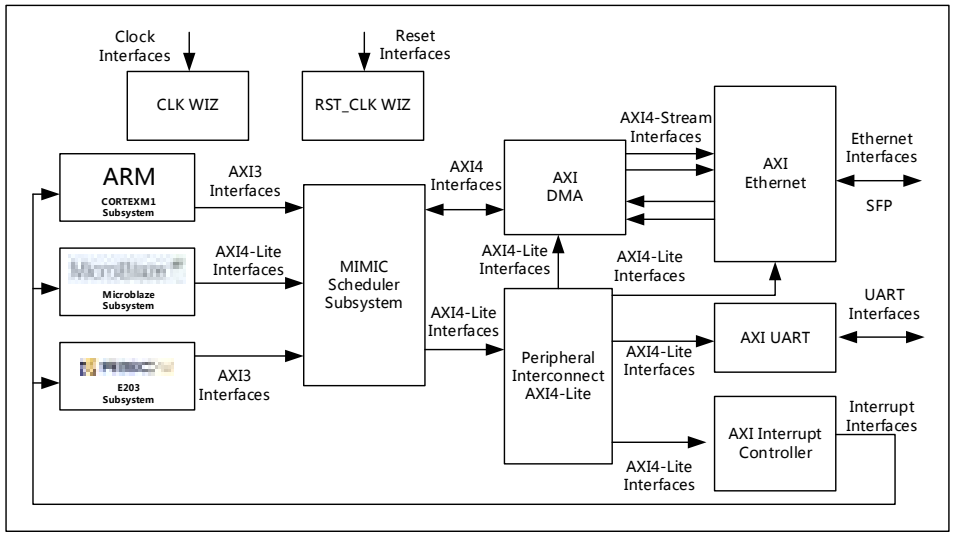


图4　FPGA原型系统架构

原型系统中各个主要组件功能描述如表1　　所示：

表1　FPGA原型系统组件情况

组件名称	组件功能
ARM Cortex M1	异构处理引擎1，智慧交通汇总信息处理
RSIC-V Hummingbird E203	异构处理引擎2，智慧交通汇总信息处理
Microblaze	异构处理引擎3，智慧交通汇总信息处理
拟态调度单元（MIMIC Schedule Subsystem）	拟态调度器IP，实现异构分发、拟态判决、协议处理等关键功能，是整个拟态系统的关键
时钟复位单元（CLK WIZ、RST_CLK WIZ）	管理全芯片时钟、复位
以太网接口	提供连接以太网接口
低速外设接口（UART）	提供低速外设接口
DMA	实现DMA功能
内部总线系统	实现内部AMBA总线互联

系统选用的开发平台是Xilinx的VC709开发板，搭载核心FPGA型号为Virtex-7 XC7VX690T-2FFG1761C，开发板提供丰富的外设接口，包括低速UART接口及以太网接口等。经评估，该开发板满足智慧交通拟态安全系统原型设计资源需求。系统中主要软硬件IP列表如表2所示：

3　原型系统FPGA工程实验

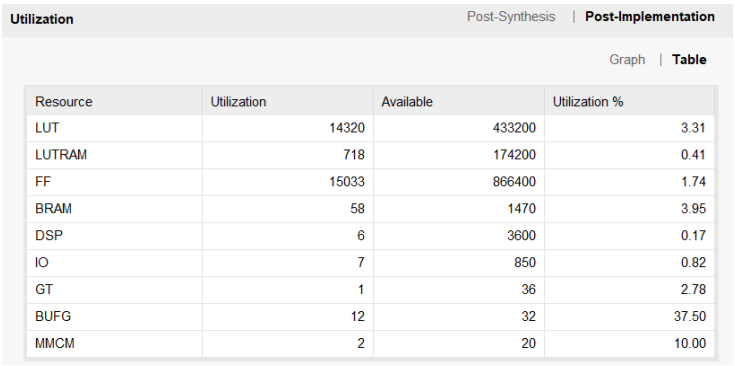
3.1　原型系统FPGA工程情况

在Xilinx Vivado工具里面建立想要的评估工程，调用Xilinx自带的相关IP，加入第三方CPU IP软核及自研的拟态调度器IP。需要特别说明的是，由于本文仅仅是进行基本系统评估，出于时间进度、人力等各方面考虑，自研拟态调度器IP部分仅仅包含了最基本的分发及判决功能，未完整实现拟态调度所有的功能，未来会进行完善。FPGA资源占用情况（基于Xilinx Virtex-7 XC7VX690T-2FFG1761C芯片）如图5所示。

由FPGA资源占用情况可以明显看出，当前原型系统在XC7VX690T FPGA中可以直接放的下，并且占用的资源比较小，未来可以把拟态调度这块的功能做大、做强，甚至可以采用低端一点的FPGA作为载体，这样整体智慧交通拟态安全系统在实现安全性的前提条件下，系统的成本、功耗相对传统的分立器件搭建的系统会大大降低。未来基于这个原型可以将整个智慧交通拟态系统芯

表2　FPGA原型系统调用IP情况

IP名称	IP属性	IP主要指标
ARM Cortex M1	CPU Core	Core Frequency:100MHz Instruction Memory:32KB Data Memory:32KB
RSIC-V Hummingbird E203	CPU Core	Core Frequency:100MHz Instruction Memory:64KB Data Memory:64KB
Microblaze	CPU Core	Core Frequency:100MHz Instruction Memory:16KB Data Memory:16KB
Microblaze Local Memory	SoC Component	Local Memory Bus V1.0, Local Memory Controller Local Memory Generator
Microblaze Debug Module	SoC Component	V3.2
AXI Direct Memory Access	SoC Component	V7.1
AXI Interconnect	SoC Component	V2.1,1 Slave,4 Master
AXI Interrupt Controller	SoC Component	V4.1
AXI Protocol Converter	SoC Component	V2.1
Processor System Reset	SoC Component	V5.0
Clocking Wizard	SoC Component	V6.0
AXI Ethernet	High Speed Interface	V7.1,AXI 1G/2.5G Ethernet,SFP
AXI UART16550	Low Speed Peripheral	V2.0,RS232



Resource	Utilization	Available	Utilization %
LUT	14320	433200	3.31
LUTRAM	718	174200	0.41
FF	15033	866400	1.74
BRAM	58	1470	3.95
DSP	6	3600	0.17
IO	7	850	0.82
GT	1	36	2.78
BUFG	12	32	37.50
MMCM	2	20	10.00

图5　原型系统FPGA资源占用情况

片化，能够进一步降低系统成本。这个也是本系统相对于分立器件搭建的智慧交通拟态安全系统最大的优势，也是本文的亮点所在。

3.12　原型系统FPGA工程仿真测试情况

系统核心功能是通过以太网接口接收智慧交通系统信息，通过拟态调度器将信息分发给三个CPU核处理，三个CPU核根据业务需求处理完信息之后把数据发还给拟态调度器，拟态调度器根据拟态安全判决算法判决三个CPU核处理返还的数据，决策得到最终的处理数据，通过以太网接口发送给智慧交通业务单元，整个拟态系统处理完成，另外通过UART串行接口监控、打印、输入相关控制、状态信息。

在仿真工具中仿真系统整体数据业务处理流程，关键的业务分发及判决波形如图6-7所示：

上行业务处理具体流程是外设接口（以太网接口或者UART接口）接收到智慧交通终端上送的数据信息，发送给拟态调度模块，拟态调度模块接收到上送信息之后，根据拟态安全理论相关算法，把上送信息进行复制，根据三个异构执行体的数据格式要求分发给三个异构引擎。图6展示的是上行业务处理中关键的业务分发部分的仿真

波形。

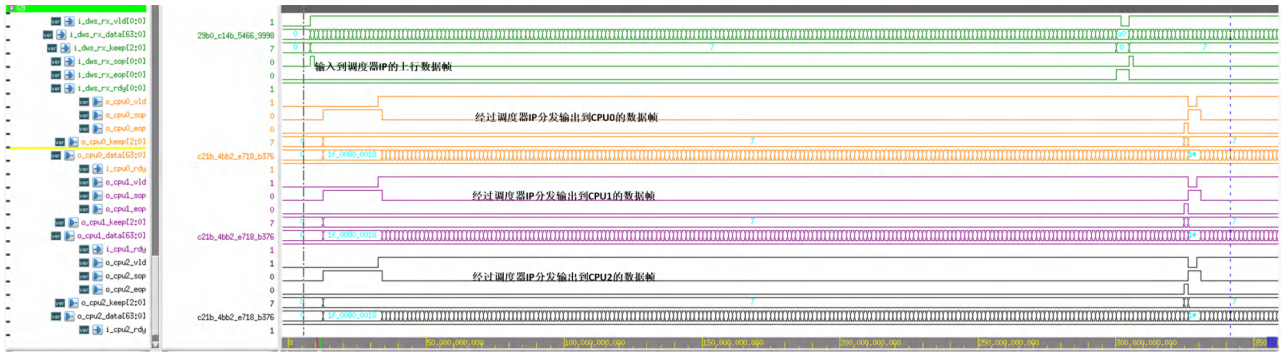


图6　上行分发仿真波形

图7　下行判决仿真波形

下行业务处理具体流程是三个异构引擎智慧交通信息按照协议及业务要求分别处理完毕返还给拟态调度模块，拟态调度模块接收到信息之后，根据拟态安全理论相关算法，对三路信息进行判决，将判决后的信息通过以太接口或者UART接口发送给智慧交通终端。如果存在部分异构引擎被黑客攻破的情况，在下行判决的时候就会发现三路信息不一致的情况，那么就会启动判决、清洗、恢复等一系列复杂的拟态处理流程，该流程在其他拟态安全调度类文章中有比较详细的描述，且不是本文论述的重点，这里就不再赘述。图7展示的是上行业务处理中关键的业务分发部分的仿真波形。

从仿真情况来看，整个原型系统外设接口、核心CPU、关键拟态调度模块、内部SoC各个组件能够按照设计要求正常工作，达到设计预期。

4　结束语

本文以拟态安全基本思想为指导，分析了当前智慧交通系统现状、安全风险及分立器件搭建拟态安全系统的弊端，引入了智慧交通拟态安全芯片及原型验证系统的必要性。根据智慧交通拟态安全芯片的需求，详细调研了市面上几种可用的开源CPU核代码、高低速外设模块及通用SoC架构，基于Xilinx FPGA开发板，选用开源的软硬核IP，搭配自研的拟态调度器IP，共同搭建了智慧交通拟态安全芯片的原型。

实验结果表明，系统架构可行，可以在FPGA系统中成功运行，实现智慧交通用拟态安全功能需求，资源占用在合理范围之内，整个系统在单片FPGA中可行，达到另外芯片系统原型验证的目标。未来可以直接基于此原型进行ASIC芯片化，进一步降低智慧交通用拟态安全的成本及功耗。由于时间限制，本文中所选用的CPU核配置没有进行详细的优化，自研拟态调度器IP也仅仅实现了基本的拟态调度功能，没有根据智慧交通需求进行系统性的完善，整体系统的面积及功耗还有一定的优化空间，这也是下一步的研究方向。

参考文献：（参考文献格式参照2015年新标准GBT 7714-2015）

［1］ 杨燕，朱焱，戴齐，李天瑞. 智慧轨道交通—实现更深入的智能化［J］. 计算机应用，2012， 32（5）： 1205-1207，1216.

YANG Y， ZHU Y DAI Q， LI T R. Smart rail transportation-an implementation of deeper intelligence ［J］. Journal of Computer Applications，2012， 32（5）： 1205-1207，1216.

［2］ 邬江兴. 网络空间内生安全［M］. 科学出版社， 2017

［3］ VC709 Evaluation Board for the Virtex-7

FPGA User Guide，UG887 （v1.6） March 11，2019.

[作者简介]
孙远航（1982—），男，硕士，中级职称，主要研究方向芯片设计。

李彧（1979—），男，博士，副高职称，主要研究方向芯片设计。

李召召（1989—），男，博士，工程师，主要研究方向芯片设计。

成诚（1984—），男，硕士，工程师，主要研究方向芯片设计。

一种基于编码信道理论的拟态调度器设计方法

欧阳玲[1，2]，贺磊[1]，宋克[1]

[1]战略支援部队信息工程大学，河南 郑州 450002；

[2]中原工学院电子信息学院 河南 郑州 451191

摘　要：本文针对 DHR 架构中的拟态括号安全性构建问题，基于编码信道理论，给出了拟态括号的各核心部件的安全设计原则，然后根据工程实现的具体场景需求，给出了面向不同应用领域的拟态括号的构建方式，并以拟态交换机的实际应用场景为例完成了包含拟态括号的拟态调度器的构建。经测试证明了所提的拟态括号构建方法完全满足拟态防御理论的白盒插装攻击抵御需求。

关键词：拟态防御、编码信道、拟态调度器

A design method of mimic scheduling adjudicator based on co-operation of hardware and software

OUYANG Ling[1,2], HE Lei[1], SONE Ke[1]

1.Information Engineering University, Zhengzhou 450002, China;

2.Zhongyuan University of Technology, Zhengzhou 450002, China

Abstract: Based on Coding Channel Theory, this paper presents the security design principles of the core components of the pseudo-state brackets, and then, according to the specific scene requirements of engineering implementation, presents the security design principles of the pseudo-state brackets, the method of constructing pseudo-parentheses for different application fields is given, and the pseudo-scheduler with pseudo-parentheses is constructed in the practical application scenario of the pseudo-parentheses. The test results show that the proposed pseudo-parenthesis construction method can meet the requirements of the pseudo-defense theory for white-box attack defense.

Key words: mimic defense; coding channel theory; mimic scheduler

1　引言

网络空间拟态防御 CMD（Cyber Mimic Defense， CMD）是邬江兴院士团队提出的一种新型的主动防御理论[1]。CMD 技术基于相对正确公理，以动态异构冗余 DHR（Dynamic Heterogeneous Redundancy，DHR）为核心架构，通过对执行体的“异构”和“冗余”设计，在绝大多数情况下，某种攻击只会影响少数异构执行体并产生输出响应，而在输出裁决时就会被当成异常输出被屏蔽；同时，结构的“动态”变换又解决了因共性缺陷或协同攻击的极少数情况下，多数执行体返回一致的攻击响应造成短期攻击成功的问题。

图 1 动态异构冗余 DHR 典型结构

为了从理论和架构上探讨 DHR 架构拟态括号构建的合理性和安全性，本文从编码信道理论入手，依据编码信道与 DHR 结构的对应关系，给出了 DHR 架构拟态括号的各核心部件的安全设计原则，然后根据工程实现的具体场景需求，给出了面向不同应用领域的 DHR 构架拟态括号的构建方式，最后选取其中一种代表性的拟态括号在拟态交换机的实际应用场景中进行了 DHR 构架的功能实现测试。

基金项目：国家核高基重大专项基金资助项目（No.2017ZX01030301）。

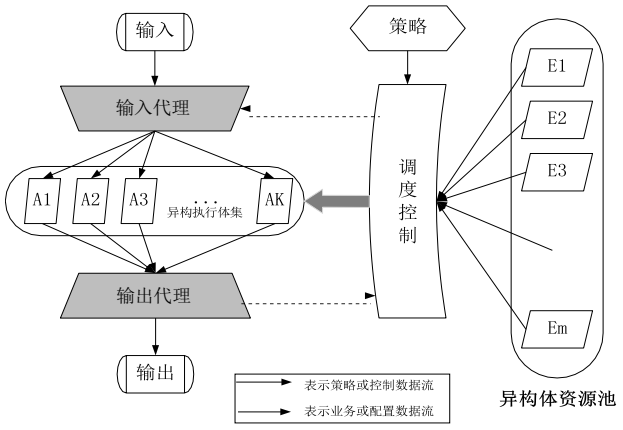


图 1　所示是 DHR 的典型结构 [2]，对 DHR 结构的研究和实现，一直都是 CMD 的研究研究热点之一。扈红超等人对拟态防御的动态异构冗余 DHR 模型若干问题进行了探讨和性能评估 [3]，朱维军等人对 DHR 结构的拟态防御自动机模型进行了研究 [4]，仝青、王禛鹏、马海龙等人基于软件实现方式，分别构建了拟态 WEB 服务器 [5]、拟态域名服务器 [6] 及拟态路由器 [7]，魏帅、宋克等人则基于硬件实现方式，构建了拟态工业处理器 [8] 和拟态交换机 [9]。构建 DHR 结构，除了功能等价的异构执行体集外，最核心的是输入代理和输出代理等环节的设计实现，这些环节是 DHR 架构的基础组件，实际上也是划分拟态界的“拟态括号”，必须在保证自身安全的前提下满足 DHR 架构的各种功能需求。

2　编码信道理论与拟态括号设计原则

2.1　编码信道理论

网络空间内生安全问题在信息系统中造成的影响表现为各种已知和未知故障，当故障被随机或非随机广义扰动激活时产生错误或失效。其中，作用于具体系统的通信随机噪声、物理性随机失效、基于内生安全问题的人为非随机攻击等扰动，在激活局部故障的同时，将导致组件或子系统的状态错误，而这些错误状态很可能造成全局性失效。具有内生或内源性安全功效的构造或算法及其体制机制使系统能够拥有内生的安全功能，该功能可以管控系统内生安全问题引发的确定或不确定安全风险。尽管内生安全问题不可避免，但通过纠正广义扰动所造成的影响，可以使被激活的故障难以发展为错误和失效。

众所周知，对于网络空间目标对象抗攻击性或攻击难度的定量分析之前缺少有效方法。原因是针对目标对象的攻击通常不是独立发生的随机性事件，其攻击效果由于目标对象的静态性、确定性、相似性和“单

向透明”等原因往往是确定的，在数学意义上表现为布尔量而非概率值，这使得抗攻击性的定量分析很难借助随机过程工具来描述。但是，对于带有多模表决/译码机制的动态异构冗余目标系统而言，“相对正确”公理首先能将针对执行体漏洞后门、病毒木马等的确定或不确定性攻击在系统效果层面变换为某种分布形式的概率事件，其次通过动态调整执行体的配置方法/编码算法、输出矢量/码字的语义丰度、裁决策略/译码算法以及变化执行体的实现算法或构造相异度，可以改变攻击效果在系统层面的概率大小和分布形态。换言之，当目标系统采用多模表决/译码机制的动态异构冗余构造时，无论是针对执行体个体的攻击事件还是执行体构件产生的随机性故障都可以在系统层面被归一化为具有概率性质的可靠性问题，从而使原本无法度量的安全防御场景可以借用成熟的概率论与数理统计方法来分析。

香农信道纠错编码理论的分析对象是“随机噪声无记忆信道”，而动态异构冗余的迭代防御场景则相当于“随机或非随机的有记忆信道”。因此，不能直接用香农理论及方法来量化分析动态异构冗余构造的安全性或广义鲁棒性，需要从信道编码理论发展出一种“编码信道”理论 [10]，以便能对动态异构冗余的“编码结构”体制机制在抑制“结构扰动噪声”方面进行形式化分析，科学指导内生安全的工程设计和实践应用活动。而编码信道理论是否成立的关键是相关存在定理的证明，需要从理论上阐明在广义扰动条件下，针对特定离散有记忆信道，如何构造合适的信道与编码来提供“正确”服务的问题。所谓“正确”的概念就是采用适当的编码与译码步骤，使具有内生安全属性的系统架构内，当存在随机或人为加性干扰时信息传递和处理的误差足够小。简而言之，编码信道理论是由内生安全构造数学模型和三个存在定理及相关定义及数学证明构成，涵盖且应当涵盖香农第二定理的内容。图 2 所示是基于编码信道的安全防御模型，其中结构编码对应拟态防御架构中的输入代理，元信道对应执行体，纠错译码对应输出裁决器，反馈控制与信道记忆消除对应反馈控制器。

我们提出了信道编码理论的 3 个存在性定理，相关定理的数学证明详见《网络空间内生安全-拟态防御

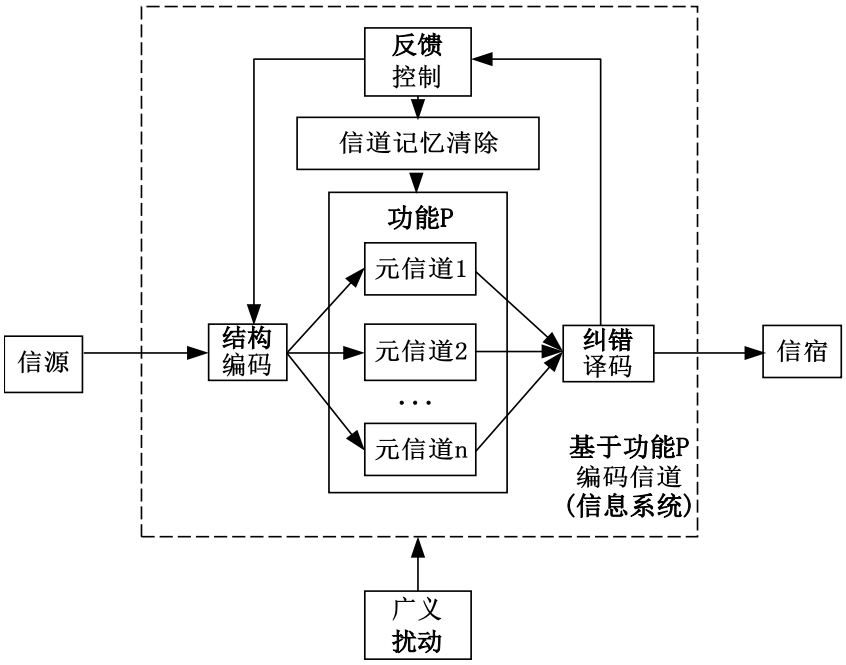


图 2　基于编码信道的安全防御模型

与广义鲁棒控制》第 15 章[10]。其中，第一定理证明了，在结构编码和纠错译码无扰动和无记忆，以及元信道有随机扰动和无记忆条件下，能够使平均译码错误概率任意小的编码信道的存在性。第二定理证明了，在结构编码和纠错译码无扰动和无记忆，以及元信道有非随机扰动和有记忆条件下，能够使平均译码错误概率任意小的编码信道的存在性。第三定理证明了，在结构编码和纠错译码有扰动和有记忆，以及元信道有非随机扰动和有记忆条件下，能够使平均译码错误概率任意小的编码信道的存在性。

编码信道存在第一定理：元信道噪声（扰动）随机到达，结构编码、纠错译码和反馈控制无扰动和无记忆，离散无记忆元信道［P（y|x）］，编码信道容量为 C，噪声随机到达，ε 为任意小正数，若编码信息传输率 R<C，存在无记忆元信道 n 足够大的编码信道，总可以在输入集合找到 $M' = 2^{nR}$ 个码字组成一个码集合（码长为 n），在一定译码规则下，可使信道输出错误概率 $Pe \leqslant \varepsilon$。

编码信道存在第二定理：元信道噪声（扰动）非随机到达，结构编码、纠错译码和反馈控制无扰动和无记忆，动态异构冗余与反馈消记忆构造后的离散有记忆编码信道［P（y|xsf）］，编码信道容量 C，

$\forall t > 0$，C（t）$\in$［Cs，$C0$］，若 t 时刻编码信息传输率 R（t）<C（t），则只要码长与编码元信道构造数 n 足够大，总可以在输入集合找到 $M = 2^{nR}$ 个码字组成一个码集合，ε 为任意小正数，在一定译码规则下，可使编码信道输出错误概率 Pe（t）$\leqslant \varepsilon$。

编码信道存在第三定理：元信道、结构编码、纠错译码和反馈控制噪声（扰动）非随机到达，结构编码、纠错译码和反馈控制有记忆且记忆可消，动态异构冗余与反馈消记忆构造后的离散有记忆编码信道［P（y|xsf）］，编码信道容量 C，$\forall t > 0$，C（t）$\in$［Cs，$C0$］，若 t 时刻编码信息传输率 R（t）<C（t），则只要码长与

编码元信道构造数 n 足够大，总可以在输入集合找到 $M = 2^{nR}$ 个码字组成一个码集合，ε 为任意小正数，在一定译码规则下，可使编码信道输出错误概率 Pe（t）$\leqslant \varepsilon$。

编码信道存在定理所研究的基本问题，就是在广义扰动条件下针对特定离散有记忆信道，如何构造合适的信道与编码来提供正确的服务。所谓“正确”的概念就是采用适当的编码与译码步骤，使存在随机与人为加性干扰时信息传递和处理的误差足够小。如用形象的语言来描述编码信道定理，就是信道提供正确服务的充要条件是能

够把“错误”关在“监督纠错”的笼子里。

总之，香农第二定理是编码信道存在定理的一种特例，编码信道存在定理能够适用于随机和非随机噪声、离散无记忆和有记忆信道，能够一体化解决系统可靠性、通信可靠性和防御可信性问题。编码信道理论为网络空间攻防博弈条件下的安全防御及可靠计算及通信建立了数学理论基础，指明了可行的发展方向，提出了一种可工程化实现的途径。

2.2 基于编码信道理论的拟态括号设计原则

根据 CMD 理论，在 DHR 的应用协议执行体基础上，进一步对拟态括号进行拟态构造设计，即将拟态

括号（分发和裁决单元）的基本功能与复杂功能分离。其中，将拟态括号“复杂功能”通过动态异构冗余方式实现，可以防御基于漏洞后门的随机/非随机扰动的影响，分解后的结构如图 3 所示。

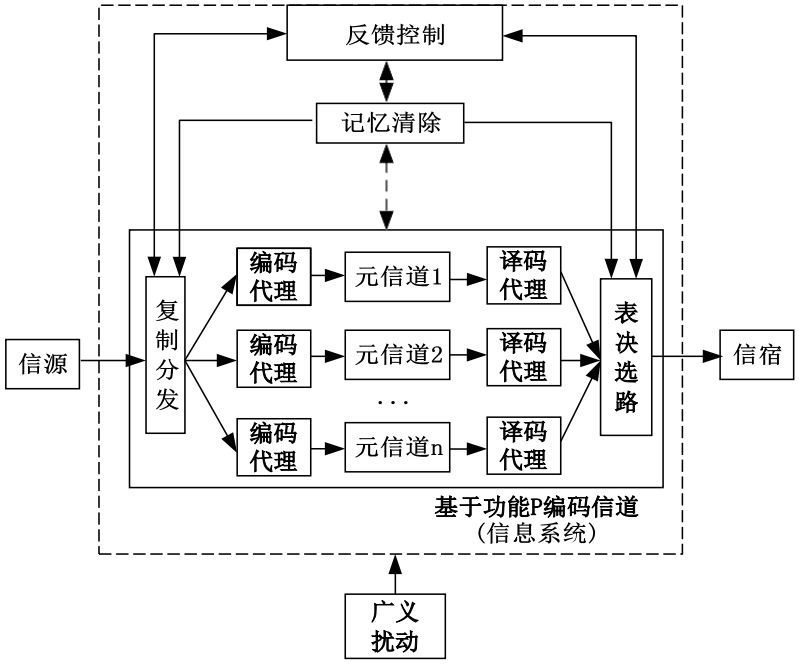


图 3　DHR 模型拟态括号功能分解结构

根据上述编码信道理论分析，拟态括号“简单功能”还必须满足两个必要条件：无后门和可消除记忆。一个安全的拟态括号必须满足如下假设条件：

假设 1

对于结构编码单元，通过对其进行拟态化设计，可以将其划分为分路器和结构编码执行体。

● 分路器功能足够简化，只需完成状态无关的输入报文复制分发。

□ 在采用分光器等物理器件实现时，假设分路器中既没有漏洞也没有后门，对于攻击等非随机扰动不具有记忆性；

□ 在采用逻辑固化/硬化/防篡改、每次处理前存储初始化等机制时，功能足够简化可进行形式化验证，假设分路器中有漏洞无后门，对于攻击等非随机扰动的记忆可消除。

● 结构编码单元除分路器之外的复杂功能由编码执行体实现。

□ 结构编码执行体采用类似元信道的方式实现，假设其中可以同时存在漏洞和后门，可以有记忆或无记忆。

假设 2

对于纠错译码单元，通过对其进行拟态化设计，可以将其划分为表决器和纠错译码执行体。

● 表决器功能足够简化，只需完成状态无关的输出报文大数表决。

□ 在采用逻辑固化/硬化/防篡改、每次处理前存储初始化等机制时，功能足够简化可进行形式化验证，假设表决器中有漏洞无后门，对于攻击等非随机扰动的记忆可消除。

● 纠错译码单元除表决器之外的复杂功能由纠错译码执行体实现。

□ 纠错译码执行体采用类似元信道的方式实现，假设其中可以同时存在漏洞和后门，

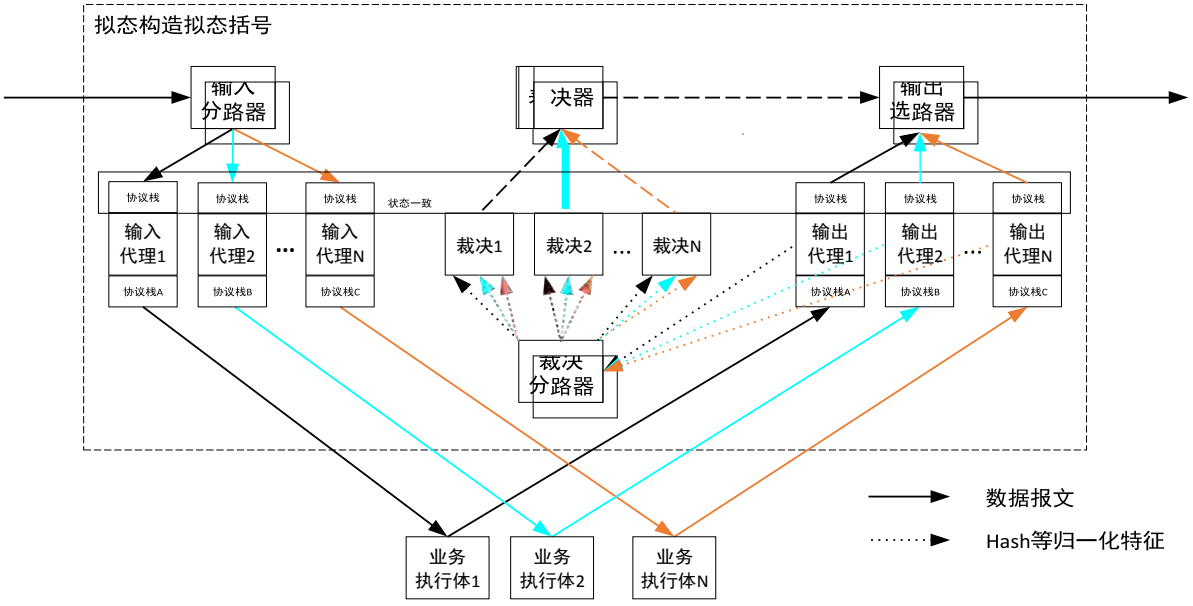
可以有记忆或无记忆。

3 拟态括号的典型构建方案

根据上述设计原则，可以总结出拟态括号的主要设计思路：1）对拟态括号进行拟态化设计，将基本功能与复杂功能分离；2）基本功能包括左括号的分路器和右括号的表决器和选路器，因为其足够简化，所以可以通过形式化验证方式确保没有后门；3）复杂功能可能同时存在漏洞和后门，可以通过 DHR 括号执行体方式实现，即输入代理执行体、裁决执行体和输出代理执行体。

3.1 拟态括号理想构建方案

满足编码信道理论要求的拟态括号的拟态构造理想体系结构如图 4 所示，主要包括输入分路器、裁决分路器、表决器、输出选路器以及输入代理、输出代理和裁决器等部件，各部件的功能及安全性要求如下：



执行体编号向量

图 4 拟态括号的拟态构造理想体系结构

输入分路器：完成用户请求报文的复制和分发，即将用户请求报文复制 N 份，并分发给输入代理执行体；要求无后门且可消除记忆，可以采用分光器等物理器件或者硬化逻辑且每次存储器使用前进行初始化等方式实现。

裁决分路器：完成执行体响应报文的归一化特征值的复制和分发，即将执行体响应报文的归一化特征值复制 N 份，并分发给裁决执行体；要求无后门且可消除记忆，可以采用分光器等物理器件或者硬化逻辑且每次存储器使用前进行初始化等方式实现。

表决器：采用大数表决方式，对输入的 N 个执行体编号向量，逐位进行择多，输出 1 个每位都是多数一致的执行体编号向量；要求无后门且可消除记忆，可以采用硬化逻辑且每次存储器使用前初始化的方式实现。

输出选路器：根据执行体编号向量，随机选择一个经过输出适配后的 1 个执行体响应报文；要求无后门且可消除记忆，可以采用硬化逻辑且每次存储器使用前初始化的方式实现。

输入代理执行体：终结用户通信连接，发起执行体连接，将用户业务请求报文进行指纹消除、随机变换等适配后送往对应的业务执行体；可以有漏洞后门但可消除记忆，根据功能复杂度采用软件或硬件异构执行体完成。

输出代理执行体：终结执行体连接，发起用户通信连接，将执行体业务响应报文进行指纹消除、随机变换等适配后，生成归一化特征值

（如 Hash 值）送往裁决分路器；可以有漏洞后门但可消除记忆，根据功能复杂度采用软件或硬件异构执行体完成。

裁决执行体：根据裁决算法，对输出代理的多个归一化特征值进行裁决，输出以执行体编号矢量为内容的裁决结果；可以有漏洞后门但可消除记忆，根据功能复杂度采用软件或硬件异构执行体完成。

3.2 拟态括号的工程化构建方案

图 4 所示的拟态括号理想构建方案是完全基于编码信道理论构建的，从机理上具备最高的安全性，适用于拟态云、拟态数据中心等大型复杂的拟态场景的构建。对于拟态交换机、拟态路由器、拟态网关或者拟态 web、拟态 DNS 服务器等网络设备或端点设备的构建，该方案存在实现成本高、结构庞杂、技术复杂等问题，需要进行合理的简化以利于工程化实现。

我们可以将图 4 中的各部件分为两类，一类是包括输入分路器、裁决分路器、表决器、输出选路器在内的非冗余部件，这些部件实现的功能相对简单，可以通过硬件逻辑实现，避免了后门的植入和操作系统、软件漏洞带来的安全风险；另一类包括了输入代理、输出代理和裁决器等部件，这些部件若需要通过软件实现比较复杂的功能，就要按照异构冗余的思想进行构建，通过大数裁决机制抵御操作系统、软件漏洞带来的安全风险。所以，拟态括号的工程化简化关键问题，就是在于对于输入代理、输出代理和裁决器的功能进行需求分析，评估能够采用硬件逻辑实现的场景，以进行整体架构的简化。

首先，对输入代理进行分析。输入代理的作用是终结用户通信连接，发起执行体连接，并将用户的业务请求数据送往异构执行体。通常情况下，输入代理不需要对用户的业务请求数据进行特殊的处理，主要完成的工作就是数据载荷的分发。因此，用户的业务请求数据完全可以做到在输入代理中的“透明传输”。所谓“透明传输”，就是输入代理不需要对数据进行深入的内容解析，这样就避免了用户数据中可能存在的恶意报文针对输入代理的攻击和破坏，同时也利于采用硬件逻辑实现输入代理的功能。因此，通过上述分析，完全可以将异构冗余的输入代理执行体进行简化，采用单一的基于硬件逻辑的执行体实现。与此同时，还可以将输入分路器的复制功能纳入到输入代理中，这样就可以得到图 5 所示的拟态括号的拟态构造输入简化体系结构。

进一步对输出代理和裁决器进行分析。由于业务执行体的异构性，通常情况下，即使针对同样的用户业务请求数据的输出，也会存在较大的形式上的差异。这种差异本质上并不是异常，如果直接由裁决器进行基于大数表决算法的比对，则有可能造成判断错误。因此，需要采用输出代理进行执行体输出数据的一致性处理，以尽力消除形式上的差异。这种数据的一致性处理和比对操作，在拟态 web 服务器、拟态 DNS

服务器以及拟态加密系统[11]，拟态云环境[12]、拟态数据防护[13]、拟态区块链[14]、拟态大数据[15]等复杂的应用场景时，通常都是采用软件的方式实现的，这就要求采用异构冗余的方式保证输入代理和裁决器的自身安全性；而在面对拟态交换机、拟态网关等协议相对固定的场景时，则可以通过硬件逻辑实现，这样就可以进一步简化拟态括号的结构，如图 6 所示。

拟态构造拟态括号

决器

输出
路器表

选

输出代理2

输出代理1

输入代理

裁决分路器

裁决N

数据报文

拟态构造拟态括号

协议栈 输出选路器

输入分路器

数据报文

图 5 和图 6 所示的两种工程化实现方案，在不降低拟态括号自身安全的前提下，降低的拟态括号的构

建难度和实现成本，在面对不同的拟态设备时，可以根据应用需求进行选择。需要说明的是，如果图 5 中将异构冗余的输出代理的功能整合进业务执行体中，并且将用于裁决的数据进行例如 Hash 运算的方式进

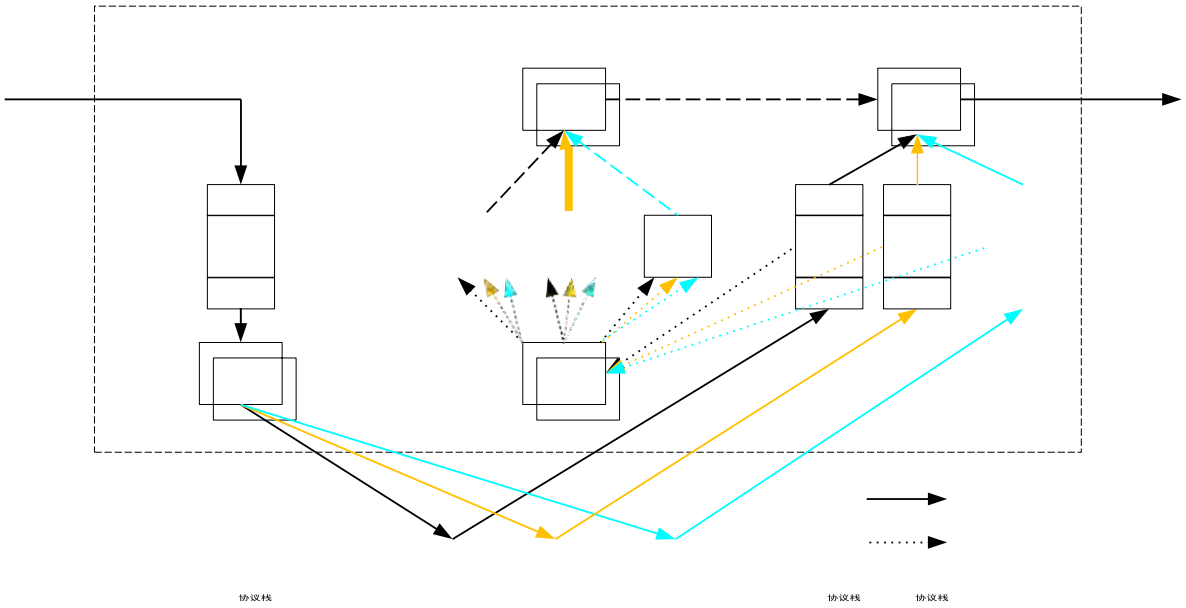


图 5　拟态括号的拟态构造输入简化体系结构

Hash 等归一化特征执行体编号向量

行归一化，则构建的拟态括号可以进一步简化。

4　拟态调度器的构建实例与测试

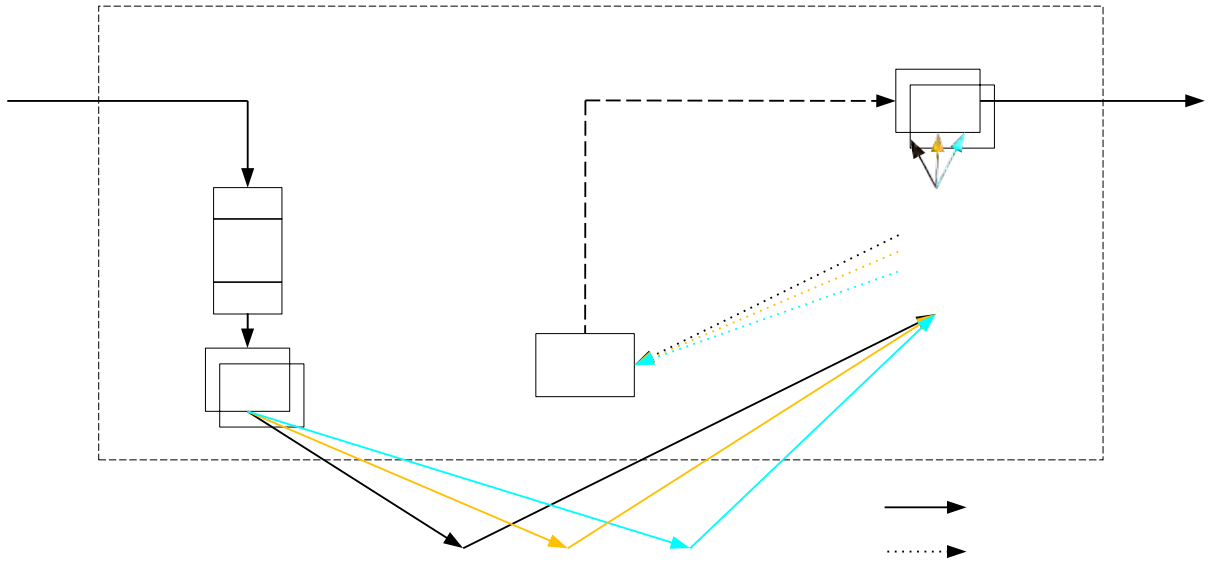
4.1　简化体系结构的拟态调度器的工程实现

在本文提到的三种拟态括号的构造体系结构中，简化的体系结构具有更为重要的工程实现意义。因为在这种体系结构下，拟态括号的核心部件均是通过硬件逻辑实现，在保证安全性的前提下，保证了拟态系统的运行效率；此外，还可将 DHR 结构中的调度控制功能集成进来，与输入代理、输出代理共同组成完整的拟态调度器系统，大大降低拟态设备构造的成本和技术难度。

图 7 构建了一种拟态调度器，主要有上行通道、下行通道及策略控制等部分组成。其中，上行通道完成用户业务数据的输入代理，主要包括 TOE 上行输入数据解链、DMA 数据缓存与传输、上行数据调度与传输、TOE 上行输出数据封装与分发等模块，完成拟态“左括号”功能；下行通道异构执行体响应数据的输出代理，主要包括 TOE 下行输入数据调度、数据缓存与预处理、数据判决比对、下行数据封装与传输等模块，完成拟态“右括号”的功能；策略度控制完成裁决比对策略及清洗调度控制，主要包括策略控制器、清洗恢复的数据缓存、传输与控制、执行体状态检测等模块，完成 DHR 结构的调度控制等功能。

4.2　基于拟态调度器的交换机验证平台

为了验证基于简化体系结构的拟态调度器的实际应用效果，本文构建了如图 8 所示的拟态交换机验证平台。该平台主要包含了拟态调度器和三个异构执行体。拟态调度器由 Xilinx zynq-7045 FPGA 实现，通过硬件逻辑实现了图 7 所示的各模块功能，其中策略控制器由内嵌的 ARM 处理器实



协议栈
输出代理
协议栈B

输入
代理

协议栈B

裁决
表决器

业务
执行体1

业务
执行体2

业 务
执 行 体

图 6　拟态括号的拟态构造简化体系结构

Hash 等归一化特征执行体编号向量

现。三个异构执行体分别由 ARM 架构、MIPS 架构和 PowePC 架构的 CPU 运行不同的操作系统组成，运行通过多样化编译后的交换机协议栈软件。

交换芯片的上行数据由 PCIe 通道交由拟态调度器，复制后透明分发到三个异构执行体，由异构协议

栈及管理软件进行处理；处理完成的数据返回拟态调度器，判决后交由 SDK 处理或直接下发交换芯片。拟态调度器将服务功能请求分配给异构执行体，再接收其输出判决得出正确的服务响应。当各个执行

体工作正常时，会给出一致的服务响应，调度器判决正确后选择一路输出。当某个执行体受到攻击产生异常输出时，调度器能够判决识别该异常，然后将多数的正常响应输出，并对异常执行体进行清洗恢复操作，将其受攻击状态复位回正常状态，以此达到抵御攻击的目的。除了异常触发的复位恢复外，调度器还会定期对三个执行体进行清洗恢复操作，以清除潜在的、未被判决发现的异常状态，保证系统的正常运行。

4.3　验证平台白盒插装测试

验证测试采用 CMD 理论中经典的白盒插装测试方法，主要包括一个执行体被攻击（差模攻击）、二个执行体被攻击（N-1 模攻击）、三个执行体被攻击（共模攻击）三种攻击场景下，测试转发表处理、二层协议解析、三层协议解析以及配置管理功能的运行情况。

1)　*差模攻击测试*

通过白盒插桩，使三号异构执行体的运行输出与一号和二号执行体不一致，开始阶段，三个异构执行体均正常运行（图 9（a））；当错误计数到达阈值时，三号执行体显示异常（图 9（b））；接着开始对其发起清洗操作，此时另外两个执行体正常工作（图 9（c））；清洗完成后，三个执行体恢复正常运行（图9（d））

（a）　（b）

2)　*N-1 模攻击测试*

通过白盒插桩使一号和二号异构执行体一致的输出错误，三号执行体正常输出。开始阶段，各执行体均正常运行（图 10（a））；不一致发生后，对三号进行清洗（图 10（b））；清洗恢复后

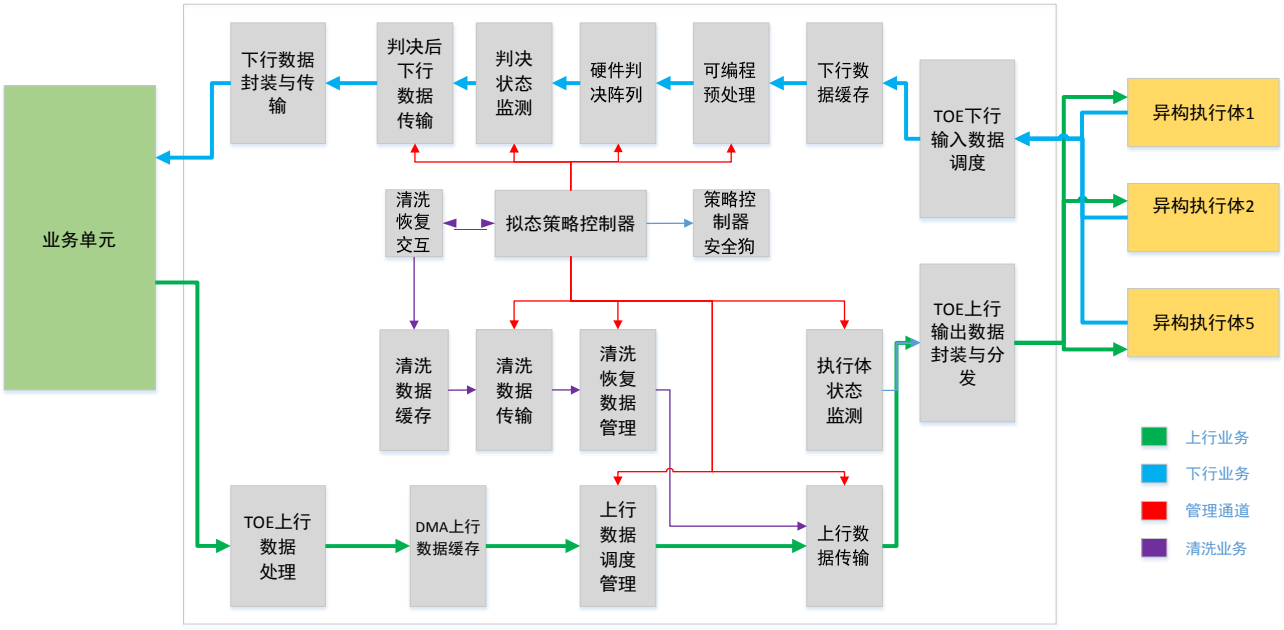


图 7　基于简化体系结构的拟态调度器

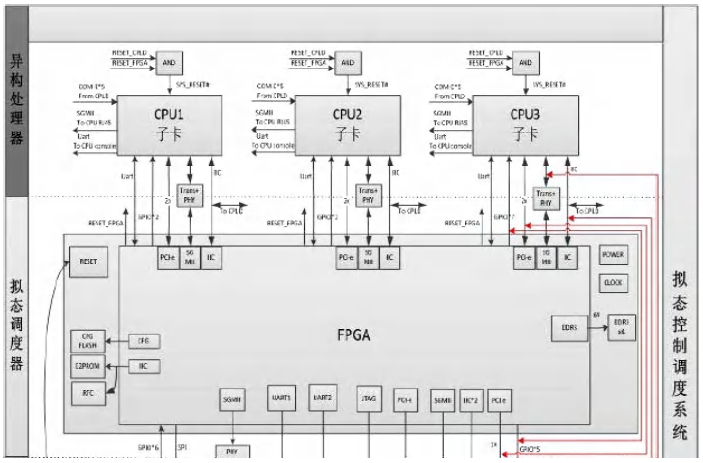
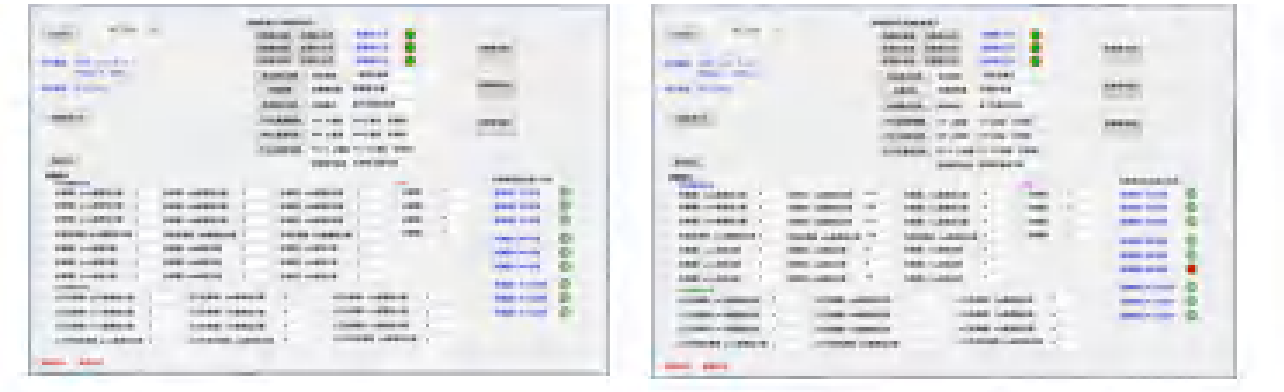
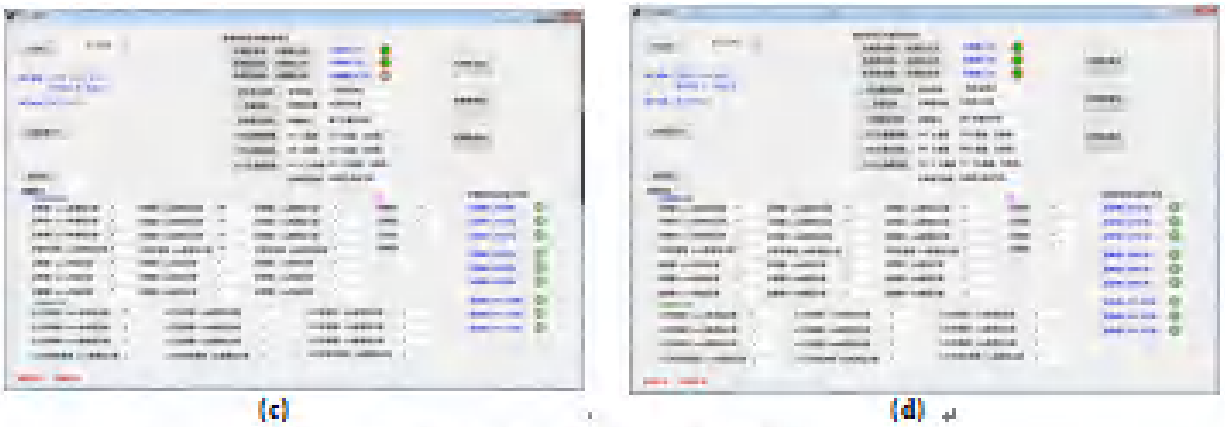


图 8　拟态调度器验证平台组成

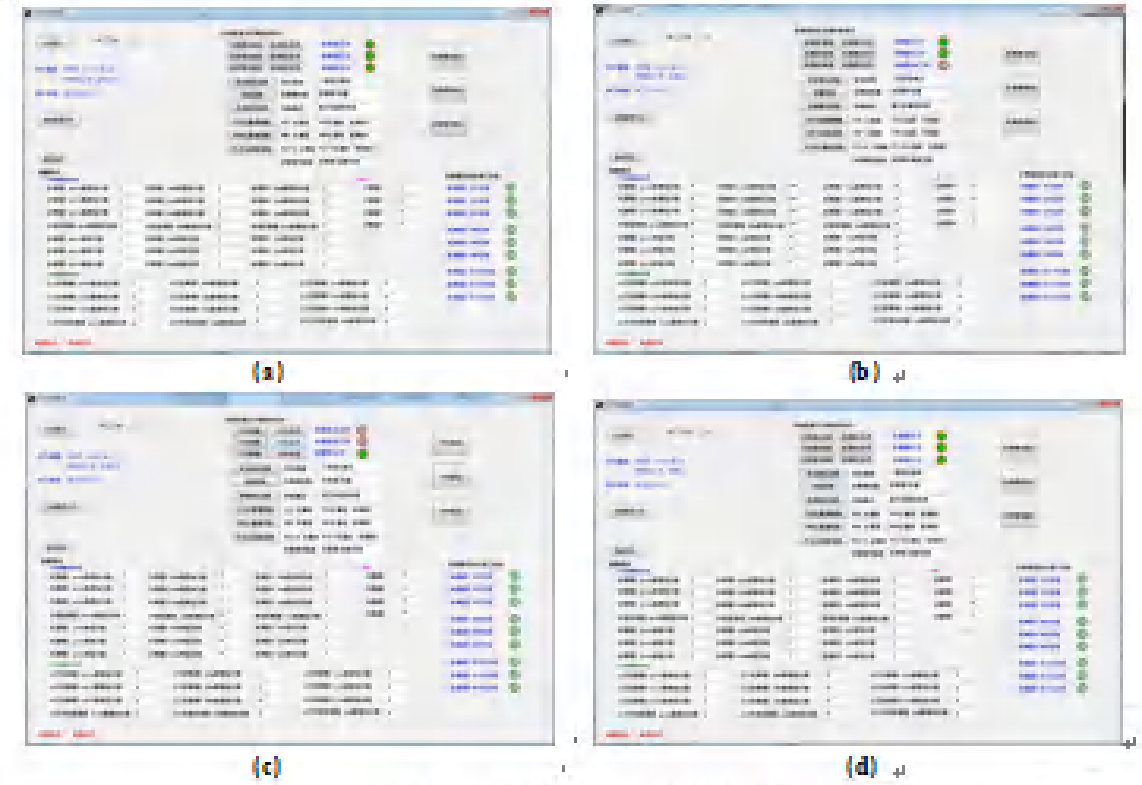


仍存在不一致，则同时清洗一和二号，此时三号工作（图 10（c））；清洗后恢复正常（图 10（d））。



(c)　(d)

图 9　基模攻击执行体运行状态图



(a)　(b)

(c)　(d)

图 10　N-1 模攻击执行体运行状态图

3) 共模攻击测试

通过白盒插桩使三个异构执行体输出一致错误。开始阶段，三个异构执行体均正常运行（图 11（a））；定时器触发，系统对三号执行体进行清洗，此时一和二号执行体工作（图 11（b））；三号恢复后出现不

一致，清洗二号执行体，此时一和三号执行体工作（图 11（c））；对一号执行体进行清洗，此时二和三号执行体工作（图 11（d）所示）；清洗完成后，三个执行体恢复正常运行（图 11（e））。

4) 测试结果对比

各个攻击模式下的拟态防御等级变化如图 12 所示。在差模下，系统仅有短时降级（由三降为二，即由三个执行体降为二个执行体同时工作）；在 N-1 模下，会出现降到一级情况（由调度策略

图 11 共模攻击执行单元运行状态图

决定，目的是缩短恢复时间），且降级时间长于差模；共模攻击下，系统降级时间最长，且会有一段“威胁无感”时间。

针对各种攻击方式，内生安全交换机的拟态运行模式与非拟态运行模式（只有一个执行体工作）输出结果对比如 13 所示。测试中对转发表（1800 测试例）、二层协议报文（1600 测试例）、三层路由协议报文（1300 测试例）以及远程管理的身份认证（500 测试例）等进行插桩测试。由结果可以看出：非拟态模式下，白盒插桩的全部攻击都能成功。而对于拟态模式，在共模攻击时由于存在一定时期的“威胁无感” 和“判决欺骗”，所以有较多的攻击达成；N-1 模攻击下，由于存在一定时期的“判决欺骗”造成了攻击逃逸，所以会有少量的攻击达成；而在差模攻击下，所有的攻击全部都被裁决机制屏蔽，所以没有任何攻击达成。

图 14 展示了稳态恢复时间。可以看出，非拟态模遭受攻击后不具备自行恢复能力；而拟态模式在遭受共模攻击和 N-1 模攻击后，均会在一段时间攻击逃逸后恢复正常，而差模攻击则一直保持正常状态。

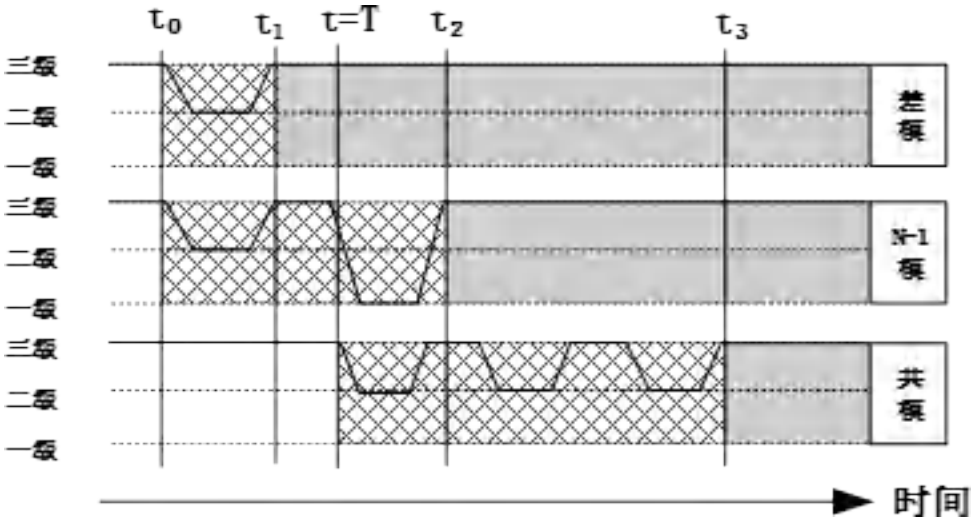


图 12　不同攻击模式的拟态防御安全等级转换

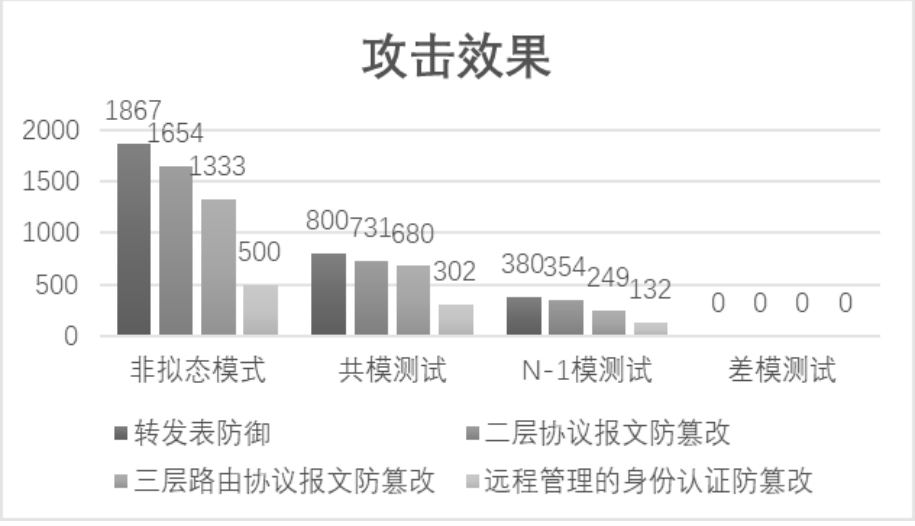


图 13　不同攻击模式下的攻击效果图

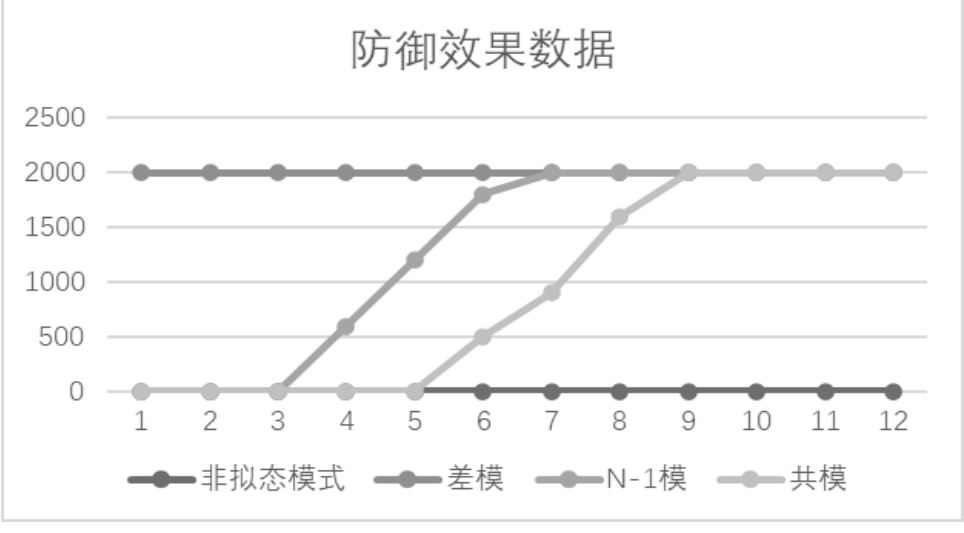


图 14　不同攻击模式下的防御效果图

5 结束语

拟态调度器将DHR架构中的输入分发、策略调度、输出裁决等功能融为一体，其功能及性能不仅决定了整个基于DHR架构的拟态防御系统能力及服务品质，而且作为DHR结构的拟态括号，其安全性直接决定了系统的安全性能。本文基于编码信道理论提出的拟态调度器设计方法，核心是关注于拟态括号的安全性设计，并根据不同的应用场景提出了多种构建方案，对拟态设备的构建具有重大指导意义。

基于本文提出的拟态调度器构建模型，可以开发面向不同应用场景的拟态调度器器件，包括各种CPU组件及FPGA平台，并可以在此基础上研发专用的ASIC或SoC集成电路芯片，对于拟态防御体系的整体产业化具有极大推动作用。

参考文献:

[1] 邬江兴. 拟态计算和拟态安全防御的原意和愿景[J]. 电信科学，2014，30(7)：1-7.

[2] 邬江兴. 网络空间拟态防御导论[M]. 北京：科学出版社，2017.

[3] 扈红超, 陈福才, 王禛鹏. 拟态防御 DHR 模型若干问题探讨和性能评估[J]. 信息安全学报，2016，1(4)：40-51.

[4] 朱维军, 郭渊博, 黄伯虎. 动态异构冗余结构的拟态防御自动机模型[J]. 电子学报，2019，47(10)：2025-2031.

[5] 仝青, 张铮, 张为华, 等. 拟态防御 Web 服务器设计与实现[J]. 软件学报，2017，28(4)：883-897.

[6] 王禛鹏, 扈红超, 程国振. 一种基于拟态安全防御的 DNS 框架设计[J]. 电子学报，2017(11)：2705-2714.

[7] 马海龙, 伊鹏, 江逸茗等. 基于动态异构冗余机制的路由器拟态防御体系结构[J]. 信息安全学报，2017，2(1)：29-42.

[8] 魏帅, 于洪, 顾泽宇, 等. 面向工控领域的拟态安全处理机架构[J]. 信息安全学报，2017，2(1)：54-74.

[9] 宋克, 刘勤让, 魏帅, 等. 基于拟态防御的以太网交换机内生安全体系结构[J]. 通信学报，2020，41(05)：18-26.

[10] 邬江兴. 网络空间内生安全-拟态防御与广义鲁棒控制[M]. 北京：中国科学出版社，2020.

[11] Shannon C E, Weaver W. The Mathematical Theory of Communication. Urbana：University of Illinois Press，1949.
Li B, ZhOU Q, Si X, et al. Mimic Encryption System for Network Security[J]. IEEE Access，2018，6：50468-50487.

[12] WANG Y, WU J, GUO Y, et al. Scientific workflow execution system based on mimic defense in the cloud environment[J]. Frontiers of Information Technology & Electronic Engineering，2018，19(12)：1522-1536.

[13] 刘彩霞, 季新生, 邬江兴. 一种基于 MSISDN 虚拟化的移动通信用户数据拟态防御机制[J]. 计算机学 报，2018，41(2)：275-287.

[14] 徐蜜雪, 苑超, 王永娟, 付金华, 李斌. 拟态区块链——区块链安全解决方案[J]. 软件学报，2019(6)：1681-1691.

[15] LIN Z, Li K, HOU H, et al. MDFS: A mimic defense theory based architecture for distributed file system[C]//2017 IEEE International Conference on Big Data (Big Data). IEEE，2017：2670-2675.

[16]

[作者简介]

欧阳玲（1978—），女，博士研究生，副教授，主要研究方向是网络空间安全和自动化控制技术。

贺磊（1974—），男，博士，副研究员，主要研究方向是网络空间安全与拟态防御技术。

宋克（1976—），男，博士，副研究员，主要研究方向是网络空间安全与集成电路设计技术。

面向VOQ缓存的故障端口环回传输容错机制

汤先拓[1]，刘冬培[1]，宋克[1]，李丹丹[2]
[1]中国人民解放军战略支援部队信息工程大学 河南郑州 450002；
[2]天津市滨海新区信息技术创新中心，天津，300000

摘　要：VOQ 缓存机制下VC 与其I/O 通道之间具有紧耦合关系，现有容错路由策略一般仅将故障定义或细化为节点或节点内部缓存和 I/O 通道，存在资源利用率和容错性能不高等问题。为此，本文构建了一种适用于VOQ缓存特点的网络故障模型和输入链路故障端口环回传输机制（FPLT），该机制可通过利用输入链路故障端口中的VC 和 I/O 通道来提高报文从最优端口执行转发的概率。实验结果表明FPLT 机制能以较小的硬件开销实现容错性能的较大提升。

关键词：虚拟输出队列、片上网络、环回传输、容错路由、网络死锁

A Faulty Ports Loopback Transmission Fault-Tolerant Mechanism for VOQ Buffer Architecture

TANG Xian-tuo[1], LIU Dong-pei[1], SONG Ke[1], LI Dan-dan[2]
1.The PLA Information Engineering University, Zhengzhou, 450002, China;
2.Information Technology Innovation Center of Tianjin Binhai New Area, Tianjin, 300000, China

Abstract: There is a tight coupling relationship between VC and its I/O channel under VOQ buffering mechanism, and the existing fine-grained fault-tolerant routing algorithms usually only refine the fault granularity to the internal port buffer and I/O channel, There are some problems such as low resource utilization and low fault tolerance performance. Therefore, this paper constructs a fine-grained network fault model according to the VOQ buffering characteristics, and then proposes a fault-tolerant routing algorithm based on loopback transmission of faulty ports (FPLT) to improve the probability of packet forwarding to the optimal port, by using the abandoned VC and I/O channels within the input fault link port. The experimental results show that FPLT can improve the fault-tolerant performance with less hardware overhead.

Key words: virtual output queue; network-on-chip; loopback transmission; Fault tolerant routing; network deadlock

1　引言

虚拟输出队列（Virtual Output Queue， VOQ）是片上输入 VC（Virtual Channel）路由器结构中一种高效能的实现方案，其特点是路由器各输入端口均为每个输出端口预留一条专用的 VC，可从根本上消除 HoL（Head-of-Line）阻塞现象［1］。相对于传统输入 VC 路由器结构而言，VOQ 路由器由于无需 VC 分配流水栈，往往具有更低的报文传输延迟。然而，VOQ 缓存机制下 VC 与其对应 I/O 通道之间存在紧耦合关系，VC 故障将导致对应 IO 通道的不可用，反之 IO 通道故障亦将导致

VC的不可用。

传统容错路由的故障表示粒度通常在节点或节点以上层次。如文献［2］提出了一种基于区域故障的容错算法，通过利用故障环和故障链对网络故障进行隔离，报文采用绕道路由来避开故障区域。基于逻辑的分布式路由算法（LBDR）［3］通过连通位来表示故障状态，并采用算法路由实现低开销的容错处理。在此类容错机制下，节点内部细微故障即可导致整个节点甚至节点区域的不可用，造成片上资源的极大浪费。为提高资源利用率，细粒度网络故障模型和容错策略开始广泛地应用于NoC容错领域。如文献［4］提出了一种基于功能故障模型的容错算法（FFBR），将故障粒度细化到节点内部的I/O通道。文献［5］针对网络中节点、链路与I/O通道故障进行了细化并提出了一种自适应容错路由算法（FFAR）。然而，这些容错算法的容错粒度虽有所细化，但其对网络中各类故障进行映射处理的方式会不可避免地造成节点中正常缓存与通道资源的浪费。为此，文献［6］提出了一种微粒度的网络故障模型，通过改进故障映射方法和采用一种基于缓存再利用的透传机制（TTBR）来提高节点内部资源利用率和容错性能，但该容错机制未对端口缓存中VC故障进行区分，在VOQ缓存结构下单个VC的故障即可导致整块端口缓存中所有VC的废弃，甚至还可能进一步造成与该缓存端口相连的crossbar中所有IO通道的废弃，因此难以有效适应VOQ缓存下对于VC和I/O通道故障的细粒度容错需求。

本文中我们将针对VOQ片上路由器容错设计，首先根据VOQ缓存特性构建一种粒度更小的网络故障模型，并以此为基础提出了一种基于输入链路故障端口环回传输机制的容错机制（FPLT）。FPLT容错机制通过借鉴TTBR机制的实现思路，并重点针对网络故障表示粒度和废弃资源“再利用”的条件等两个方面进行了优化处理，以期进一步提高网络资源的利用率和容错性能。

2　故障端口环回传输机制

2.1　网络故障模型

基于VOQ缓存的片上通信架构中可能发生的故障主要包括链路故障、节点故障两类，其中节点故障根据故障位置和功能的不同可细分为VC故障、通道故障和控制故障等［7］［8］。在片上网络中，节点内部的端口缓存和IO通道是硬件开销的最大来源，而其它控制组件（如路由、开关分配逻辑等）硬件开销一般较小，发生故障的概率亦相对较小。为此，本文主要考虑VC、IO通道、链路故障等三种类型。网络故障的诊断通常在节点内部采用内建自测试方法来进行［9］。

针对不同的故障类型，传统容错路由算法通常采用统一的故障映射方法来对其进行功能处理，如将链路故障映射为当前节点和下级节点中与该故障链路相连端口对应的所有IO通道的故障，将缓存故障映射为与该缓存相连端口对应的所有IO通道的故障。为此，网络中的链路、VC、IO通道故障将统一映射为节点中12条I/O通道的故障，如图1（左图）所示。这种故障映射方法虽可减少存储故障信息所需缓存空间、简化算法复杂度，但将导致资源的极大浪费。为了提高片上资源利用率，本文将对于网络中的链路、VC和I/O通道故障进行区分和存储。由于本地端口发生故障通常需丢弃整个节点，因此我们在节点中仅关注N、S、W、E四个转发方向的VC、I/O通道和链路的故障状态。

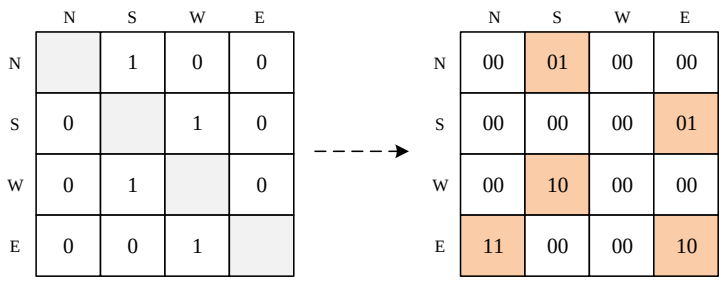


图1　VC与I/O通道故障表示

（1）对于节点内部的VC与I/O通道故障，故障信息以矩阵形式表示，如图1（右）所示。其中纵、横轴代表N、S、W、E四个输入输出方向。根据VOQ的紧耦合缓存特性，我们可将VC与其对应的Crossbar I/O通道的故障状态存储在同一矩阵单元｛VC故障，通道故障｝中。每个节点共需要32 bit缓存用来存储各端口VC与I/O通道的故障信息。

（2）对于节点之间的链路故障，故障信息主要记录N、S、W、E方向四条链路的故障状态，单方向的链路故障表示结构为｛输入链路故障、输出链路故障｝，如图2所示。每个节点需要8bit缓存用来存储四个转发方向的链路故障信息。

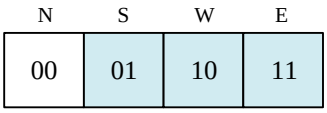


图2　链路故障表示结构

具体实现时，每个网络节点都需要增加一个40 bit缓存用于存储当前节点的VC与I/O通道故障状态和其4个转发方向的链路故障状态。

2.2　环回传输机制

故障端口环回传输（Fault Port Loopback Transmission，FPLT）机制的基本思路是采用节点内部“二次转发”的方法，重分利用因输入链路发生故障而弃用的端口中正常VC和I/O通道来尽可能地增加报文在网络故障状态下从最优端口执行转发，减少报文在网络层面进行绕道路由甚至回退至上级节点的概率。在FPLT机制下，当报文因节点内部I/O通道故障无法获得最优转发时，报文可先转发至其他存在输入故障链路的端口，再采用端口环回传输，通过“二次转发”发送至其最优转发端口。如图3所示，到达N端口报文的最优转发端口为S而N－>S通道发生故障，此时报文将无法利用从N－>S通道直接转发至端口S，但在FPLT机制下该报文可首先传输至输入链路故障端口W并根据其最优转发方向缓存在相应的VC中，然后按照正常的路由流水过程转发至其最优端口S，以期增加报文从最优端口执行转发的概率。

具体实现时，路由器N、S、E、W四个方向输出与输入端之间将增加一条额外的数据通道，称之为“环回通道”（Loopback Channel，LC），如图3所示。环回通道中主要包括1个多路转接器（MUX）和1个多路分离器（DeMUX），用于控制输入输出端口间数据流的传输。报文利用FPLT机制执行转发时将携带一个环回标识（Loopback_Flag）用于控制环回通道中DeMUX的连接关系，而当前端口链路的故障状态则可用来控制环回通道中MUX的的连接关系。当报文采用这种“二次转发”的方式到达某输入故障链路端口的输出端后，将利用环回通道传输至该端口的输入侧。报文到达输入侧后将按照其最优转发端口选择存入相应的VC中，其后报文将按照正常的路由流水方式，依次经历开关分配、开关传输流水处理后到达其最优转发端口。此外，为不影响正常报文的路由转发，交叉开关分配时我们指定“二次转发”的报文具有最低的优先级。

报文执行FPLT机制的条件主要包括：（1）报文所在端口与最优转发端口之间的通道故障。（2）节点内部存在输入链路故障端口（即某端口的输入链路故障）；（3）报文所在端口至输入链路故障端口的VC和通道都没有故障；（4）输入链路故障端口至报文最优转发端口的VC与通道都没有故障。

不同于TTBR机制，FPLT机制针对节点内部的VC与通道故障进行分别定义与表示，同时对于废弃资源“再利用”的条件进行了弱化。TTBR机制需要端口输入输出链路均故障时对应端口弃用资源才可被复用，在此条件下端口缓存和通道资源“再利用”概率将大大减少。而FPLT机制仅要求某方向输入链路故障时端口内部的VC和通道资源即可能被报文所复用，进而提高资源复用概率和报文的最优转发概率。

2.3　死锁避免

转向模型是片上网络中广泛采用的一种规避网络死锁的方法［10］。然而，传统的转向模型的引入通常会引起网络的不公平现象。针对这一问题，研究人员在传统转向模型基础上提出了一种奇偶转向模型［9］，其规定网络中所有奇数列节点禁止E -> N和E-> S转向，偶数列节点禁止N-W和S-W转向。本文中即选用奇偶转向模型来避免网络的死锁，在具体实现上仅需将某些被禁止转向映射为I/O通道故障即可，如图4所示。

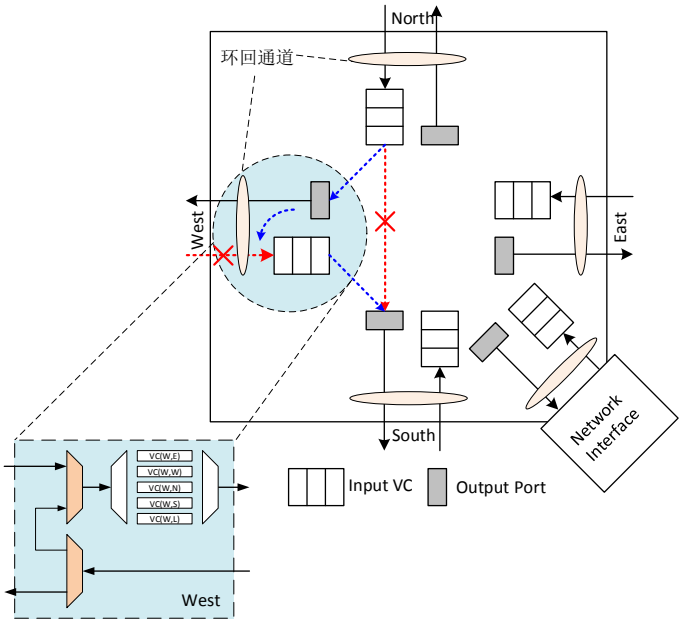
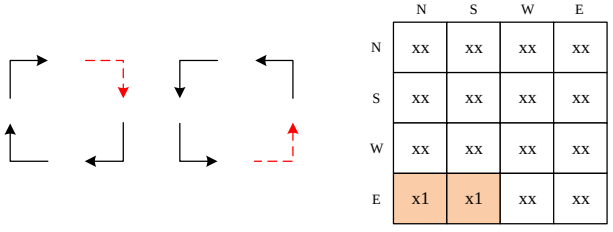
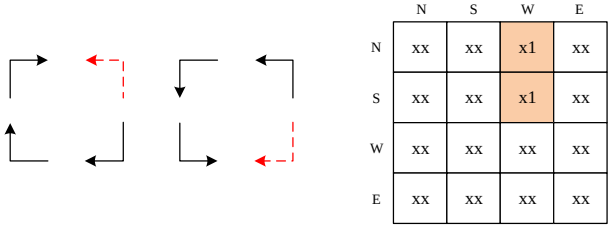


图3　故障端口环回传输机制



（a）奇数列转向模型



（b）偶数列转向模型

图4　奇偶转向模型与故障表示

2.4　算法实现

根据VOQ缓存特性，本文在实现上将采用一种超前路由机制。具体实现时，我们首先将转发端口定义成4个优先级Prio_x（x=1，2，3，4），其中Prio_1为优先级最高（最优转发端口），Prio_4为优先级最低，具体分配原则如下：（1）最短路径转发端口优先；（2）直向传输优先；（3）X方向优先于Y方向；（4）报文输入方向具有最低的优先级。执行路由计算时，每个中间节点都需要记录当前节点与其相邻节点的故障信息，每个节点需要200 bit缓存来存储当前和相邻4个节点的故障状态，其中Fault（X），X∈｛E，W，N，S｝分别代表与四个相邻节点的故障状态。详细选路过程如算法1所示。

算法1

FPLT路由算法

在FPLT容错机制下，当报文至Prio_x（1 ⩽ x ⩽ 4）级转发端口I/O通道故障时，依次对次优级端口执行路由选路，直至搜索完毕回传至上级节点。

3　实验结果与分析

3.1　实验平台搭建

实验平台搭建时，本文分别对FFBR、TT-BR、FPLT三种细粒度容错机制在基准VOQ路由器结构的基础上进行了硬件实现。具体网络结构和模拟配置如表1所示。网络模拟过程中，本文选用Uniform_random、Bit_reverse、Shuffle、Transpose四种合成负载和Barnes，Cholesky、FFT应用踪迹信息来作为负载注入。网络模拟时假设本地端口相关通道和缓存不存在故障，并在网络非边界节点中按照10%、20%和30%的概率随机注入VC、通道和链路故障。

3.2　人工合成负载

图5所示为人工合成负载下三种容错机制在

```
已知: Cur - Current Node Coordinate;
      Dest - Destination Node Coordinate;
      RC - Output direction for Current Node;
      Fault(X) - Fault Information of Output Port X (其中 X ∈ {E, W, N, S});
求:   Dir - Output Direction for Downstream Node;
      Loopback_port - Loopback Port for Packet Traversal;
      Loopback_flag - Loopback Flag for Packet Traversal;
 1: if (RC == E) then
 2:         downstream_inport ← W;
 3:         downstream_node ← (Cur.x + 1, Cur.y);
 4: else if (RC == W) then
 5:         downstream_inport ← E;
 6:         downstream_node ← (Cur.x − 1, Cur.y);
 7: else if (RC == N) then
 8:         downstream_inport ← S;
 9:         downstream_node ← (Cur.x, Cur.y + 1);
10: else if (RC == S) then
11:         downstream_inport ← W;
12:         downstream_node ← (Cur.x, Cur.y − 1);
13: else
14:         downstream_inport ← Empty;
15:         downstream_node ← Empty;
16: end if
17: if (downstream_node == Dest) then
18:         dir ← Local;
19:         Loopback_port ← Empty;
20:         Loopback_flag ← 0;
21:         return;
22: else
23:         for (x = 1; x <= 4; x + +) do
24:                 Opt_outport ← Port(prio_x);
25:                 if (Fault(RC)[Link(Opt_outport, 1)]) then
26:                         continue;
27:                 else
28:                         if (Fault(RC)[Path(downstream_inport, Opt_outport)]) then
29:                                 dir ← Opt_outport;
30:                                 Loopback_port ← Empty;
31:                                 Loopback_flag ← 0;
32:                                 return;
33:                         else
34:                                 for (each Fault(RC)[Link(i, 0)] = 1) do
35:                                         Fault(1) ← Fault(RC)[Path(downstream_inport, Link(i))];
36:                                         Fault(2) ← Fault(RC)[Path(Link(i), Opt_outport)];
37:                                         Fault(3) ← Fault(RC)[VC(downstream_inport, Link(i))];
38:                                         Fault(4) ← Fault(RC)[VC(Link(i), Opt_outport)];
39:                                         if (Fault(1) | Fault(2) | Fault(3) | Fault(4)) then
40:                                                 continue;
41:                                         else
42:                                                 dir ← Opt_outport;
43:                                                 Loopback_port ← Link(i);
44:                                                 Loopback_flag ← 1;
45:                                                 return;
46:                                         end if
47:                                 end for
48:                         end if
49:                 end if
50:         end for
51: end if
```

表1　网络结构和模拟配置

模拟参数	配置
拓扑结构	2D-MESH
网络规模	8×8
端口VC数目	5
虚通道深度	4-flit
Flit数据位宽	131bit
分配策略	基于优先级的轮询仲裁

不同故障率下报文从最优端口获得转发的概率（即最优转发概率）对比。网络模拟时，负载注入率设置为 0.02 flits/cycle/node。当网络无故障时，报文传输时均将沿着最短路径依次经历各中间节点直至到达目的节点，三种容错机制的最优转发概率均可达到100%。当网络故障率不断提高，报文从最优端口执行转发的概率均呈下降趋势，但

FPLT 机制在任意故障率下都要明显优于其他容错机制。当网络中存在故障时，FFBR 机制下报文最优转发端口对应的 I/O 通道故障时将会转发至其他次优端口甚至回退至上级节点；TTBR 机制采用“透传”实现了节点中部分废弃缓存与通道资源（因链路故障导致）的再利用，但其未对 VC 故障进行有效区分，且资源复用的条件比较严苛，将大大地降低端口缓存和通道资源的“再利用”概率，因此对于报文最优转发概率的提升将愈发有限；而 FPLT 机制下报文最优转发端口对应的 I/O 通道存在故障时，在对 VC 故障进行细分和弱化资源复用条件的基础上将具有更大的概率从其最优端口获得转发。如图所示，在 Uniform_random、Bit_reverse、Transpose、Shuffle 通信下，当故障率依次为 10%、20%、30% 时，FPLT 机制的报文最优转发概率较 FFBR、TTBR 容错机制可平均增加 7.6%、2.8%，12.3%、6.9% 和 15.5%、11.5%；

图 6 所示为人工合成负载下三容错机制在不同故障率下的网络吞吐率对比。当网络无故障时，报文传输时均将沿着最短路径获得成功转发，不同容错机制网络吞吐率将等于基准结构下吞吐率的大小。当网络故障率不断提高时，网络中大量存在的故障将使得三种容错机制的网络吞吐率明显降低，且容错粒度越大下降的幅度将更加明显。由图可知，由于对网络故障进行更好的细分和资源利用，FPLT 机制在任意故障率下的网络吞吐率都要优于 FFBR、TTBR 两种容错机制，且故障率越高提升的相对幅度将越为明显。在 Uniform_Random、Bit_Reverse、Transpose、Shuffle 通信下，FPLT 机制的网络吞吐率在故障率为 10%、20%、30% 时较 FFBR、TTBR 三种容错机制可平均增加 6.4%、3.4%，11.9%、7.3% 和 17.7%、12.1%。

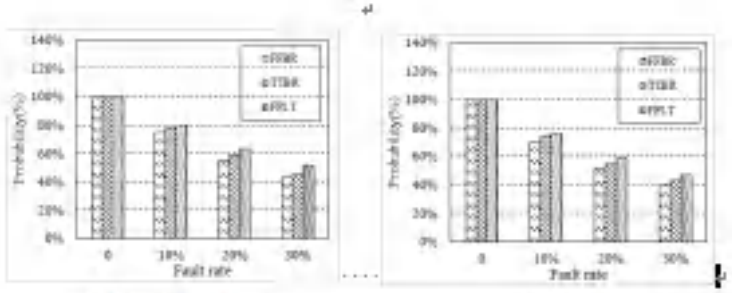
图 10 所示为应用踪迹负载下三种容错机制在不同故障率下的平均延迟对比。当网络无故障时，网络平均延迟仅与报文源节点与目的节点之间的平均距离相关，三种容错机制在 3 种应用通信中具有均等的平均延迟。当网络故障率不断提高时，不同容错机制下报文的平均延迟将快速增加，但由于 FPLT 机制的最优转发概率更高，具有相对较小的平均跳步数，因此其平均延迟的增长幅度要明显低于其他两种容错机制。如图所示，在 Barns、Cholesky、FFT 踪迹负载下，FPLT 机制下报文平均延迟在故障率为 10%、20%、30% 时较 FFBR、TTBR 两种容错机制可平均降低 5.4%、7.9%、13.4% 和 2.7%、4.9%、9.6%。

3.3 硬件实现开销

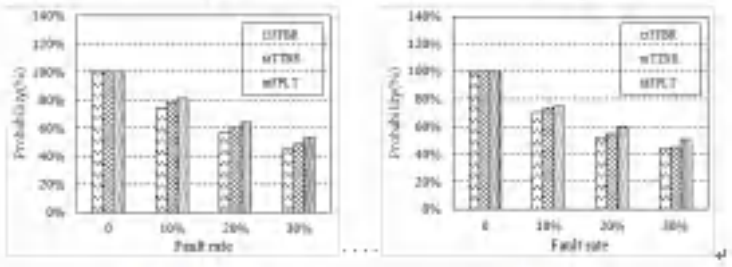
本文在 TSMC 65nm 工艺环境下使用 Synopsys Design Compiler 对 FFBR、TTBR、FPLT 三种容错路由器进行综合以评估其硬件开销。表 1 所示为不同路由器结构硬件开销对比。由于 FPLT 路由器在路由计算时需要关注更多的网络故障细节，因此相对其他容错结构其关键路径延迟均会有一定程度的增加。面积增大的原因在于 FPLT 路由器中需要更大的缓存空间来存储当前节点和相邻节点的故障信息，环回通道亦会带来一定的面积开销。

4 结束语

现有容错算法存在容错粒度大、故障映射粗犷、容错性能不高等问题，难以适应 VOQ 缓存结构下片上网络的高效容错需求。为此，本文首先构建了一种适用于 VOQ 缓存特性的网络故障模型，并以此为基础提出了一种端口环回传输机制，该机制通过利用输入链路故障端口中 VC 和 I/O 通道来提高报文的最优转发概率。实验结果表明 FPLT 机制能以较小的硬件开销实现网络容错性能的较大优化。

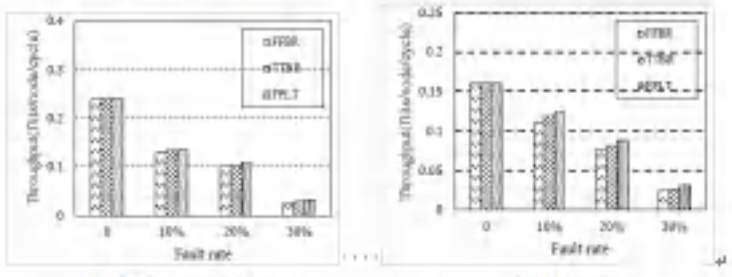


(a) Uniform_random (b)Bit_reverse

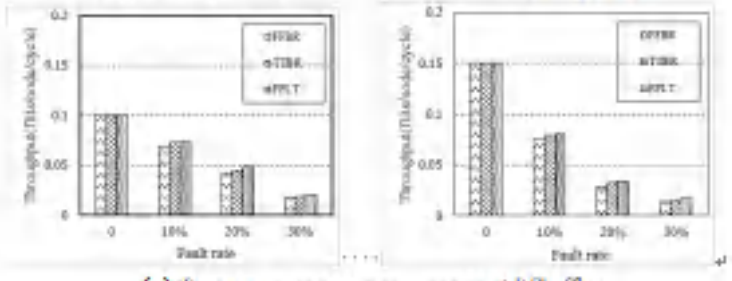


(c) Transpose (d)Shuffle

图 5 人工合成负载下不同容错机制的最优转发概率对比



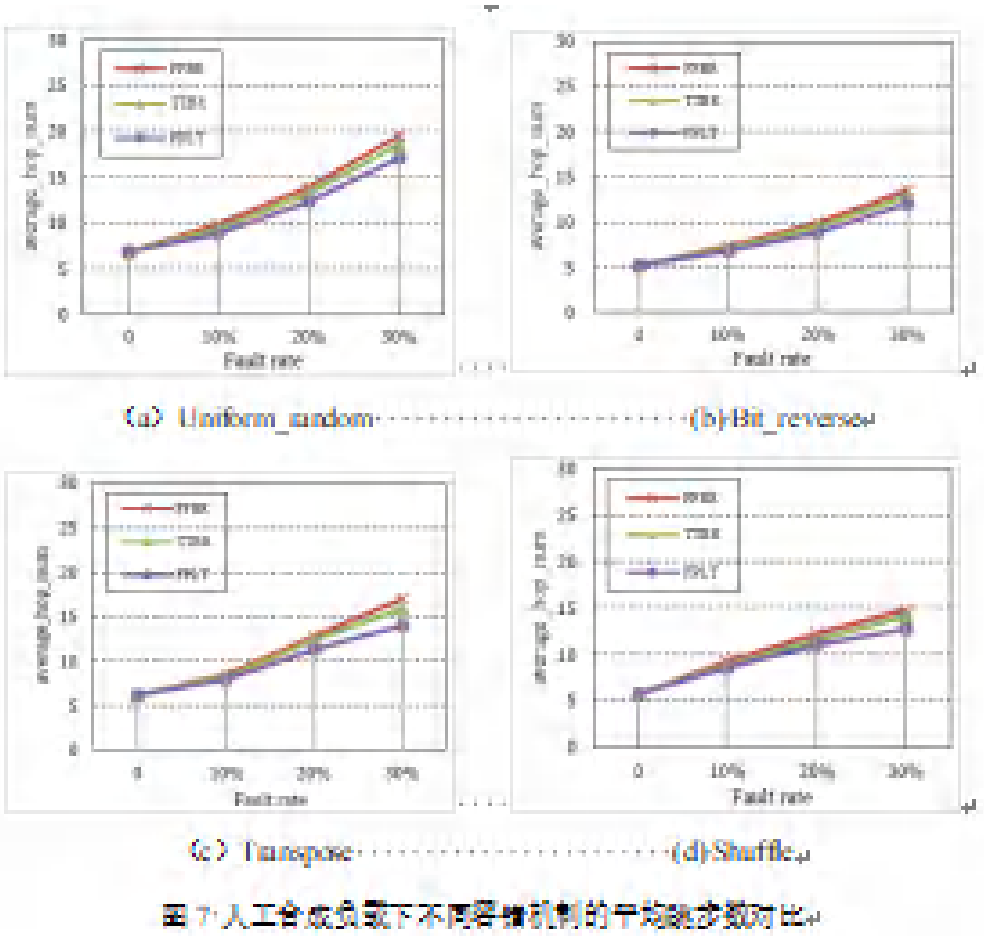
(a) Uniform_random (b)Bit_reverse



(c) Transpose (d)Shuffle

图 6 人工合成负载下不同容错机制的网络吞吐率对比

图 7　所示为人工合成负载下三种容错机制在不同故障率下的平均跳步数对比。当网络无故障时，三种容错机制下报文传输时均将沿着最短路径获得成功转发，其平均跳步数仅与报文源节点与目的节点之间的平均距离相关。当网络故障率不断提高时，不同容错机制下报文的跳步数都将快速增加，但由于FPLT机制下报文具有更大的概率转发至其最优（或次优）端口，可一定程度上减少从低优先级端口进行绕道路由甚至回退至上级节点网络传输的发生，因此其在不同故障率下的平均增长幅度都要明显低于其他两种容错机制。如图所示，在Uniform_Random、Bit_Reverse、Transpose、Shuffle通信下，FPLT机制的报文平均跳步数在故障率为 10%、20%、30% 时较FFBR、TTBR容错机制可平均减少7.4%、2.9%，10.7% 、7.4% 和14.5%、11.5%。一般来说，在相同网络注入率和故障率下，更低的平均跳步数即意味着更低的网络平均延迟。



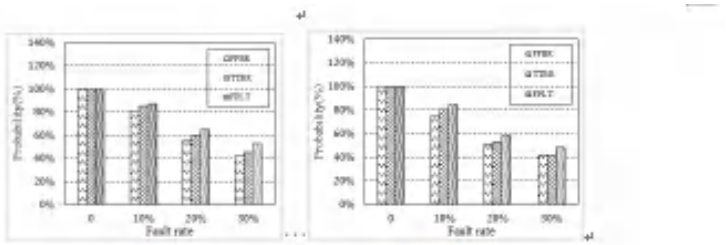
(a) Uniform_random (b)Bit_reverse

(c) Transpose (d)Shuffle

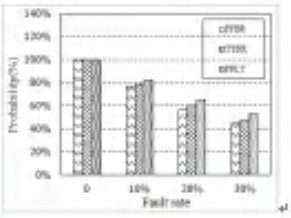
图 7 人工合成负载下不同容错机制的平均跳步数对比

图8　所示为在Barns、Cholesky与FFT应用踪迹负载下三种容错机制的报文最优转发概率对比。当网络无故障时，当网络无故障时，报文传输时均将沿着最短路径依次经历各中间节点直至到达目的节点，最优转发概率均可达到100%。当网络故障率不断提高时，由于FPLT机制相对于FFBR机制可通过端口环回传输的方式使得报文具有更大的概率转发至其最优（或次优）端口，减少报文在网络中进行绕道路由的次数，而FPLT机制相对于TTBR机制对于端口VC缓存故障进行了进一步细化，同时对端口资源“再利用”的条件进行了优化，因此FPLT机制下报文的最优转发概率在不同故障率下都要显著高于FFBR、TTBR容错机制。如图所示，在Barns、Cholesky、FFT踪迹负载下，FPLT容错机制的最优转发概率在故障率为10%、20%、30%时较FFBR、TTBR容错机制可平均提高7.9%、3.7%，13.3%、9.3%和16.8%、13.6%。

3.3 应用踪迹负载

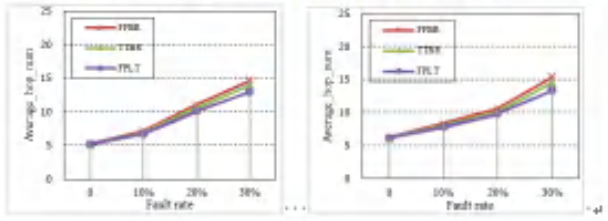


(a) Barns (b)Cholesky

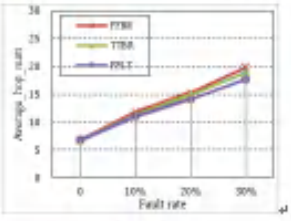


(c)FFT

图 8 应用踪迹负载下不同容错机制的最优转发概率对比



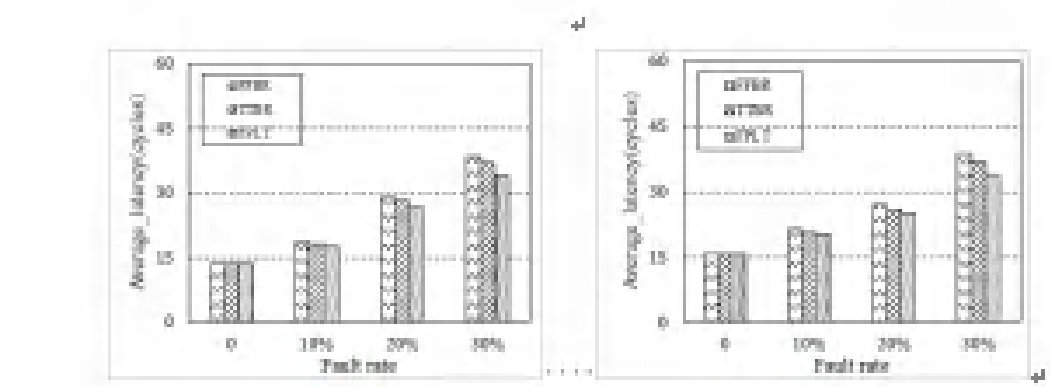
(a) Barns (b)Cholesky



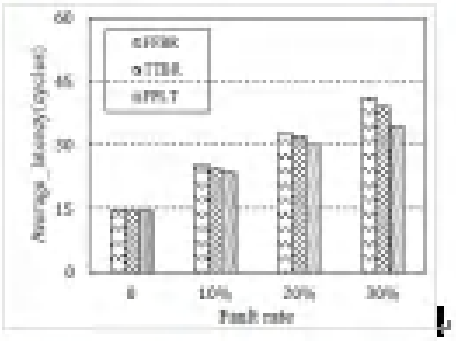
(c)FFT

图 9 应用踪迹负载下不同容错机制的平均跳步数对比

图 9　所示为应用踪迹负载下三种容错机制在不同故障率下的平均跳步数对比。当网络无故障时，三种容错机制下报文传输时均将沿着最短路径进行路由转发，其平均跳步数仅与报文源节点与目的节点之间的平均距离相关。当网络故障率不断提高时，不同容错机制下报文的跳步数均将快速增加，但由于FPLT机制下报文具有更大的概率转发至其最优（或次优）端口，可一定程度上减少从低优先级端口转发或回退至上级节点网络传输的发生，因此其在不同故障率下的平均增长幅度均要明显低于其他两种容错机制。如图所示，在Barns、Cholesky、FFT踪迹负载下，FPLT 机制的报文平均跳步数在故障率为 10%、20%、30% 时较 FFBR、TTBR 两种容错机制可平均减少 5.6%、2.9%，8.3% 、4.8% 以及 12.2%、7.4%。一般来说，在相同网络注入率和故障率下，更低的平均跳步数即意味着更低的网络平均延迟。



(a) Barnes　　(b)Cholesky



(c)FFT

图 10 应用程序负载下不同容错机制的平均延迟对比

表2　不同路由器结构硬件开销对比

Router	Critical delay(ns)	Area(μm²)
FFBR	1.57	330660
TTBR	1.63	337660
FPLT	1.71	347345

参考文献:

[1] Nguyen S T, Oyanagi S. A Low Cost Single-Cycle Router Based on Virtual Output Queuing for On-Chip Networks [C]. In The 13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools. 2010: 60 - 67.

[2] Boppana R V, Chalasani S. Fault-Tolerant Routing with Non-Adaptive Wormhole Algorithms in Mesh Networks [C]. In Proceedings of the ACM/IEEE Conference on Supercomputing. 1994: 693 - 702

[3] Flich J, Rodrigo S, Duato J. An Efficient Implementation of Distributed Routing Algorithms for NoCs [C]. In the 2nd ACM/IEEE International Symposium on Networks-on-Chip. 2008: 87 - 96

[4] 郑焱, 王红, 杨士元. 基于功能故障模型的 NoC 容错路由[J]. 计算机研究与发展. 2010, s1: 147-152
ZHENG Y, WANG H, YANG S Y. A Fault-tolerance routing algorithm of NoC based on functional fault model[J]. Journal of Computer Research and Development. 2010, s1: 147-152

[5] 陈庆强, 罗兴国, 张帆等. 基于故障节点再利用的细粒度NoC容错路由算法[J]. 计算机应用研究, 2012(07):192-194+223.
CHEN Q Q, LUO X G, ZHANG F. Fine grained fault tolerance routing algorithm of NoC based on reuse of partly defective switches[J]. Application research of computers, 2012(07):192-194+223.

[6] 张士鉴, 韩国栋, 沈剑良. 基于故障链路缓存再利用的NoC容错路由算法[J]. 计算机辅助设计与图形学学报. 2014, 26 (1): 131-137
ZHANG S J, HANG G D, SHEN J L. Fault-tolerant routing algorithm of NoC based on bufffer reuse of faulty links[J]. Journal of computer-aided design & computer graphics. 2014, 26 (1): 131-137

[7] Jun S, Wang L T, et al. Review on Fault-Tolerant NoC Designs[J]. Journal of Electronic Science & Technology. 2018, 016 (003): 191-221.

[8] Werner S, Navaridas J, Lujan M. A Survey on Design Approaches to Circumvent Permanent Faults in Networks-on-Chip [J]. ACM Computing Surveys. 2016, 48 (4): 59:1 - 59:36

[9] Hans-Joachim W, Martin R. Multi-Layer Test and Diagnosis for Dependable NoCs [C]. In Proceedings of the 9th International Symposium on Networks-on-Chip. New York, NY, USA, 2015: 5: 1 - 5:8

[10] Janfaza V, Baharlouei E. A new fault-tolerant deadlock-free fully adaptive routing in NOC [C]// IEEE East-west Design & Test Symposium. IEEE, 2017:1-6.

[作者简介]

汤先拓（1985—），男，博士，助理研究员，主要研究方向

为新型网络体系结构，高效能NoC设计

刘冬培（1985-），男，博士，助理研究员，主要研究方向为软件定义互连、SoC芯片设计

宋 克（1976-），男，博士，副研究员，主要研究方向为网络空间拟态防御、片上系统体系结构设

李丹丹（1992-），女，工学学士，助理工程师，主要研究方向为拟态调度及判决

基于本体模型的容器云高生存力保障机制研究

罗论涵[1]，解维[1]，余新胜[1]，徐李定[1]，应飞[2]

[1]中国电子科技集团公司第三十二研究；

[2]同济大学电子与信息工程学院，上海 201808

摘　要：针对云环境下重要信息系统应用安全防护需求，本文引进信息系统生存力理论，结合本体化建模方法，研究了容器云威胁识别、威胁抵抗、功能恢复技术，形成了一套高生存力容器云框架。通过实验仿真证明，基于本框架构建的容器云具有高生存力，容器应用服务漏洞和后门攻击发现准确率和拦截效率比达到99.9%，异常容器重构时间小于10s，说明容器云即使面对恶意攻击时也能够屏蔽威胁，不影响容器云上的应用服务正常运行。

关键词：高生存力、容器云、本体化建模、高生存力控制单元

Research on high survivability defense mechanismof container cloud platform based on ontology

Luo Lunhan[1], Xie Wei[1], Yu Xinsheng[1], Xu Liding[1], Ying Fei[2]

1.The 32nd Research Institute of China Electronics Technology Group Corporation, Shanghai 201808, China;

2.School of Electronic and Information Engineering, Tongji University,Shanghai 201804, China

Abstract: Aiming at the security protection for important information system applications in cloud environment, this paper constructed a high viability container cloud framework, introduced the information system survivability theory, and studied the threats recognition, functional recovery, and adaptive technology based on ontology modeling. The experimental simulation proved that the container cloud do have high viability, presenting by the accuracy and efficiency of the container application service bugs and back door intercepting ratio reached 99. 9%, the abnormal containers reconstruction time was less than 10 s, which means that our container cloud can stay robust and effective even in the face of malicious attacks.

Key words: High survivability; Container cloud; Ontology modeling; High survivability control unit

1　引言

容器是一种轻量、灵活的虚拟化资源编排方式，它实现了应用运行环境的标准化构造，支持计算、存储、网络资源按需弹性伸缩，具备可移植、低开销、弹性伸缩等特点，目前已经成为云应用部署最高效的手段。

容器云的安全性与可靠性直接影响云应用能否稳定运行，由于虚拟化平台使用容器技术将宿主机物理资源池化，如果宿主机遭受了攻击，其上的所有虚拟容器均暴露在攻击范围之内，从而造成应用服务的异常或存储信息的丢失，甚至威胁应用数据安全。正是由于虚拟容器和宿主机的依赖关系，使得网络攻击者能够通过对虚拟容器，甚至是运行在其上的应用服务进行扫描和探测获取其文件系统、操作系统乃至内核的详细信息，进而使宿主机的操作系统等信息被暴露在攻击者的视野之内。一旦宿主机的基本信息被探测到，则物理服务器上的所有虚拟化容器面临被一网打尽的窘境。考虑到虚拟化容器具有占用空间小，启动速度快等特点，一般虚拟化平台上都会运行着大量负载着各种应用服务，开放着大量端口的

基金项目：上海市科学技术委员会项目"拟态容器安全云平台研究"(18511104402)

虚拟化容器，一旦其中个别容器上出现了后门或漏洞，导致被攻击者探测到并加以利用，很有可能在极短的时间内就能够使所有的虚拟化容器处于瘫痪的状态。

目前通过增加防火墙策略，添加审计模块，更新病毒库等手段，虽然能够在一定程度上减少攻击者通过虚拟化容器切入攻击的几率，然而在操作系统以及应用服务中，均或多或少的存在个别未知的后门和漏洞，倘若传统的安全防护手段未能做到面面俱到，那么被攻破的几率仍旧存在，尤其是在虚拟化平台大面积部署和使用的环境下，这种风险将会以几何级数增长，进而使整个虚拟化平台连同上面运行的各种信息系统及服务都处在了濒临崩溃的边缘，因此需要找到一种切实可行的方法从根本上提升虚拟化容器的可靠性。

本文引进信息系统生存力理论，结合本体化建模方法，构建了高生存力容器云框架，通过网络、应用和数据的分层，将系统结构划分为系统单元与单元之间的关系组成，并通过单元之间的关系来实现信息单元传递。基于对容器云本体化模型的分析，本文形成了以改造容器调度管理单元保障容器云结构层与功能层安全的思路，在传统容器云架构上增加高生存力控制逻辑单元，提升容器云威胁识别抵抗、功能恢复重构能力，从而弥补系统的安全缺陷，有效提高了重要信息系统面对恶意攻击时的生存力。

2　国内外研究现状

国内外学者对云计算安全从结构、策略、加密等形式进行了研究。SawantS.等［1］引入高效的第三方审核员（TPA）保障云计算中的数据可靠性。He等［2］构建了云计算的安全体系结构，针对云计算面临的安全威胁，从云计算用户和服务提供商的角度提出了相应的策略。RonaldPetrlic［3］将软件供应商和用户对云计算安全、隐私保护的DRM系统的需求结合起来。使用代理重新加密云数据安全。冯登国等［4］分析了云计算对信息安全领域中技术、标准、监管等各方面带来的挑战。熊金波等［5］提出一种基于角色对称加密的云数据安全去重方案，实现分层结构下的云数据授权去重。郝斐等［6］设计并实现了一款云存储安全增强系统，对用户上传至云存储的数据进行加密保护。

Alireza Sadighia等［7］提出一个基于上下文环境信息的本体警报关联框架，该框架使用本体来表达和存储警报信息，并采用本体逻辑规则来关联和过滤不相关警报，降低了误报率。Igor Kotenko等［8］提出一种基于安全指标本体的分析方法，该安全指标本体包括6个组成部分：拓扑指标、犯罪指标、攻击指标、系统指标、代价指标和零日漏洞攻击指标。张连华等［9］构建了网络脆弱性本体的核心模型。肖云等［10］基于知识库系统的入侵从顶级本体、领域本体、任务与行为本体、应用本体等四个层次建立了基于本体的网态知识库模型。

在生存力方面，Robert等［11］对可生存性分析方法作了研究，提出了一种定性的分析方法SNA（survivable network analysis），对系统的3R（resistant，recognize，recover）特性进行层次分析，对可生存性评估具有很好的参考价值。Irving Vitra Paputungan等［12］进行的生存能力相关的恢复过程建模。Andrew P. Moore等［13］进行的信息安全与生存性的攻击建模。刘密霞等［14］用模糊Petri网构建了攻击失效模型，推理系统遭受到攻击后到达不同状态的概率，进而计算出系统在受到攻击后的可生存性。林雪纲等［15］通过层次化的方式从可抵抗性、可识别性和可恢复性3个方面进行了量化计算。

目前研究尚未基于本体进行容器云建模并开展生存力研究，本文将分析容器云关键组件与运行关系，结合本体化建模方法，形成高生存力容器云框架，经过攻防验证分析容器云生存力。

3　高生存力容器云构建技术研究

3.1　高生存力容器云基础架构

容器云是以容器为资源分割和调度的基本单位，封装整个软件运行时环境，为开发者和系统管理员提供用于构建、发布和运行分布式应用的平台，能够为微服务架构的应用服务提供灵活便捷的部署和运行环境。信息系统生存力基础理论，是对信息系统自身如何健康、长久存在能力的简称。生存力构建技术，涉及系统全生命周期上的生存力形成模型，包括网络、软件、数据的生存力组件；生存力监控流程，涉及系统在运行过程

中对于生存力的维持，生存力提升的流程。本文基于信息系统生存力理论，构建高生存力容器云架构，以保障容器云平台上应用服务的鲁棒性。

高生存力容器云具备以下功能层：

图1高生存力容器云架构

1） 容器资源层：构建基础设施资源池，为虚拟化容器提供计算、存储和网络资源，经过容器层的封装，形成可供信息系统微服务部署和运行的容器应用、容器存储和容器网络资源。

2） 容器集群层：基于CPU、内存、亲和性等多维度的容器资源自动和手动调度，通过服务发现、负载均衡等微服务架构辅助功能保障系统的弹性伸缩。

3） 高生存力控制逻辑单元：实现容器云平台高生存力调度，构建微服务-容器资源、容器资源-基础设施环境的对应关系，在识别到应用服务、容器资源异常或失效等威胁时，用于指导容器云平台重新调度部署。

4） 支撑管理层：支持多类型容器集群的管理，提供镜像仓库实现容器镜像的上传、下载和管理，通过持续集成工具实现基于容器镜像的打包和集成服务，通过编排工具实现多种类型的应用单元的组合编排和部署服务，通过监控实现容器的健康检查、性能监测等功能。

3.2 容器云本体化建模

在计算机领域，本体可以在语义层次上描述知识，可以看成描述某个学科领域知识的一个通用概念模型。通常本体包含5个基本的建模元语，这些元语分别为：类、关系、函数、公理和实例，根据信息系统的具体情况，本体化建模可以使用部分或全部元语。

本文面向虚拟化容器高生存力构建需求，构建容器本体化模型：以K8S为例，将容器组件根据功能划分为管理类、调度类和服务类，管理类组件由集群控制中心和高生存力控制中心组成，负责集群运行控制以及构建容器高生存力特性；调度类组件由负载调度器和负载代理组成，承担pod到节点，service到pod的路由功能；服务类组件提供计算、存储、网络资源，构成容器服务。在实际的容器结构中，管理类、调度类、服务类组件可以进一步实例化，如下图所示：

高生存力控制中心为容器云生存力保障的关键组件，由容器调度管理器与威胁识别器组成。容器调度管理器监听apiserver中的组件运行状态，并维护kube-proxy中的Pod调度规则，威胁识别器监听应用在Pod中的运行状态，发现异常Pod则进行下线，并通知kubelete新建Pod。

以云容器类与实例的组成关系为基础，本文对容器应用中实例之间的运行关系进行建模，归纳了管理、监听、维护、调度、通知、报告、读写、运行于等典型实例关系。其中，管理是指功能或者资源、运行配置的关系；监听是指状态监控关系；维护是指实例状态管理操作；调度是资源选择操作；通知是指满足预设条件时的通信关系；读写是指对数据库或者文件的读写操作；运行于是指一个实例为另一个实例的载体。容器云实例关系如下图所示：

分析K8S中关键组件功能及运行关系，包括kube-controller manager，apiserver，kubeschedul-er，service，kube-proxy，pod，kublete，kube-con-troller manager，node，ectd等。

kube-controller manager：集群内部的管理控制中心，负责node，pod副本，服务端点（end-

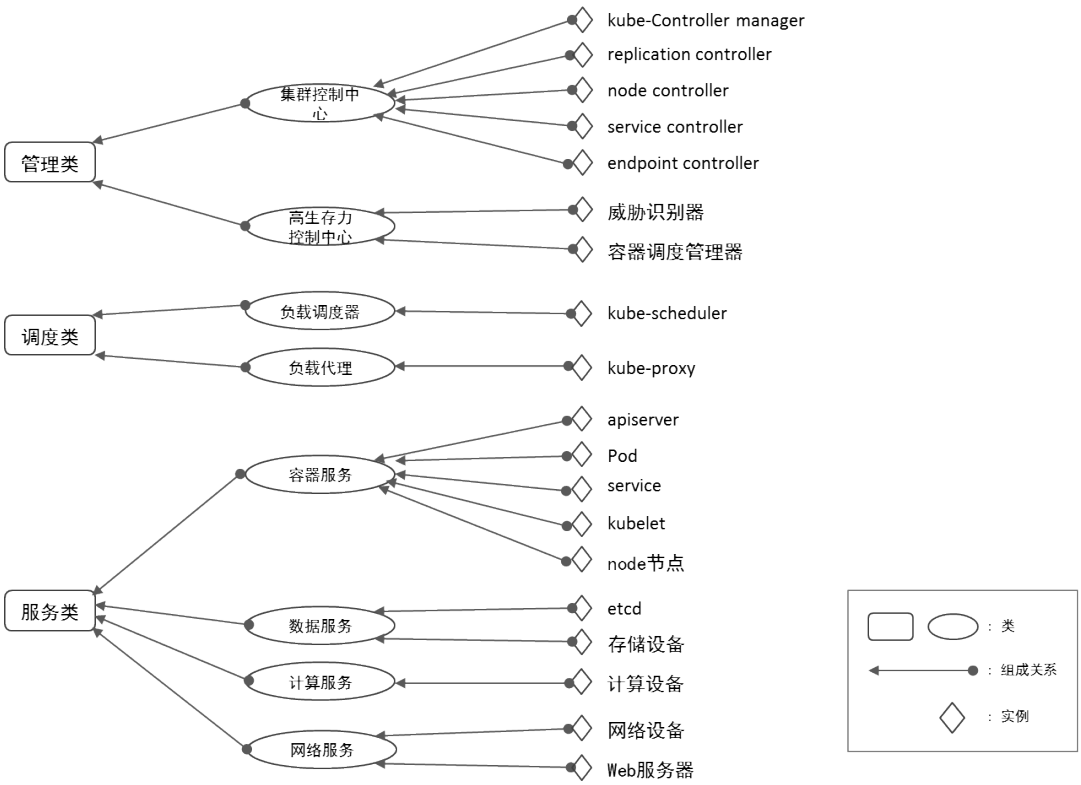


图2云容器类与实例组成关系

point）、命名空间、服务账号、资源定额的管理，当某个node停止工作时，kube-controller manager执行自动化修复流程，确保集群始终处于预期的工作状态。

node controller： 通过apiserver实时获取node的相关信息，管理和监控集群中的各个Node节点的相关控制功能。

replication controller： 保证集群中关联的Pod副本数保持预设值。

endpoint controller： endpoints表示了一个Service对应的所有Pod副本的访问地址， endpoints controller负责生成和维护所有endpoints对象的控制器，监听Service和对应的Pod副本的变化。

service controller： 是kubernetes集群与云平台之间的一个接口控制器。Service Controller监听Service变化，确保云平台相应的创建、删除Service对应实例及更新路由转发表。

apiserver：提供K8S中pod， Service等资源对象的增删改查，是K8S的数据总线和数据中心。

service：提供负载均衡和服务自动发现。

kube-proxy：为Pod创建代理服务，从apiserver获取所有server信息，并根据server信息创建代理服务，路由和转发server到Pod的请求。

pod：是K8S管理的最小单位级，它是一个或多个容器的组合。在pod中，所有容器都被统一安排和调度，并运行在共享的上下文中。Pod是具体应用的逻辑主机，包含业务相关的多个应用容器。

kubelet：在启动时会通过apiserver注册自身的节点信息，并定时向apiserver汇报状态信息，apiserver接收到信息后将信息更新到etcd中。

etcd： 用于共享配置和服务发现的分布式、一致性KeyValue型存储系统。

endpoint： service关联pod的ip地址和端口。

kube-sheduler： 决定集群调度的容器。

node：物理机或虚拟机资源。

K8S中关键组件运行关系如下图所示：

基于实例之间的关系类型开展容器云安全性分析，如表格所示。

通过容器云本体化建模分析可知，提高容器云安全性，构建高生存力需重点设计容器调度管理器与威胁识别器，使容器云具备在安全威胁情况下也能保障功能完整的能力。本文构建的容器调度管理器与威胁识别器基于冗余副本思想，利用容器云可快速构建、部署、下线容器机制，为

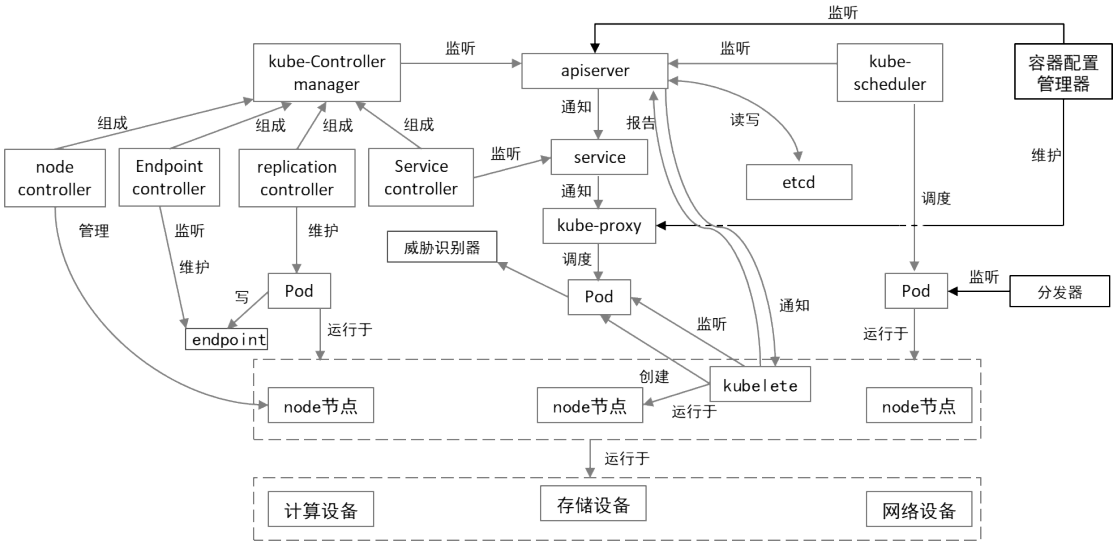


图3 容器云实例关系图

表格 1　容器云实例关系与安全性分析

关系类型	关系说明	安全性影响
实例A运行于实例B	实例B为实例A的运行基础	实例B安全运行状态依赖于实例A安全运行状态
实例A监听实例B	实例A获悉实例B安全运行状态	可配合威胁识别器发起容器云威胁识别、威胁抵抗、功能恢复,恢复容器云运行状态,提升安全性
实例A维护实例B	实例A根据实际情况修改实例B运行状态	无安全性影响
实例A通知实例B	实例A通知实例B进行容器云操作	无安全性影响
实例A读写实例B	实例A对实例B进行数据库读写	无安全性影响
实例A调度实例B	实例A选择实例B为服务Pod,根据威胁识别轮换服务Pod	可配合监听、威胁识别器提升系统安全性
实例A管理实例B	实例A为实例B提供资源、运行配置	无安全性影响
实例A组成实例B	实例A为实例B的组成部分	实例B安全运行状态依赖于实例A安全运行状态

应用构建多运行容器副本，通过资源监控、威胁识别器监控容器副本运行状态，当监测到容器云中存在Pod未正确响应，威胁识别器执行应用服务的重组、重构与重建，将副本Pod作为服务容器，并对异常Pod下线清洗，或通过容器编排依据功能迁移列表将部分功能迁移至其他容器中，实现高生存力容器云识别力、抵抗力、恢复力闭环。

3.3　容器云威胁识别技术

3.3.1　基于资源监控的识别

通过对云平台的整体架构中的计算、存储、网络等基础设施资源进行持续监控，评估系统内承担各功能的容器数量及其工作状态，根据构成系统的各容器及网络连接状况，识别容器云及其上应用服务的威胁。

对结构上的监控由各个工作节点上的协调器负责。为了更好地进行容器应用的生命周期及物理资源池的资源管理，它将收集并汇总任务的运行时性能和消耗的实时资源数据。

协调器收集的应用运行时数据主要包括两个方面：

1）操作系统通过procfs接口暴露的统计信息，包括进程消耗的RAM和CPU，运行时间等。

2）更底层的与CPU硬件性能相关的计数器，包括缓存未命中，由于DRAM访问或I/O造成的停顿，CPI等。

子协调器还将把收集到的运行数据共享给父协调器，这使得高生存力容器云能够在获取单个任务的运行报告的基础上，以物理节点或虚拟机节点为单位进行汇总，甚至分析同一机器上运行的不同任务之间的影响，从而达到对容器云威胁识别的目的。

3.3.2　基于威胁识别器的识别

在容器云平台环境中，信息系统应用服务通常以封装在一个或多个pod中的微服务形式进行部

署。威胁识别器实时监测容器云平台及其上应用服务在运行过程中的潜在问题，包括：因为存储资源问题所导致的始终处于“Container Creating”状态的pod；因为容器CNI网络通信故障所导致的始终处于“Terminating”状态的pod；因为微服务自身或其依赖关系异常所导致的频繁重启的pod。

威胁识别器对长时间（如10分钟）不能达到“Ready”状态的pod，以及在一定时间内频繁重启（如1小时内重启5次以上）的pod发出告警信息。威胁识别器在获取告警信息的基础上，识别出功能异常的应用服务和容器资源，并通知容器云平台执行容器清洗或删除，同时通过容器管理系统中的node controller和replication controller记录容器状态的变化，以执行后续容器应用、副本或者节点的重启与恢复过程。

3.4 容器云威胁抵抗技术

高生存力容器云以灵活多变的配置结构以及调度算法间不确定性映射关系为核心，以容器多副本、动态轮换机制提升防御场景为恢复方法，隐匿了恶意攻击对系统的干扰，达到保障容器及其上应用服务健壮、安全的目的。高生存力容器云能凭借其对自身架构的重构，提升容器及其上应用服务的高生存力，确保基本功能完整，系统信息不被篡改和破坏，既能对明确特征的威胁实施精确防范，又能对不确定性威胁主动抵抗。

3.5 容器云功能恢复技术

集群控制中心的node controller和replication controller作为编排调度模块实现容器云功能恢复，集群控制中心通过配置文件指定一个容器本体单元有多少个副本在集群中运行，并且保证运些副本的期望状态与当前状态一致。如果当前宿主机节点异常或者容器本体单元运行终止，编排调度模块就会进行相应的容器本体单元重调度。功能资源管控组件持续跟踪高生存力容器云中的功能实现矩阵和容器本体单元的对应信息，同时收集所有容器本体单元的资源负载情况。发现与功能实现矩阵不相匹配的容器本体单元，会根据功能转移矩阵的信息将其分发到其他可供转移功能的容器本体单元中，同时根据容器本体单元的定义分发系统资源。通过集群控制中心的编排调度和资源调配，支持容器云在副本侧、容器侧、节点侧发生异常时按需恢复或重建。

3.6 实验与仿真

通过模拟各层次遭受攻击的环境，对高生存力容器云进行黑盒测试。将容器云接入模拟攻击环境，使用嗅探工具对容器应用服务进行漏洞扫描，并在此基础上实施攻击，以检验高生存力容器云对APT攻击的防御效果。进一步在虚拟化平台上的应用服务中注入预先设计的后门和漏洞，进行注入测试，以验证虚拟化平台的对未知漏洞的主动防御效果。最终进入虚拟化平台内部，检测容器快速重构的时间间隔是否符合预期。通过模拟硬件故障和网络故障，来测试应用服务是否能在硬件故障的情况下依旧保持高生存力，测试原理如图4所示。

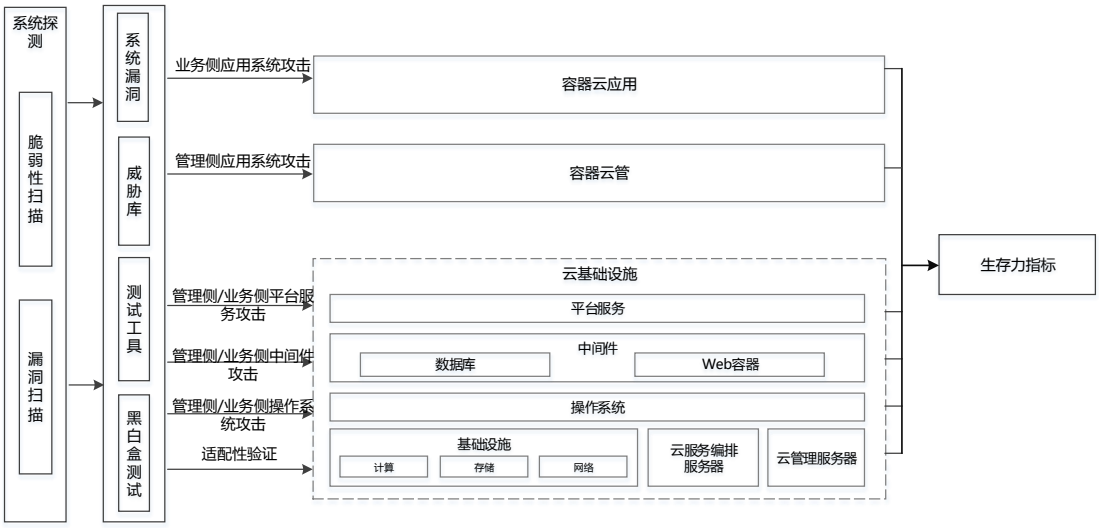


图4 容器云生存力验证原理框图

开展具备高生存力控制中心的容器云与原生容器云生存力验证。使用攻击工具，开展人工渗透测试进行系统威胁扫描与漏洞分析，通过构造漏洞验证系统安全防御能力，包括威胁识别力和抵抗力。利用漏洞、后门、病毒等手段针对容器云CPU，操作系统，数据库，web服务器、前端系统等不同系统层次攻击时的防御能力。围绕贴近系统资源和用户的管理侧和业务侧，设计针对云应用部署、数据访问、运行监控、前端页面展示等关键云功能的攻防验证方法与测试用例，在不采用外挂式安全防护手段时，原生容器云被利用漏洞进行攻击时，无法正常运行，系统不具备威胁识别力、抵抗力与恢复力。具备高生存力控制中心的容器云生存力指标如表格2所示。

表格2　具备高生存力控制中心的容器云生存力指标

威胁类别	一级指标	二级指标	指标数值	描述
利用漏洞的网络攻击	识别力	目标发现准确率	99.9%	识别漏洞数/总确定漏洞数
	抵抗力	拦截效率比	99.9%	成功拦截次数/总拦截次数
	恢复力	恢复功能比	100%	恢复功能数/失效功能数
		单个容器恢复平均耗时	<10s	恢复耗时

表格1为经过高生存力改造后的容器云生存力指标模型生存力指标，通过数据分析可得，高生存力容器云针对容器应用服务漏洞和后门攻击有主动防御效果，目标发现准确率和拦截效率比达到99.9%，说明高生存力容器云即使遭遇漏洞和后门攻击，也能够根据系统的威胁识别力和抵抗力屏蔽威胁影响，不影响运行在容器云上的服务。高生存力容器云具备对各层次攻击下维持信息系统生存力的能力，包括对基础设施环境资源的重组，容器运行环境的重建，以及应用服务的重构，并且其切换时间<10s。

4　总结与展望

本文形成了一套基于本体分析的高生存力容器云框架，通过构建容器云本体化模型确立了围绕高生存力容器控制逻辑单元的容器生存力加固技术，研究了容器云威胁识别、威胁抵抗、功能恢复技术：容器云威胁识别和威胁抵抗技术以状态监听为基础，以多容器副本机制达到内生威胁识别和抵抗能力，容器云功能恢复技术基于高生存力容器控制逻辑单元协同容器云管理调度单元，实现容器云在副本侧、容器侧、节点侧发生异常时按需恢复或重建，最终形成识别力、抵抗力、恢复力闭环。实验仿真环节验证了依据本文框架构建的容器云具有高生存力。

5 参考文献:

[1] Sawant S S , Soujanya S , Babu G C . SAFETY SUPERVISION OF FOCUSSING VITAL, AREAS IN CLOUD COMPUTING [J]. International Journal of Computer & Electronics Research, 2013, 2(4).

[2] He Z , He Y . Analysis on the security of cloud computing [C]// PIAGENG 2010: Photonics and Imaging for Agricultural Engineering. International Society for Optics and Photonics, 2011.

[3] Petrlic R . Proxy Re-Encryption in a Privacy-Preserving Cloud Computing DRM Scheme [C]// The 4th International Symposium on Cyberspace Safety and Security (CSS 2012). Springer-Verlag, 2012.

[4] 冯登国, 张敏, 张妍,等. 云计算安全研究[J]. 软件学报，2011，22(1):71-83.

[5] 熊金波, 张媛媛, 田有亮, et al. 基于角色对称加密的云数据安全去重[J]. 通信学报，2018，v. 39;No. 371(05):63-77.

[6] 郝斐, 王雷, 荆继武, 等. 云存储安全增强系统的设计与实现[J]. 信息网络安全，2012(3):38-41.

[7] Sadighian A, Fernandez J M, Lemay A, et al. ONTIDS: A highly flexible context-aware and ontology-based alert correlation frame work[C]//Foundation & Practice of Security. Switzerland : Springer-Verlag，2014:161-177.

[8] Kotenko I, Saenko I, Polubelova O, et al. The ontology of metrics for security evaluation and decision support in SIEM systems [C]// International Conference on Availability，Reliability and Security. Regensburg: IEEE，2013:638-645.

[9] 张连华,张洁,白英彩. 基于ontology 的安全漏洞分析模型[J]. 计算机应用与软件，2006,23(5) : 121-123.

[10] 肖云,王选宏. 基于网络安全知识库的入侵检测模型[J]. 计算机应用研究,2009,26(3) : 1079-1081.

[11] ROBERT J,ELLISO N,C et alRICHARD, Survivable network system analysis:a case smdy[J]. IEEE Software,1999,16(4):70-77.

[12] Irving Vitra Paputungan, AbdullahAzween. Survivability assessment: modeling a recovery process[C], Seminar nasional aplikasi teknologi informasi, 2007

[13] Moore A P, Ellison R J, Linger R C. Attack modeling for information security and survivability [OL], Technical note CMU/SEI-2001-TN-001, 2001.

[14] 刘密霞,张玉清,洪毅. 基于模糊推理的网络可生存性的建模与分析[J]通信学报,2009,30(1):31-37.
[15] 林雪纲, 许榕生, 熊华, et al. 一种信息系统生存性的量化分析框架[J]. 电子与信息学报, 2006,28(9).

[作者简介]

罗论涵（1985-），女，博士，中级工程师，主要研究方向：网络安全防御，信息安全。

解维（1986-），女，硕士，中级工程师，主要研究方向：网络安全防御，信息系统架构。

余新胜（1979-），男，硕士，高级工程师，主要研究方向：网络安全防御，信息安全，信息系统架构。

徐李定（1993-），男，学士，工程师，主要研究方向：网络安全防御，云计算。

应飞（1987-），男，硕士，高级工程师，主要研究方向：网络安全防御，云计算。

基于相对时间的异构执行体程序状态同步方法

苏野[1]，魏帅[2]，姚领彦[1]，谭立波[1)]

[1]天津市滨海新区信息技术创新中心，天津 300450；

[2]信息工程大学信息技术研究所，河南 郑州 450002

摘　要：近年来，基于内生安全理念的拟态防御技术取得了长足进展，在一系列测试和比赛中显示了较强的防御能力。拟态防御系统采用异构执行体执行相同任务，并对结果进行比较，但是不同执行体受调度等因素影响会造成输出结果顺序不同甚至完全不一致，如路由删除问题等，因此异构执行体中程序状态的同步是拟态防御技术需要解决的关键问题，如果采用分布式同步算法需要至少3M+1个执行体和大量的同步消息，并且对现有程序需要进行大量改动，基于此，本文提出基于相对时间的异构执行体程序状态同步方法，基于拟态防御系统异构执行体不协同的原则，通信代价较小，具有较好的扩展性，且对程序改动较小，只需在主循环中进行少许改动。理论分析和实验结果证明，基于相对时间的异构执行体程序同步方法能够有效的解决多个异构执行体程序状态同步问题。

关键词：拟态防御、异构、同步、相对时间

Program state synchronization method of heterogeneous executors based on relative time

Su Ye[1], Wei Shuai[2], Yao Lingyan[1], Tan Libo[1)]

1.Information Technology Innovation Center of Tianjin Binhai New Area, Tianjin 300450, China;

2.Institute of Information Technology, Information Engineering University, Zhengzhou 450002, China

Abstract: The mimic defense technology based on dynamic heterogeneous redundancy plays an important role in network security research. The prerequisite for a successful mimic judgment is that the synchronization state of the programs running is same in the heterogeneous executors. The existing synchronization methods cannot effectively solve such problems. Based on this, this paper proposes a program state synchronization method of heterogeneous executors based on relative time. By synchronizing the relative time of program execution events in heterogeneous executors, the consistency of program execution status is realized. Theoretical analysis and experimental results prove that the program state synchronization method of heterogeneous executors based on relative time can effectively solve the problem of program state synchronization of multiple heterogeneous executors.

Key words: mimic defense; heterogeneous; synchronization; relative time

1　引言

如何根据有毒带菌的器件构建安全可靠的系统，邬江兴等提出了拟态防御的思想，采用动态异构冗余的思路打破漏洞，利用通常依赖的相似确定静态执行环境，具体模型如图1所示，外界输入通过输入代理分发给异构执行体，异构执行体执行相同的任务，并将结果发送给调度器，调度器采用大数裁决等策略对执行结果进行裁决，最终产生出一个认为是正确的输出结果[1~3]，调度器可以根据各执行体的表现进行相应的反馈控制，如某个执行体产生错误，则对其进行清洗，待其清洗完成之后再将其加入工作队列，裁决相关参数和系统运行状态信息可通过反馈控制器进行控制和查看。

虽然拟态系统可以有效防御已知或者未知漏

基金项目：国家核高基重大专项基金资助项目（No.2017ZX01030301）。

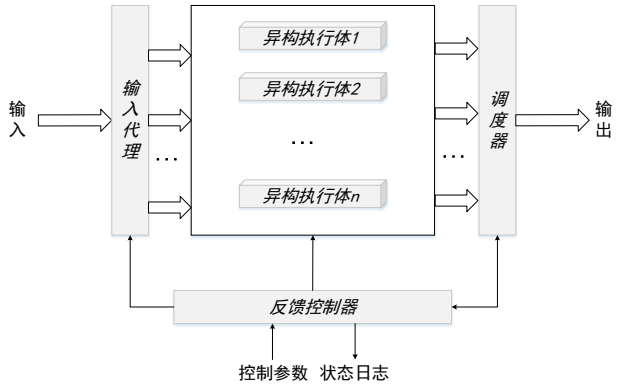


图1　拟态防御架构图

洞/后门的攻击[4-5]，但是异构执行体通常是处理器上配置操作系统，多程序并行执行，异构系统中进行调度和切换会造成输入执行的不确定性，最终会使系统产生不一样的输出，虽然这些输出对于单个执行体都是正常的，但是会造成拟态防御系统[6]中调度器难以进行有效判决，尤其是调度器出于安全因素采用硬件实现，逻辑功能较为简单的情况下。以单执行体中某个进程为例，如图2所示。进程存在多个状态，状态之间的变迁由读事件、定时器事件及其他事件引起，其他事件也由读事件和定时器事件引起，只要控制住读事件和定时器事件顺序即可保证状态机一致，而定时器事件的添加和超时基于本地系统时间，同时各事件通过IO复用检测函数（select、poll、epoll）触发执行，各个执行体由于时钟和调度策略等原因使得异构执行体执行超时事件和读事件乱序，从而导致状态机失步。以路由协议ospf为例，ospf邻接关系建立过程有down、init、2-way、exstart、exchange、loading、full七种状态，状态间的切换是通过内部定时器事件与外部报文read事件触发的，在邻接关系建立过程中处理各事件的顺序的不同会导致异构执行体ospf邻居所处的状态的不同。如图3所示，异构执行体邻居状态处于exchange，交换database数据库信息，由于异构执行体LSU定时器到期和程序调度时机的不一致，执行体1和执行体2先发送LSU报文，择多判决，调度器发送LSU报文给远端，远端回复LSAck报文，执行体1和执行体2接收LSAck报文后邻居状态变为full，执行体3先接收LSAck报文并丢弃，后发送LSU报文，因为未接收到与LSU对应的LSAck报文而一直重新发送LSU报文，邻居状态依然为exchange。

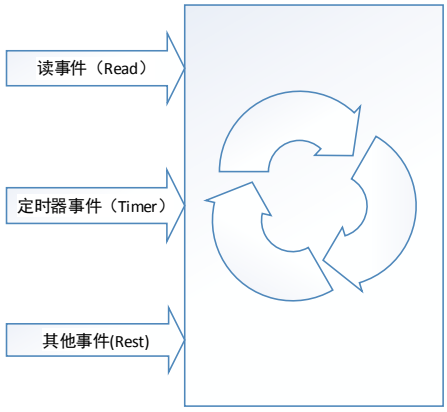


图2　进程状态转移图

在容错系统中复制状态机采用Paxos、ZAB、Raft[7-9]等共识算法解决状态机同步问题，但是这些方法各个执行体之间需要相互通信，与拟态防御系统各执行体之间不能协同的原理相悖，消息通信复杂且增加不必要的延时。由于各个执行体不可信任，所以属于拜占庭问题，需要更多的执行体和更复杂的通信算法才能保证系统的可靠。

在异构执行体程序运行状态同步方法的探索中提出过异构执行体系统时间同步方法、部分关键事件同步方法、事件整队列同步方法，其中

异构执行体系统时间同步方法定期向调度器同步系统时间来保证各执行体定时器事件执行顺序的一致性，该方法能够一定程度上解决调度简单的程序状态的同步问题，但对于调度复杂的程序则不适用；

部分关键事件同步方法采用的是程序的部分关键事件向调度器同步后执行来保证程序状态的一致性，该方法对于复杂程序的同步结果优于系统时间同步方法，但受同步信息丢包影响很大，且为区分事件及其响应，程序需要做大量修改，同时交互量较大的程序难以保证所有事件执行的顺序一致，从而导致程序状态失步；

事件整队列同步方法将事件队列发送给调度器，调度器将共有事件发送给异构执行体来保证异构执行体事件执行的一致性，该方法的同步结果优于前两种方法，但为保证异构执行体事件执行的一致性需要进行阻塞申请，因此需要保证同步申请消息不能丢包，增大了工程实现的复杂性，同时该方法需要区分事件及将事件重新排序，

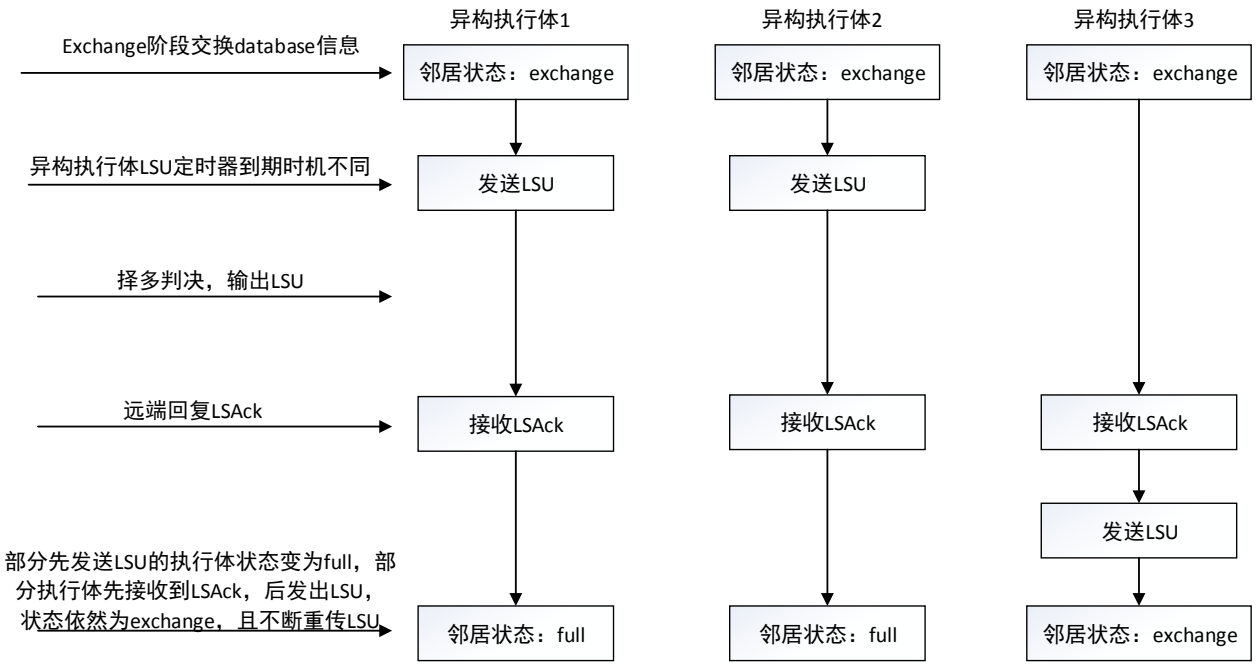


图3　ospf邻居状态变化图

增加实现的复杂性，每一次事件检测循环均需要同步一次，交互量大导致程序的执行效率降低。

本文首先提出基于相对时间的异构执行体程序同步的拟态系统模型，该模型在拟态防御系统各执行体之间不能协同的前提下提出了基于相对时间的异构执行体程序同步方法，该方法充分考虑了异构执行体事件执行顺序的一致性、同步申请量、工程应用环境的丢包等问题，通过理论分析和实验来证明该方法的可行性。

2　基于相对时间的异构执行体程序状态同步的拟态系统模型

基于相对时间的异构执行体程序状态同步的拟态系统模型包括输入代理、异构执行体池、调度器、反馈控制器组成，模型结构如图4所示：

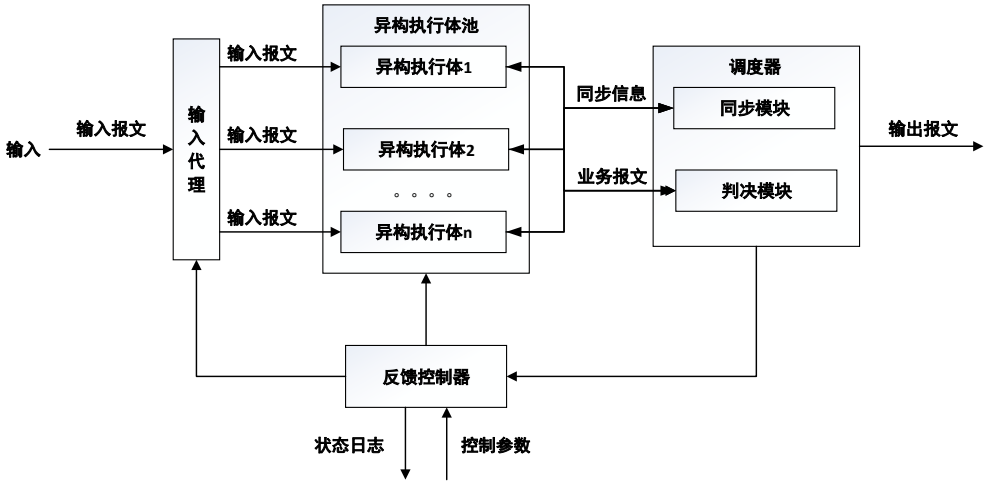


图4　基于相对时间的异构执行体程序同步的拟态系统模型

输入代理是用户请求的入口，将用户请求复制多份分发给异构执行体池。异构执行体池是由异构的、多样的、冗余的执行体组成，对用户的请求进行响应，同时将响应结果发送给调度器。调度器包括同步模块和判决模块，同步模块用于接收异构执行体发送的同步信息，根据一定的同步策略将同步结果返回给异构执行体。判决模块是用户请求响应的最终出口，为保证对用户输出

结果的一致性和正确性，将对多个响应进行投票表决，根据表决策略产生最后输出。同时调度器可以根据各执行体的表现进行相应的反馈控制，如某个执行体长期不发送同步申请或输出报文与其它子卡不一致，则对其进行清洗，待其清洗完成之后再将其加入工作队列，裁决相关参数和系统运行状态信息可通过反馈控制器进行控制和查看。

基于相对时间的异构执行体程序状态同步方法在拟态系统中起着至关重要的作用，是保证异构执行体中程序状态同步、输出报文一致性的关键。程序进程存在多个状态，状态之间的变迁由读事件、定时器事件及其他事件引起，其他事件也由读事件和定时器事件引起，各事件通过IO复用检测函数（select、poll、epoll）触发执行，基于相对时间的异构执行体程序同步方法通过控制住读事件和定时器事件顺序来保证状态机一致，从而控制异构执行体状态一致，进而控制异构执行体发出的报文一致，方便拟态防御系统中调度器进行有效判决，并控制整个系统s的正常运行。

3 实现方案

考虑一个通用拟态防御系统的异构冗余池，符号表示如表1所示，并由以下定义进行形式化描述。

3.1 方法的参数确定

1） 定时器事件时间基准

异构执行体中进程定时器事件的添加和超时基于本地系统时间，异构执行体定时器事件1添加的时间t_{11}、$t_{12}\cdots t_{1n}$，超时时间t'_{11}、$t'_{12}\cdots t'_{1n}$，定时器事件2添加的时间t_{21}、$t_{22}\cdots t_{2n}$，超时时间t'_{21}、$t'_{22}\cdots t'_{2n}$，由于异构执行体上的时钟和程序调度的不同导致定时器事件到时的时机会有偏差，异构执行体1的（t'_{11}- t_{11}）>（t'_{21}- t_{21}），即异构执行体1定时器事件2先执行，异构执行体2的（t'_{12}- t_{12}）<（t'_{22}- t_{22}），即异构执行体2定时器事件1先执行，从而导致不同执行体定时器到时的顺序不一致，执行顺序不一致，最终导致异构执行体程序状态不同步。

基于相对时间的异构执行体同步方法中定时器事件添加和超时基于的是从调度器申请的时间*RT = RT_end-RT_start*，异构执行体定时器事件*i*添加的时间均是*RT*，超时时间均为*RT+Pi*，判断定时器超时的时间基准均为从调度器申请的*RTr*，定时器事件执行顺序一致，加上对外部read事件的统一管控，实现异构执行体中timer事件、read事件执行顺序的一致，进而保证程序运行状态的一致。

表1

符号	含义
t_{ij}	异构功能等价体*i*的第*j*个定时器事件添加时间
t'_{ij}	异构功能等价体*i*的第*j*个定时器事件超时时间
RT	相对时间
RDi	异构功能等价体*i*，$1 \leqslant i \leqslant n$，read事件指示
EN	异构执行体个数
ENr	有效异构执行体个数
TVi	定时器事件*i*，$1 \leqslant i \leqslant n$
RV	read事件
RDr	调度器的读时间响应结果，$RDr = \sum_{i=1}^{ENr} RDi$。
NODEsn	同步次数为*sn*的同步信息节点
SNs	同步申请次数
SNr	有效执行体申请信息中的共有的*SN*的最大值
RT_start	调度器中*RT*初始时间
RT_end	调度器同步信息发送事件
R_start	调度器中SN_s节点添加的初始时间
R_now	调度器当前时间
INDEXi	*NODEsn*中执行体的置位标识
INDEXset	*NODEsn*中执行体的置位个数
RTr	调度器返回的相对时间*RTr = RT_end−RT_start*
Ti	定时器事件*i*的到期时间
Pi	定时器事件*i*周期
EV	$EV = \{ev1, ev2\cdots evm\}$，事件集合
T	$T = \{T1, T2\cdots Tn\}$，定时器事件到期时间集合
t_now	当前时间
Twait	等待时间
Qready	ready 队列
Qrest	rest队列
Ms	同步消息
Mr	同步响应消息

2） 事件顺序控制

进程状态之间的变迁由read事件、timer事件引起，控制住read事件、timer事件的执行顺序才能保证异构执行体状态机一致。基于相对时间的异构执行体程序同步方法在每一次IO复用检测函数（select、poll、epoll）触发执行时向调度器申请定时器事件的相对时间及read事件的同步，事件

的添加顺序为先将 $RT + Pi < RTr$ 定时器加入 Q_{ready} 后，再判断 $RDr = EN$（其中 $RDr = \sum_{i=1}^{ENr} RDi$）来保证执行体均收到read事件后将read事件加入 Q_{ready}，保证了事件执行顺序的一致性，从而保证程序运行状态一致。

3） 申请方式

申请方式有阻塞申请和非阻塞申请两种，阻塞申请下，程序接收不到同步响应信息会一直阻塞不执行，非阻塞申请下，即使程序没有立刻接到回复也会继续执行。在实际工程应用中，丢包是不可避免的问题，同步信息的丢包会导致程序阻塞等待无法继续执行，从而导致程序不能正常运行。

基于相对时间的异构执行体同步方法采用的是非阻塞同步方式，每一次IO复用检测函数触发执行时向调度器发送的同步信息中携带同步标识*SNs*，调度器收到同步信息后查找同步响应队列中包含所有有效执行体的最大标识*SNr*的响应消息发送给该执行体，该执行体通过对比同步标识*SNr*是否与*SNs*相等来确定后续执行过程，避免由于丢包导致的程序不能正常运行。

4） 同步信息等待时间阈值

异构执行体程序运行时每一次IO复用检测函数触发执行均会向调度器申请时间，若某个异构执行体程序异常，同步申请中断，调度器端若无限等待，则所有执行体程序均会等待，因此调度器端需要添加同步信息等待时间的阈值。

根据异构执行体程序运行时每秒钟IO复用检测函数触发执行的次数统计，以路由协议ospf为例，根据交互量的不同，执行次数范围为10~300，因此基于相对时间的异构执行体程序同步方法中采用1s为同步信息等待时间的阈值。调度器端在接收到同步标识为*SNs*的同步信息后记录时间*R_start*，若在1s中以内，即*R_now-R_start* >1时未收到某一执行体的同步申请，即*NODEsn*中某些执行体的*INDEXi*= 0，则证明该异构执行体出现异常，可将该执行体的同步信息从同步队列中删除，判断其它执行体的同步信息，同时清洗异常的执行体。

3.2 方法的实现流程

基于相对时间的异构执行体程序状态同步方法的具体流程如下：

（1） 异构执行体向调度器发送同步申请信息

执行体判断是否有读事件，将读标识RDs置位，发送包含读标识RDs、申请次数SNs的同步申请信息给调度器，以定时器事件最小到时时间为IO复用检测函数的等待时间。

（2） 调度器接收同步申请信息，返回同步申请响应

调度器接收执行体同步申请信息后，若没有*NODEsn*节点则添加，并记录创建时间*R_start*，若已有则将相应的执行体标识*INDEXi*置位，判断*NODEsn*的存在时间*R_now - R_start*是否大于1s，若*R_now - R_start*>1，则根据*NODEsn*信息计算*SNr*、*RDr*、*RTr*、*ENr*（$SNr = SNs$，$RDr = \sum_{i=1}^{n} RDi$，$RTr = R\_now - RT\_start$，$ENr = INDEXset$）发送*Mr*，清洗*NODEsn*中*INDEXi*=0的执行体。

（3） 异构执行体未收到同步响应信息的处理

异构执行体若在等待时间内未收到调度器发送的同步响应信息，以相同的申请次数SNs发送同步申请信息给调度器，以定时器事件最小到时时间为IO复用检测函数的等待时间。

（4） 异构执行体收到同步响应信息的处理

异构执行体若在等待时间内收到调度器发送的同步响应信息，则先判断*SNr*与*SNs*是否相等，若$SNr = SNs$，则根据收到的*RTr*进行计算，将定时器事件的超时事件加入ready队列*Qready*中，同时判断*RDr*与*ENr*是否相等，若相等则将read事件加入ready队列*Qready*中，程序处理ready队列。若$SNr \neq SNs$，则将同步响应消息丢弃。

3.3 异构执行体端的执行流程

基于相对时间的异构执行体程序状态同步拟态系统异构执行体端的流程如图5所示：

执行体端算法实现如下：

其中*Ms*中含有*RTs*，*RD*，*SNs*

3.4 调度器端的执行流程

基于相对时间的异构执行体程序状态同步的拟态系统调度器端的流程如图6所示：

调度器端算法实现如下：

其中*Mr*中含有*RDr*，*RTr*，*SNr ENr*。

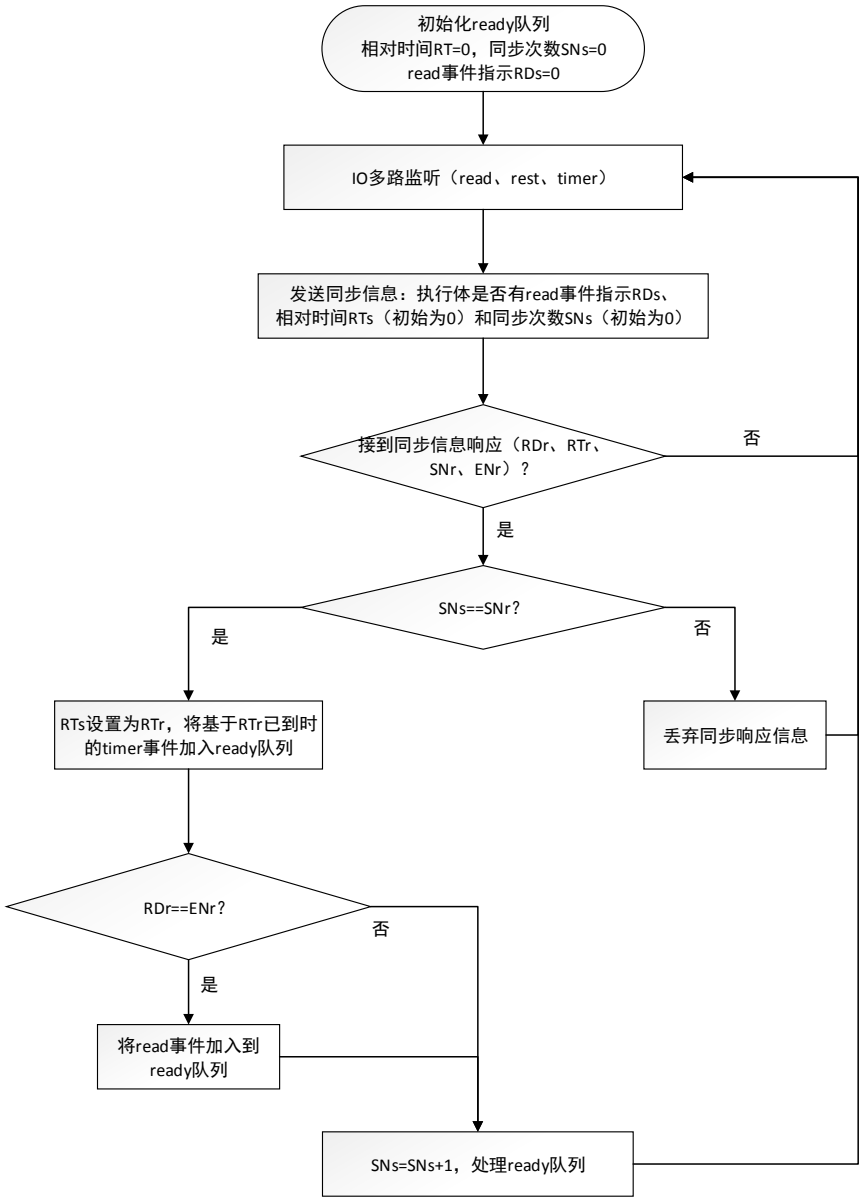


图5　基于相对时间的异构执行体同步方法-执行体端处理流程

4　实验结果及分析

路由协议ospf邻居存在多个状态，邻接关系建立过程有down、init、2-way、exstart、exchange、loading、full七种状态，状态之间的变迁由读事件、定时器事件引起，各事件通过IO复用检测函数触发执行，因此在邻接关系建立过程中处理各事件的顺序的不同会导致异构执行体ospf邻居所处的状态的不同，下面以ospf路由协议为例将基于相对时间同步方法与异构执行体系统时间同步、部分关键事件同步、事件整队列同步等方法的同步效果进行对比，验证基于相对时间的异构执行体程序同步方法，√表示状态同步：邻居状态一致、database一致、路由条数一致；×：邻居状态不一致、database不一致、路由条数不一致中的一个。

实验配置环境如表2所示：

Ospf协议进程运行在三个异构执行体上，修改quagga-1.2.4中ospf源码，增加向调度器申请定时器添加与超时的时间基准、read事件和timer事件的执行顺序向调度器同步功能，调度器通过fpga编程实现同步信息的接收及响应。表3给出基于相对时间同步方法与异构执行体系统时间同步、部分关键事件同步、事件整队列同步等方法在程序

Algorithm 1	基于相对时间的同步算法-执行体端
Input：初始RT、初始SN、初始RD，M_r	
Output：异构执行体同步结果	
1	$Qready$= 0 //初始化ready队列
2	RT = 0,RD = 0,SNs = 0//初始化基准时间，read事件标识，申请次数
3	select，$Twait$ = min (T) - RT//等待时间出发，等待时间为定时器事件到期最小值。
4	Send(Ms) //发送Ms给调度器
5	Recv(Mr) //接收Mr
6	if SNr == SNs//同步返回次数与申请次数相同
7	go to line 10
8	else:
9	go to line 3
10	if RTr - RT > Pi
11	add(TVi) //将到期的定时器事件加入$Qready$中
12	if RDr == ENr
13	add(RV) //将读事件加入$Qready$中
14	Execute($Qready$)//执行ready队列中事件
15	SNs = SNs + 1，RT = RTr，go to line 3

修改量方面的对比。

4.1 路由容量场景测试

路由容量的测试场景采用的是测试仪向异构执行体发送不同数量的LSA，异构执行体中的ospf程序接收到LSA后进行路由计算，随着测试仪向异构执行体发送的LSA条目数的增多，ospf协议与外界交互的报文数增加，从而导致ospf内部的read事件与定时器事件的数量增加，在此场景下对比各种同步方法的同步效果。

测试仪分别向异构执行体发送3000条、5000条、8000条LSA，测试结果如表4所示：

针对于路由容量的多次测试，3000条路由学习，各方法均能实现，对于5000条以上路由学习，异构执行体系统时间同步方法由于受不同执行体获取时间的时机不一致影响变大，事件执行顺序不一致从而导致状态失步，其它方法均能实现；8000条路由学习，由于交互的LSU报文增多，read事件数与定时器事件数随之增大，部分关键事件同步方法出现未同步的事件与read事件执行顺序不一致，从而导致状态失步，事件整队列同步方法与基于相对时间同步方法不受事件乱序的影响均能实现路由学习。

4.2 多邻接关系建立场景测试

多邻接关系的测试场景，异构执行体中ospf角色为DR，测试仪模拟4个设备与异构执行体建立邻接关系，随着每个设备中具有的LSA数量的增加，ospf内部的read事件与定时器事件的数量在短时间内急剧增加，在此场景下对比各种同步方法的同步效果。

测试仪每个设备具有1000条LSA、2000条LSA、3000条LSA等场景的异构执行体程序状态，结果如表5所示：

针对多邻接关系的不同LSA数量的多次测试，分别具有1000条LSA的4个设备的邻接关系建立，系统时间同步方法由于短时间内报文交互量大导致执行体获取时间不一致，从而导致状态完全失步；分别具有2000条LSA的4个设备的邻接关系建立，由于短时间内交互量的增加引起的read事件数与定时器事件数增大，部分关键事件同步方法出现未同步的事件与read事件执行顺序不一致，从而导致状态失步；分别具有3000条LSA的4个设备的邻接关系建立，事件整队列同步方法由于同步次数的增加，事件的处理时间超过了报文交互要求的最长时间导致邻接关系不能正常建立，从而导致状态失步；虽然随着报文交互量的增加read事件数和定时器数量会随之增加，但基于相对时间同步方法中的timer事件的添加和超时基于的是同步的相对时间，定时器事件的增加不会导致同步次数增加，同步次数受read事件增加影响，但同步次数增加带来的事件处理的时间延迟远小于报文交互要求的时间，因此基于相对时间同步方法能够稳定的实现状态同步。

4.3 同步消息丢失场景测试

在实际的工程实践中，链路丢包是常见现象，通过调度器随机不接收及连续不接收某些执行体同步申请消息、调度器随机不发送及连续不发送同步响应信息四个场景对比各种同步方法的同步效果，测试结果如表6所示：

针对同步消息丢失的场景，异构执行体采用系统时间同步、部分关键事件同步和事件整队列等方法均会受同步信息丢包的影响，从而导致程序状态失步，基于相对时间的异构执行体同步方法不受少量同步信息丢包的影响，对于连续多次的丢包现象能够及时发现该异构执行体出现问题，

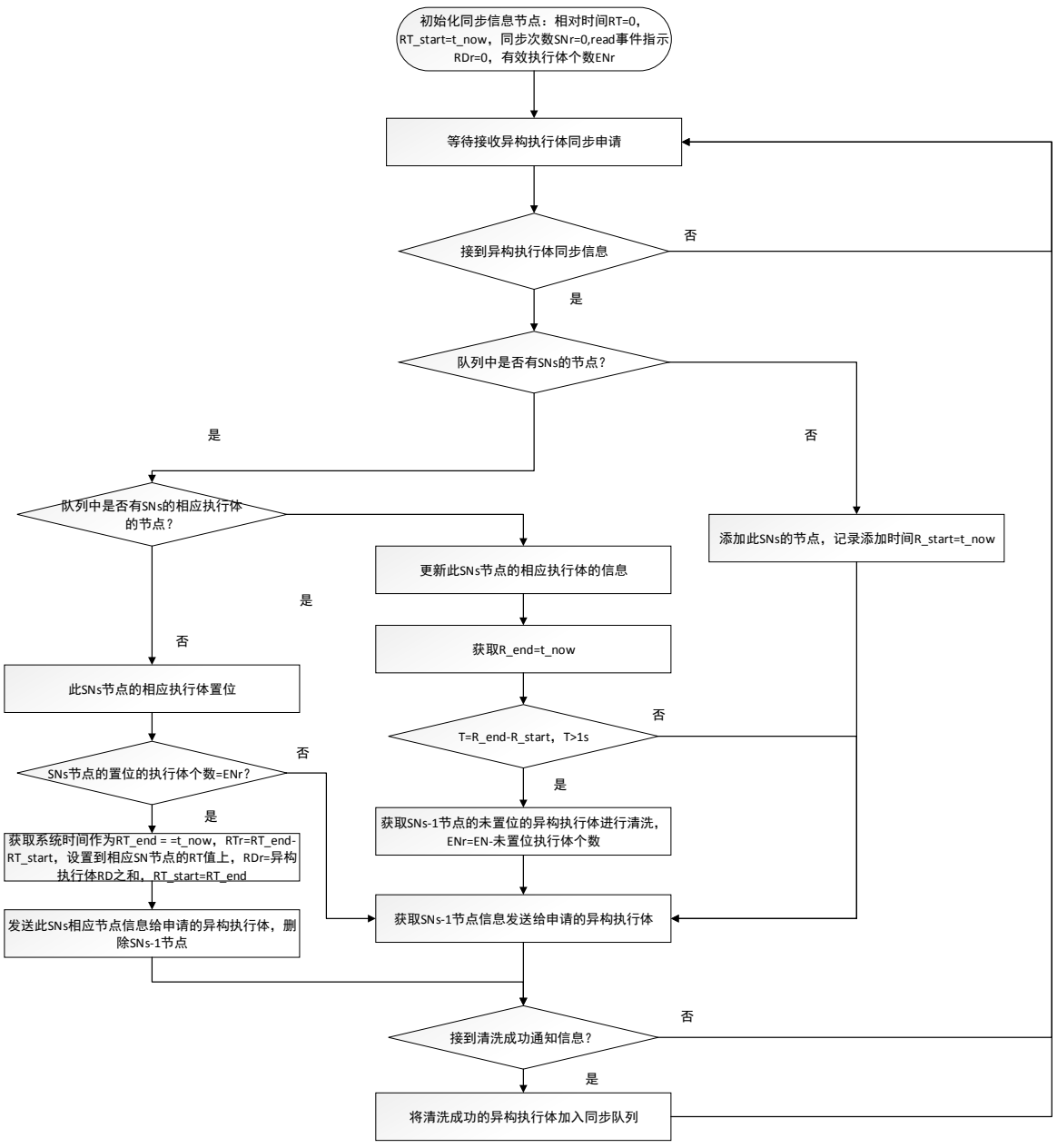


图6　基于相对时间的异构执行体同步方法-调度器端处理流程

从而启动该执行器清洗恢复工作。

4.4　实验结果分析

以上通过路由容量、多邻接关系建立、同步信息丢包等场景的多次测试，将基于相对时间同步方法与异构执行体系统时间同步、部分关键事件同步、事件整队列同步等方法的同步效果进行对比，实验结果显示，在基于相对时间的异构执行体程序同步方法在与外界大量交互、read事件与定时器事件的数量较大的环境下能够实现复杂路由协议ospf程序运行状态的一致，能够有效解决同步过程丢包带来的影响，同时能够灵活的随子卡状态的变更改变同步策略。

5　结束语

本文介绍了基于相对时间的异构执行体程序状态同步方法以及其在拟态系统中的关键作用。讨论了Paxos、ZAB、Raft等共识算法解决状态机同步方法不能应用在拟态防御系统。通过与异构执行体系统时间同步、部分关键事件同步、事件整队列同步等方法在ospf协议应用上的同步效果进行实验对比，验证基于相对时间的异构执行体程序状态同步方法能够有效的使异构执行体的状

Algorithm 2：基于相对时间的同步算法-调度器端

Input：初始$RT=0$，$Ms(RDr=0, ENr)$

Output：同步结果M_r

1 $RT=0$，$RDr=0$，EN//初始化基准时间、read事件标识，

2 Recv(Ms) //接收同步申请信息

3 if $NODEsn$ == NULL

4 Create($NODEsn$),Insert(Ms,R_start), goto line 12//创建节点，插入Ms，记录R_start

5 else:

6 Insert(Ms,0)，goto line 7

7 if R_now-R_start>1s

8 send(Mr)//发送Mr，Mr中的$SNr=SNs$, $RDr=\sum_{i=1}^{ENr} RDi$（$INDEXi\neq 0$），RTr，$ENr=\sum_{i=1}^{EN} INDEXi$。

9 If $INDEXi$ == 0

10 Clear(i) //清洗$INDEXi=0$的子卡，

11 else:

12 $INDEXi=1$，goto line 13

13 Compute(RDr,SNr,RTr) //$RDr=\sum_{i=1}^{ENr} RDi$，$RTr=RT\_end-RT\_start$，$SNr=SNs-1$

14 Send(Mr)，go to line 2

表2

名称	型号	实现方式
异构执行体1	X86	quagga-1.2.4
异构执行体2	PowerPC	quagga-1.2.4
异构执行体3	ARM	quagga-1.2.4
调度器	690t	Fpga
测试仪	Spirent测试仪	Spirent测试仪模拟测试设备

态实现同步。在以后的工作中要优化现有方法，降低同步次数，提高处理效率。研究在拟态防御系统中异构执行体程序完全同步的方案，减少由于状态机不同步所导致的错误表决和纠错。

表3

	系统时间同步	部分关键事件同步	事件整队列同步	基于相对时间同步
修改内容	增加系统时间申请及修改	增加事件区分标识、执行时增加事件同步申请及响应	增加事件区分标识、同步申请、事件排序	增加相对时间申请
改动量	小	大	大	小

表4

	系统时间同步	部分关键事件同步	事件整队列同步	基于相对时间同步
3000条LSA	√	√	√	√
5000条LSA	×	√	√	√
8000条LSA	×	×	√	√

表5

	系统时间同步	部分关键事件独立	事件整队列同步	基于相对时间同步
1000条LSA	×	√	√	√
2000条LSA	×	×	√	√
3000条LSA	×	×	×	√

表6

	系统时间同步	部分关键事件同步	事件整队列同步	基于相对时间同步
随机不接收	×	×	×	√
连续不接收	×	×	×	√
随机不发送	×	×	×	√
连续不发送	×	×	×	√

参考文献:

[1] 邬江兴. 网络空间拟态防御原理[M]. 北京：科学出版社，2018.
WU J X. Cyberspace mimic defense[M]. Beijing：Science Press，2018.

[2] 邬江兴. 网络空间拟态防御研究[J]. 信息安全学报，2016，1(4)：1-10.
WU J X. Research on cyber mimic defense[J]. Journal of CyberSecurity，2016，1(4)：1-10.

[3] 扈红超, 陈福才, 王禛鹏. 拟态防御 DHR 模型若干问题探讨和性能评估[J]. 信息安全学报，2016，1(4)：40-51.
HU H C, CHEN F C, WANG Z P. Performance evaluations on DHR for cyberspace mimic defense[J]. Journal of Cyber Security，2016，1(4)：40-51.

[4] 宋克,刘勤让,魏帅,张文建,谭力波. 基于拟态防御的以太网交换机内生安全体系结构[J]. 通信学报，2020，41(5)：18-26.
SONG K,LIU Q R,WEI S,ZHANG W J,TAN L B. Endogenous security architecture of Ethernetswitch based on mimic defense [J]. Journal on Communications，2020，41(5)：18-26.

[5] 魏帅,于洪,顾泽宇,张兴明. 面向工控领域的拟态安全处理机架构[J]. 信息安全学报，2017，2(1)：54-73.
WEI S,YU H,GU Z Y,ZHANG X M. Architecture of mimic security processor for industry control system[J]. Journal of Cyber Security，2017，2(1)：54-73.

[6] 刘勤让,林森杰,顾泽宇. 面向拟态安全防御的异构功能等价体调度算法[J]. 通信学报，2018，39(7)：188-198.
LIU Q R,LIN S J,GU Z Y. Heterogeneous redundancies schedulingalgorithm for mimic security defense [J]. Journal on Communications，2018，39(7)：188-198.

[7] 祝朝凡,郭进伟,蔡鹏. 基于Paxos的分布式一致性算法的实现与优化[J]. 华东师范大学学报(自然科学版) ,2019,(5):168-177.
ZHU C F,GUO J W,CAI P. Implementation and optimization of a distributed consistency algorithm based on Paxos[J]. Journal of East China Normal University(Natural Science)，2019,(5):168-177.

[8] 呼美玲. 基于RDMA的高性能Paxos算法的设计与实现[D]. 浙江:浙江大学，2018.
HU M L, Design and Implementation of High-Performance Paxos on RDMA[D]. Zhejiang：Zhejiang University，2018.

[9] 王江,章明星,武永卫,陈康,郑纬民. 类Paxos共识算法研究进展[J]. 计算机研究与发展，2019，56(4)：692-707.
WANG J,ZHANG M X,WU Y W,CHEN K,ZHENG W M. Paxos-like Consensus Algorithms:A Review[J]. Journal of Computer Research and Development，2019，56(4)：692-707.

[作者简介]

苏野（1987-），女，硕士，天津市滨海新区信息技术创新中心工程师，主要研究方向为计算机软件、网络体系结构。

魏帅（1984- ），男，博士，信息工程大学助理研究员，主要研究方向为计算机软件、网络体系结构。

姚领彦（1986-），女，硕士，天津市滨海新区信息技术创新中心工程师，主要研究方向为计算机软件、网络体系结构。

谭力波（1981- ），男，本科，天津市滨海新区信息技术创新中心高级工程师，主要研究方向为网络空间安全、集成电路设计、软件定义互联。

面向拟态判决的最小异常值判决方法

谭力波[1]，贾广瑞[2]，张文建[3]，欧阳玲[3]，宋克[1]

[1]天津市滨海新区信息技术创新中心，天津 300450；

[2]河北工业大学 电子信息工程学院，天津 300401；

[3]战略支援部队信息工程大学 信息技术研究所，郑州 450002

摘　要：为简化常规拟态判决时存在语义相同语法不同需要进行数据归一化的过程，本文提出了最小异常值判决方法。首先，使用网络流量异常检测的方法训练了两个神经网络异常检测模型，用于对拟态判决器输入数据异常检测；其次，构建了基于拟态防御系统的异常检测数据集，进行了功能仿真实验和安全性仿真实验。实验结果表明，基于异常检测的拟态判决方法能够解决语义相同语法不同的问题，与多数表决算法相比本文算法安全性更好。

关键词：拟态判决、数据归一化、异常检测

Resource Mobility based Hybrid Task Planning in Space Information Network

——(1. Information Technology Innovation Center of Tianjin Binhai New Area, Tianjin, 300457, China; 2. School of Electronic and Information Engineering, Hebei University of Technology, Tianjin, 300401, China; 3. Information Technology Research Institute, Strategic Support Force Information Engineering University, Zhengzhou 450002, China)

Abstract: In order to simplify the process of data normalization with the same semantics and different syntax in conventional mimic judgment, this paper proposes a minimum outlier judgment method. First, two neural network anomaly detection models were trained using the method of network traffic anomaly detection to detect anomaly in the input data of the mimic decision device; secondly, an anomaly detection data set based on the mimic defense system was constructed, and functional simulation experiments were performed. Safety simulation experiment. The experimental results show that the mimic judgment method based on anomaly detection can solve the problem of the same semantics and different grammars. Compared with the majority voting algorithm, the security of this algorithm is better.

Key words: mimic judgment; data normalization; anomaly detection

1　引言

信息安全与网络安全越来越受到人们的重视，拟态防御技术[1]作为网络内生安全的一项新兴技术逐渐走向成熟，成为一种重要的网络安全防御手段。拟态防御系统主要是通过构建动态异构冗余（Dynamic Heterogeneous Redundancy，DHR）架构来实现系统的内生安全机制，如图1所示，通过动态调度策略对在线执行体不定时的切换和拟态裁决器的判决，保证系统的安全性和可靠性。

目前，网络安全领域的拟态防御系统都是采用异构操作系统、异构处理器和异构运算器等组件来构建拟态防御的异构系统[2]。由于拟态系统中各执行体组件的异构化，导致在进行拟态判决时判决器输入的数据具有相同的语义但数据的语法不同，主要体现在数据分组头中包含多种协议，

基金项目：国家核高基重大专项基金资助项目（No.2017ZX01030301）

每种协议为了考虑可扩展性都设定的自定义字段、保留字段、可选字段；数据负载中由于操作系统的不同协议栈的不同导致数据段顺序的不同，这些都对拟态判决造成了很大的困扰[3]。

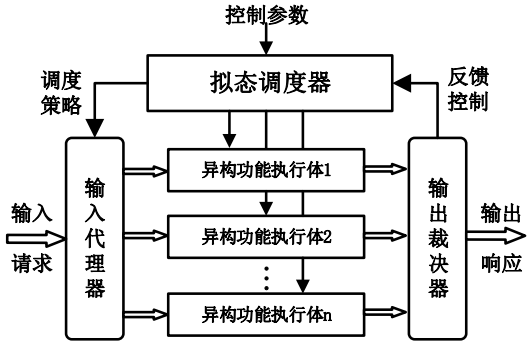


图1　拟态防御系统通用架构图

拟态判决器的策略已经被提出许多，主要以多数判决为主并增加辅助判决的机制，例如全体一致判决法，多数判决法，复数判决法，中值判决法，n中取2判决法，一致性判决法等[4-5]。其中，多数一致判决算法最具有代表性，基本思想是将多数相同的裁决的结果作为判决器输出，从而达到屏蔽一般错误的效果。然而，当前研究拟态判决算法都是基于多执行体输出结果完全一致的情形[6]，当多数执行体输出结果出现语义相同语法不同的情形时，需要对数据进行归一化处理。常用的归一化处理时利用软件或硬件电路的方法实现对数据协议的解析，对数据中自定义字段、保留字段、可选字段进行归一化处理，对数据乱序的内容进行重新排序。因此，当前在进行拟态判决时进行的归一化处理严重降低了系统判决的效率，同时由于归一化操作需要对数据进行深度的报文解析等操作，过程比较复杂。此外，由于不同的拟态系统传输数据不同，因此对不同系统的归一化过程不具有通用性。

对于现有拟态系统的判决过程中需要对数据进行归一化处理，增加了系统的复杂性且适用性较差的问题。本文从另一种角度出发通过构建深度学习模型对数据进行归一化处理和语义分析，借鉴深度学习网络流量数据异常检测的方法[7]，构建了基于最小异常值的拟态判决方法。首先，构建了基于深度学习的异常检测模型；然后，根据拟态防御系统的语义相同语法不同问题创建了特定的数据集，用于对深度学习异常检测模型的训练；最后，将训练好的模型用于对拟态判决器输入数据的异常检测和最小值判决输出最终结果。利用深度学习的异常值检测方法能够避免归一化的处理的复杂过程，此外通过异常检测的最小异常值判决能够提高系统的安全性。

通过实验仿真结果表明，本文提出的最小异常值判决方法与原有多数表决方法相比提高了判决的准确率，使得系统的可靠性有了较大的提升。此外，本文提出的方法能够降低拟态判决时归一化的复杂度，并且对于不同的拟态系统适用性较强，只需要根据不同系统使用对应的数据集训练模型即可，操作简单通用性较强。

2　研究思路

深度学习异常检测的过程具有对数据高层次抽象化的优点，能够避免在以往进行拟态判决时的归一化处理过程。通过建立合适的数据集对深度学习模型训练，经过不断训练和学习的过程能够将语义相同语法不同的数据经过数据抽象达到归一化处理，最终根据抽象结果的异常值进行判决。在简化归一化过程的同时，系统具有一定的异常检测（错误感知）能力，在一定程度上提升系统的安全性。因此，面向拟态判决的最小异常值判决模型构建过程主要分为以下三个部分：

（1）数据集的构建。当前拟态防御系统主要针对网络数据传输过程的安全和网络设备的安全，主要包含网络服务器，路由器，交换机等网络设备的拟态化。因此，建立用于拟态防御系统的异常检测数据集[8]，需要符合拟态设备的数据特性。即，数据采集于真实网络环境，网络具有异构性，数据具有全面性，且需要包含各种常见的网络异常流量。

（2）异常检测模型的构建。异常检测模型用于对各个在线异构执行体输出数据的检测，因此需要耗费一定的资源和计算能力，所以异常检测模型应该在保证检测能力需求的情况下尽可能的小，以满足拟态系统的性能需求。此外，拟态系统具有异构冗余的特性，因此需要有多个功能相同架构不同的异常检测模型，以满足系统安全性需求。

（3）最小异常值判决算法。将构建好的异常检测模型用于拟态判决，判决过程中要对异常检

测模型进行有针对性的调整以满足异构型。在简化数据归一化的前提下保证系统的安全性可靠性。同时，由于异常检测模块具备错误检测能力，因此系统具有共模逃逸检测能力。

综上所述，本文研究将数据集的构建、异常检测模型的构建、最小异常值判决算法三方面展开，简化拟态判决数据归一化过程，提高系统的有效性；增加异常检测模块的错误感知功能，提高系统的可靠性。

3　最小异常值判决算法

假设拟态防御系统在线执行体数量为n，最小异常值判决算法描述如下：

（1）各执行体的输出数据表示为集合 $D=\{d_1,d_2,d_3,\cdots,d_n\}$，$d_i$ 代表每个执行体的输出结果。由于，各异构执行体的输出结果存在语意相同语法不同的问题，所以即使输出正常结果也存在某些数据字段不同的情况。本文假设 d_i 互不相同，需要进行相应的数据处理后才能进行判决。

（2）将执行体的输出结果 d_i 进行神经网络异常检测，得到每个执行体输出结果的异常值 $Y=\{y_1,y_2,y_3,\cdots,y_n\}$，因为异常检测为二分类模型，分类器为Sigmoid函数输出结果，如图2所示，所以 $0\leqslant y_i\leqslant 1$ 的数值。

（3）最小异常值判决时判决时选择异常值 $Y=\{y_1,y_2,y_3,\cdots,y_n\}$ 的最小值 $y_{\min}$ 对应的数据结果 d_i 输出。

（4）当所有执行体输出结果异常值的最小值 $y_{\min}\geqslant 0.6$ 时认为所由执行体结果都出现异常，拟态系统发生共模逃逸。其他情况判决失效，需要重新判决。

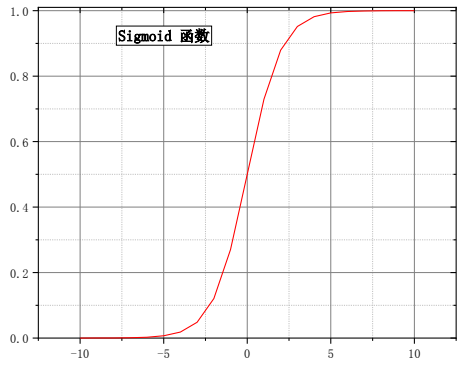


图2　Sigmoid函数示意图

4　深度学习异常检测模型

本文研究的异常检测方法是基于有监督学习的分类的方法，即通过对正常数据于异常数据的区分来进行异常检测，需要使用有数据集和标签集的先验知识库来对深度学习模型训练[9]。建立最小异常值判决模型需要构建适用于拟态系统的数据集，以及根据相应的异常检测任务建立神经网络模型并训练来满足最小异常值判决的需求。

4.1　基于拟态防御系统的数据集构建

由于拟态防御系统本身为网络安全防御的重要基础，目前缺少公开的具有权威性的拟态防御系统相关数据集。能够用于拟态防御系统的异常检测数据集需要具备数据真实性，络流量数据为真实网络中采集；需要具备同种网络设备的异构性，考虑多个异构执行体的输出；需要具备数据的完整性，采集网络架构中从发送端到接收端完整的数据流；需要具备异常数据多样性，在网络传输中加入多样化的攻击流量。此外，由于个人建立的数据集具有一定的局限性，因此本文选取了目前网络异常检测认可度较高的ISCX-2012数据集[10]。

ISCX-2012数据集为加拿大新不伦瑞克大学的信息安全中心在2012年发布的入侵检测数据集[11]，包含了构建网络中所有传输的原始流量数据，其中包含4中异常流量Brute Force SSH，DDoS，HttpDos，Infiltrating，对于研究网络异常检测具有重要意义。数据集的构建过程共包括组建的两个主要网络，即攻击网络和被攻击网络。攻击网络和被攻击网络共5个部门，包括多台不同操作系统不同组织架构的服务器、路由器、计算机、交换机等场见的网络设备，共进行7天的网络流量捕获较为全面的涵盖了当前网络攻击所有必要条件和最新的攻击类型。

可以将数据集构架过程中的同种设备不同操作系统和不同架构类比为拟态防御系统中的多个异构执行体，此外，该数据集具备数据真实性和完整性、设备异构性、异常多样性等特点基本满足本文需求，能够用于训练异构执行体的异常检测模型。

异常检测模型的输入需要固定大小固定形式的数据，因此在进行训练前需要对数据进行预处

理。预处理过程主要分为按网络数据流量的切分和数据大小的选取两阶段。**阶段一**，数据流量的切分需要按照一定的粒度对网络流量进行离散化，常用的网络流量切分方式有基于TCP连接、流、会话、服务、主机五种切分粒度[12]。根据拟态防御系统需要和目前主流研究方法，本文选择使用流（Flow）的切分粒度进行网络流量异常检测，即按照具有相同五元组（源IP、源端口、目的IP、目的端口、传输层协议）信息进行粒度切分。**阶段二，**神经网络模型的输入需要固定大小和形式的数据输入，对于不同的流（Flow）对于长度不同，因此需要对输入模型的流（Flow）同一长度处理。参考文献[13]，本文按照长度1000字节进行固定长度处理，大于1000字节进的流（Flow）进行截断保留前1000字节，小于1000字节进的流（Flow）进行用（0x00）补长。

表1　ISCX-2012数据集预处理统计

类别	训练集	测试集	占比
Normal	890726	593811	97.27%
BFSSH	4179	2785	0.46%
HttpDoS	2090	1392	0.23%
DDoS	12673	8448	1.38%
Infiltrating	6027	4017	0.66%
合计	915695	610453	100%

表1展示了对网络攻击流量进行预处理后的统计结果，其中正常数据占比97.27%，异常数据占比2.73%，将数据集按照训练集60%和测试集40%的比例随机划分。

4.2　神经网络模型的构建

基于拟态防御中动态异构冗余的思想，本文创建了两个深度学习异常检测模型用于最小异常值判决。由于预处理后的每条网络数据流不是很大，且分类任务比较明确只需轻量级的神经网络就能满足基本需求。

基于一维卷积神经网络（CNN）的异常检测模型。1D-CNN模型示意图如图3所示，主要包括输入层，卷积层，池化层和全连接层。输入层：网络输入为长度1000字节（Byte）的一维数据流（Flow）；

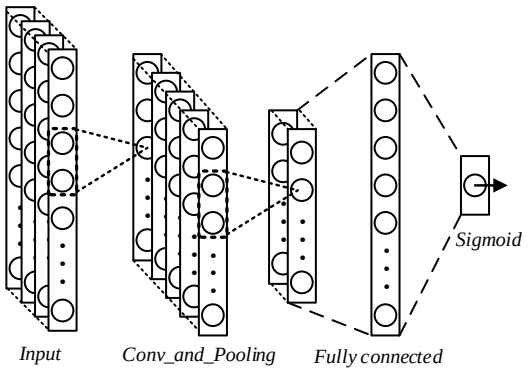


图3一维卷积神经网络示意图

卷积层：将一维数据与卷积核进行卷积运算提取数据特征，使用卷积核最大的优势是权值共享减少网络参数避免过拟合现象，计算公式为$x_k^l = \sum_{i=1}^{N_{l-1}} 1Dconv(w_{ik}^{l-1}, s_i^{l-1}) + b_k^l$. 其中$x_k^l$为第$l$层第$k$个神经元的输入，$w_{ik}^{l-1}$为第$l$-1层第$i$个神经元与第$l$层第$k$个神经元的卷积核，$s_i^{l-1}$为第$l$–1层第$i$个神经元的输出，$b_k^l$为第$l$层第$k$个神经元的偏置。此外，每个卷积运算后为了加速收敛使用（rectified linear unit， ReLU）激活函数，计算式为$y_k^l = f(x_k^l) = \max(0, x_k^l)$。池化层：为了减少特征空间和网络参数的计算量，防止过拟合，通常卷积层之后进行池化操作，本文中使用最大值池化$s_k^l = \max_{(k-1)H+1<j<kH}(s_j^{l-1})$，其中H为卷积核大小。输出层使用Sigmoid分类器将数据转换为输出一个0~1概率值，用概率值表示数据异常值。

基于循环神经网络（RNN）的异常检测模型。循环神经网络的示意，如图4所示，由于网络数据流是关于时间先后顺序特征的数据，因此使用循环神经网络（RNN）检测将能取得不错的效果。

循环神经网络的输入层为1000字节（Byte）的数据，在输入网络前首先对每一字节进行One-hot编码，使用256长度的向量表示每一个字节数据。如图所示，在每个时间戳，网络层接受当前时间戳的输入 和上一个时间戳的网络状态向量h_{-1}，经过$h_t = f_\theta(h_{t-1}, x_t)$运算后后得到当前时间戳的新状态向量，并写入内存状态中，其中 代表了网络的运算逻辑， 为网络参数集。在每个时间戳上，网络层均有输出产生y，y = （），即将网络的状态向量变换后输出。循环神经网络一般使用xh 、 hh 和偏置来参数化网络，并按照$h_t = \tanh(w_{hh}h_{t-1} + w_{xh}x_t + b)$，其中用tanh为激活函数。经过三层RNN的运算后将输出通过全连接神经网络输出，由于时进行异常数据与正常数据的

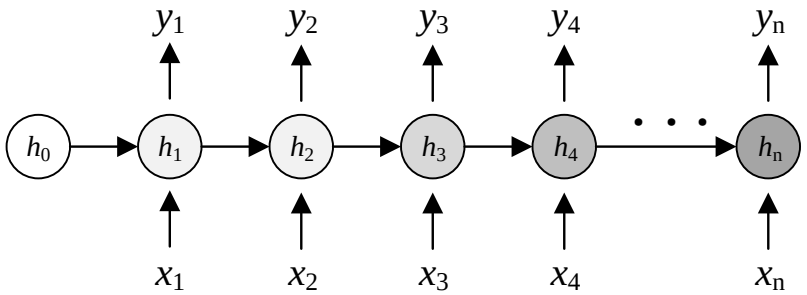


图4　循环神经网络示意图

分类因此选用sigmoid函数作为分类器的运算函数输出分类结果。

4.3　神经网络模型的评估

模型的评估标准为常用的三个指标[14]精度（Accuracy）用ACC代表，主要评价异常检测的整体效果；查准率（Precision）用PRE代表，查全率（Recall）用REC代表，主要评价对异常数据的检测效果。计算公式如下，其中，TP为真正例、TN为真负例、FP为假正例、FN为假负例，具体含义的的描述可以从表2的混淆矩阵中得出。

表2　混淆矩阵的定义

	实际类：X	实际类：Not X
预测类：X	TP	FP
预测类：Not X	FN	TN

$$accuracy,ACC = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

$$precision,PRE = \frac{TP}{TP + FP} \tag{2}$$

$$recall,REC = \frac{TP}{TP + FN} \tag{3}$$

数据集的40%为测试集如表1所示，将测试集用于本文两种神经网络模型1D-CNN和RNN的测试，测试结果如图5所示。对于ACC准确率，1D-CNN模型的准确率为96.328%，RNN模型的准确率为97.754%，表明对异常检测分类的识别效果循环神经网络（RNN）模型较好。对于异常数据PRE查准率和REC查全率，1D-CNN模型的PRE=95.783%，REC=97.532，RNN模型的PRE=95.368%，REC= 96.431%，表明1D-CNN模型对异常流量的识别效果更好。

由于拟态防御系统在进行判决时大多数情况下输出为正常流量，因此在多个执行体输出时对某一个正常流量异常检测时识别错误基本不会影响判决结果。所以，在最小异常值判决时本文更加关心对异常数据流量的检测率，综合两个模型的三方面指标，1D-CNN模型更加适合最小异常值判决时使用。

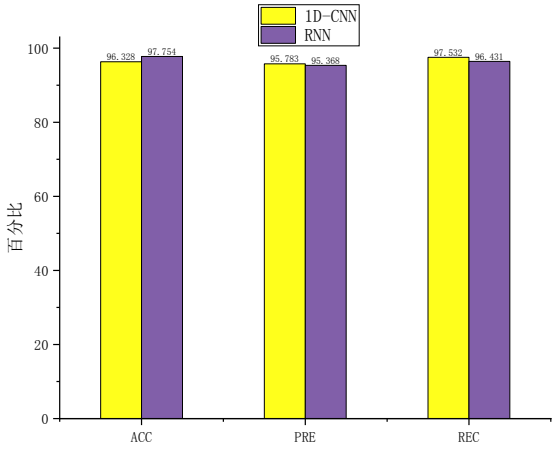


图5　神经网络模型1D-CNN和RNN的评价

5　实验评估与分析

本文研究的最小异常判决方法是基于拟态防御系统中各异构执行体输出结果进行异常检测，主要目的是简化拟态判决时的数据归一化过程，解决语义相同语法不同的问题，并且在一定程度上提高现有多数表决的输出正确率。

5.1　仿真数据集的构建

最小异常值判决是把所有在线异构执行体输出结果分别进行异常检测，因此需要构建在线执行体输出结果数据，且要考虑数据中存在语义相同语法不同的情况。本文假设在线执行体数n=5，选取ISCX-2012数据集的40%测试集中的部分数据用于仿真数据集的构建。如表1所示，测试集中异常流量共16642个数据流，假设仿真测试集中异常流量占比为15%，因此仿真数据集中每个执行体

输出数据为110946个数据流，构建仿真数据集的过程主要分为两部分，如图6所示。

步骤一：从测试集中随机选取110946个正常数据流用于仿真数据集构建，将正常数据复制五份作为5个执行体的输出数据。由于执行体输出数据存在语义相同语法不同的情形，本文假设同一执行周期所有执行体输出正常数据都存在语义相同语法不同。使用数据包解析的方法[15]对每各执行体的输出数据流进行处理，主要处理数据包的自定义字段、保留字段或可选字段，另外改变每一个数据流中数据包的顺序，使得同一数据流的五份数据语义相同但语法不同。将构建好的只有正常数据流量的数据集命名为**仿真数据集I**，用于仿真实验。

步骤二：假设5个异构执行体被攻击的概率相同，即输出异常数据的占比都为15%。采用均匀分布[16]的方法将每个执行体输出正常数据的15%随机替换为测试集中的异常数据。将构建好的仿真数据集命名为**仿真数据集II**。

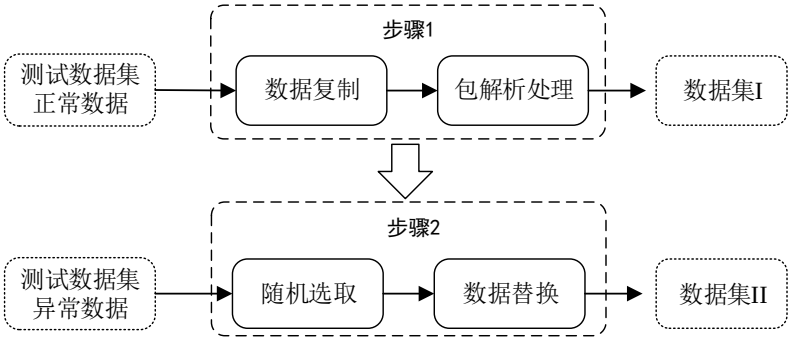


图6仿真数据集构建过程

仿真数据集如表3所示，共有五个执行体的输出数据，每个执行体输出数据中都包含Normal，BFSSH，HttpDoS，DDoS和Infiltrating五种数据流量且各流量占比相同。

表3　仿真数据集的数据内容统计

类别	执行体1	执行体2	执行体3	执行体4	执行体5	占比
Normal	94304	94304	94304	94304	94304	85.01%
BFSSH	2785	2785	2785	2785	2785	2.51%
HttpDoS	1392	1392	1392	1392	1392	1.25%
DDoS	8448	8448	8448	8448	8448	7.61%
Infiltrating	4017	4017	4017	4017	4017	3.62%
合计	110946	110946	110946	110946	110946	100%

5.2　实验仿真和结果分析

5.2.1　对语义相同语法不同的实验仿真

针对异构执行体输出数据存在语义相同语法不同的情形，以往拟态判决方法需要归一化处理，本文方法是将数据进行异常检测来简化数据归一化的过程。实验仿真首先验证异常检测模块对语义相同语法不同问题的归一化处理效果，也就是对**仿真数据集I**的同一执行周期输出的语义相同数据进行异常检测，查看输出异常值是否相同。

评价标准为，**误检率（False detection rate，FDR）**，拟态判决实验中出现对语义相同语法不同的正常数据异常检测，输出异常值大于0.5的概率。仿真数据集I中共有110946组语义相同语法不同的正常数据，统计对每一组数据异常检测输出值大于0.5的个数。仿真实验使用训练好的1D-CNN和RNN两个异常检测模型，实验一选用五个执行体输出数据都用1D-CNN异常检测，实验二选用五个执行体输出数据都用RNN异常检测，仿真结果如图7所示。

在实验一（1D-CNN）的110946次仿真实验中，语义相同语法不同的正常数量流量，误检数为1的次数为2843，误检数为2的次数为2718，误检数为3的次数为2432，误检数为4的次数为214，误检数为5的次数为30。总的误检数据流个数为16581，误检率为2.989%。在实验二（RNN）的110946次仿真实验中，语义相同语法不同的正常数量流量，误检数为1的次数为1864，误检数为2的次数为1568，误检数为3的次数为2182，误检数为4的次数为101，误检数为5的次数为5。总的误检数据流个数为11975，误检率为2.158%。通过对比实验结果表明，RNN模型的异常检测对正常

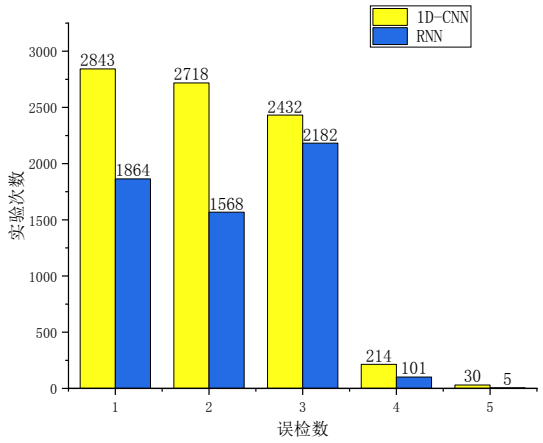


图7　误检次数仿真结果图

流量的识别效果更好总的误检率只有2.158%，能够满足实际应用需求。此外，避免了以往拟态判决数据归一化过程，简化了判决过程。

5.2.2　最小异常值判决的实验仿真

将仿真数据集II进行最小异常值判决的实验仿真，主要是验证本文所用方法对拟态判决正确率的提升，选择与多数表决算法做对比实验。如图8所示，判决算法评价标准为：

1）正确结果被拟态判决通过的概率（CAA，Correctness And Agreement）；

2）正确结果没有被拟态判决通过的概率（CAF，Correctness And Failure）；

3）错误结果被拟态判决通过的概率（IAA，Incorrectness And Agreement）；

4）错误结果没有被拟态判决通过的概率（IAF，Incorrectness And Failure）；

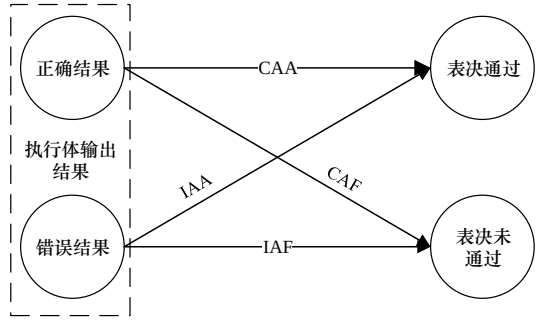


图8　判决算法评价指标

仿真数据集II中正常流量占比85%，异常流量占比15%，且同一执行周期输出数据为$\{d_1, d_2, d_3, d_4, d_5\}$。实验一：多数表决算法（Majority Voting Algorithm，MVA）[3]，正确结果被拟态判决通过（CAA）的比例为93.0721%，正确结果没有被拟态判决通过（CAF）的为6.9273%，错误结果被拟态判决通过（IAA）的比例为25.9862%，错误结果没有被拟态判决通过（IAF）的比例为74.0138%。实验二：基于1D-CNN的最小异常值表算法，正确结果被拟态判决通过（CAA）的比例为97.187%，正确结果没有被拟态判决通过（CAF）的为2.813%，错误结果被拟态判决通过（IAA）的比例为5.753%，错误结果没有被拟态判决通过（IAF）的比例为94.247%。实验三：基于RNN的最小异常值表算法，正确结果被拟态判决通过（CAA）的比例为96.594%，正确结果没有被拟态判决通过（CAF）的为3.406%，错误结果被拟态判决通过（IAA）的比例为9.634%，错误结果没有被拟态判决通过（IAF）的比例为90.366%。实验结果如图9所示。

实验结果表明，本文两种异常检测模型（1D-CNN和RNN）的四项评价指标均优于多数表决算法，其中错误结果被拟态判决通过（IAA）的比例降低了78%，错误结果没有被拟态判决通过（IAF）的比例提升了27.4%。1D-CNN和RNN的最小异常值判决结果对比实验表明1D-CNN异常检测模型的CAA、IAF指标优于RNN异常检测模型，1D-CNN异常检测模型更加适合用于拟态判决器的最小异常判决算法。

6　结束语

为简化拟态判决方法避免以往判决过程中的数据归一化过程提高判决效率，本文提出了最小异常值判决算法，训练了1D-CNN和RNN两个异常检测模型，用最小的数据异常值来判决最终输出结果。构建了基于拟态防御架构的仿真数据集，验证了本文所用方法的有效性和可靠性，通过与多数表决算法的对比实验验证了本文算法能够提升拟态判决的安全性。

由于目前还没有关于拟态判决相关的较为全面的数据集，本文在数据集构建和仿真阶段都是基于网络异常检测常用数据集ISCX-2012做的改进，因此实验结果存在一定的局限性。此外，本文所提出的方法比较依赖于异常检测模型的可靠程度。计划在下一步的工作中研究性能更好的异

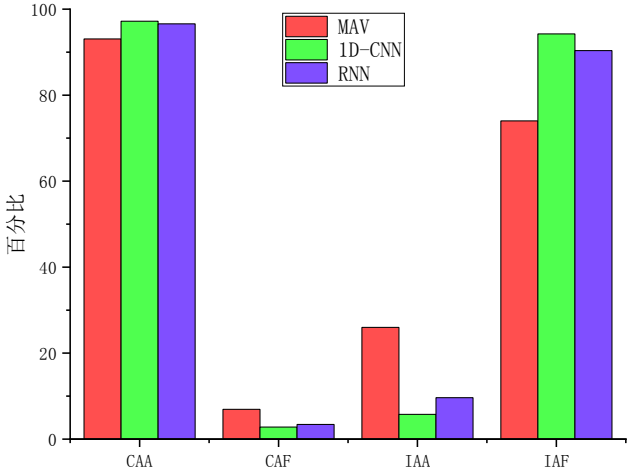


图9　本文算法与多数表决法对比实验

常检测模型，构建较为全面的拟态数据集用于实验仿真。

参考文献:

[1] 邬江兴. 网络空间拟态防御原理: 上册[M]. 2 版. 北京:科学出版社，2018:1-3,185-186

[2] 邬江兴. 网络空间拟态防御研究[J]. 信息安全学报,2016,1(4):1-10.

[3] 张文建, 宋克, 谭力波, et al. 面向拟态判决的可编程语义解析方法[J]. 通信学报，2020，041(004):62-69.

[4] 欧阳城添,王曦,郑剑． 自适应一致表决算法[J]. 计算机科学，2011，38(7)：130-133.

[5] 党小超. 基于碰撞识别的优先级仲裁策略[J]. 计算机工程与应用，2012,48(27)：74-79.

[6] 李卫超, 张铮, 王立群, 等. 基于拟态防御架构的多余度裁决建模与风险分析[J]. 信息安全学报，2018，3(5)：68-78.

[7] 马若龙. 基于卷积神经网络的未知和加密流量识别的研究与实现[D]. 2018.

[8] Bhuyan M H , Bhattacharyya D K , Kalita J K . Network Anomaly Detection: Methods, Systems and Tools[J]. IEEE Communications Surveys & Tutorials，2014，16(1):303-336.

[9] Dainotti A , Pescape A , Claffy K C . Issues and future directions in traffic classification[J]. Network，IEEE，2012，26(1):p. 35-40.

[10] Canadian Institute for Cybersecurity, Intrusion detection evaluation dataset (ISCXIDS2012)［EB/OL］(2011. 12)［2020-8-21］https://www. unb. ca/cic/datasets/ids. html

[11] ShiraviAli, ShiraviHadi, TavallaeeMahbod, Toward developing a systematic approach to generate benchmark datasets for intrusion detection, Computers & Security, Volume 31，Issue 3, May 2012, Pages 357-374, ISSN 0167-4048, 10. 1016/j. cose. 2011. 12. 012

[12] Lecun Y , Bottou L . Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE，1998，86(11)：P. 2278-2324.

[13] Wei Wang, Xuewen Zeng, Xiaozhou Ye, Yiqiang Sheng and Ming Zhu，"Malware Traffic Classification Using Convolutional Neural Networks for Representation Learning，" in the 31st International Conference on Information Networking (ICOIN 2017)，pp. 712-717，2017.

[14] Sharafaldin I , Lashkari A H , Ghorbani A A . Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization[C]// 4th International Conference on Information Systems Security and Privacy. 2018.

[15] microsoft-message-analyzer-operating-guide［EB/OL］(2016. . 12)［2020-8-21］https://docs. microsoft. com/ en-us/message-analyzer/microsoft-message-analyzer-operating-guide

[16] 武兆琪,张帆,郭威,卫今,谢光伟. 一种基于执行体异构度的拟态裁决优化方法[J/OL]. 计算机工程：1-8. http://kns. cnki. net/kcms/detail/31. 1289. T P. 20191016. 1009. 002. html.

[作者简介]

谭力波（1981– ），男，天津市滨海新区信息技术创新中心高级工程师，主要研究方向为网络空间拟态防御系统设计、交换结构设计及计算机结构设计。

贾广瑞（1995—），男，硕士，主要研究方向为拟态防御原理，网络安全，深度学习流量异常检测。

张文建（1987– ），男，信息工程大学博士生、助理研究员，主要研究方向为网络空间安全、网络可编程设计、集成电路设计。

动态异构赋能拟态防御架构安全性量化方法

赵玉风[1，2]，张铮[1，2]，季新生[1]

[1]信息工程大学 郑州 450001；

[2]数学工程与先进计算国家重点实验室（信息工程大学）郑州 450001

摘　要：动态异构冗余机制可以有效提高系统安全性，但其动态异构冗余组件实际增加了系统攻击面以及可被攻击者利用资源的数量，而移动攻击面模型并不能给出合理的解释。针对移动攻击面在动态异构冗余机制下的局限性，一种动态异构赋能的探测面度量方法从攻击者的准备出发，对系统的安全性进行分析。首先对异构加持下冗余系统的探测面做出分析，其次量化动态性引发的探测面改变。接着在此基础上构建安全度量模型对拟态防御架构的安全性进行量化分析。最后实验验证这种安全量化方法可以合理地解释动态异构冗余特性对安全性的贡献来源。

关键词：拟态防御架构、安全量化、动态分析、异构冗余

Security Quantification Method for Mimic Defense Architecture powered by Dynamic Heterogeneity

Zhao Yufeng[1,2], Zhang zheng[1,2], Ji sheng[1]

1.Information Engineering University, Zhengzhou 450001;

2.State Key Laboratory of Mathematical Engineering and Advanced Computing (Information Engineering University), Zhengzhou 450001

Abstract: The mechanism of dynamic heterogeneous redundancy could effectively improve the security of the system, but the components of dynamic heterogeneous redundancy actually increase the number of the attack surface of the system and the resources that can be exploited by the attacker, but the shift attack surface model could not give a reasonable explanation. Aiming at the limitation of shift attack surface under dynamic heterogeneous redundancy mechanism, a dynamic enabling detection surface measurement method is proposed to analyze the security of the system from the perspective of the preparation of the attacker. Firstly, the probe surface of redundant system under heterogeneous loading is analyzed, and secondly, the probe surface change caused by dynamics is quantified. Then a security measurement model is built to quantify the security of the mimic defense architecture. Finally, experiments show that this security quantification method can reasonably explain the contribution of dynamic heterogeneous redundancy to security.

Key words: mimic defense architecture; security quantification; dynamic analysis; heterogeneous redundancy

1　引言

网络系统的静态性是攻防不对称的原因[1]，打破这种不对称现状，保护系统的安全性的一个重要举措就是系统动态变化机制的引入。系统安全防护中动态性的典型应用，主要集中在移动目标防御[2]和拟态安全防御[3]。

移动目标防御技术的安全性分析采用攻击面技术以及移动攻击面技术，通过对系统动态改变前后的方法、数据和通道等相关资源的改变进行分析[4]，以得到系统的安全可靠性。而拟态防御技术以异构冗余动态机制为基础，这种机制的安全性并不能由移动攻击表面理论给出。系统采用动态可重构功能等价的冗余资源，在这种情况下

基金项目：本课题得到国家自然科学基金项目（61521003），国家重点研发计划（Grant No.2018YF0804003, and Grant No. 2017YFB0803204）以及网络通信与安全紫金山实验室资助。

通讯作者：张铮（ponyzhang@163.com）

系统原有的方法、数据和通道并未发生改变，而整体的攻击面却增大了。但是工程实践与理论证明，异构冗余架构对于基于未知漏洞后门等的攻击具有十分显著的防御效果，这与移动攻击表面理论“缩小攻击表面有利于安全性”[5][6]的说法相悖。说明移动攻击表面理论不能刻画此类环境下的网络攻击，需要引入新的描述方法。

本文从攻击者的角度出发对这种现象做出解释，攻击者在进行攻击之前首先要对被攻击对象的环境和基本情况进行分析，这个过程就是扫描探测的过程，冗余系统显著增加了攻击者在攻击之前的准备，即需要更多的努力去探测环境，因此本文从这个角度入手，对动态异构冗余机制进行安全分析。首先对异构加持下冗余系统的探测面做出分析，其次量化动态性引发的探测面改变。由于资源的变化可能会引起风险的增加，所以在构建安全度量模型时将资源改变的风险增益也考虑在内。最后通过实验验证这种安全量化方法，解释动态异构冗余特性对安全性的贡献来源，并通过现网测试对安全性进行进一步的验证。

主要贡献：

• 本文提出一种动态冗余系统安全度量的新思路，从攻击者攻击准备的角度给出系统安全性的考虑。

• 本文的安全度量模型验证了冗余系统安全性的贡献来源，解决了移动攻击表面对冗余机制的局限性。

• 本文对拟态防御架构的安全性进行了详细分析，不仅从理论层面验证了拟态防御架构的安全性，还结合现网测试佐证拟态防御架构的安全性。

第2节给出静态和动态探测面的基本定义，第3节构件拟态防御安全度量模型，实验验证部分主要分为两部分，模型分析以及现网验证（第4节），最后给出总结。

2 探测面

攻击者在进行攻击之前首先要进行扫描探测，获得可利用的漏洞或者易被攻击的资源，这是攻击进行的前提和基础。

定义1

探测面

探测面是在攻击者发动攻击时对攻击目标的信息探测范围，被攻击对象S的探测面用T_S表示。

相较非拟态系统，攻击者对拟态系统进行攻击时需要付出更多的努力，首先扫描的对象成倍增加，对于非异构冗余而言，还可能存在攻击迁移的情况，但是异构冗余进一步打破了攻击者的设想，异构组件之间相同的漏洞数量很少，攻击迁移的情况弱化，攻击者甚至需要扫描所有上线的冗余组件。其次，攻击中还会有持续保留攻击的现象，由于系统维护的复杂性和耗时性，系统部署之后往往会保持很长一段时间[1]，一旦攻击出现，利于攻击的环境就会维持很久，使得系统长时间暴露在危险之中。动态性使得系统不断变换其软硬件配置，即便攻击成功，攻击者所依赖的环境也会被会及时清洗，迫使攻击者重新进行扫描。因此，拟态系统的安全性分析从两方面进行，静态探测面和动态探测面。

2.1 静态探测面

攻击者在攻击非拟态或者单系统时，只需要扫描系统本身的组件就可以，但是对于拟态系统而言，由于同时运行多个等价组件并由这些组件得到的结果共同决定输出结果，因此需要扫描所有上线的组件。静态探测面用熵的思想进行定义，冗余强度越大，系统随机性越强，熵越大，故探测面越。非冗余系统的静态探测面为0，但这并不意味着非冗余系统没有探测面，只是没有额外的探测面。

定义2

静态探测面

静态探测面指冗余系统对探测面的增益效果，冗余使得系统整体探测面增大，但是增大的程度并非最佳，因为冗余又分相似冗余和非相似冗余（即，异构冗余）。相似冗余之下，虽然探测面增加，但是攻击者还是可以利用相似漏洞进行迁移从而攻击系统。而异构冗余相似漏洞出现的概率很小，甚至为0，此时攻击者被迫对每一个异构冗余组件进行扫描，因此需要付出更大的努力。故此，冗余系统中冗余组件的随机性越大，静态探测面越大。

度量随机性的指标，最常用的就是信息熵，熵即不确定性的量度，系统越随机，其熵越大[7]。

熵的极值原则：

所有可以被探测到的组件同等概率时熵达到最大值。

拟态防御架构的探测面主要集中在上线执行体中，拟态防御架构的执行体集用 $A=\{A_1,A_2,\cdots,A_n\}$ 表示。静态探测面度量用 $T(A)$ 表示，某个执行体的静态探测面为 $T(A_i)$。

$$T(A)=H(A)=\sum_{A_i\in A}p(A_i)\log_{10}(1/p(A_i)) \quad (1)$$

根据熵的极值原则由于拟态防御架构的上线执行体是功能等价，即被攻击者探测到以及被攻击者利用的概率相同，因此静态探测面度量满足 $T(A)=\sum_{A_i\in A}T(A_i)\leqslant \log_{10}(1/p(A_i))$。

2.2 动态探测面

定义3

动态探测面

静态探测面度量用 $T(A)$ 表示，此处的探测面增益是异构冗余引起的。然而，动态也会对探测面产生作用，攻击面的改变作为动态性对探测面影响的因素，用 $\Delta T=C(\Delta AS)$ 表示。

其中，动态探测面用 ΔT 表示，主要由于可被攻击资源的改变引起，只要有资源的改变，探测扫描机制就需要重新运行，以寻找新的可利用漏洞。动态探测面用资源改变数量来定义，当在线执行体发生清洗的时候，在线执行体被备份执行体替换。对于拟态防御架构而言，由于执行体之间的异构性，只要发生执行体的替换清洗，就会发生资源的改变。而其余动态变化的防御机制或者系统架构，有的动态变化也许并不会引起资源的改变，此时需要具体情况具体分析。因此，动态探测面用资源变化的情况进行分析。

系统中可被探测的资源变化的数量用 C 表示，对于多个冗余执行体 A_j 有 $\Delta T=C(\Delta AS)=\sum_j\Delta AS(A_j)$，对于某个执行体 A_j 其资源变化情况为 $\Delta AS(A_j)$，每个执行体都是由多个层级组成的，每个层级都可能会发生变化，所以某执行体的资源变化由该执行体所有层级资源变化情况决定 $\Delta AS(A_j)=\sum_i^m(AS^j{}_{i\_after}-AS^j{}_{i\_before})$。

3 拟态防御安全性度量模型

根据动静探测面分析，结合移动攻击面度量中的资源风险评价本节给出拟态防御架构的安全度量模型，该模型适合所有动态异构的系统。

拟态防御的安全性度量 MS 用探测面度量和资源风险增益来决定。

$$MS=T(A)+\Delta T-I(\Delta ASM) \quad (2)$$

探测面改变会对攻击者的攻击产生阻力，攻击者需要付出更多的努力才能发起攻击，但是由于每一种资源的风险不同，存在一种现象，当攻击面改变将低风险的资源改变为高风险的资源时，虽然攻击者付出的额外的努力，但是会得到较好的收益，因此在安全度量模型中需要将资源的风险考虑进去。在（2）中，$T(A)$ 为静态探测面度量，即异构冗余带来的安全增益，ΔT 为动态探测面度量，即动态冗余为系统安全性所做的贡献，$I(\Delta ASM)$ 为资源风险增益，如果资源风险增加则 $I(\Delta ASM)>0$，若资源风险不变有，$I(\Delta ASM)=0$，如果资源风险降低，则 $I(\Delta ASM)<0$。

对于N冗余系统，每个冗余组件的资源层级为M，则该系统的资源风险增益为 $I(\Delta ASM)=\sum_i^n\sum_j^m\Delta ASM_j^i$。$\Delta ASM_j^i$ 表示第 i 个执行体第 j 层资源的变化情况。

4 实验验证

本文研究动态赋能的安全性量化方法以拟态防御架构为研究对象，其中拟态防御架构[8]中上线执行体以三冗余[9][10]执行体图1为例进行研究。

上线执行体三台，即系统冗余度为3，如果上线执行体出现被攻击的情况被表决器监测到，就通过调度器用异构组件集中的执行体进行替换，以保证上线执行体的安全性以及输出结果的正确性。

本节从安全度量模型和白盒安全测试两个方面进行拟态防御架构的安全性分析。首先是在4.1节中对本文提出的动态异构赋能安全量化方法进行验证。

4.1 安全度量模型分析验证

本文的安全度量模型主要用到三个参数，静态探测面度量、动态探测面度量以及资源风险增益度量，通过对三个参数的分析最终得到安全模型的分析。

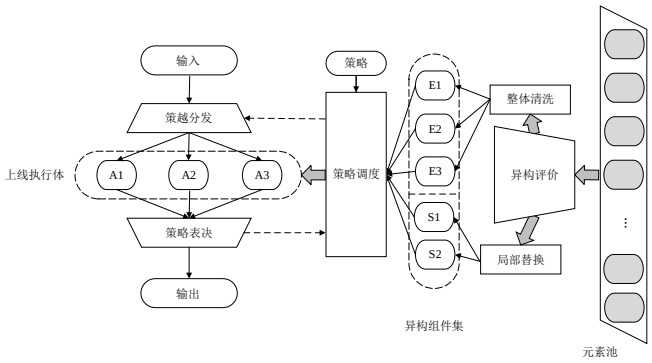


图 1　拟态防御架构工程图

首先是静态探测面度量分析，三冗余执行体的静态探测面度量情况如图 2 所示。鉴于静态探测面度量基于熵理论进行分析，本文给出三冗余系统的静态探测面度量情况。

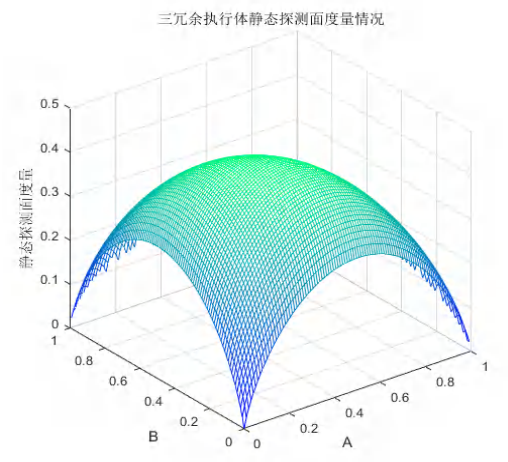


图 2　三冗余执行体系统的探测面度量情况

$$T(A)=H(A)=\sum_{i}^{3}p(A_i)\log_{10}(1/p(A_i))\leqslant$$

$\log_{10}(3)$，因此 3 冗余系统的静态探测面最大为 $\log_{10}(3)$，而只有每个执行体被攻击的几率相同时才可以保证系统可以取到最大的探测面值。

条件一、拟态防御架构的上线执行体之间功能等价，并且互相独立，不仅同时接受数据，最终结果也取决于所有结果的表决，因此其受到攻击的几率是相同的。

条件二、拟态防御架构中的上线执行体之间异构冗余，保证冗余的基础上差异性的最大化，因此与相似冗余相比随机性更大，相应的熵值更大。

根据以上两个条件，可知拟态防御架构可以达到静态探测面的最大值，给出安全性的最大贡献。

其次是动态探测面的度量，动态探测面度量依赖于资源的改变数量，变化的拟态执行体层级主要为软件层的资源变动，如图 3 所示。数据库和应用层进行随机标签处理，可以保证每一个执行体的这两层资源都是不同，因此只要执行体进行更换，就满足 $C(\Delta AS)\geqslant 2$。如果上线执行体进行整体替换即需要改变三个执行体，那么 $C(\Delta AS)\geqslant 6$。而虚拟机操作系统层和服务器软件层则根据具体替换情况进行分析。

综上分析可知，动态探测面 $C(\Delta AS)\geqslant 0$，即动态冗余对系统安全性的增益为积极正面的，故可以增加攻击者的攻击负担，从而提高系统安全性。

第三个参数资源风险度量根据 CVSS 漏洞评分系统 [11] 的情况（表 1）进行赋值，通过分析不同资源的漏洞占比 [12]，给出每种资源的平均风险值，并根据资源的变动，计算风险值的变化情况。如果风险增加，则利用攻击者，如果风险减少或者不变则对系统安全有利。

拟态防御架构中应用层和数据层都进行了随机标签化处理，本文中认为其风险值是相同的，所以即便执行体改变引起这两层的资源发生变化，风险值也不会有所改变。因此，攻击面改变引起的资源风险变化集中在服务层和操作系统层（表 2），不同操作系统的风险不同，同样不同服务层软件的风险也不相同，如果使用低风险替换高风险执行体的比例比较高或者用软件风险差别不大的执行体进行替换，那么安全性就可以得到保障。

执行体动态改变引起攻击面资源改变，执行

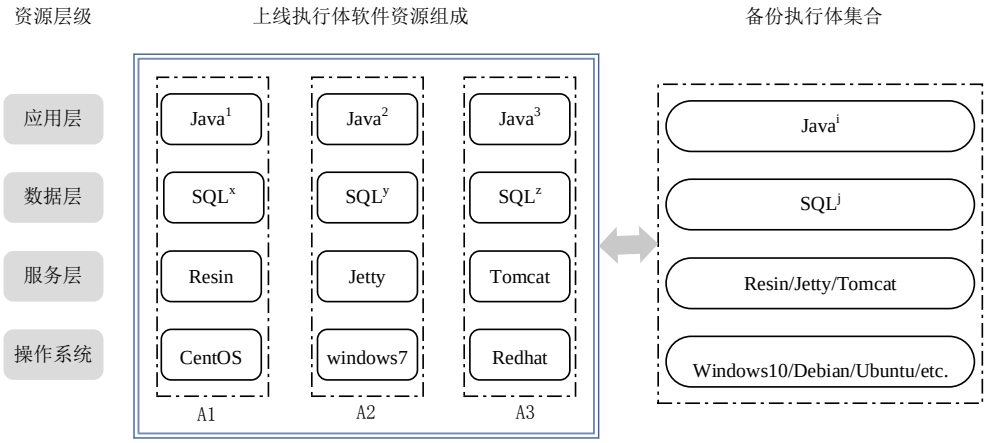


图 3　拟态防御架构执行体资源层级图

表 1　CVSS 评分规则

RATING	CVSS SCORE
None	0.0
Low	0.1 – 3.9
Medium	4.0 – 6.9
High	7.0 – 8.9
Critical	9.0 – 10.0

表 2　资源风险值

资源层级	软件类别	CVSS 风险评分
服务层	Resin	5.3
	Jetty	5.3
	Tomcat	5.2
操作系统	Windows XP	7.9
	CentOS	7
	Windows7	7
	Red hat	6.5
	Windows10	6.5
	Debian	6.3
	Ubuntu	6.2

体替换有两种形式，局部清洗替换某一个，整体替换将上线执行体整体清洗。在这两种情况下，资源风险增益情况如图 4 所示。

整体清洗时，低风险占比 99%，局部替换时，资源引起的风险增量如果按照 CVSS 等级分析，低风险占比 100%。因此，即便资源动态改变，资源的风险降低还是占绝对优势的。

综上对安全度量进行分析，局部替换的情况下 $MS > 0.6$，整体清洗的情况下 $MS > 1.4$。在实际操作中，整体清洗攻击者需要对所有上线环境进行重新分析，而局部替换攻击者只需要对部分环境进行分析，所以安全度量满足实际情况。由此，动态异构赋能的拟态防御架构对安全性有一定的增益。

4.2　安全方法与程度对比

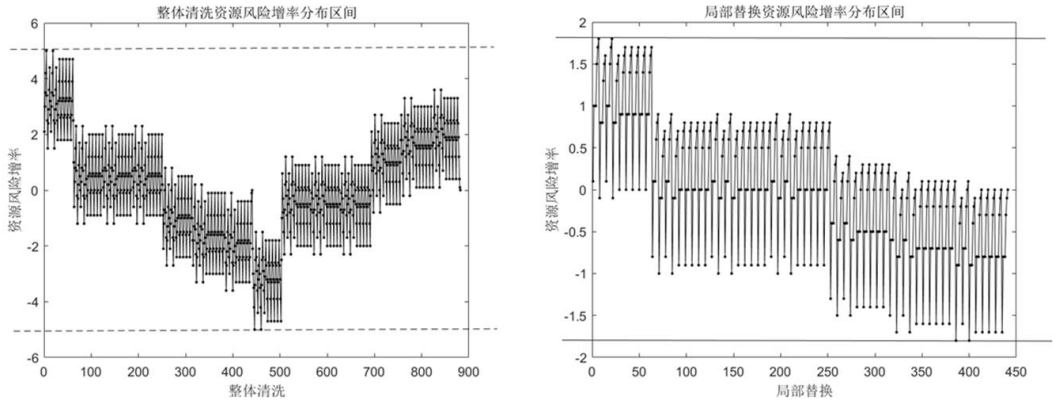
本文提出的方法与已有度量方法相比主要优势有二：

其一，与之前研究中的资源权值分析方法相比，本文研究中的资源风险权值由组件本身的漏洞风险平均值决定，更具客观性。

其二，从攻击者攻击所付出的努力出发对动态异构机制的安全性进行分析的探测面度量方法，较移动攻击面度量更能反映拟态防御架构的安全性所在。首先移动攻击面度量方法主要是从不同的角度对系统的安全性进行分析，提出一种在某种情况下更适合的移动方式；而探测面度量方法直接将攻击付出的努力进行量化，付出努力越多，系统安全性越高。其次，移动攻击面度量方法是主要依赖于资源权值的变化，只要资源权值发生了改变，系统的风险就随之发生变化。而实际情况中，资源权值的变化并不与系统风险成比例，因为系统的安全并不仅仅由系统暴露的可被攻击的资源决定，还关系到攻击者捕获这些资源的历程，资源的变动会带来系统的风险，但同时给攻击者带来负担。

在静态探测面对比中用一个简单的案例将未拟态普通架构、简单冗余、以及拟态动态异构冗余的静态探测面度量进行分析，如图 6 所示。

未进行拟态的架构探测面不会发生增益，因此对系统安全的保障没有拟态架构深刻。探测面度量越小，攻击者需要付出的额外努力越少，在



(a) (b)

图 4 资源风险增益分布图

根据资源风险情况得到动态情况下资源风险增益分布情况，整体清洗时 $I \in [-5, 5]$（图 4 （a））， 局部替换时 $I \in [-1.8, 1.8]$（图 4 （b））。根据 CVSS 评分系统，即便在资源风险增率最大的情况下，即系统动态变化引起高风险的资源替换低风险的资源的情况，也只是中等危险的程度，并且无论是整体清洗还是局部替换，99% 以上的情况，资源风险增益都是处于低风险区段（图 5）。

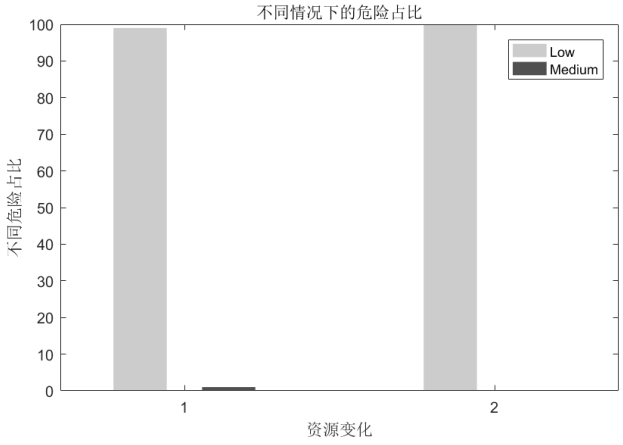


图 5 不同动态情况下的风险占比

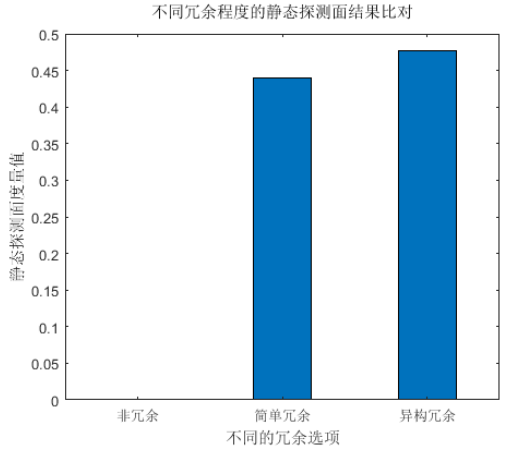


图 6 静态探测面对比

探测面度量为 0 的情况下攻击者只需要进行常规攻击扫描就可以不需要付出额外的精力和时间去应对探测面的增加。简单冗余比如入侵容忍系统的应用，在这种情况下，每个执行体暴露的程度不同，因此，对攻击者而言，探测面会增加，但是增加的情况并不是最好的。

因此，拟态防御架构的探测面度量比非拟态架构，甚至简单冗余架构的探测面度量都要大，攻击者对进行拟态防御架构设计的网络设备进行攻击时，需要付出更多的时间和精力。这是从异构冗余的角度进行分析，由于系统的动态变化，使得探测面在运行时还会进一步变得复杂。

动态探测面度量主要与移动目标防御技术比对，移动目标防御技术执着与系统配置的动态改变，但是并没有保证动态性的最大化，而异构加持下的动态变化，每次都保证动态前后组件差异的最大化。而动态探测面相比移动攻击面度量更

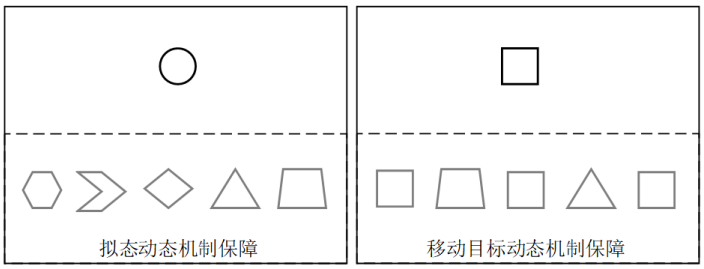


图 7 动态机制差异

好地体现了这一点。用资源变化的数量对动态性进行程度分析。

由于拟态组件的异构性，使得资源变化数量能保证大于0，而移动目标防御技术并不能给出此项保证。如图 7所示，拟态动态机制可以保证替换元素之间差异的最大化，以及同类替换的最小化，而移动目标防御技术并不能保证这一目标。

综上，探测面度量从冗余异构和动态异构两个角度对拟态防御技术的安全性进行分析，相比已有的入侵容忍技术以及移动目标防御技术面，拟态防御技术是安全性程度更高。同时相比已有的度量方法，探测面度量针对性更强，更能体现拟态防御架构的安全增益。

4.3 白盒测试安全性分析

通过白盒渗透测试对系统的稳定和执行体的安全性进行验证，进一步佐证动态赋能拟态防御架构的安全性。实验将测试程序部署于异构冗余执行体中，通过“白盒插桩”注入测试例的方式，评估拟态防御架构的安全性。本节中的测试主要关注对执行体注入危险代码之后，执行体的自清洗功能是否可以有效避免危险出现，实际组网部署的简易图如图 8所示。

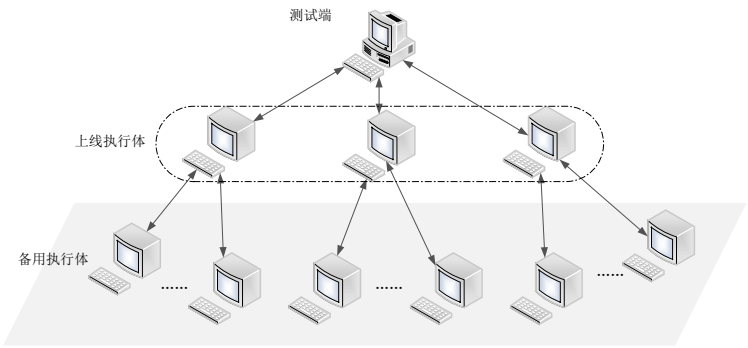


图 8 实网部署简易架构

实验方案如表 3所示，首先测试拟态防御表决功能验证，当对某个执行体的内容进行篡改时，判断表决器的最终输出结果。其次是动态反馈功能，除了发现执行体问题时的局部替换和整体清洗，拟态防御架构还有定时清洗的功能，当开启动态反馈功能时，如果执行体出现异常，可以在一定时间内将执行体恢复正常。最后是差模共模攻击验证，差模共模攻击验证与动态反馈功能验证类似，都是直接给上线执行体注入攻击，验证是否可以恢复正常。

通过测试验证，拟态防御架构具有功能完备性和安全可靠性。

5 结束语

本文提出一种安全分析方法用以分析动态赋能的拟态防御架构，安全模型基于攻击前的探测准备度量以及资源风险评测。分别对静态探测面（异构冗余为探测面所做的贡献），动态探测面（动态冗余对探测面的贡献）以及系统动态变换带来的资源风险增益进行分析，最终得到安全度量的分析。模型验证中得到拟态防御架构安全性的贡献来源以及安全模型的正确性验证。通过对拟态防御架构安全模型的分析以及白盒注入测试的验证，给出拟态防御架构是安全可靠的。

表 3　测试方案

测试项	测试方案	测试情况	测试结果
表决功能	对某执行体进行篡改，观察表决器结果	篡改一个执行体，表决器正常执行体，篡改多个执行体是表决器监测异常结果	通过
动态反馈功能	对某执行体进行篡改，观察系统自动修复功能	系统可以在一定时间内恢复正常	通过
差模攻击	对某一个上线执行体进行篡改，观察其修复情况	一段时间之后可以保证上线执行体正常	通过
N-1模攻击	对某两个上线执行体进行篡改，观察其修复情况	至多出现一次威胁，并且可以成功清洗	通过
共模攻击（N模攻击）	对所有上线执行体进行篡改，观察其修复情况	可以逐步退化为差模攻击，并最终清洗成功	通过

本系统模型证明了拟态防御架构的安全性以及其安全性的来源，能够满足动态系统的安全分析，不仅对拟态防御架构的动态安全性有效，同时也适用于所有动态系统的安全度量分析。

参考文献：

[1] Zhuang R, DeLoach S A, Ou X. Towards a Theory of Moving Target Defense [C]//Proceedings of the First ACM Workshop on Moving Target Defense. ACM，2014：31-40.

[2] Zhou Yuyang, Cheng Guang, Guo Chunsheng, Dai M. Survey on attack surface dynamic transfer technology based on moving target defense. Journal of Software，2018，29(9)：2799-2820 (in Chinese).
周余阳, 程光, 郭春生, 等. 移动目标防御的攻击面动态转移技术研究综述[J]. Journal of Software，2018，29(9). ）

[3] Wu Jiangxing. Construction of endogenous security in National Information Cyberspace by Mimic Defense Technology [J]. Information and Communication Technology，2019 (6)：2.
邬江兴. 拟态防御技术构建国家信息网络空间内生安全[J]. 信息通信技术，2019 (6)：2. ）

[4] Manadhata P K, Wing J M. An attack surface metric [J]. IEEE Transactions on Software Engineering，2010，37(3)：371-386.

[5] Ghavamnia S, Palit T, Benameur A, et al. Confine: Automated system call policy generation for container attack surface reduction [C]// Proceedings of the International Conference on Research in Attacks，Intrusions，and Defenses (RAID). 2020.

[6] Ghavamnia S, Palit T, Mishra S, et al. Temporal system call specialization for attack surface reduction [C]//Proceedings of the 29th USENIX Security Symposium. 2020.

[7] Shannon C E. A mathematical theory of communication [J]. Bell system technical journal，1948，27(3)：379-423.

[8] WU Jiangxing. Principles of mimic defense in cyberspace: generalized robust control and endogenous security (Volume 1) [M]. Beijing：Science Press，2018.
邬江兴. 网络空间拟态防御原理: 广义鲁棒控制与内生安全. 上册[M]. 科学出版社，2018. ）

[9] Wei Shuai, Yu Hong, and Gu Zeyu, et al. "Architecture of Mimic Security Processor for Industry Control System，" Journal of Cyber Security，2017，2(01)：54-73.
魏帅, 于洪, 顾泽宇, et al. 面向工控领域的拟态安全处理机架构[J]. 信息安全学报，2017，2(01)：54-73. ）

[10] Zhang Jiexin, Pang Jianmin, and Zhang Zheng，"Quantification Method for Heterogeneity on Web Server with Mimic Construction，" Journal of Software，2020，31(02)：564-577.
张杰鑫,庞建民,张铮. 拟态构造的Web服务器异构性量化方法[J]. 软件学报，2020，31(02)：564-577. ）

[11] The Common Vulnerability Scoring System (CVSS). https：//ixyzero. com/blog/archives/3368. html.

[12] CVE security vulnerability database. https：//www. cvedetails. com/top-50-product-cvssscore-distribution. php.

[13] Xiong Xinli, Zhao Guangsheng, Xu Weiguang, LI Bo. System attack surface based MTD effectiveness assessment model. Journal of Tsinghua University (Science and Technology)，2019，59（4）：276-283.
熊鑫立, 赵光胜, 徐伟光, 李渤. 基于系统攻击面的动态目标防御有效性评估方法. 清华大学学报(自然科学版)，2019，59(4)：276-283. ）

[14] Zhang Z, Wang LQ, LI WC. Research on formal model for an information system's attack surface with dissimilar redundant architecture [J]. Journal on Communications，2018，2.
张铮, 王立群, 李卫超. 面向非相似余度信息系统的攻击面模型[J]. 通信学报，2018，2. ）

[15] Huang K, Yang L, Fu R, et al. HASN: A hierarchical attack surface network for system security analysis [J]. China Communications，2019，16(5)：137-157.

[作者简介]

季新生（1968—），男，教授，博士生导师，主要研究方向为网络安全，新一代移动通信技术。

[作者简介]

赵玉风（1996—），女，数字工程与先进计算国家重点实验室硕士生，主要研究方向为主动防御和网络安全。

张铮（1976—），男，博士，副教授，硕士生导师，主要研究方向主动防御技术和高性能计算。

Distributed Asynchronous Learning forMultipath Data Transmission Based on P-DDQN

Liu Kang[1], Quan Wei[1], Gao Deyuan[1], Yu Chengxiao[1], Liu Mingyuan[1]

1.School of Electronic Information Engineering, Beijing Jiaotong University, Beijing 101416, China;

2.19111028, weiquan, gaody, chengxiaoyu, 18111025}@bjtu.edu.cn

Key words: Distributed asynchronous learning; Multipath data transmission; Deep reinforcement learning; P-DDQN.

Abstract. In wireless networks， adaptive packets scheduling can efficiently handle the high dynamic nature and unpredictability of network data transmission. Therefore， we propose a distributed asynchronous deep reinforcement learning framework for adaptive multipath packet scheduling. In framework， we propose an asynchronous prioritized replay double deep Q-learning packets scheduling algorithm （ADPS）， which mainly uses prioritized replay double deep Q-learning network （P-DDQN） to perceive and analyze the path states， for dynamic adaptive packet scheduling. Detailedly， the framework contains two parts， local asynchronous packet scheduling （Laps） and distributed cooperative control center （DC3）. Each Laps is a learner and they have their own ADPS for packets scheduling. Learners can initiative or timed trigger P-DDQN learning for parameter exchange in DC3. Based on learners' weight changes， DC3 will adaptively updates network parameters so as to accelerate P-DDQN learning for Laps. In experiments， we make performance comparison with Random weight （Random） and Round－Robin （RR） method. ADPS is 2.18% and 1.14% better than Random and RR in throughput， respectively. ADPS outperforms Random and RR by 1.87% and 1.27% in RTT performance， respectively. The packet loss ratio of ADPS is 23.39% and 23.04% reduced than Random and RR， respectively. Further， ADPS has 2.34 and 2.14 times better than Random and RR on the stability of data transmission， respectively.

1 Introduction

In recent years， with the development of the Fifth Generation （5G） cellular network and the Internet-of-Things （IoT）， there are 5.7 billion mobile users by 2023 compared to 5.1 billion in 2018. Simultaneously， it is expected there are 14.7 billion mobile IoT device connections in the Internet by 2023［1］. The rapid growth of devices heralds the explosion of wireless network data. Correspondingly， how to guarantee quality-of-service （QoS） of data transmission has become the focus of research.

Due to the complexity and differences of performance indicators in the wireless networks， especially for data transmission， researchers propose many efficient methods. For example， based on the heuristic model， some researchers used information-aware method for throughput or power management［2-3］. In recent years， the rapid development of machine learning has inter adaptive peculiarity which is fit for the variation of network states. Therefore，

ML-based methods have made great progress in network optimization [4-7]. Compared with many outcomes, the prospective law prediction of ML-based methods is fitting for the high dynamic nature of wireless network optimization [8].

In addition, due to the development of device and technology, the multipath transmission has become the focus of data transmission optimization. Especially, the combination of multipath scheduling and machine learning makes further improvement of efficient data transmission [9-12]. However, the rapid increase of wireless connections in the Internet makes network states more complex and time-varying. Therefore, researchers proposed multiple algorithms fusion to solve problems. For example, deep Q-learning networks (DQN) not only realize the adaptive management for the dynamics of network sates but also cater to different joint objectives optimization [13-14]. In addition, double DQN, as an evolution of DQN, uses twice Q-learning over neural networks to find the optimal action, which can further accelerate the optimization process and reduce delay [14-16]. Moreover, the large scale and big data of wireless networks make the training of machine learning methods more complex and time-consuming. Therefore, researchers designed clustering and distributed learning to wireless network optimization [17-18].

In this paper, we propose a distributed asynchronous deep reinforcement learning framework for multipath data transmission in order to improve stability, throughput, and to reduce packet loss ratio, RTT. In this framework, there are two working parts, local asynchronous packet scheduling (Laps) and distributed cooperative control center (DC3). Laps is performing for multipath packets scheduling. We propose an asynchronous prioritized replay double deep Q-learning packet scheduling algorithm (ADPS) to help each learner (Laps) for multipath packet scheduling management. ADPS performs packets scheduling over multiple paths according to the network states at the executing timeslot based on P-DDQN. Facing the large scale of learners, we use DC3 to make cooperative learning for more precise packets scheduling so as to fit the network states. In DC3, each learner will choose a suitable learner to learn the neural network parameters of P-DDQN by dynamic weights adjustment with the feedback of throughput. In a word, the learner will make multipath packets scheduling with ADPS. Then, learners will initiative/timed call DC3 to make cooperative learning so as to enhance the scheduling accuracy. The main contributions are as follows.

We propose a distributed asynchronous deep reinforcement learning framework for multipath adaptive packets scheduling. The framework also supports distributed cooperative learning to enhance the stability of large scale data transmission.

We design ADPS to make specific packets scheduling management. ADPS uses P-DDQN to make efficient adaptive scheduling policy adjustment for the high dynamic nature of data transmission.

We use Programming Protocol-independent Packet Processors (P4) and bmv2 switch to implement a multipath data transmission system for experimental comparison.

The rest of this paper is organized as follows. Section 2 introduces some related works. The details of our distributed asynchronous deep reinforcement framework, network model, and algorithm are presented in Section 3. Section 4 gives a simulation for the framework and algorithm. Section 5 makes some conclusions.

2 RELATED WORKS

Comprehensive and efficient data transmission management is the main part to realize network optimization. Therefore, some researchers used queue-aware and channel-estimator methods to maximize throughout [2] and realize optimal transmission power allocation [3], respectively. Similar heuristic models also make some other outcomes. However, methods of the heuristic models always take action until something happened, which is hysteretic.

In addition， researchers also propose data clustering machine learning algorithms to reduce data transmission ［4］. In the following， Sagduyu et.al. used deep learning to guarantee the security of data transmission ［5］. Al-Tous et.al. used reinforcement learning to design a power control algorithm for reducing transmission delay ［6］. Certainly， Su et. al. inserted deep reinforcement learning into routing select algorithm for improving transmission sharing and enhancing throughput ［7］. In addition， Yu and Gabriel et. al also proposed multipath transmission method based on network coding ［9］ and multipath randomly schedule for unstable channels ［10］， respectively. Outstandingly， The combination of machine learning and multipath transmission flexibly make multi-objective joint optimization based on multiple indicators ［11-12］.

Furthermore， to enhance the adaptivity of methods， Xu et. al used deep Q-learning networks （DQN） to realize the adaptive management for the dynamics of network sates， so as to cater for different joint objectives optimization ［13］. In addition， Pan et.al used double DQN to find the optimal control action， which can further accelerate the optimization process and reduce delay ［14］. Alsadi et. al also proposed Mini-Batch Gradient Descent Method to accelerate the training based on gradient adjustment. Such as Stochastic Gradient Descent and Batch Gradient Descent method ［15］. Similarly， Liu et. al proposed a method to train samples with priority. The bigger the reward， the more important the sample， which makes the training process accelerated ［16］. According to the high efficiency of distributed learning and precise of machine learning， Lyu et. al designed distributed machine learning to realize online data partitioning ［17］. Certainly， the accelerate training researches also improved the practical application of distributed machine learning ［18］.

3 FRAMEWORK AND MODEL DESIGN

3.1 Distributed Asynchronous Deep Reinforcement Learning Framework

For improving the accuracy of packet scheduling， we design a distributed asynchronous deep reinforcement framework， shown in Fig. 1. The framework consists of two parts， local asynchronous packet scheduling （Laps） and distributed cooperative control center （DC3）.

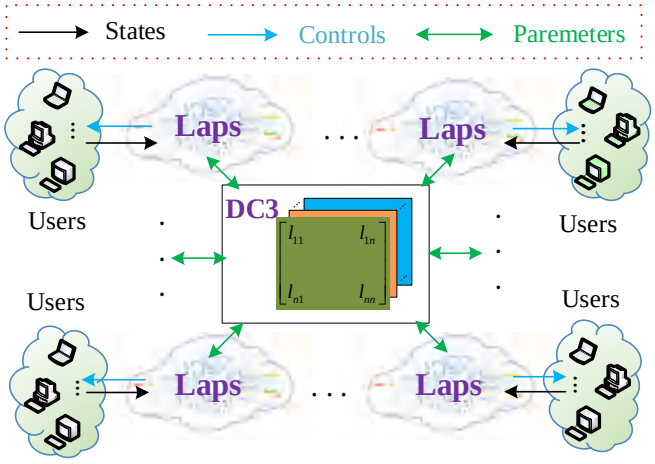


Fig. 1 The distributed asynchronous deep reinforcement framework.

The Laps is working like a learner in the local of users， shown in Fig. 2. In Laps， we design two parts to perform the packet scheduling. Firstly， the online packet scheduling model uses the trained P-DDQN to execute online scheduling. The online packet scheduling will schedule all arrival packets over multiple paths based on corresponding ratios （Action） which made by P-DDQN with inputs （States） at each *scheduling timeslot* （ST）. Secondly， the Offline double deep Q-learning network

（P-DDQN） training based the samples in *Experience Pool* to train the P-DDQN. This part contains three components， *Observer*， *Experience Pool* and *Trainer*. *Observer* is to collect state s_t and action a_t at each t timeslot. Then， due to reward function of P-DDQN， we get$r_t = reward(s_t)$. By the time going， *Observer* gets many tuples$\{s_t, a_t, r_t, s_{t+1}\}$， which are stored in *Experience Pool* by queue stack. *Experience Pool* is a memory with finite space. *Trainer* performs the training of P-DDQN based on samples which are chosen from *Experience Pool*.

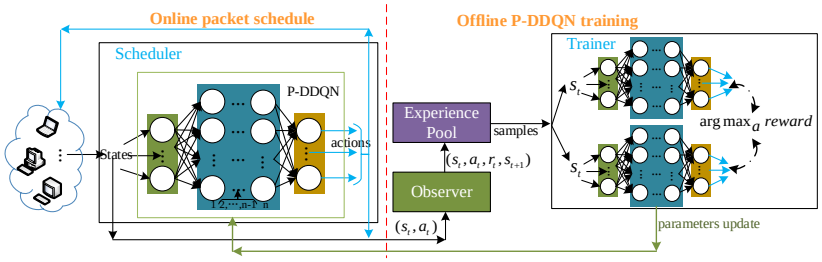


Fig. 2 The process of local asynchronous packet scheduling （Laps）.

The DC3 is working for parameters update of P-DDQN for all learners. Each learner can initiative or timed request parameters update by DC3. Then， learners send their own parameters to DC3 and waiting for the back of exchange and learned parameters. In DC3， all learning weights of each learner constitute an update matrix table$L_i \triangleq \{l_{ij};\ i,j = \{1,2,\cdots,N\}\}$. In addition， $l_{ij} \in [0,1]$, $\sum_{i=1}^{N} l_{ij} = 1$ and $\sum_{j=1}^{N} l_{ij} = 1$. $l_{ij} = 0$ means learner i and learner j are not connected and they do not learn from each other. Otherwise， $l_{ij} > 0$. $\sum_{i=1}^{N} l_{ij} = 1$and $\sum_{j=1}^{N} l_{ij} = 1$ mean the total learning rate of any learner with other learners is 1. Each learner will choose one learner’s parameters which has the biggestl_{ij} for learning by the probabilityη. Otherwise， randomly chooses one learner’s parameters to learn.

3.2 Network Model

Facing the problem of data transmission optimization， realizing dynamic adaptive packet scheduling is an efficient solution. Therefore， based on P-DDQN， we design a distributed learning schedule method to make efficient packet scheduling， which can enhance the data transmission performance.

Suppose there are N learners to make packet scheduling based on our framework. The working process can be divided into discrete timeslots as$t \in \{1,2,\cdots,T\}$. According to state s_t at timeslot$t$， learner will has an action $a_t = \{p_m;\ m \in \{1,2,\cdots,M\}\}$ to make packet scheduling for M paths based on the policy$policy(s_t)$. $p_m \in [0,1)$and $\sum_{m=1}^{M} p_m = 1$， p_m means the packet scheduling ratio for m-th path （The total number of packet scheduling can adaptively setting）. Therefore， realizing optimal packet scheduling is to find the optimal control policy based on P-DDQN. Certainly， how to get an optimal trained P-DDQN is the following problem. We will find the optimal $policy(\theta^Q)$ by maximizing the reward in the long training time， as Eq. （1）：

$$policy(\theta^Q) = \arg\max_{\theta^Q} \limsup_{t\to\infty} \frac{1}{t}\mathbb{E}\left[\sum_{i=1}^{N} reward(s_t)\right] \tag{1}$$

where θ^Q are parameters of P-DDQN， which constitutes the scheduling policy. However， in this paper， we want to realize the network optimization of enhancing throughput， reducing delay （RTT）， and cutting down packet loss. Therefore， we formulate packet scheduling as an optimization problem：

Maximize

$$throughput \tag{2}$$

Subject to：

$$a_t \in policy(s_t|\theta^Q) \tag{3}$$

$$delay < \bar{d}, packet\ loss < \bar{p} \tag{4}$$

where $\bar{d}$ and $\bar{p}$ are constant. *throughput* = $\mathbb{E}\left[\sum_{t=1}^{T}\sum_{m=1}^{M} t\, h_{m,t}\right]$, where $th_{m,t}$ is throughput of path m at timeslot$t$. But in a practical network environment, some network properties are interactional and subjected to each other. Therefore, for making full use of dynamic and the nature of fitting of P-DDQN, we formulate the overall optimization problem to maximize throughput, minimize delay and packet loss, which subject to bandwidth and congestion window size. Based on Lagrange multiplier method, and Eq. (1) – Eq. (4), we get the final optimization reward function Eq. (5):

$$reward\,(r_t^{th},r_t^{rtt},r_t^{lo},r_t^{bw},r_t^{cw}) = \alpha_1 r_t^{th} - \alpha_2 r_t^{rtt} - \alpha_3 r_t^{lo} + \alpha_4 r_t^{bw} + \alpha_5 r_t^{cw} \quad (5)$$

where α_1, α_2, α_3, α_4, α_5 are weights of each input and their values range is (0, 1).

$r_t^{th} = \sum_{m=1}^{M} th_{m,t}$ is the throughput of all paths. $th_{m,t}$ is throughput of path m at timeslot t in ST.

$r_t^{rtt} = \frac{1}{M}\sum_{m=1}^{M} rtt_{m,t}$ is the average RTT of all paths. $rtt_{m,t}$ is average RTT of path m at timeslot t in ST.

$r_t^{lo} = \sum_{m=1}^{M} lo_{m,t}$ is the packet loss of all paths. $lo_{m,t}$ is packet loss of path m at timeslot t in ST.

$r_t^{bw} = \frac{1}{M}\sum_{m=1}^{M} bw_{m,t}$ is the average bandwidth of all paths. $bw_{m,t}$ is bandwidth of path m at timeslot t in ST.

$r_t^{cw} = \frac{1}{M}\sum_{m=1}^{M} cw_{m,t}$ is the average congestion window size of all paths. $cw_{m,t}$ is congestion window size of path m at timeslot t in ST.

Above all, according to$s_t = \{th_t, rtt_t, lo_t, bw_t, cw_t\}$, we can perform Laps for the training of P-DDQN based on$\{s_t, a_t, r_t, s_{t+1}\}$.

3.3 Working Management and Algorithm Description

In local asynchronous packet scheduling (Laps), we choose P-DDQN as the core to schedule packets, because the convolution layer of neural network structure in P-DDQN can deeply efficient approach the coupling degree of all inputs in $s_t = \{th_t, rtt_t, lo_t, bw_t, cw_t\}$. Then, P-DDQN uses a full connection layer to simulate the interaction of all paths. The main process of Laps is the training process in *Trainer.*

For P-DDQN, its purpose is to find optimal neural network parameters θ^Q by maximizing the reward function in a long time training. The training process based the interaction of two neural networks, execute-network $Q(s,a|\theta^Q)$ and comparer-network$Q'(s,a|\theta^{Q'})$. For execute-network $Q(s,a|\theta^Q)$, we use state-action value function $\mathcal{Q}(s,a)$ to make iterative optimization so as to find an optimal action, as

$$a_{t+1}^{max} = \arg\max_a \mathcal{Q}(s_{t+1},a|\theta_t^Q) \quad (6)$$

Likewise, comparer-network uses the a_{t+1}^{max} to calculate cumulate target reward, as

$$R_t = r_{t+1} + \gamma \mathcal{Q}'(s_{t+1},a_{t+1}^{max}|\theta_t^{Q'}) \quad (7)$$

where γ is a constant. Thereafter, based on Eq. (8), we adjust the parameters θ_t^Q by minimizing the mean square error,

$$Loss(\theta_t^Q) = \mathbb{E}\left[(R_t - \mathcal{Q}(s_t,a|\theta_t^Q))^2\right] \quad (8)$$

Finally, we make Batch Gradient Descent method (BGD) [15] and Prioritized Replay DQN [16] into consideration, we get

$$Loss(\theta_t^Q) = \frac{1}{k}\sum_{j=1}^{k}\omega_j (R_j - \mathcal{Q}(s_j,a|\theta_j^Q))^2 \quad (9)$$

where ω_j is prioritized weights of sample j. j is the size of mini-batch samples from *Experience Pool*.

***Algorithm* 1**: The Asynchronous Prioritized Replay Double Deep Q-learning packets scheduling Algorithm (ADPS).

Input: Learning rateγ, update rateε, random probabilityψ, Experience Pool$P$.

Output: Execute-network parametersθ^Q.

1. Randomly initialize execute-network$Q(\theta^Q)$.

2. Initialize comparer-network$Q'(\theta^{Q'})$, where$\theta^{Q'} \leftarrow \theta^Q$.

//*****Online packet scheduling in Scheduler by time series t.******//

3. Records current statess_t， and calculates action a_t by $Q(\theta^Q)$ with probability ψ or randomly select actiona_t.

4. Performs packet scheduling witha_t， records states s_{t+1} and calculatesr_t.

5. Formats tuple{ s_t, a_t, r_t, s_{t+1}}， makes initialize current sample$j$'s priorityω_j， and stores them in P by tree structure andω_j.

//*****Offline P-DDQN training in Trainer. *******//

6 Loop: For *iteration* = 1,⋯,T do

1） Randomly extract k samples from P based on tree structure.

2） Calculates cumulate target reward of k samples， as

$$R_j = \begin{cases} r_{j+1} + \gamma \mathcal{Q}'(s_{j+1}, a_{j+1}^{max} | \theta_t^{Q'}), & \textit{Laps is active}, \\ r_{j+1}, & \textit{Laps is dormant}. \end{cases}$$

3） Update θ^Q by minimizing the mean square error， as

$$Minimize \frac{1}{k} \sum_{j=1}^{k} \omega_j \, (R_j - \mathcal{Q}(s_j, a | \theta_j^Q))^2.$$

4） Updates parameter： $\theta_t^{Q'} \leftarrow \varepsilon \theta_t^Q + (1 - \varepsilon)\theta_t^{Q'}$.

5） Updates current samples' tree structure by $\omega_{j, j \in k} = \left| R_j - \mathcal{Q}(s_j, a | \theta_j^Q) \right|$.

6） **If** Laps is dormant

7） Break.

8） **EndIf**

7 EndFor

For each learner， Fig. 2 shows the overall process of local P-DDQN training and packet scheduling. We also propose the asynchronous prioritized replay double deep Q-learning packets scheduling algorithm （ADPS） to show detailed steps. In Algorithm 1， there are two threads， which respectively execute online packet scheduling in *Scheduler* by time series t and offline P-DDQN training in *Trainer*. In *Scheduler*， online packet scheduling （line 1-5） not only do packet scheduling based on excute-network $Q(s, a|\theta^Q)$ and inputss_t， but also collect samples for *Experience Pool* by *Observer*. *Experience Pool* has finite space， when new samples come， some stale data will be dropped. In *Trainer*， offline P-DDQN training of execute-network $Q(s, a|\theta^Q)$ is executed （line 6， 7）. According to second Q-network prediction based on samples from *Experience Pool* to update the parameters θ^Q by T iteration.

When each learner executes Laps in local， DC3 is to maintain weight matrix table for distributed cooperative learning. DC3 can timed or requested by learner perform learning. When DC3 calls all learners to make update learning， each learning i will send their Q-network parameters $\theta_{i,\tau}^Q$ and feedback $h_{ij,\tau}$ to DC3.

$$h_{ij,\tau} = \sum_{t=\tau-1}^{\tau} throughput_t \Big/ throughput_{\tau-1} \qquad (10)$$

where $h_{ij,\tau}$ is the feedback of learner i to j at τ-th update. DC3 will firstly update weight matrix table based on all $h_{ij,\tau}$ by

$$l_{ij,\tau} = h_{ij,\tau} * l_{ij,\tau} \qquad (11)$$

Certainly， the weight matrix table should following the constraint by $\sum_{i=1}^{N} l_{ij} = 1$ and $\sum_{j=1}^{N} l_{ij} = 1$. In the following， each learner will select max weight of learner j for parameter update with probability$\eta, \eta \in (0, 1)$， or randomly select learner$k$， as

$$\theta_{i,\tau}^Q = \begin{cases} \mu\theta_{i,\tau}^Q + (1-\mu)\theta_{j,\tau}^Q & with\ probability\ \eta, \\ \mu\theta_{i,\tau}^Q + (1-\mu)\theta_{k,\tau}^Q & Otherwise, \end{cases}$$

where $\mu \in [0, 1]$ is the learning rate. Finally， DC3 sends all θ_τ^Q to their own learners which will update their parameters for better scheduling.

4 EXPERIMENTS AND ANALYSIS

4.1 Experiments Configurations

To verify the validity and performance of our framework and algorithm， we implement a data transmission system with bmv2 switch by programming P4 （P4 is a high-level language for programming protocol-independent packet processors， which works like OpenFlow （https：//p4.org/）.）. We systematically make comparing simulations over the Random weight method （Random）， Round Robin （RR）， and our method （ADPS）.

Settings and Network Parameters: The experiments are performed on a Linux machine (CPU: Intel i5-8265 3.2GHz; Memory: 8GB; OS: 64-bit Ubuntu 16.04). Multipath is made by WiFi and LTE networks respectively using Linux etc. Such as LTE: bandwidth 8Mbps, RTT 50ms; WiFi: bandwidth 16Mbps, RTT 70ms. The values of the notations of our framework and algorithm are shown in Table1.

Table. 1 The settings of notations in experiments.

Notations	Values	Notations	Values	Notations	Values	Notations	Values
N	10	M	2	α_1	1	k	50
α_2	0.7	α_3	0.7	α_4	0.7	P	2000
α_5	0.8	τ	1s	ST	1s	γ	0.9
η	0.9	ψ	0.9	μ	0.8	ε	0.9

4.2 Performance Analysis

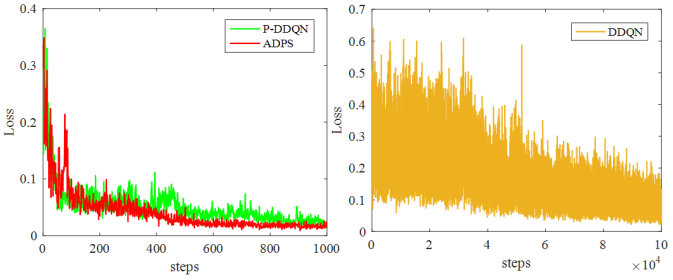


Fig. 3 The converge comparison of ADPS, P-DDQN, DDQN

To verify the validity of the overall framework, we make a comparison of DDQN, prioritized replay DDQN (P-DDQN), and ADPS (Trainer+DC3). Fig. 3 shows the converge comparison of ADPS, P-DDQN, DDQN. The Loss in Fig. 3 means the average training loss of all learners over every step. The smaller the Loss is, the better the algorithm approaches the pattern which hidden in samples. In Fig. 3, the loss of the training process has a downtrend of three algorithms. In addition, the loss of P-DDQN and ADPS approach to 0.01 level after 1000 training steps. However, DDQN needs 100000 training steps to make the loss close to 0.01 level. In short, the training delay of P-DDQN and ADPS are approximately 10 times shorter than DDQN, which means P-DDQN and ADPS have a better real-time capability facing the dynamic nature of the network. Considering the real-time of scheduling system, we make the training step as 1000 steps in Table 1. Whereas, the smallest average loss of them after 1000 steps training are ADPS: 0.0412, P-DDQN: 0.0515, DDQN: 0.1667. Therefore, ADPS has a smallest loss and faster training process. Comparing with P-DDQN, our distributed learning framework (DC3) is also helpful for ADPS.

In the following, we use the P4 system to make performance comparison on RTT, packet loss ratio and throughput of Random, RR and ADPS respectively. We respectively insert these three methods into our framework for comparison. In the experiment, the client continuously sends packets over two paths. The scheduling ratio of packets over paths is controlled by methods. At the beginning of every 1 second, the scheduling ratio will be changed by control method. In other words, the interval of work is 1 second. Fig. 4 shows the RTT comparison of these methods over the interval of work. In Fig.4, the average and variance of RTTs by RR are 68.245466 and 13.922727. That of Random are 67.763326 and 11.626504. However, that of ADPS are 66.968760

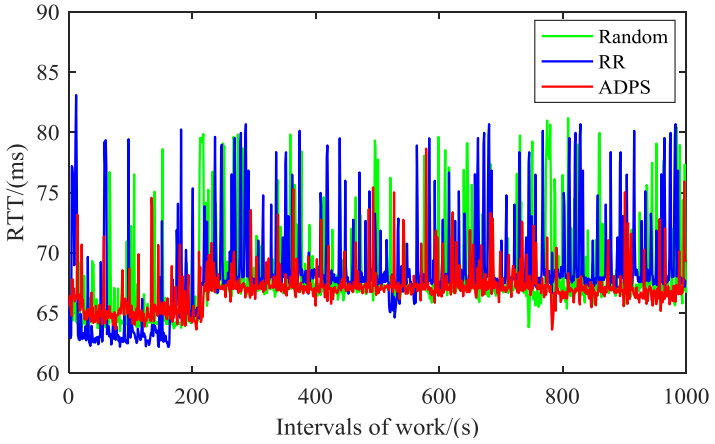


Fig. 4 The RTT comparison of Random， RR and ADPS.

and 2.901513. Making an overall comparison on RTT， ADPS outperforms Random and RR by 1.87% and 1.27%， respectively. The stability of ADPS is improved by 79.13% and 75.04% than Random and RR， respectively. Clearly， ADPS has great improvement on RTT and system stability.

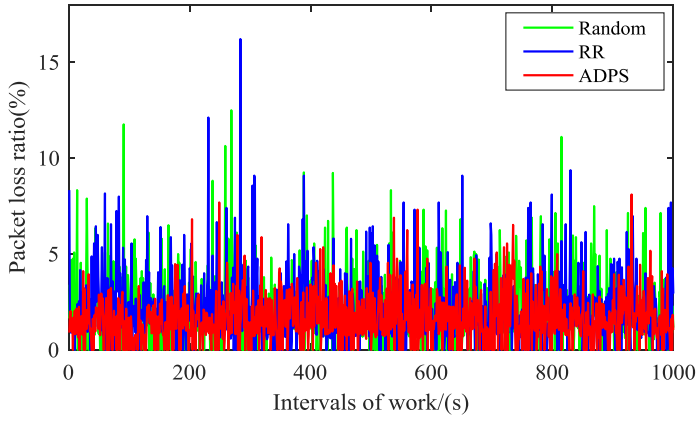


Fig. 5 The packets loss ratio comparison of Random， RR and ADPS.

Fig. 5 shows the packet loss ratio over the interval of work. On the whole， ADPS almost shows a lower packets loss ratio than Random and RR. Specifically， the average and variance of loss ratio by Random are 2.18% and 3.068. That of RR are 2.17% and 3.121. However， that of ADPS are 1.67% and 1.631. Making an overall comparison on packet loss ratio， ADPS outperforms Random and RR by 23.39% and 23.04%， respectively. The stability of ADPS is improving by 46.31% and 47.74% than Random and RR， respectively. Therefore， ADPS has the lowest packet loss ratio and has minimal fluctuation.

Fig. 6 shows the throughput over the interval of work. Specifically， the average and variance of throughput by Random are 94.295 and 809.123975. That of RR are 95.538 and 944.876556. However， that of ADPS are 96.078 and 779.467916. Therefore， ADPS outperforms Random and RR by 2.18% and 1.14% on throughput， respectively. The stability of ADPS outperforms Random and RR by 3.66% and 17.5%， respectively. Obviously， ADPS has better performance on throughput and stability.

Finally， experimental results show that ADPS has better performance on RTT， packet loss ratio and throughput than Random and RR. Particularly， ADPS has more prominent advantages in system sta-

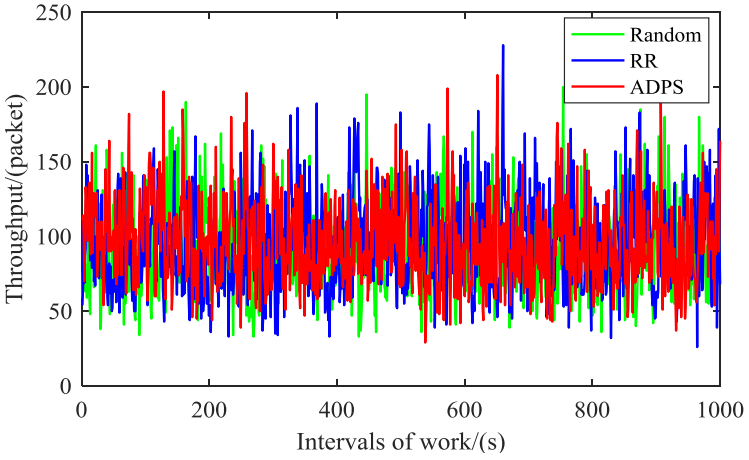


Fig. 6 The throughput comparison of Random, RR and ADPS.

bility. Overall, ADPS has 2.34 and 2.14 times better than Random and RR on the stability of data transmission, respectively. In addition, with the improvement of equipment performance and environment, the advantages of ADPS will be clearer.

5 Conclusions and future work

This paper proposes a distributed asynchronous deep reinforcement learning framework for adaptive multiple packet scheduling. The framework contains two parts, Laps and DC3. Laps is working for online packet scheduling and P-DDQN training. DC3 is distributed learning for learners based on Laps. The combination of framework and ADPS makes the learning process are faster and more precise. In addition, comparing with default methods, ADPS has a better performance on throughput, delay, and packet loss. Certainly, system stability is also a better advantage of ADPS for data transmission.

References:

[1] Cisco, Cisco Annual Internet Report (2018 – 2023) White Paper. URL: https://www. cisco. com/c/en/us/solutions/collateral/serviceprovider/visual-networking-index-vni/white-paper-c11-741490. html. Last accessed 26 May 2020.

[2] DimitriouI. and PappasN. . Stable Throughput and Delay Analysis of a Random Access Network With Queue-Aware Transmission [J]. IEEE Transactions on Wireless Communications, 2018, 17(5): 3170 – 3184.

[3] MaJ. , ZhangS. , LiH. , Zhao and VN. . C. M. Leung. Interference Alignment and Soft-Space-Reuse Based Cooperative Transmission for Multi-cell Massive MIMO Networks [J]. IEEE Transactions on Wireless Communications, 2018, 17(3): 1907-1922.

[4] VaidehiS. K. and V. . Clustering and Data Aggregation in Wireless Sensor Networks Using Machine Learning Algorithms [C]// 2018 International Conference on Recent Trends in Advance Computing (ICRTAC). IEEE Computer Society, 2018: 109-115.

[5] SagduyuY. E. , ShiY. and ErpekT. . IoT Network Security from the Perspective of Adversarial Deep Learning [C]// 2019 16th Annual IEEE International Conference on Sensing, Communications, and Networking (SECON). IEEE Computer Society, 2019: 1-9.

[6] Al-Tous and IH. . Barhumi. Distributed Reinforcement Learning Algorithm for Energy Harvesting Sensor Networks [C]// 2019 IEEE International Black Sea Conference on Communications and Networks (BlackSeaCom). IEEE Computer Society, 2019: 1-3

[7] SuY. , LuX. , ZhaoY. , HuangL. and DuX. . Cooperative Communications With Relay Selection Based on Deep Reinforcement Learning in Wireless Sensor Networks [J]. IEEE Sensors journal, 2019, 19(20): 9561- 9569.

[8] ChenM. , ChallitaU. , SaadW. , YinC. and DebbahM. . Artificial Neural Networks-Based Machine Learning for Wireless Networks: A Tutorial [J]. IEEE Communications Survey Tutorial, 2019, 21(4): 3039-3071.

[9] YuF. . A Multi-path Transmission Method for Wireless Sensor Network Based on Network Coding [C]// 2018 International Conference on Virtual Reality and Smart Systems (ICVRIS). IEEE Computer Society, 2018: 505-508.

[10] GabrielF. , RischkeJ. , FitzekF. H. P. , MhleisenM. and LohmarT. . No Plan Survives Contact with the Enemy: On Gains of Coded Multipath over MPTCP in Dynamic Settings [C]// 2019 IEEE Wireless Communications and Network Conference (WCNC). IEEE Computer Society, 2019: 1-8.

[11] LiW. , ZhangH. , GaoS. , XueC. , WangX. and LuS. . SmartCC: A Reinforcement Learning Approach for Multipath TCP Congestion Control in Heterogeneous Networks [J]. IEEE Journal Selected Areas Communications, 37(11), 2019: 2621-2633.

[12] ZhaoH., ZhangM., YuH., MaoT. and ZhuH.. Multipath TCP Path Scheduling optimization Based on Q-Learning in Vehicular Heterogeneous Networks [C]// 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP). IEEE Computer Society, 2018: 1-5.

[13] XuF., YangF., BaoS. and ZhaoC.. DQN Inspired Joint Computing and Caching Resource Allocation Approach for Software Defined Information-Centric Internet of Things Network [J]. IEEE Access, 2019, 7(1): 61987-61996.

[14] PanJ., WangX., ChengY. and YuQ.. Multisource Transfer Double DQN Based on Actor Learning [J]. IEEE Transactions Neural Networks Learned System, 2018, 29(6): 2227-2238.

[15] AlsadiM. S., GhnematR. and AwajanA.. Accelerating Stochastic Gradient Descent using Adaptive Mini-Batch Size [C]// 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS). IEEE Computer Society, 2019: 1-7.

[16] LiuB., YeX., GaoY., DongX., WangX. and LiuB.. Forward Looking Imaginative Planning Framework Combined with Prioritized Replay Double DQN [C]// 2019 5th International Conference on Control, Automation and Robotics (ICCAR). IEEE Computer Society, 2019: 336-341.

[17] LyuX., RenC., NiW., TianH., LiuR. P. and DutkiewiczE.. Optimal Online Data Partitioning for Geo-Distributed Machine Learning in Edge of Wireless Networks [J]. IEEE Journal on Selected Areas in Commutations, 2019, 37(10): 2393-2406.

[18] LimE., AhnS. and ChoiW.. Accelerating training of DNN in distributed machine learning system with shared memory [C]// 2017 International Conference on Information and Communications Technology Convergence (ICTC). IEEE Computer Society, 2017: 1209-1212.

About the authors

Liu Kang was born is in 1992. He is currently pursuing the Ph. D. degree with the Department of Electronic Information Engineering, Beijing Jiaotong University, Beijing, China. His research interests include next generation Internet, deep reinforcement learning and network communication and programmable data plane (Email: 19111028@bjtu. edu. cn).

Wei Quan received the Ph. D. degree in communication and information system from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2014. He is currently an Associate Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University (BJTU). He has published more than 20 articles in prestigious international journals and conferences, including the IEEE Communications Magazine, the IEEE WIRELESS COMMUNICATIONS, the IEEE NETWORK, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE COMMUNICATIONS LETTERS, IFIP Networking, the IEEE ICC, and the IEEE GLOBECOM. His research interests include key technologies for network analytics, future Internet, 5G networks, and vehicular networks. He is also a member of ACM and a Senior Member of the Chinese Association of Artificial Intelligence (CAAI). He serves as an Associate Editor for the Journal of Internet Technology (JIT), Peer-to-Peer Networking and Applications (PP-NA), and IET Networks, and as a technical reviewer for many important international journals (Email: weiquan@bjtu. edu. cn).

Deyun Gao received the B. Eng. and M. Eng. degrees in electrical engineering and the Ph. D. degree in computer science from Tianjin University, China, in 1994, 1999, and 2002, respectively. He spent one year as a Research Associate with the Department of Electrical and Electronic Engineering, The Hong Kong University of Science and Technology. He then spent three years as a Research Fellow with the School of Computer Engineering, Nanyang Technological University, Singapore. In 2007, he joined the Faculty of Beijing Jiaotong University as an Associate Professor with the School of Electronics and Information Engineering and was promoted to a Full Professor, in 2012. In 2014, he was a Visiting Scholar with the University of California at Berkeley, USA. His research interests are in the area of the Internet of Things, vehicular networks, and the next-generation Internet (Email: gaody@bjtu. edu. cn).

Chengxiao Yu was born in Dalian, Liaoning, China, in February 1993. He received the B. E. degree in communication and information systems from Beijing Jiaotong University, Beijing, China, in 2015. He is currently pursuing the Ph. D. degree with the School of Electronic and Information Engineering, the National Engineering Lab for Next Generation Internet Technologies (NGIT), Beijing Jiaotong University (BJTU), Beijing, China. His research interests include concurrent multipath transfer (CMT), machine learning (ML), high-speed railway (HSR) networks, and programmable data plane (Email: chengxiaoyu@bjtu. edu. cn).

Mingyuan Liu received the bachelor's degree in communication engineering, in 2018. He is currently pursuing the Ph. D. degree with the School of Electronic and Information Engineering, Beijing Jiaotong University. His current research interests include future networks, software defined networking

（SDN）， and cybersecurity.

A High Performance Accelerator for Lattice Generation Automatic Speech Recognition Decoding

SUN Huajie, YIN Shouyi, LIU Leibo, WEI Shaojun

Department of Microelectronics and Nanoelectronics, Tsinghua University,Beijing 100084, China

Key words: Lattice generation accelerator; speech recognition decoder; automatic speech recognition

Abstract. Currently， many Artificial Intelligence （AI） technologies and heterogeneous computing are performed in the cloud， and the acceleration with FPGAs is a major trend. In this field， a large part of cloud-based applications are automatic speech recognition （ASR）. Until now， there are a lot of research on acoustic models and language models but few on the decoder of speech recognition system. This paper proposes a high performance accelerator for lattice generation automatic speech recognition to accelerate the process of speech decoding. Based on the Zynq UltraScale+ MPSoC： EG （ZU19 EG） device， we completed the proposed accelerator’ s implementation. The experimental result shows that the accelerator has a maximum frequency of up to 20 MHz， the power consumption of which is only 1.92 W. Compared to the decoder implemented based on CPU， the proposed accelerator can achieve 17.5× speedup.

1 Introduction

In recent years， thanks to AI and Internet of Things （IoT） technologies， automatic speech recognition （ASR） technology has been further developed. At present， many ASR systems have been widely used in smart phones， entertainment devices， PCs， etc. With the introduction of various ASR technologies， the interaction between humans and machines is becoming more natural and comfortable， ［1-3］.

As shown in Fig. 1， many Artificial Intelligence （AI） technologies and heterogeneous computing are performed in the cloud currently， and a large part of these applications are speech recognition. In the Large Vocabulary Continuous Speech Recognition （LVCSR） systems， in order to improve the recognition accuracy， the acoustic model and the language model are often designed very cleverly， but at the same time， the complexity of the system is also increased. The large search graph and complex algorithm exacerbate the system's computing power and date storage space requirements， so the LVCSR system is difficult to implement in resource-constrained mobile devices and more suitable for implementation on the cloud server for better performance ［4-6］.

In the field of speech recognition， people have done a lot of research on acoustic models and language models. In order to reduce the size of the search graph and improve the decoding speed， the literature ［7］ proposed the EESEN speech recognition framework. The CTC-RNN acoustic model has more obvious advantages than the traditional HMM-GMM model. The literature ［8］ proposed an En-

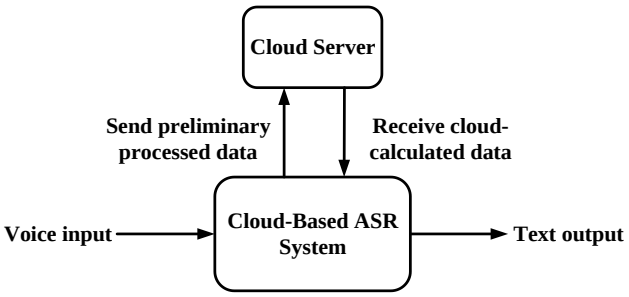


Fig. 1 A cloud-based automatic speech recognition system.

coder-Decoder network with an attention mechanism, which is significantly simpler than the HMM-DNN. Combined with a trigram language model, literature [8] shows excellent performance. And the literature [9] presented a Neural Speech Recognizer without pronunciation lexicon and decoder. Using the CTC-LSTM acoustic model, [9] achieves better performance than conventional context-dependent phone-based systems.

At the same time, however, there are few researches on the decoder of speech recognition system. In a LVCSR system, the speech decoding process occupies a large amount of time due to the large size of the search graph and the complexity of the decoding algorithm. For improving the data throughput of the speech recognition server so as to support more clients, it is necessary to use hardware acceleration for the decoding.

Based on previous analysis, we have conducted some research on decoder under the EESEN framework and designed a software and hardware combined high performance ASR decoding accelerator. Besides, a FPGA-based accelerator is proposed for accelerating the key steps of the decoding process. The proposed ASR decoding accelerator is based on weighted finite state transducer (WFST) and the novelties of the decoding accelerator can be listed as follows:

Unhashing logic: Hash logic does not exist in our circuit structure to increase Lattice generation efficiency.

A hardware-friendly way to states' read and write: Without changing the performance of speech recognition, we have changed the order of reading and writing of the state and reduced the complexity of the logic.

Relative partial insertion sorting: We propose a new sorting algorithm. The sorting algorithm can seamlessly connect upstream circuits and implement pipelined operations, saving a large number of idle clock cycles.

The rest of the paper is organized as follows. Section 2 introduces the proposed ASR decoding accelerator in detail. Section 3 describes the experimental results of the ASR decoding accelerator. Finally, we conclude the paper in Section 4.

2 A High Performance Accelerator for Lattice Generation ASR Decoding

2.1 Proposed accelerator framework

The framework of the proposed ASR decoding accelerator is shown in Fig. 2. In this design, the accelerator is divided into hardware part and software part. The software part is based on ARM, which is used for pruning Lattice, searching the best path (i.e., the path with the lowest total cost) and outputting the recognized text finally. However, Lattice generation using software methods is often time-consuming and takes up a lot of memory, so in the hardware part we designed a high performance Lattice generation VLSI structure (viz. Lattice Generator) which can significantly increase the efficiency of Lattice generation. Furthermore, to simplify the VLSI and reduce the number of useless clock cycles, we rearrange the WFST search graph and spilt it into emitting part and none-emitting part. The emitting part is for storing the emitting arcs with none-zero in-

put label， whereas the none-emitting arcs with zero input label are stored in the none-emitting part. It is worth mentioning that the WFST search graph needs frequent read during the Lattice generation process， so actually it is implemented by on-chip ROM instead of DRAM for reducing the number of off-chip memory access and speeding up the Lattice generation process in some way. In addition， the final cost of each final state is stored in DRAM， which will be used for pruning Lattice in software part.

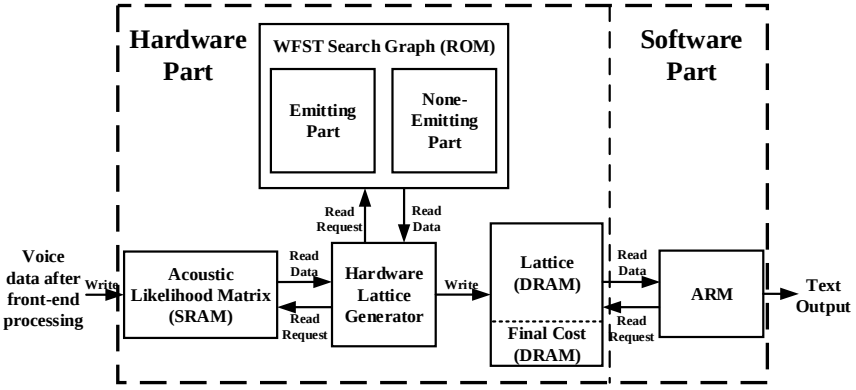


Fig. 2 The framework of the proposed ASR decoding accelerator.

On the whole， for each voice， the acoustic likelihood matrix will receive a set of voice data after front-end processing. With the WFST search graph and the acoustic likelihood matrix， the hardware part will generate tokens and links for each frame of the input voice and write them into an off-chip DRAM during the next frame generation process. At the end， all tokens and links will form the complete Lattice and then the software part will perform its operation and output the recognized text. At the same time， the hardware part will generate another Lattice for the next voice. This hard-soft pipeline mechanism can improve speech recognition efficiency in the case of multiple voice input.

2.2 Key technologies of the Accelerator

1) Unhashing logic

There are many hash tables or hash buckets in the software implementation. They are mainly used to find or write tokens. However， hash logic inevitably has a hash collision. And we often need to use hash functions to avoid collisions， so that the data can be correctly found or written. But these processes often take a lot of time. Because one hash often cannot find the data we want. In terms of hardware， it will cause many idle clock cycles， affecting the performance of the entire ASR decoding accelerator， especially in the token generation phase. In addition， a bad hash function may cause a low filling factor and waste a lot of memory space in hardware. But a high filling factor hash function usually has very complicated logic that is hard to implement in hardware.

Hence， there is no hashing logic in our design. In the design， a similar indirect addressing is used to complete the token search， as shown in Fig. 3. The input signal “state_dest” is the arrival state of the current arc. “state_dest” will read data from

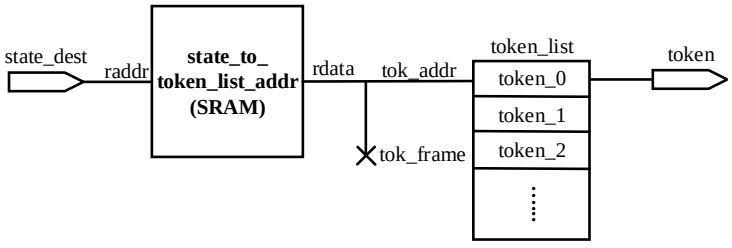


Fig. 3 A similar indirect addressing.

"state_to_token_list_addr" SRAM as an address. The read data consists of "tok_addr" and "tok_frame", respectively. The "tok_addr" is the address of the query token in "token_list" SRAM and the "tok_frame" is the frame when the query token is stored. If "token_frame" is not equal to current frame or invalid, it means the token corresponding to "state_dest" doesn't generate in the current frame yet; otherwise, the desired token can be obtained from "token_list" through "tok_addr". According to the algorithm, this mechanism can complete the update of the total cost of the token without hashing logic and greatly improves the speed of Lattice generation compared with hash logic. However, this method is more suitable for small vocabulary ASR systems; otherwise, the size of "state_to_token_list_addr" will be very large.

2) A hardware-friendly way to states' read and write

In the original algorithm, the states of all emitting tokens in the current frame should be pushed into a state list one by one. And in the none-emitting phase, a state will be read from the tail of state list for the generation of none-emitting tokens each time, as shown in Fig. 4 (a).

In addition, when the total cost of a token is updated or some token is generated for the first time, the state of the corresponding token will be written in the tail of the state list, as shown in Fig. 4 (b). The new state should not cover the "State_1003", because all the current frame states will be used for the next frame's tokens and links generation.

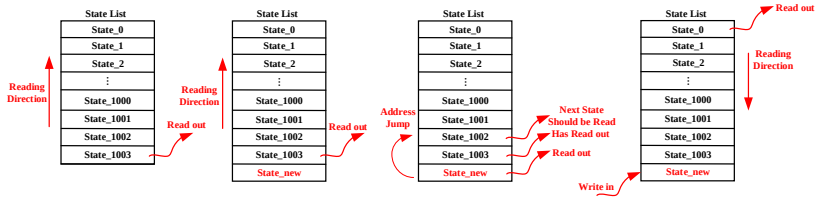


Fig. 4 A hardware-friendly way to states' read and write.

(a) Read the last state out (b) Write new state (c) Address jump (d) New reading direction

However, this reading mechanism will cause the generation of reading address to be very complicated. As shown in Fig. 4 (c), in order to improve efficiency, the "state_1003" should not be read again after reading the "State_new", so there is an "address jump". And with the token generation, this kind of address jump will be more and more. Regrettably, this address jump is not easily implemented in hardware because there is a lot of historical information that needs to be recorded.

To fix this problem, a hardware-friendly way to states' read and write is proposed. As shown in Fig. 4 (d), in terms of states write, this method still writes the new states to the tail of state list; but in terms of states reading, this method will read the states from the head of state list for token generation. In this way, state reading becomes more convenient and the logic of this part of the circuit is simplified. Most importantly, this state reading method does not have any side effects on Lattice generation, because each state processing is independent and the reading order does not affect the Lattice generation result.

3) Relative partial insertion sorting

According to the algorithm, we need to sort the total cost of the current frame's token in a small to large order before the next frame's token generation. However, how to sort the total cost as soon as possible is a big problem. Usually one voice will be divided into several hundred frames and the number of tokens per frame is in the range of thousands to tens of thousands, so if the sorting consumes a large number of clock cycles, it will bring a large delay to the entire circuit. The best way is to pipeline the sorting circuit, so the sorting of total cost can pipeline with its generation. In order to reduce the number of

pipeline stages， the insertion sorting is considered. This kind of sorting circuit can get the sorting result in one clock cycle at the fastest. However， there is still another key issue. As mentioned before， the token’s total cost is updating before the states in state list are all read， i.e.， if a smaller total cost of some token is produced in the current frame， it will take place of the old one. As shown in Fig. 5， this change will cause an error in the insertion sorting result.

One solution is to sort the total cost after it is stable. But as mentioned before， it will reduce the efficiency of speech recognition. To fix this problem， the relative partial insertion sorting algorithm is proposed. As show in Fig. 6， this sorting algorithm has two kinds of input data， i.e.， new smaller value and relative value. And the relative value is the difference between the old value and the new smaller value.

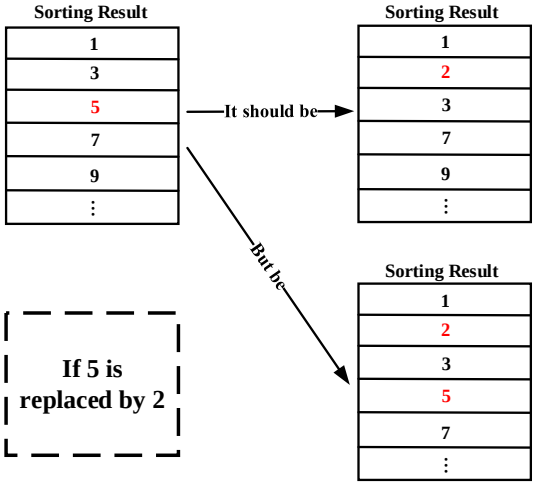


Fig. 5 The sorting error.

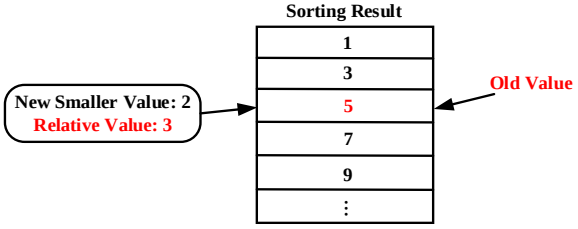


Fig. 6 The relative partial insertion sorting inputs.

The sorting algorithm is divided into three steps totally. First， each value in the sorting result needs to be subtracted from the input new smaller value and a difference is obtained， as shown in Fig. 7（a）. Second， compare the relative value with the difference and get the comparison results. Finally， insertion sorting is performed on the sorting result where the difference is not greater than the relative value portion and the correct sorting result is obtained， as shown in Fig. 7（b） and Fig. 7（c）， respectively.

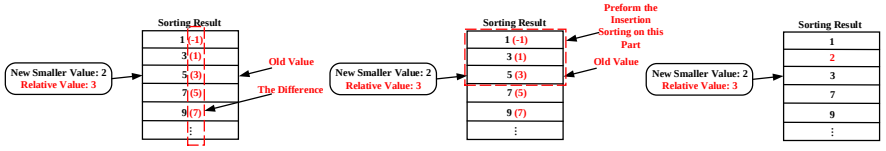


Fig. 7 The relative partial insertion sorting steps.

(a) The difference (b) The partial insertion sorting (c) The correct sorting result

3 Evaluation Results

Table 1 FPGA Resource Utilization of Accelerator

Resource Utilization	Resource				
	LUT	*LUTRAM*	*FF*	*BRAM*	*DSP*
Accelerator	294,609	43,735	82,094	1,178	1

In order to facilitate the connection between the software part and the hardware part of the accelerator， we used the Zynq UltraScale+ MPSoC： EG（ZU19EG） device to complete the entire accelerator implementation. Based on the VIVADO platform， we completed the synthesis and P&R of the Lattice Generator. The experimental result shows that the Lattice Generator can operates at 20 MHz to complete the generation of Lattice. The FPGA resource utilization result is shown in Table 1. Due to the large number of sorting result （up to 5000 supported） and the large WFST search graph （about 9.17 MB）， the relative partial insertion sorting circuit takes up most of the used LUT and the WFST search graph takes up most of the used BRAM.

Table 2 Real-Time Factor Comparison Between the Proposed decoding accelerator and the CPU-Based Decoder

Decoder	RTF
[7]	0.64
Proposed ASR decoding accelerator	0.0365

In addition， the experimental results show that the total power consumption of the Lattice Generator is 1.92 W， which is far less than the result implemented by the GPU ［10］.

Compared the proposed ASR decoding accelerator with that of CPU-based ［8］， the results show that the accelerator proposed in this paper can improve the real-time speed of speech recognition by 17.5 times， as show in Table 2.

4 Conclusions and future work

In this work， a high performance automatic speech recognition decoding accelerator is proposed. In this accelerator， we designed a hardware Lattice Generator to accelerate Lattice generation. Compared to software， this Lattice Generator does not have hash logic. In addition， we present a hardware-friendly way to states' read and write. Both of them reduced the complexity of the logic and increased Lattice generation efficiency. Besides， the relative partial insertion sorting algorithm proposed saved a large number of idle clock cycles. To facilitate the connection between the software part and the hardware part of the proposed accelerator， we used the Zynq UltraScale+ MPSoC： EG （ZU19EG） device to complete the entire accelerator implementation. The experimental result shows that the Lattice Generator has a maximum frequency of up to 20MHz. And the power consumption of the Lattice Generator is only 1.92 W which is far less than the result implemented by the GPU. Finally， compared the proposed accelerator with that of CPU-based， the accelerator proposed can achieve ×17.5 real-time speed. In future studies， we will consider a new token sorting method and embed the instruction hash logic core to further improve the performance of the decoding process.

References:

[1] HORI T, NAKAMURA A. Speech recognition algorithms using weighted finite-state transducers［M］. Morgan and Claypool Publishers，2013.

[2] YU D, DENG L. Automatic speech recognition: A deep learning approach［M］. Springer-Verlag，2015.

[3] YAZDANI R, SEGURA A, ARNAU J, et al. Low-Power Automatic Speech Recognition Through a Mobile GPU and a Viterbi Accelerator［J］. IEEE Micro，2017，37(1)：22-29.

[4] ASSEFI M, WITTIE M, KNIGHT A. Impact of Network Performance on Cloud Speech Recognition［C］//Proc. Int. Conf. Comput. Commun. Networks. Las Vegas，NV，United states，vol. 2015-October，article number：7288417，2015.

[5] TWIEFEL J, BAUMANN T, HEINRICH S, et al. Improving Domain-independent Cloud-Based Speech Recognition with Domain-Dependent Phonetic Post-Processing［C］//Proc. Natl. Conf. Artif. Intell. Quebec City，QC，Canada，vol. 2，2014：1529-1535.

[6] SHIRAZ M, GANI A, KHOKHAR R H, et al. A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing［J］. IEEE Commun. Surv. Tutor.，2013，15(3)：1294-1313.

[7] MIAO Y, GOWAYYED M, METZE F. Eesen: end-to-end speech recognition using deep rnn models and wfst-based decoding［C］// IEEE Workshop Autom. Speech Recognit. Underst. Scottsdale，AZ，United states，2015：167-174.

[8] BAHDANAU D, CHOROWSKI J, SERDYUK D, et al. End-to-end attention-based large vocabulary speech recognition［C］//IEEE Int. Conf. Acoust. Speech Signal Process Proc. Shanghai, China, vol. 2016-May，2016：4945-4949.

[9] SOLTAU H, LIAO H, SAK H. Neural speech recognizer: acoustic-to-word lstm model for large vocabulary speech recognition［C］//Proc. Annu. Conf. Int. Speech. Commun. Assoc. Stockholm，Sweden，vol. 2017-August，2017：3707-3711.

[10] LEE M, HWANG K, PARK J, et al. FPGA-based low-power speech recognition with recurrent neural networks［C］//IEEE Workshop Signal. Process. Syst. SiPS Des. Implement. Dallas，TX，United states，2016：230-235.

About the authors

Huajie SUN received his B. E. degree from the School of Information Science and Technology， Southwest Jiaotong University， Chengdu， China， in 2018. He is currently pursuing the master's degree with the Institute of Microelectronics， Tsinghua University， Beijing， China. His research interests include Automatic Speech Recognition and mobile computing.（Email： shj18@mails. tsinghua. edu. cn）

Shouyi YIN ［corresponding author］ received the B. S.，

M. S.， and Ph. D. degrees in electronic engineering from Tsinghua University， Beijing， China， in 2000， 2002， and 2005， respectively. He has worked at Imperial College London， South Kensington Campus， London， U. K.， as a Research Associate. He is currently an Professor with the Institute of Microelectronics， Tsinghua University. His research interests include mobile computing， wireless communications， and system on chip （SoC） design. （Email：yinsy@tsinghua. edu. cn）

Leibo LIU received the B. S. degree in electronic engineering and the Ph. D. degree from the Institute of Microelectronics， Tsinghua University， Beijing， China， in 1999 and 2004， respectively. He is currently a Professor with the Institute of Microelectronics， Tsinghua University. His current research interests include reconfigurable computing， mobile computing， and VLSI digital signal processing. （Email：liulb@tsinghua. edu. cn）

Shaojun WEI was born in Beijing， China， in 1958. He received the Ph. D. degree from the La Faculté Polytechnique de Mons， Mons， Belgium， in 1991. He became a Professor at the Institute of Microelectronics， Tsinghua University， in 1995. His main research interests include VLSI system on chip （SoC） design， electronics design automation （EDA） methodology， and application-specific integrated circuit （ASIC） design for communications. Dr. Wei is a Senior Member of the Chinese Institute of Electronics （CIE）. （Email：wsj@tsinghua. edu. cn）

基于VIP的RapidIO交换芯片功能验证设计与实现

张丽，宋克，沈剑良，刘冬培
中国人民解放军战略支援部队信息工程大学，河南 郑州 450002

摘　要：串行RapidIO是一种高性能、低引脚数的基于报文可靠交换的互连体系结构，其芯片规模将达千万门晶体管，相应的验证技术的复杂程度也逐步提高。针对RapidIO交换芯片的功能验证，提出了一种基于SRIO VIP的RapidIO交换芯片功能验证方法，并采取UVM搭建了相应的电路验证平台，通过了大量测试用例的仿真验证。结果表明该方法可以很好地验证并保证RapidIO交换芯片的协议一致性，有效地提升了验证的效率和准确性，缩短了验证周期。

关键词：SRIO VIP、功能验证、RapidIO交换芯片、UVM验证平台

Functional verification design and implementation of RapidIO switch chip based on VIP

ZHANG Li, SONG Ke, SHEN Jian-liang, LIU Dong-pei
The PLA Information Engineering University, Zhengzhou 450002, China

Abstract: Serial RapidIO is a high-performance, low-pin-count, reliable message-based interconnect architecture. RapidIO switch chip size will reach tens of millions of transistors, and the corresponding verification technology will gradually increase in complexity. For functional verification of RapidIO switch chip, a functional verification method of RapidIO switch chip based on SRIO VIP was proposed, and the corresponding circuit verification platform was built by using the UVM, which passed the simulation verification of a large number of test cases. The results show that this method can well verify and guarantee the protocol consistency of RapidIO switch chip, effectively improve the efficiency and accuracy of verification, and shorten the verification period.

Key words: SRIO VIP; functional verification; RapidIO switch chip; UVM verification platform

1　概述

验证技术是集成电路设计的重点和难点，其复杂度随着芯片规模的增大以指数的方式增加，功能验证缺陷导致的芯片设计失败或重新流片的比例逐步提高[1-2]。RapidIO交换芯片的规模将达千万门晶体管，针对这样大规模的芯片设计，需要采用先进的仿真和验证技术，在模块级、芯片级、系统级等多个层次开展基于覆盖率、功能、形式化等驱动的多种验证手段，以及硬件仿真器、FPGA仿真等支持的多种验证平台，以保证芯片设计的功能正确性，降低流片风险。本文针对RapidIO交换芯片的功能验证，为保证芯片的协议一致性，采用基于Cadence公司的SRIO VIP（Verification IP）的RapidIO交换芯片功能验证方法，并搭建了相应的UVM验证平台，实现了其协议一致性验证。

2 RapidIO交换芯片结构

RapidIO交换芯片由SerDes高速通道模块、RapidIO控制器模块、Switch Fabric模块三个大的部分组成。SerDes高速通道模块主要完成串行数据与并行数据之间的转换，应用于RapidIO等协议的物理层数据传输；RapidIO控制器模块主要完成报文解析，实现数据传输控制；Switch Fabric模块是RapidIO交换芯片的核心模块，主要实现多个端口之间的无阻塞数据交换，其主要验证目标就是交换能力，包括各种转发表、单播包、组播包。

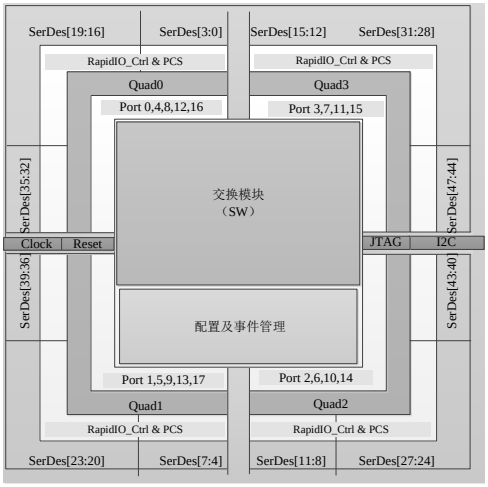


图1 RapidIO交换芯片结构框图

Figure1 Structural block diagram of RapidIO switch chip

图1描述了RapidIO交换芯片的总体结构框图，包含时钟、复位、I2C、JTAG、18个srio port、12个transceiver、48条lane和1个switch_fabric，每个srio_port又包含BUF、配置模块、包路由、包生成与包捕获、包追踪与包过滤模块等，端口支持1x、2x、4x的端口宽度，每条lane速率支持1.25/ 2.5/3.125/5/6.25Gbaud，支持RapidIO Gen2协议，并可与带有RapidIO Gen1与Gen2的端点设备对接，比如微处理器、DSP、FPGA、ASIC等。

3 SRIO VIP介绍

SRIO VIP是由Cadence公司开发的一款用于验证SRIO协议的IP核，可利用其已有的UVM基础类进行快速搭建适合自身要求的验证环境，其可充当激励发生器，用于SRIO协议的一致性和违例检查，并且协议功能是可配置的[3-5]。

在使用SRIO VIP时，首先需要根据验证平台所使用的是SystemVerilog版本还是UVM版本，生成对应的编译好的VIP库文件，然后在相应的验证平台脚本文件中将环境变量指向该VIP库文件的位置，后续才可以使用SRIO VIP。然后，根据实际项目要求，对SOMA（specification of modeling architecture）文件进行配置[3]以完成VIP与实际DUT的一致。这里的SOMA文件通过PureView工具来完成配置，主要包含SRIO logic model设置，SRIO serial physical model设置，SRIO通用设置以及SRIO环境设置，例如lane的个数，active_link_width参数，初始化端口宽度，idle选择等都可以在此配置完成。其次，SRIO VIP中transaction包含由VIP产生的各种通信数据并输入到DUT中，其UVM中的类为denaliSrioTransaction。Transaction的层次包括logical层、Physical层、Transport层以及Base Transaction，在编写测试用例的过程中，需要根据需要选择相应的通信字段进行设置。最后，需要在验证平台的testbench中实例化VIP的接口文件，才可以成功将VIP链接到验证平台中，并进行相应的仿真操作。

4 UVM验证方法

UVM（Universal Verification Methodology）是一个以SystemVerilog类库为主体的验证平台开发框架，验证工程师利用其可重用组件可以构建具有标准化层次结构和接口的功能验证环境[6-7]。

UVM验证平台的主要组件[8]包括：1）env，派生自uvm_env，它是一个容器，用于实例化scoreboard、reference model等组件。2）in_agent/out_agent，派生自uvm_agent，也是一个容器，用于实例化driver和monitor。由于在输出端口中不需要驱动任何信号，只需要监测信号，所以out_agent中只需要实例化monitor。3）monitor，派生自uvm_monitor，监测DUT的输入输出信号。用于收集DUT的端口数据，并将其转换成transaction交给后续的reference model和scoreboard组件进行处理。4）reference model，派生自uvm_component，用于完成和DUT相同的功能，一般采用高级语言描述，其输出被scoreboard接收，用于和DUT输出相比较。5）scoreboard，派生自

uvm_scoreboard，根据DUT的输出来判断DUT的行为是否与预期相符合，这里即对DUT的输出和reference model的输出进行比较。6）driver，派生自uvm_driver，负责驱动transaction，自身并不产生transaction。7）transaction，派生自uvm_sequence_item，是一个抽象的概念，用于描述各个组件间传递的信息，它是各个组件传递信息的基本数据结构。8）sequencer，派生自uvm_sequencer，用于检测仲裁队列中是否有某个sequence发送transaction的请求，以及检测driver是否申请transaction。9）sequence，派生自uvm_sequence，用于产生transaction，即产生激励。sequence用于创建一个transaction的实例，并将其送给sequencer。sequence机制是UVM中最重要的机制之一，包含sequence和sequencer。sequence用于产生transaction；sequencer则用于承载sequence产生的transaction并转交给driver使用。

5　功能验证平台设计与实现

5.1　UVM验证平台搭建

RapidIO交换芯片功能验证涉及到多种验证手段及验证功能点，这里主要关注与RapidIO 2.1协议相关的功能验证点，而与之相关的则为RapidIO单象限子系统验证。采用UVM搭建了基于SRIO VIP的RapidIO单象限子系统验证平台，其总体结构框图如图2所示，主要包括三个层次，自底向上依次为：Testbench、TestCase和Module Top。

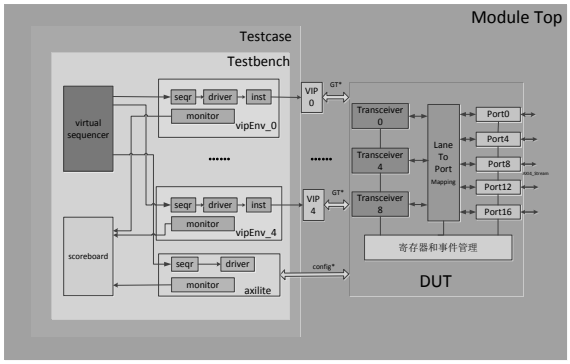


图2　RapidIO单象限子系统UVM验证平台总体结构

Figure2　RapidIO single quadrant subsystem UVM verificational the overall platform architecture

Testbench中包含了每个TestCase中的公有部分。为了便于与DUT顶层设计的信号进行对应，在RapidIO单象限子系统UVM验证平台的Testbench中，共设计了最多5个用于各个srio_port收发包的vipEnv_i（i=0，1，…，4），以及1个用于配置各个srio_port内部寄存器的axilite_i。这里代码中最终是几个srio_port，需要根据外部的配置情况进行动态设置。每个axilite_i和srioport_i中都包含各自的sequencer（图中简写为seqr）、driver和monitor，另外，对于srioport_i由于需要在外部和SRIO VIP进行连接，由VIP充当激励产生器，故这里还包含instance（简写为inst）。除了axilite_i和srioport_i之外，验证平台的Testbench中还包含virtual sequencer和scoreboard，virtual sequencer用于调度管理axilite_i和srioport_i的sequencer，在每个测试用例中与一个virtual sequence相对应。scoreboard根据各个axilite_i中monitor收集的配置信息，以及srioport_i中monitor收集的收发包信息，完成各个srioport收发包的统计和自动对比功能。在验证平台中，我们根据相关monitor收集到的配置信息和发包信息，计算出预期的收包信息，再与monitor实际收集到的收包信息进行自动对比，输出对比结果。这里没有单独使用reference model，因为已经把reference model的功能内嵌到scoreboard中。

TestCase定义了当前的测试用例与哪一个virtual sequence相对应。整个RapidIO单象限子系统UVM验证平台可以包含无穷多个TestCase，每个TestCase的类名必须不一样，每个TestCase对应的virtual sequence也要求不一样。

Module Top为子系统UVM验证平台的最顶层，包含了最多5个SRIO VIP，以及RapidIO单象限源代码DUT，并通过实例化将它们连接。通过仿真脚本中的语句“+UVM_TEST_NAME= XXX”来确定当前仿真执行哪一个TestCase。

5.2　内部通信设计

UVM中的通信是通过各种端口来实现的，端口包括PORT、EXPORT、IMP和analysis_port、analysis_export[8]。其中PORT、EXPORT、IMP的通信是一种一对一的通信，PORT和EXPORT端口只负责转发transaction，没有存储作用，除了转发操作之外不作其他操作；该transaction一定要由后续的某个组件进行处理，即IMP端口；然后IMP端口调用相关的任务（如put函数）对transaction进行处理。而analysis_port和analysis_export是

UVM中两种特殊的端口，可以连接多个IMP，是一种一对多的通信，更像一个广播。

除此之外，应用最多的是使用FIFO进行通信，FIFO的本质是一块缓存加两个IMP，如在monitor和scoreboard之间采用FIFO进行通信，monitor仍然为主动发送数据，socreboard接收数据的方式，由使用IMP被动地从agent接收变成使用get主动从FIFO获取，其可以按照自己的节奏工作。

RapidIO单象限子系统验证平台中，根据相关monitor收集到的配置信息和发包信息发送到scoreboard的FIFO中，然后在scoreboard中完成各个srioport收发包的统计和自动对比功能。monitor和scoreboard之间的各种端口的连接如图3所示。

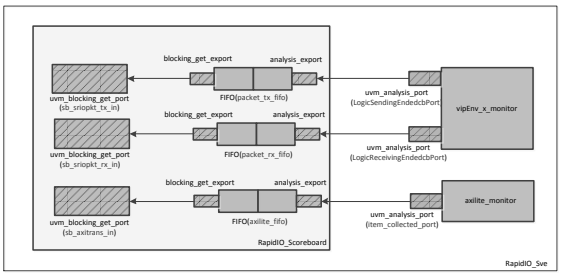


图3　内部通信连接示意图

Figure3　Schematic diagram of internal communication connection

这里涉及到UVM验证平台中的端口通信问题，需要强调的是，这里FIFO中的analysis_export和blocking_get_export虽然名字中有关键字export，但是其类型却是IMP。

5.3　基本数据结构transaction

transaction是UVM验证平台中各组件间流动的基本数据结构。RapidIO单象限子系统验证平台中包含两个transaction：axilite_trans和viptransceiver_transaction。

axilite_trans中定义了3个字段，axilite_addr、axilite_data和axilite_rd_wt，分别表示axilite总线操作的地址、数据和操作类型。通过对相关字段施加约束可以完成相应的配置操作。

viptransceiver_transaction继承于SRIO VIP的transaction，包括logical层、Physical层、Transport层三个层次的transaction字段定义。

5.4　Sequence的调用

验证平台中的Sequence与测试用例密切相关，根据验证平台的结构描述，验证平台中测试用例调用各Sequence的层次关系如图4所示。在run_subsrio脚本指定当前运行RapidIO_Test.sv中的某一个测试用例，在验证平台顶层会自动调用该测试用例；在RapidIO_Test.sv中指定当前的测试用例与哪一个virtual sequence相对应。

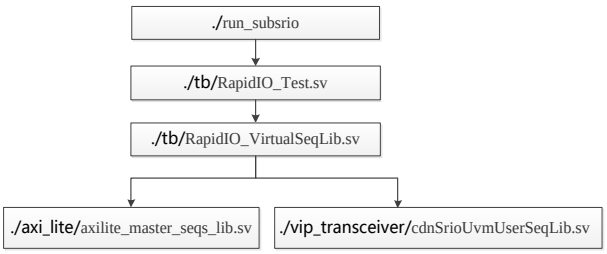


图4　测试用例调用各sequence的层次关系

Figure4　the test case calls the hierarchy of each sequence

在RapidIO_VirtualSeqLib.sv中，包含多个类，每个类都定义了一个测试用例的核心执行代码，它将axilite配置方式和srioport发包方式加载到具体的某个sequencer之上，如图5所示。在axilite_master_seqs_lib.sv中，定义axilite配置方式的sequence（cseq_xxx_1）；在cdnSrioUvmUserSeqLib.sv中，定义srioport发包方式的sequence（tseq_xxx）。

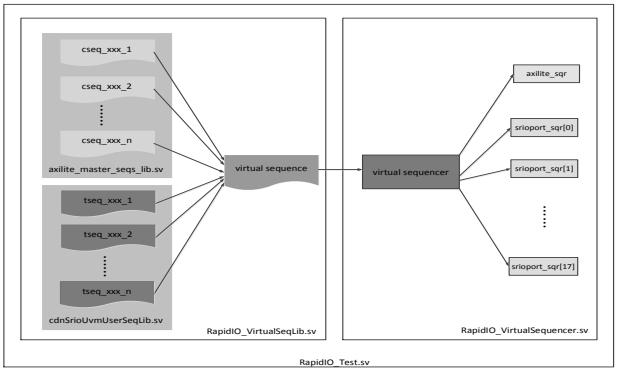


图5　virtual sequence的调用关系

Figure5　the invocation relationship of the virtual sequence

5.5　测试用例编写

UVM验证平台搭建完毕后，需要根据前期制定的验证功能点，编写相应的测试用例，用于验证RapidIO交换芯片的功能。新增加一个测试用例主要就是编写sequence类，包括如下步骤：

1）首先进行vip_transceiver中的sequence的编写，这里我们定义一个写序列操作即cdnSrioUvmUserNWriteSeq，其包字段包括如图6所示（除此

之外还有读序列、Doorbell、Message、维护包等操作可以设置）。该sequence位于cdnSrioUvmUserSeqLib.sv文件中。

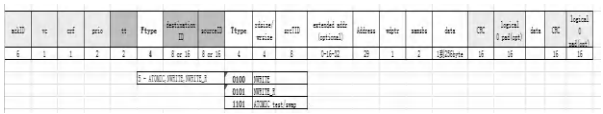


图6　Nwrite包字段

Figure6　Nwrite package fields

2）在RapidIO_VirtualSeqLib.sv文件中编写新的sequence类，该类实例化第一步中所编写的NWriteSeq，并在该类中确定控制他们的调用顺序。

3）编写一个新的测试用例的类mycase1_nwriteSeq，并通过config_db机制中的set函数调用第二步的sequence。

4）所有的sequence都编写完毕，就可以通过在仿真脚本中调用来实现仿真。这里在脚本run_subsrio中调用mycase1_nwriteSeq类进行一个写数据包的测试用例仿真。

以上是编写一个测试用例的步骤方法，按照此方法，根据验证功能点一览表，对端口控制器主通路，从包类型、控制符、IDLE、加解扰等，以及包路由、流控、包生成与包捕获、包追踪与包过滤等角度编写相应的测试用例，开展验证。

5.6　仿真验证与分析

在基于SRIO VIP的RapidIO交换芯片UVM验证平台上，根据实际需要进行了多种模式的不同组合配置，采用Xilinx的Transceiver，Serdes以及自主开发的代码进行了交叉组合，针对RapidIO的功能和协议一致性问题进行了详尽的仿真验证，包括数据包单播/多播功能，维护包功能，包路由、包生成与包捕获、包追踪与包过滤以及BUF等功能；以及RapidIO协议所定义的逻辑层、传输层和物理层三层分级体系结构所规定的全部功能。与Xilinx的Transceiver、Serdes进行不同组合，并配置为不同链路宽度、速率等，验证结果均满足协议一致性要求，具体见表1。通过对代码覆盖率和功能覆盖率[9]进行统计，统计结果显示代码覆盖率为可解释的100%，功能覆盖率为100%，满足验证要求。

表1　RapidIO交换电路多种模式测试结果

对接模式	通道速率	端口宽度	IDLE模式	测试结果
Serdes、Xilinx Transceiver、单端口	3.125G	1X	IDLE1	通过
	5G	4X		
	6.25G	2X		
	1.25G			
	2.5G	1X		
Serdes、自开发代码、单象限（5Port）	3.125G	2X	IDLE1	通过
	5G	4X		
	6.25G			

6　结束语

为更好的保证RapidIO交换芯片的协议一致性，确保芯片符合RapidIO 2.1协议，在项目验证过程中引入了Cadence公司的SRIO VIP，并搭建了相应的基于SRIO VIP的UVM验证平台，进行了大量测试用例的仿真验证，结果显示该芯片完全满足RapidIO 2.1协议，支持RapidIO 2.1协议定义的RapidIO互连特性，并兼容RapidIO 1.3协议，且能与通用RapidIO接口设备进行安全可靠的互连。采用基于VIP的RapidIO交换芯片功能验证，很好地验证并保证了RapidIO交换芯片的协议一致性，有效地提升了验证的效率和准确性，缩短了验证周期，为后续同类验证可提供参考借鉴。

参考文献:

[1] 郭炜. SoC设计方法与实现[M](3版). 北京：电子工业出版社，2017.

[2] 柴远波. 现代SoC设计技术[M]. 北京：电子工业出版社，2009.

[3] Cadence. Cadence SRIO VIP user guide[Z]，2015.

[4] Cadence. SRIO VIP user interface reference for UVM Systemverilog[Z]，2015.

[5] Cadence. VIP Setup Scripts Reference[Z]，2015.

[6] Accellera, Universal Verification Methodology (UVM) 1.1 User Guide[S], 2011.

[7] Ray S. The UVM Primer: An introduction to the Universal Verification Methodology[M]. Boston Light Press，2013.

[8] 张强. UVM实战[M]. 北京：机械工业出版社，2014.

[9] 罗莉, 何鸿君, 窦强,徐炜遐. 覆盖率驱动的芯片功能验证设计与实现[J]. 计算机工程与科学，2013，35(1)：36-40.

[作者简介]

张丽（通讯作者） 女，(1982-)，硕士，副研究员。研究方向：软件定义互连、SoC芯片设计。E-mail：yezizhang2003@126.com

宋克 男，（1976-），硕士，副研究员。研究方向：网络空间拟态防御、片上系统体系结构设计。

沈剑良 男，（1982-），博士，副研究员。研究方向：软件定义互连、网络安全、新型网络体系结构设计。

刘冬培 男，（1985-），博士，助理研究员。研究方向：软件定义互连、高效能计算、SoC芯片设计。

云计算环境下的资源分配策略研究

倪思源[1]，扈红超[2]，刘文彦[2]，梁浩[2]
[1]郑州大学，中原网络安全研究院，郑州 450000；
[2]战略支援部队信息工程大学，郑州 450000

摘　要：云计算凭借其按需索取、按需付费的优势广泛被使用。如何满足用户的需求、合理的分配云资源是当前研究的热点。本文首先介绍了云计算的研究背景，然后对云计算中现存的资源分配策略进行了分类并展开介绍和对比，最后总结了当前云计算资源分配面临的挑战以及未来的研究方向。
关键词：云计算、资源分配、虚拟化

Resource Allocation Strategy in Cloud Computing Environment

Ni Siyuan[1], Hu Hongchao[2], Liu Wenyan[2], Liang Hao[2]
1.Zhongyuan Network Security Research Institute, Zhengzhou University, Zhengzhou 450000, China;
2.Information Engineering University, Zhengzhou 450000, China

Abstract: Cloud computing is widely used by virtue of its advantages of requesting and paying on demand. How to meet the needs of users and allocate cloud resources reasonably is a hot topic of current research. This article first introduces the research background of cloud computing, then classifies, introduces and compares existing resource allocation strategies in cloud computing, and finally summarizes the current challenges facing cloud computing resource allocation and future research directions.
Key words: Cloud computing; resource allocation; virtualization

1　引言

云计算给我们的日常生活提供了便利的网络服务，如电子邮件、在线视频流和在线存储，近年来凭借其按需收费、即付即用的优势被广泛使用。云计算可提供对计算资源的廉价又便捷的访问，但是云提供商必须高效地管理、提供和分配这些资源。因此供应商必须通过的合理的分配机制，在用户需求的同时提高利润空间。由于互联网和云服务的使用率越来越高，传统的静态资源分配和管理机制已经无法适应用户的需求[1]，动态的资源分配得到了广泛的关注[2][3][4]。但是动态的资源分配在给用户带来便利的同时也引入了新的问题，并引起了广泛的研究。本文对现有的资源分配问题的解进行了分类与对比，并总结了资源分配所面临的问题与挑战，展望了未来云计算资源分配的研究方向。

2　云计算的背景简介

2.1　云计算服务模型

云计算的服务模型包含三种类型，如表2.1所示，分别是软件即服务（Software as a Service，SaaS）模型、平台即服务（Platform as a Service，PaaS）模型和架构即服务（Infrastructure as a Service，IaaS）模型[5][6][7]。

在软件即服务（Software as a Service， SaaS）模型中，云消费者不是使用本地运行的应用程序，而是使用云提供商在云基础架构上运行的软件服务。维护和管理云消费者使用的软件服务是云提供商的工作。云提供商可以根据软件的数量和使用时间来收费。Salesforge.com和阿里云等都采用SaaS模型。

基金项目：国家自然科学基金创新群体项目(61521003)，国家自然科学基金项目(62002383)

在平台即服务（Platform as a Service， PaaS）模型中，云平台提供了一个基于互联网的应用开发环境，开发人员可以在该环境上创建和部署应用程序。PaaS提供了可以运行应用程序和服务的平台。消费者不需要控制底层的云基础架构，包括网络、服务器、操作系统和存储单元，但可以控制已部署的应用程序。Google应用引擎、Microsoft Azure和RightScale等都采用了PaaS模型。

在架构即服务（Infrastructure as a Service，IaaS）模型中，云提供商管理大量计算资源，包括存储和处理能力。云消费者拥有对操作系统、存储单元、已部署的应用程序以及部分网络组件（例如主机防火墙）的控制能力。由于可以大大降低硬件成本，IaaS也被称作硬件即服务（HaaS）模型。Amazon Web Services、OpenStack和Eucalyptus等都采用了IaaS模型。

表2.1　云计算服务模型分类

服务模型	主要功能	主要特点	应用示例
SaaS模型	云消费者通过Web浏览器等其他客户端使用云提供商在云基础架构上运行的软件服务，根据软件的数量和使用时间来计费	客户不需要管理和控制云基础设施，客户仅需要对应用进行有限的、特殊的配置	Salesforge.com、腾讯云和阿里云等
PaaS模型	提供一个基于互联网的应用开发环境，包括应用编程接口和运行平台等（数据库、文件系统和应用环境等）	客户不需要管理和控制云基础设施，但可以控制已部署的应用程序	Google Cloud、Microsoft Azure和RightScale等
IaaS模型	提供大量计算资源、存储和处理能力，客户可以部署和运行任意软件，包括操作系统和应用软件等	客户不需要管理和控制云基础设施，但可以控制操作系统、存储单元、已部署的应用程序以及部分网络组件（例如主机防火墙）	Amazon Web Services、OpenStack和Eucalyptus等

2.1　云计算部署模型

在云计算中，根据物理位置和分布的变化采用了不同的部署模型，如图2.1所示，可以分为公有云（Public cloud）模型、私有云（Private cloud）模型、混合云（Hybrid cloud）模型和社区云（Community cloud）模型四种类型。

公有云（Public cloud）部署模型是一种为公共用户提供服务的方式，其中的所有云服务均以按需付费的方式提供给公众。企业和组织可以采用公有云部署模型来降低硬件和软件成本。由于开放的环境，公有云也面临着许多安全和性能问题，包括数据安全性、应用安全性和高扩展性等[8] [9] [10] [11]。IBM， Sun Cloud和Google应用引擎[8]等都采用了公有云部署模型。

私有云（Private cloud）部署模型是专门为终端用户或单个组织提供服务的，与公有云相比，具有更高的安全性和服务费用。私有云部署模型通常存在于组织内部，并不向公众提供服务。

混合云（Hybrid cloud）部署模型由两个或多个不同的云组成，这些云可以是公有云也可以是私有云[8] [9] [10]。利用混合云，可以实现差异化的数据存储，通过将敏感数据存储在私有云中，其他类型的数据存储在公有云中，提升了数据安全性，并节省了成本开销。

社区云（Community cloud）部署模型是一种为拥有共同利益的个人或公司提供云服务的方式，社区云的资源由所有社区用户共享。社区云部署模型也是通常存在于组织内部，并不向公众提供服务。

3　资源分配技术

3.1　资源分配技术分类

云计算中的资源分配是一种对云用户和应用程序间的可用资源进行有效分配的处理方式。资源分配是云计算应用中的挑战之一，特别是在基于IaaS的云计算服务模型中。应用资源分配技术以后，基于IaaS的云计算服务模型能够提供一下几点好处：1）效率高，用户无需安装或更新软硬件即可访问应用程序，具有低成本和高效性；2）灵活性强，用户可以通过云平台访问世界上任何系统的应用程序和数据资源；3）限制性低，通过云平台能够很容易的实现对多媒体和网站的使用。

云计算中的资源分配技术主要分为静态资源分配和动态资源分配两类。静态资源分配技术是一种将固定资源分配给云用户和应用程序的方式。在静态资源分配技术中，云用户预先知道应用程

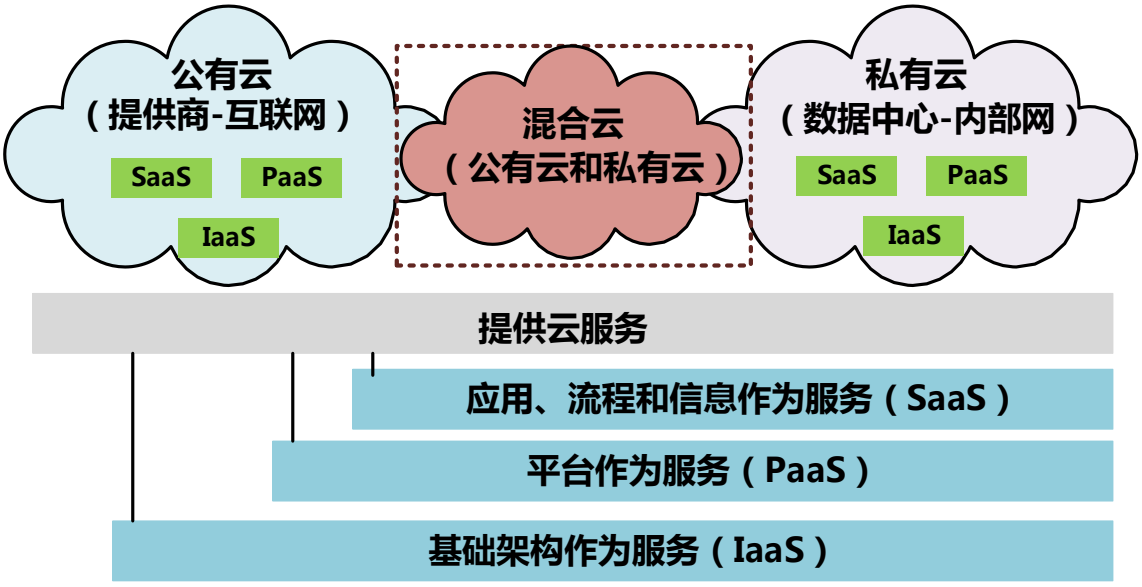


图2.1 云计算部署模型类型

序所需资源实例的数量以及请求资源的类型，并能确定应用程序的负载峰值。然而，由于资源分配方式固定，在静态资源分配技术中，应用程序存在对计算资源的过度利用或低效利用的问题，导致资源分配效率受限于非高峰时段的资源利用率，不具备经济效益优势[12][13]。

动态资源分配技术是一种按需分配资源给云用户和应用程序的方式，能够解决静态资源分配技术过度利用或低效利用计算资源的问题。然而，由于动态分配方式的局限性，可能存在访问失败的问题，云服务提供商必须从不同的云数据中心分配资源[14][15]。

3.2 资源分配技术的研究现状

当前对资源分配的研究大致分为以下五类：基于经济、基于感知、基于服务水平协议（SLA）和基于利用率的资源分配。

3.2.1 基于经济的资源分配

根据经济状况来合理划分云计算资源得到了广泛的关注与研究。许多公共IaaS提供商如Amazon EC2［16］针对动态的资源变化需求提出了由市场驱动的资源分配算法，云提供商基于拍卖机制以实现最大收益的同时最小化成本。

文献［17］介绍了一种基于市场的资源分配机制，该机制利用模型预测控制（MPC）算法解决了为不同虚拟机类型分配资源的问题。该机制通过给不同的服务水平划分不同的价格以满足客户需求，并最大限度地减少能源消耗。文献［18］改进了基于市场的分配，开发了一个基于双边组合拍卖的模型，不仅允许用户和提供商交易当前的服务，也允许用户和提供商提前交易未来的服务。文献［19］提出了一种基于密封竞价拍卖的云资源分配模型，用户向云服务提供商提出预期的价格，云服务提供商收集用户的出价并以此来确定价格。这种机制提供了资源的有效配置，但不能保证利润最大化。

3.2.2 基于感知的资源分配

基于感知的资源分配包括基于网络感知与基于能源感知。文献［20］提出了一种基于网络感知的分配方法。该方法考虑了多数据中心环境，并提出了处于不同数据中心环境下的虚拟机，对同一请求等待时间最短的算法，以及使数据中心间通信量和机架间通信量最小化的算法。 FairCloud［21］定义了三个云网络服务需求：最低保证，高利用率和网络比例。最低保证规定，云用户应该能够获得保证的最小带宽。高利用率要求在需要时使用可用带宽。网络比例性意味着带宽应该在云用户之间与其支付成比例地分配。该研究对以上三种需求进行了折中，并提出了相应的分配算法。

Beloglazov等［22］简要介绍了云计算中的能量感知研究。该研究定义了对节能云管理的基本原则，同时提出了一种虚拟机部署和动态迁移的算法，以降低利用CPU时引起的能耗的。然而现存的对能量感知的研究大多集中在CPU功耗上。

Ye等人［23］在此基础上考虑了网络资源的能耗，同时最大限度地提高了负载平衡和资源利用率。

3.2.3 基于服务水平协议(SLA)的资源分配

SLA是一种规定服务提供者和消费者之间QoS的协议［24］。Popovici等人根据SaaS提供商的QoS参数（如提供的负载和价格）提出了一种资源分配模型［25］。但该分配模型并没有考虑用户的需求。对于多云环境，Soodeh Farokhi［26］从SaaS级别、约定的SLA和服务提供商条件的角度，提出了一种针对多云环境的资源分配的模型。提出的模型使用服务提供商的QoS参数进行SLA违规检测和监视。

很少有模型同时考虑云提供商和用户双方。Wu等人［27］提出了一种在最小化违反SLA的同时最小化基础设施成本的资源分配算法。该算法通过合并管理相似的虚拟机，节约了50%的成本。

Garg等［28］训练了一个人工神经网络，通过预测应用程序的工作量，来相应地分配虚拟机。并根据实时监视资源使用情况，将虚拟机迁移到更好的服务器来解决由于预测错误而导致的任何违反服务级别协议的情况。该研究仅考虑了CPU的资源分配。

3.2.4 基于利用率的资源分配

动态管理虚拟机可以克服由于将固定资源分配给应用程序和服务而导致的资源利用不足，最大限度地利用资源并最小化成本。Lin等人［29］设计一个根据实际需求调整虚拟机的算法。该算法通过监测和预测应用的需求，以实现提高资源利用率、降低成本的目的。Pandit等人［30］提出了一种基于模拟退火的资源分配算法。采用多参数装箱算法以减少资源的浪费。该模型提高了云计算资源利用率，降低了成本。Gabay和Zaourar［31］根据启发式算法将云计算中不同类型的资源抽象为维度，提出了一种多为维资源分配算法。Generi［32］在此基础上针对云基础设施异构性，提出了一种异构云基础设施资源分配算法，提高了异构云服务器间的负载平衡，提高了资源利用率。

如表3.2所示，本文对比了上文中提到的资源分配策略所使用的方法、优势和缺点。

3.3 资源分配技术面临的挑战

云计算中的资源分配技术的研究仍处于早起阶段，在研究不断进步的同时，一些现有问题尚未得到充分解决，下面对资源分配技术面临的挑战进行讨论。

1）虚拟机迁移问题：当用户需要切换到另一个云提供商以获得更好的数据存储时，将面临虚拟机迁移问题。

2）资源控制问题：由于资源是用户从远程服务器租用的，因此通常缺乏对资源的控制机制。

3）能耗效率问题：大型的云数据中心在具备各类计算操作功能的同时，也需要消耗大量电能，这将导致大量的碳排放，造成环境污染问题。

4）并行作业的调度问题：计算领域中的并行作业包括非独立作业和独立作业两种类型，前者需要提前协商一致，后者可以同时使用多个虚拟机。

5）降低成本并最大程度地利用资源：重要的是要处理在云运营成本方面的问题，需要在满足的资源分配的同时，最大限度地利用所有资源。换句话说，服务提供商必须为用户提供低成本服务。

6）保持高可用性：如果某项工作需要长时间运行，需要进行数小时的计算，那么则必须确保云中资源的可用性。因此，需要一些技术来自动处理资源中的任何中断，并将作业切换到可用资源。

7）弹性问题：在云计算中，弹性是指可以动态处理资源需求的程度。对资源的需求可能会随时间的增加而增加，云服务商应该具备自动检测需求的能力，以及确定满足这些需求所需的必要资源。

3.4 资源分配技术未来研究方向

关于资源分配问题，已经进行了大量研究，并在云计算领域提出了许多解决方案。但是，仍然存在一些问题和挑战需要进一步研究，并且尚未找到适用于大多数云环境的最佳解决方案。

根据先前研究的文献，得出如下挑战发现：

1）在最大化资源分配利用率时，由于大多数模型都会影响QoS，有必要减少用户违反SLA的行为，需要能够降低成本并保持较高效率的资源分配算法。

2）需要一种适用于不同云环境的资源分配框架，以减轻异构云中分配的复杂性。

表3.2　云环境中各种资源分配技术及其优缺点之间的比较

文献	使用的方法	优势	缺点
[17]	模型预测控制(MPC)算法	它提供了最大化的收入,以满足客户的期望。它使能耗最小化。	缺乏对未来利润的预测。
[18]	基于双边组合拍卖的模型	允许用户和提供商交易当前和未来的服务。	资源管理较为复杂
[19]	密封竞价	提高资源利用率	不能保证利润最大化
[20]	基于网络感知	最小化通信量	缺乏处理动态资源请求的能力。
[21]	多目标约束分配网络资源	兼顾到三种网络资源分配需求	资源利用率不高
[22]	基于能源感知	降低CPU能耗	未考虑网络资源
[23]	基于能源感知与网络资源感知	提高了负载平衡率与资源利用率	管理较为复杂
[25]	多维资源分配(MDRA)模式	提高了资源利用率,降低了数据中心的成本。	当用户需求增加时,节能效率低下
[26]	为多云系统中的SaaS提供商提供服务	在检测违规的同时,找到最能满足用户需求的基础设施资源。	未考虑延迟
[27]	针对SaaS提供商提出的资源分配算法	降低了SaaS提供商的成本和SLA违规的数量。	算法的总体性能较低,客户满意度不高。
[28]	通过神经网络预测要分配到资源	动态迁移虚拟机,减少SLA协议的违反	仅考虑CPU
[29]	一个使用CloudSim工具包的基于阈值的系统资源动态分配	可以最大限度地利用资源并将成本降至最低	缺乏对物理资源开销的考虑
[30]	模拟退火的装箱算法	有助于降低成本,解决多层次的资源分配问题	缺乏处理动态资源请求的能力。
[31]	异构装箱算法解决资源分配问题	正式定义了异构装箱问题,解决了异构云资源分配问题。	负载平衡率较低
[32]	根据服务器与虚拟机间的相似性,选择虚拟机的部署位置	提出了一种判断资源利用率的指标,提高了异构云的资源利用率	未考虑动态的部署虚拟机

3）资源分配需要将云消费者的成本降至最低，并为云提供商带来最大的利润。对于云提供商而言，对有限资源的有效利用和管理非常重要。

4）需要考虑云资源中的负载平衡并以最佳方式调度工作负载，以便满足用户的QoS要求并通过增强资源的使用来最大化利润。

云计算中资源分配的未来研究方向应解决上述每个挑战，并尝试实施最佳实践模型。

4　结束语

本文针对当前云计算中的资源分配问题展开了研究，首先将资源分配技术分为动态资源分配与静态资源分配，然后对比分析了基于经济、基于感知、基于服务水平协议（SLA）和基于利用率的资源分配策略，最后对当前资源分配面临的挑战进行了分析，并展望了资源分配技术的未来研究方向。

参考文献:

[1] Li J, Qiu M, Niu J W, et al. Adaptive resource allocation for preemptable jobs in cloud systems [C]//2010 10th International Conference on Intelligent Systems Design and Applications. IEEE，2010：31-36.

[2] Naha R K, Garg S, Chan A, et al. Deadline-based dynamic resource allocation and provisioning algorithms in fog-cloud environment[J]. Future Generation Computer Systems，2020，104：131-141.

[3] Tseng F H, Wang X, Chou L D, et al. Dynamic resource prediction and allocation for cloud data center using the multiobjective genetic algorithm[J]. IEEE Systems Journal，2017，12(2)：1688-1699.

[4] Yuan X, Min G, Yang L T, et al. A game theory-based dynamic resource allocation strategy in geo-distributed datacenter clouds[J]. Future Generation Computer Systems，2017，76：63-72.

[5] Mukundha C, Vidyamadhuri K. Cloud computing models: a survey [J]. Adv. Comput. Sci. Technol.，2017，10(5)：747-761.

[6] Muñoz-Escoí F D, Bernabéu-Aubán J M. A survey on elasticity management in PaaS systems [J]. Computing，2017，99（7）：617-656.

[7] Bhamare D, Jain R, Samaka M, et al. A survey on service function chaining[J]. Journal of Network and Computer Applications，2016，75：138-155.

[8] Mell P, Grance T. The NIST definition of cloud computing (draft) [J]. NIST special publication，2011，800(145)：7.

[9] Hu P, Dhelim S, Ning H, et al. Survey on fog computing: architecture, key technologies, applications and open issues [J]. Journal of network and computer applications，2017，98：27-42.

[10] Zissis D, Lekkas D. Addressing cloud computing security issues[J]. Future Generation computer systems，2012，28(3)：583-592.

[11] Stein L D. The case for cloud computing in genome informatics[J]. Genome biology，2010，11(5)：207.

[12] Sonkar S K, Kharat M U. A review on resource allocation and VM scheduling techniques and a model for efficient resource management in cloud computing environment[C]//2016 International Conference on ICT in Business Industry & Government (ICTBIG). IEEE，2016：1-7.

[13] Ge Y, Ding Z, Tang M, et al. Resource Provisioning for MapReduce Computation in Cloud Container Environment[C]//2019 IEEE 18th International Symposium on Network Computing and Applications (NCA). IEEE，2019：1-4.

[14] Xiao Z, Song W, Chen Q. Dynamic resource allocation using virtual machines for cloud computing environment[J]. IEEE transactions on parallel and distributed systems，2012，24(6)：1107-1117.

[15] Lyazidi M Y, Aitsaadi N, Langar R. Dynamic resource allocation for Cloud-RAN in LTE with real-time BBU/RRH assignment[C]//2016 IEEE international conference on communications (ICC). IEEE，2016：1-6.

[16] Amazon E C. Amazon web services [J]. Available in: http://aws.amazon. com/es/ec2/(November 2012)，2015.

[17] Zhang Q, Zhu Q, Boutaba R. Dynamic resource allocation for spot markets in cloud computing environments [C]//2011 Fourth IEEE International Conference on Utility and Cloud Computing. IEEE，2011：178-185.

[18] Melissaris T, Anagnostopoulos I, Soudris D, et al. Agora: Agent and market-based resource management for many-core systems[C]//2016 IEEE International Conference on Electronics，Circuits and Systems (ICECS). IEEE，2016：400-403.

[19] Hosseinalipour S, Dai H. A two-stage auction mechanism for cloud resource allocation [J]. IEEE Transactions on Cloud Computing，2019.

[20] Santos J, Wauters T, Volckaert B, et al. Towards network-aware resource provisioning in kubernetes for fog computing applications [C]//2019 IEEE Conference on Network Softwarization (NetSoft). IEEE，2019：351-359.

[21] Popa L, Kumar G, Chowdhury M, et al. FairCloud: Sharing the network in cloud computing [C]//Proceedings of the ACM SIGCOMM 2012 conference on Applications，technologies，architectures，and protocols for computer communication. 2012：187-198.

[22] Beloglazov A, Abawajy J, Buyya R. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing[J]. Future generation computer systems，2012，28(5)：755-768.

[23] Son S, Jung G, Jun S C. An SLA-based cloud computing that facilitates resource allocation in the distributed data centers of a cloud provider [J]. The Journal of Supercomputing，2013，64(2)：606-637.

[24] Popovici F I, Wilkes J. Profitable services in an uncertain world[C]//SC'05: Proceedings of the 2005 ACM/IEEE conference on Supercomputing. IEEE，2005：36-36.

[25] Farokhi S. Towards an SLA-based service allocation in multi-cloud environments [C]//2014 14th IEEE/ACM International Symposium on Cluster，Cloud and Grid Computing. IEEE，2014：591-594.

[26] Wu L, Garg S K, Buyya R. SLA-based resource allocation for software as a service provider (SaaS) in cloud computing environments[C]//2011 11th IEEE/ACM International Symposium on Cluster，Cloud and Grid Computing. IEEE，2011：195-204.

[27] Garg S K, Toosi A N, Gopalaiyengar S K, et al. SLA-based virtual machine management for heterogeneous workloads in a cloud datacenter [J]. Journal of Network and Computer Applications，2014，45：108-120.

[28] Sharma N, Maurya S. SLA-Based Agile VM Management in Cloud & Datacenter [C]//2019 International Conference on Machine Learning，Big Data，Cloud and Parallel Computing (COMITCon). IEEE，2019：252-257.

[29] Seth S, Singh N. Dynamic threshold-based dynamic resource allocation using multiple VM migration for cloud computing systems [C]//International Conference on Information，Communication and Computing Technology. Springer，Singapore，2017：106-116.

[30] Pandit D, Chattopadhyay S, Chattopadhyay M, et al. Resource allocation in cloud using simulated annealing[C]//2014 Applications and Innovations in Mobile Computing (AIMoC). IEEE，2014：21-27.

[31] Gabay M, Zaourar S. Vector bin packing with heterogeneous bins: application to the machine reassignment problem [J]. Annals of Operations Research，2016，242(1)：161-194.

[32] Mergenci C, Korpeoglu I. Generic resource allocation metrics and methods for heterogeneous cloud infrastructures [J]. Journal of Network and Computer Applications，2019，146：102413.

[作者简介]

倪思源（1995年），女，硕士，研究方向为：云计算、网络安全。

扈红超（1982），男，博士，，国家数字交换系统工程技术研究中心研究员，研究方向为：云计算、网络安全。

基于DPDK的P4软件交换机优化研究

张雪，王洪超，高德云，北京交通大学下一代互联网互联设备国家工程实验室

摘　要：软件定义网络（Software Defined Network， SDN）及可编程协议无关报文处理（Programming Protocol-Independent Packet Processors， P4）协议的提出促进了P4交换机的发展与创新，研究学者相继提出了基于硬件平台的P4交换机、基于X86架构软件平台的P4交换机等方案。但这些P4交换机存在着一定的缺点或不足，基于硬件平台的P4交换机缺乏可移植性，而基于X86架构软件平台的P4交换机转发速度较慢。因此，对P4交换机的优化及加速成为了当前P4交换机的一个重要研究课题。本文提出了BMv2（Behavioral Model Version 2）双层架构加速优化方案，使用数据平面加速开发套件（Data Plane Development Kit， DPDK）实现高速数据报文收发，架构下层为DPDK层，实现可移植功能，架构上层为CORE层，实现通用及高级定义功能。实验结果表明，在三种测试场景下，改进的P4交换机端到端时延性能提升了3倍，64Bytes至512Bytes长度的数据报文转发性能都达到了1000Mbps，吞吐量大大提升。

关键词：软件定义网络、P4软件交换机、DPDK、加速优化、时延、吞吐量

Research on Optimization of P4 Software Switch Based on DPDK

ZHANG Xue, QUAN Wei, QIN Shuai, WANG Hong-Chao

Abstract: The introduction of Software Defined Network (SDN) and Programming Protocol-Independent Packet Processors (P4) protocols has promoted the development and innovation of P4 switches. Researchers scholars have successively proposed hardware-based platforms P4 switches and P4 switches based on the X86 architecture software platform, etc. However, these P4 switches have certain shortcomings or deficiencies. The P4 switches based on the hardware platform lack portability, while the P4 switches based on the X86 architecture software platform have a slow forwarding speed. Therefore, the optimization and acceleration of P4 switches have become an important research topic of current P4 switches. To this end, this paper proposes a BMv2 (Behavioral Model Version 2) two-layer architecture acceleration optimization solution, using the Data Plane Development Kit (DPDK) to achieve high-speed data message transmission. The lower layer of the architecture is the DPDK layer, which implements the portable function, and the upper layer of the architecture is the CORE layer, which realizes the general and advanced definition functions. The experimental results show that in the three test scenarios, the end-to-end delay performance of the improved P4 switch is increased by 3 times, and the forwarding performance of data packets from 64Bytes to 512Bytes in length reaches 1000Mbps, and the throughput is greatly improved.

Key words: Software Defined Network; P4 software switch; Data Plane Development Kit; acceleration optimization; delay; throughput

1　引言

可编程协议无关报文处理语言P4[1-2]作为数据平面特定领域编程语言，具有可重配置性、协议无关性、平台无关性三大特点，它的出现解决了OpenFlow数据平面可扩展性差的问题。借助P4语言，P4交换机在实现传统交换机功能的基础上进一步支持了VxLAN[3]、RCP[4]等新兴网络协议。不仅如此，P4交换机还可以支持带内网络遥测、分布式计算、大流检测、带状态负载均衡[5-6]等复

基金项目：国家重点研发计划-战略性国际科技创新合作重点专项(2018YFE0206800)。

杂功能。

P4语言的提出与可编程数据平面的发展，使自定义数据平面报文处理逻辑、网络设备灵活支持各种新协议和新功能等目标成为可能[7]。但对于如何使现有台支持P4语言[8]、如何设计并优化编译器等问题，P4语言联盟并没有给出相应解决方案，因此学术界对P4语言的落地和优化作出了一系列研究[9]。其中P4语言联盟的BMv2交换机模型提供了一种P4行为模型。但BMv2存在着诸多的问题，如使用libpcap进行收发包、使用Linux网卡驱动、单一线程、单一收发队列、不必要的内存拷贝等，这些问题导致BMv2交换机性能相比于OVS等其他软件交换机，性能较差，达不到商用交换机的要求。

为提高P4软件交换机的性能，本文使用数据平面加速套件对BMv2交换机进行改进，进行P4软件交换机加速研究，最终提出了一种基于DPDK改进的交换机BMv2_DPDK。文章的主要工作包括以下三个部分：

（1） 在深入研究BMv2交换机程序结构后，对其进行分层归类，提出双层架构程序模型。

（2） 架构上层实现了P4语言数据报文处理模型，将有锁队列优化为无锁队列，队列性能提升一倍。

（3） 架构下层使用DPDK实现高速数据报文收发，将相关代码编译成动态链接库（Dynamic-link Library， DLL），通过将动态链接库封装成类，向上层提供接口并屏蔽底层硬件实现。

本文第2节介绍P4交换机一些相关研究与工作；第3节进行P4软件交换机加速研究，详细阐述基于DPDK的P4软件交换机加速优化设计。第4节是实验性能与分析，测试分析了BMv2_DPDK与BMv2交换机在不同使用场景下的性能。第5节对工作进行总结与展望。

2 相关工作

学术界和工业界对P4交换机的研究可分为硬件平台实现与软件平台实现两方面。硬件平台包括可编程交换机、FPGA、GPU等，软件平台包括基于X86架构硬件实现的软件交换机等。硬件平台实现与软件平台实现有各自不同的特点，如硬件平台中的可编程交换机的性能较高，软件平台中的软件交换机则具有更强的灵活性。由于虚拟机通常使用软件交换机来完成互相通信，因此P4软件交换机拥有良好的应用前景。目前有不少的研究工作尝试通过P4语言编译器设计实现软件平台中的数据包处理逻辑。软件交换机的实现基于大量的内核相关代码，如果想要对软件交换机进行更改或升级，需要对软件交换机的底层硬件平台及系统内核代码等实现细节有相当深入的了解。

在FPGA方面，Wang H等人设计了适用于FPGA的P4报文解析模型以及相应的代码转换编译器[10-11]；Benaček等人基于P4语言数据报文处理模型，以流水线方式构成适用于FPGA的报文解析模型，所设计的编译器可以将P4语言代码转化为能够在FPGA上运行的VHDL硬件语言代码[12-13]。

在GPU方面，有研究者将P4程序部署在GPU与CPU组成的混合架构上[14-15]，实验结果表明该系统可以实现较高的查找速度与较小的报文延迟。

在软件平台方面，有研究者基于如今应用较广的软件交换机Open vSwitc（简称OVS）[16]提出了可编程的、协议无关的软件交换机架构PISCES[17]，PISCES交换机中实现相同功能的代码量比OVS交换机节省约40倍；针对基于x86 CPU架构开发的高性能数据包处理软件平台VPP（Vector Packet Processing），有研究者设计并实现了PVPP[18-19]编译器，实验结果表明，使用VPP运行P4程序具有较高的并行处理能力和较高的可扩展性；还有研究者提出了基于主动队列管理（AQM）的P4软件交换机T4P4S[20]，其科研团队的实验测试结果表明，T4P4S在时延及带宽上具有较高的二层转发及三层路由性能。

虽然基于FPGA及GPU硬件的P4编译器及解决方案拥有较高的性能，但是目前将P4语言代码编译成FPGA硬件编程语言具有设计调试难度大、部分代码库不可移植等缺点，基于CPU与GPU的混合架构对于系统也出现兼容性较差，无法适用X86架构的系统等问题。另一方面，基于Linux系统软件平台实现的PISCES、P4VPP、T4P4S交换机的底层硬件实现都是基于DPDK进行设计，由此可见，通过软件平台实现的P4交换机使用DPDK作为收发包程序是发展趋势。而目前因商业化的原因，PISCES交换机及PVPP交换机已经不是开源项目。因此本文选择使用对P4语言有较好支

持的BMv2交换机，并且使用DPDK替换BMv2交换机中libpcap收发包套件，实现底层硬件控制，达到BMv2交换机加速优化的目的。

3 基于DPDK的P4软件交换机优化研究

DPDK（Intel Data Plane Development Kit）是英特尔（Intel）为网络开发人员提供的数据平面加速开发套件[21]，其提供Intel architecture（IA）处理器架构机器下高速数据包处理驱动及库函数。与一般数据包收发套件不同，DPDK以用户态运行在用户空间，绕过了内核态的Linux内核协议栈，通过多种技术手段实现高速数据包的处理，实现在IA处理器平台下多种不同的应用，如数据包处理、控制处理[22-23]等。有许多研究将DPDK应用在网络设备[24-25]，因为DPDK相比于libpcap等数据包处理驱动，在性能上有较大的优势，故考虑使用DPDK对BMv2交换机进行加速，提升BMv2带宽，降低其延时，从而达到提高BMv2交换机性能的目的，将使用DPDK优化加速后的BMv2交换机称为BMv2_DPDK交换机。

文章引入双层架构，将交换机分为CORE层和DPDK两层实现。其中DPDK层做为架构下层，向下驱动底层网卡，向上屏蔽下层操作细节，满足可移植设计原则；CORE层做为架构上层负责流水线管理、P4程序中的动作匹配等数据包处理操作，满足通用、高级定义的设计原则。

3.1 BMv2_DPDK子类设计

当前BMv2代码使用C++编写，耦合较高，通过构造多种类以较小的颗粒度来实现P4管道逻辑。在设计BMv2_DPDK双层架构代码时首先需要对构成交换机的基类进行双层架构的解耦。通过对BMv2交换机程序分析与研究，其主要包括以下几类：

（1）simple_switch类：通过维护队列及入口处理线程、出口处理线程及数据包发送线程实现P4语言数据包处理模型；

（2）switch类：为simple_switch类的父类，进行交换机初始化（如数据包出口，数据包生存时间），进行函数接口初始化，设置数据包计数器等；

（3）dev_mgr类：为switch类的父类，进行数据报文的发送和接收与端口管理；

（4）runtime_interface类：为switch类的父类，进行BMv2 Runtime的交互（如查询数据包入口数量、获取数据包类型等）；

（5）Queue类：进行再次封装，使用mutex锁减少死锁或程序异常状态的出现；

（6）Packet类：实现对数据包的封装；

（7）PHV类：基于P4语言，存储数据包解析内容。

基于以上BMv2交换机程序的分析与总结，根据其继承关系及功能抽象将各个子类划分在DPDK层或CORE层。如果类的内容更接近底层硬件级别的概念，与网卡驱动相关或者能够实现跨平台移植，则归类为DPDK层。若涉及到P4数据包处理模型或转发处理逻辑，则归类为CORE层。文章分类结果如下图1。以虚线作为抽象级别阈值，分为CORE层与DPDK层，越往下类的抽象程度越低，与硬件的结合程度越高。

3.2 DPDK层优化设计

在优化程序设计中，DPDK层负责网卡驱动、DPDK初始化、提供应用程序接口等功能，包括Send_packet类、Receive_packet类等。DPDK使用C语言的多线程编程，BMv2使用C++编写。故将DPDK库封装成动态链接库进行调用，DPDK_API动态链接库直接使用DPDK驱动网卡，主要包括接收和发送数据包、从数据包读取数据用以流表匹配、更改数据包实现转发功能以及为BMv2提供接口等功能。之后再使用C++类调用DPDK_API动态链接库中的函数封装成DPDK层类。DPDK层结构如图2所示。

优化后DPDK程序通过动态链接库中实现队列数据报文的高速收发过程如下：

（1）EAL层（Environment Abstraction Layer，底层资源的抽象层）初始化，注册中断信号处理函数。

（2）初始化逻辑内核，为每个逻辑内核绑定端口（网卡），且每个端口只对应一个逻辑内核。

（3）初始化内网卡，检查网卡类型及连接状态。

（4）完成初始化之后，启动逻辑内核上的收发线程（主内核除外），开始进行数据报文收发。通过调用rte_eth_rx_burst（）接口函数接收数据报文，通过调用CORE层中receive（）回调函数将数

图1　BMv2_DPDK子类设计

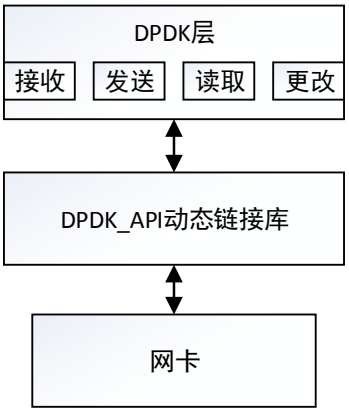


图2　DPDK层结构

据报文传入CORE层处理。通过逻辑内核发送队列中的数据报文，若是CORE层使用突发方式发送数据报文则不须经过逻辑内核处理。最后检查是否传入SIGINT信号，若传入SIGINT信号，则各个逻辑内核退出循环，反之则进入下一次数据报文收发。

3.3　CORE优化设计

CORE层负责实现与硬件无关的抽象P4逻辑转发，转发逻辑只受P4程序影响，由P4编译器生成的JSON代码控制，因此CORE层代码需要具有可靠性。头类型和头文件都有基本类声明，状态寄存器的默认配置、查找表以及解析结果，都可通过P4 Runtime模块进行读取。在P4数据包处理模型中，需要对数据包进行解析与逆解析。标头的类型结构从数据包中提取出，用于控制引流或计算时需要，其所在字段及大小在程序中枚举给出，并且在数据包的提取过程中需要将其从网络字节序转换到主机字节序。CORE层负责实现P4程序中数据包的动作匹配和数据流的控制等操作，而实际的数据包存储及查找在DPDK层中实现。

完成DPDK动态库链接后，对BMv2_DPDK程序进行测试发现程序出现跑飞及死锁现象。经过调试分析到是由于DPDK收包速度过快而交换机程序队列传输速度较慢，导致程序出现错误。因此对BMv2队列进行测试，在测试程序中创建两个数据包收发线程并使用一个队列进行连接。队列中保存整数类型数据，一个线程将数据压入队列，另一个线程从队列弹出数据。通过时间戳和压入弹出的总数据量，得到在两个线程之间通过push和pop传输单个项目的时间为50ns。

当前BMv2队列使用有锁队列存在两个设计问题：（1）每个队列实例中都存在互斥锁，当多个线程同时争用，会产生高昂的时间代价，即使没有争用，线程获得互斥锁的时间代价也在几十纳秒级别；（2）队列中还定义了notify变量，在每次push及pop操作时都会通知另一等待的线程，线程的notify的时间花销也比较大。

为了解决以上问题，对CORE层进行了无锁队列优化，push始终在队列前面操作，pop始终在队列后面操作，使用固定大小的数组及两个索引，其中一个索引指示队列生产者的下一个可用插槽，另一索引指示队列消费者的下一个已就绪的插槽。该数组作为循环数组，消费者索引永远不能超过生产者索引，当两个索引重合，表示数组为空，当消费者索引正好位于生产者索引前，表示数组已满。除了生产者索引和消费者索引外，生产者

和消费者各自还有一个事件索引。此事件索引用于通知其他线程当前线程需要被通知的时间。例如当生产者线程速度过快线程已满时，生产者索引值为队列中间，则生产者线程进入等待状态，且将在队列为半空时得到通知。如下图3展示了事件索引的触发过程。

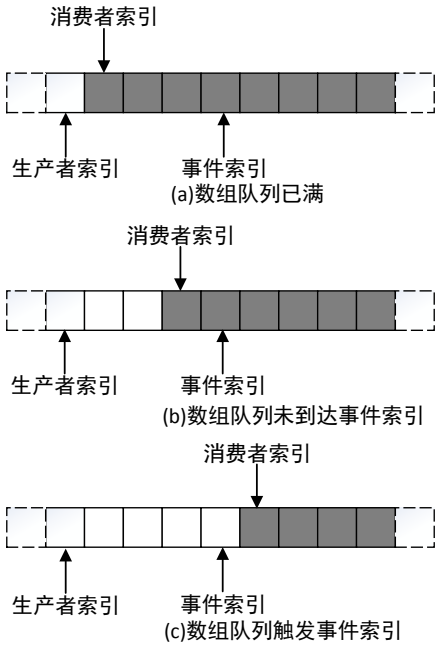


图3　数组队列事件索引触发过程

使用双线程对改进的队列进行测试得到一个项目经过压入和弹出需要的时间为40ns。在测试过程中记录队列通知的次数，得到生产者线程发送每个数据包的平均通知数为0.6，而消费者线程接收每个数据包发送的通知数为1.6*10^（-6）。从测试结果可以看出，若能减少生产者线程的通知数，队列的速度还可以进一步得到提升。由于生产者线程给消费者线程发送通知的数量与消费者线程在空队列条件下的等待次数成正比，因次本文采用的解决方法为：当队列为空时，让消费者线程等待数个时间，再次检查队列是否为空，若队列还为空状态，则消费者线程进入等待状态，等待生产者线程发送通知。通过采取再次检查队列的策略，生产者线程压入每个数据包的平均通知数量大大减少。其中，生产者线程每个数据包的平均通知数量为1.2*10^（-6），消费者线程每个数据包的平均通知数量为8*10^（-4）。

再次对优化过的无锁队列进行测试，得到每个项目压入和弹出需要的平均时间为25ns，相比于初始队列的50ns有了很大的提升，同时也解决了交换机运行时因队列速度过慢而产生的跑飞、死锁问题。

4　实验与性能分析

本章测试分析了BMv2_DPDK与BMv2交换机在二层转发、三层路由、负载均衡三种不同使用场景下的性能，进行测试的数据报文为64Bytes、128Bytes、256Bytes及512Bytes，包含最短MAC头部及IPv4头部，使用随机生成的数据补齐数据报文。测试结果显示，BMv2_DPDK交换机在多个场景下的性能都优于BMv2交换机。

4.1　实验环境搭建

首先在服务器上分别安装BMv2及BMv2_DPDK交换机，并且使用流量生成器和流量接收器连接交换机，搭建网络测试拓扑，测试不同场景下交换机的性能。

流量生成器及测试用服务器使用如下表1所示的配置。对每个服务器都进行DPDK绑定和Intel千兆网卡的配置。由于传统的netperf/iperf工具在64Bytes的小包下，数据包收发速度很难达到千兆，所以需要寻找其他网络测试工具来进行吞吐量测试。最终经过多方的选择比对，本文使用pktgen-dpdk [26] 网络测试工具。pktgen-dpdk是一个基于DPDK开发的网络性能测试套件，与iperf，Qperf等一般网络测试工具相比较，pktgen-dpdk产生封包的速度更快。另外由于pktgen-dpdk基于DPDK进行开发并且运行在用户空间，所以它能更精准的计算出网络的性能，也可以对网络进行更大的压力测试。

表1　服务器配置

服务器硬件	硬件版本
核心处理器(CPU)	Intel(R) Xeon(R) E5-1660 v4 3.2GHz
内存	2×4GB DDR4 SDRAM
系统	Ubuntu 16.04 Desktop
网卡	Intel 82580eb I340-T4 千兆网卡
流量测试软件	pktgen-dpdk 3.4.0

4.2　实验拓扑设计

本文搭建了三种不同模型用于测试BMv2_DPDK及BMv2的性能：二层转发、三层转发以及简单的负载均衡。

A. 简单的二层转发拓扑

交换机使用相同的二层转发 P4 代码，在交换机内部维护一张目的 MAC 表用于报文匹配，将数据报文通过匹配到的端口转发出去，如果传入的目的 MAC 地址未知则丢弃相应的数据报文。

B. 三层路由拓扑

交换机使用相同的三层路由 P4 代码，测试软件发送 IPv4 数据报文，交换机使用最长前缀匹配规则对接收到的数据报文进行 IP 匹配，更换数据报文的源 MAC 和目的 MAC 并通过对应的网口转发出去，在这种情况下需要用到两个流表，交换机根据一个流表对数据包 IP 进行最长前缀匹配，返回下一跳的标识符并递减数据包中的 TTL 字段，根据第二个表下一跳标识符更新源 MAC 及目的 MAC，通过对应的端口转发。

C. 简单的负载均衡拓扑

通过模仿 web 前端的功能实现简单的负载均衡，交换机将流量根据不同 IP 分配给不同的后端服务器，交换机使用相同的负载均衡 P4 代码，交换机收到的流量依据数据包源 IP 进行分配，负载均衡管道是依据两个流表进行转发，先是根据最长前缀匹配对数据包 IP 进行匹配以确定后端服务器，然后根据后端服务器设置目标 MAC 及转发端口。

4.3 实验结果分析

本文首先测试交换机的端到端时延，再使用 dpdk-pktgen 生成数据包流量测试交换机在三种场景下的带宽。

4.2.1 端到端时延测试分析

如下图 4 显示了交换机端到端时延测试结果。在三层路由拓扑中通过使用 ping 指令测试端到端时延，得到 BMv2 交换机的平均端到端时延为 0.82ms，BMv2_DPDK 交换机的端到端时延为 0.21ms。可见 BMv2 使用 DPDK 加速后，时延性能提升了 3 倍，这与无锁队列优化结果保持一致。

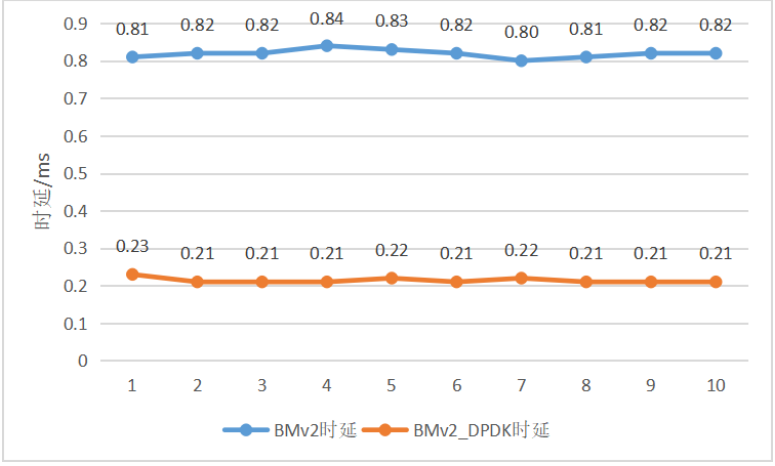


图4 拓扑交换机时延图

4.2.2 吞吐量测试分析

A. 交换机二层转发带宽测试结果

如下图 5 显示了二层转发带宽测试结果。BMv2_DPDK 交换机在 64Bytes、128Bytes、256Bytes 及 512Bytes 长度的数据报文测试中都达到了千兆网卡的性能。BMv2 交换机转发 64Bytes 长度的数据报文带宽为 83.2Mbps，即每秒传输约 1.3M。计算可得，BMv2_DPDK 交换机在 64Bytes 小包的转发性能提升了至少九倍。

如下图 6 显示了交换机三层路由带宽测试结果。BMv2_DPDK 交换机在三层路由带宽测试中同样达到了千兆网卡的性能。BMv2 交换机 64Bytes 长度的数据报文带宽为 79.1Mbps，相比于二层转发带宽，由于增加了 IP 层的处理，BMv2 交换机的三层路由带宽减少了 4.1Mbps，每秒转发的数据包数量减少了 65.6k。可见，BMv2_DPDK 在 64Bytes 小包的转发性能提升了至少九倍。

如下图 7 显示了交换机简单负载均衡带宽测试结果。BMv2_DPDK 交换机在负载均衡场景中的带宽依旧达到了千兆网卡的性能。BMv2 交换机性能与三层路由场景带宽一致。

在时延测试中，BMv2_DPDK 性能明显优于

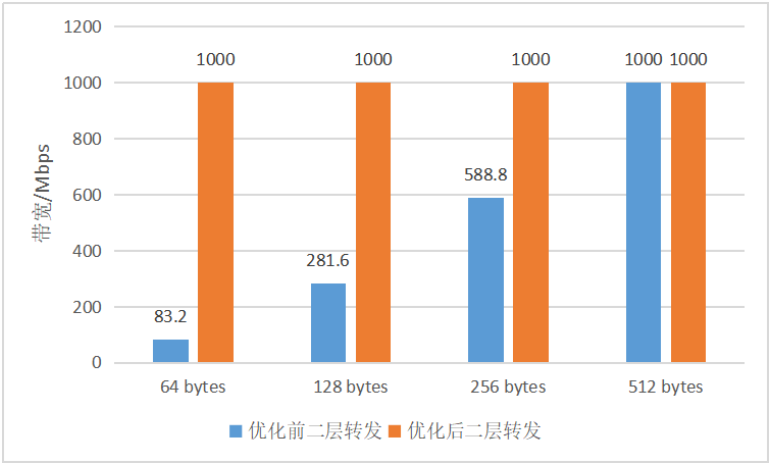


图5　交换机二层转发带宽图

B. 交换机三层路由带宽测试结果

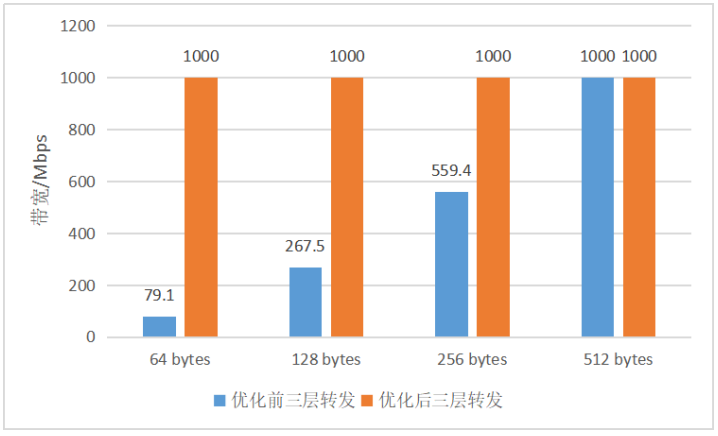


图6　交换机三层路由带宽图

C. 交换机简单负载均衡带宽测试结果

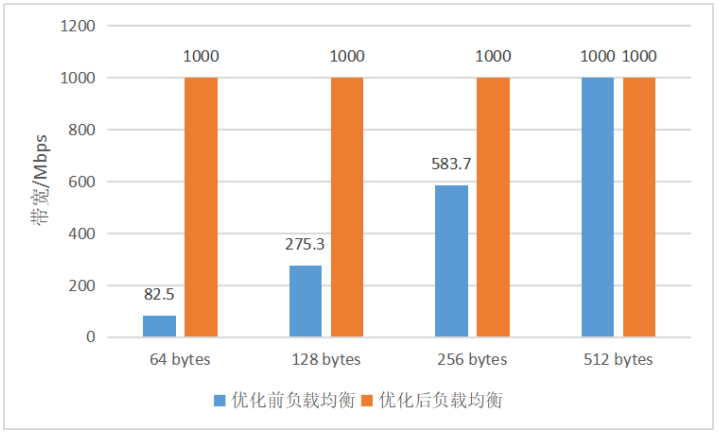


图7　交换机负载均衡带宽图

BMv2。BMv2_DPDK 交换机在三个应用测试场景中，其 64Bytes 至 512Bytes 长度的数据报文带宽性能都达到了千兆。BMv2 交换机在三个应用测试场景中，64Bytes 长度的数据报文转发的带宽在 1.3Mpps，128Bytes、256Bytes 的数据报文的带宽分别为 2.2Mpps、2.3Mpps。可见随着数据报文的增长，BMv2 交换机的每秒转发的数据报文数有所增加。BMv2_DPDK 相比于 BMv2，在 64Bytes、

128Bytes、256Bytes长度的数据报文性能至少分别提升9倍、2.5倍、0.7倍。但受网卡性能限制，512Bytes长度的数据报文转发带宽结果与方案持平，后续将改用1G或10G网卡针对1024Bytes和1500Bytes数据包进行测试。

5 结束语

本文主要针对P4软件交换机加速进行了研究，提出了基于DPDK对BMv2软件交换机进行加速的方案。通过双层架构设计，在CORE层中实现P4语言数据报文转发抽象模型，并且对原BMv2交换机中的队列进行了优化；在DPDK层中实现了底层网卡驱动，并能够向上层提供应用接口程序。实验测试结果表明，BMv2_DPDK交换机相比于BMv2交换机，其端到端时延性能提升了3倍，带宽也有明显的提升。

P4软件交换机由于灵活性和可编程性受到了广大学者的研究与关注，因此本文对基于DPDK的BMv2加速优化进行了研究探索。在未来的工作中，将会使用更高测试仪与工具针对当前方案与近年其他研究方案进行对比，同时针对如何在DPDK层对数据报文进行更复杂处理这一问题（例如将IPv4数据报文更改成IPv6数据报文）进行研究。

参考文献:

[1] Bosshart P, Daly D, Gibb G, et al. P4: Programming protocol-independent packet processors［J］. ACM SIGCOMM Computer Communication Review，2014，44(3)：87-95.

[2] Sivaraman A, Kim C, Krishnamoorthy R, et al. Dc. P4：Programming the forwarding plane of a data-center switch［C］// Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research. Santa Clara，USA，2015：2-9.

[3] Mahalingam M , Dutt D , Duda K , et al. Virtual eXtensible Local Area Network (VXLAN): A framework for overlaying virtualized layer 2 networks over layer 3networks[J]. 2014.

[4] Dukkipati N . Rate Control Protocol (RCP): Congestion control to make flows complete quickly［J］. These instructions，2007（8）：256-257.

[5] Sivaraman V, Narayana S, Rottenstreich O, et al. Heavy-hitter detection entirely in the data plane［C］//Proceedings of the Symposium on SDN Research. Santa Clara，USA，2017：164-176.

[6] Popescu D A, Antichi G, Moore A W. Enabling Fast Hierarchical Heavy Hitter Detection using Programmable Data Planes［C］// Proceedings of the Symposium on SDN Research. Santa Clara，USA，2017：191-192.

[7] Duncan R, Jungck P. PacketC language for high performance packet processing［C］// Proceedings of High Performance Computing and Communications. Seoul，Korea，2009：450-457.

[8] Brebner G, Jiang W. High-speed packet processing using reconfigurable computing[J]. IEEE Micro，2014（1）：8-18.

[9] 林耘森箫, 毕军, 周禹, 张程等. 基于P4的可编程数据平面研究及其应用[J]. 计算机学报，2019，42(11)：2539-2560.
Lin yunsenxiao, Bi jun, Zhou yu, et al. Research and Application of Programmable Data Plane Based on P4［J］. Chinese Journal of Computers，2019，42(11)：2539-2560.

[10] Wang H, Soulé R, Dang H T, et al. P4FPGA: a rapid prototyping framework for p4［C］//Proceedings of the Symposium on SDN Research. Santa Clara，USA，2017：122-135.

[11] Wang H, Lee K S, Shrivastav V, et al. P4FPGA: High Level Synthesis for Networking[C]//Proceedings of the SIGCOMM Posters and Demos. Salvador，Brazil，2016：620

[12] Benek P, Pu V, Kubtov H. P4-to-VHDL: Automatic generation of 100 gbps packet parsers[C]//Proceedings of Field-Programmable Custom Computing Machines. Washington DC，USA，2016：148-155.

[13] Benáček P, Puš V, Kubátová H, et al. P4-To-VHDL: Automatic generation of high-speed input and output network blocks［J］. Microprocessors and Microsystems，2018，56：22-33.

[14] Li P, Luo Y. P4GPU: Acceleration of programmable data plane using a cpu-gpu heterogeneous architecture［C］//Proceedings of High Performance Switching and Routing. Yokohama，Japan，2016：168-175.

[15] Li P, Luo Y. P4GPU: Accelerate packet processing of a p4 program with a cpu-gpu heterogeneous architecture［C］//Proceedings of the 2016 Symposium on Architectures for Networking and Communications Systems. Ithaca，USA，2016：125-126.

[16] Pfaff B, Pettit J, Koponen T, et al. The design and implementation of open vswitch[C]//Proceedings of Symposium on Networked Systems Design and Implementation. Santa Clara，USA，2015：117-130.

[17] Shahbaz M, Choi S, Pfaff B, et al. Pisces: A programmable, protocol-independent software switch［C］//Proceedings of the Conference of the ACM Special Interest Group on Data Communication. Salvador，Brazil，2016：525-538.

[18] Choi S, Long X, Shahbaz M, et al. The Case for a Flexible Low-Level Backend for Software Data Planes[C]//Proceedings of the First Asia-Pacific Workshop on Networking. Hong Kong，China，2017：71-77.

[19] Choi S, Long X, Shahbaz M, et al. PVPP: A Programmable Vector Packet Processor［C］//Proceedings of the Symposium on SDN Research. Santa Clara，USA，2017：197-198.

[20] Vörös P, Horpácsi D, Kitlei R, et al. T4P4S：A Target-independent Compiler for Protocolindependent Packet Processors" [C]//IEEE HPSR. 2018：17-20.

[21] 唐宏, 柴桌原, 任平, 王勇. DPDK应用基础[J]. 电信科学，2016，32(08)：170.
Tang hong, Chai zhuo-yuan, Ren ping, Wang yong. DPDK application basics［J］. Chinese Journal of Telecommunication Science，2017，33(S1)：268.

[22] 李伟成．基于DPDK的报文采集系统的研究与实现[D]．北京邮电大学，2016.
Li wei-cheng. Research and implementation of data acquisition system based on DPDK[D]. Beijing University of Posts and Telecommunications，Beijing，2016.
[23] 杨军．基于DPDK的数据包查表技术的设计与实现[D]．电子科技大学，2015.
Yang jun. Design and implementation of packet lookup technology based on DPDK[D]. University of Electronic Science and Technology of China，Chengdu，2015.
[24] Muhui Y, Yihua H. OVS-DPDK with TSO feature running under docker [C]// 2018 International Conference on Information Networking (ICOIN). 2018
[25] Hans W. DPDK-based implementation of application-tailored networks on end user nodes [C]// International Conference & Workshop on the Network of the Future. IEEE，2014.
[26] Nötzli A, Khan J, Fingerhut A, et al. P4pktgen: Automated test case generation for p4 programs [C]//Proceedings of the Symposium on SDN Research. Los Angeles，USA，2018：5-11.

[作者简介]

张雪（1998年—），女，硕士研究生，下一代互联网、软件定义网络、智慧标识网络。

覃帅（1995年—），男，硕士研究生，下一代互联网、软件定义网络、智慧标识网络。

权伟（1987年—），男，工学博士后，北京交通大学副教授，下一代互联网及智慧车联网、智慧标识网络、软件定义网络。

王洪超（1982年—），男，工学博士，北京交通大学副教授，未来网络体系及智慧车联网、空天地网络、5G和车载网络等。

高德云（1973年—），男，工学博士后，北京交通大学教授，物联网、车辆网络、下一代互联网领域和智慧标识网络等。

SecIngress: An API Gateway Framework to Secure Cloud Applications Based on N-variant System

ZHOU Dacheng, CHEN Hongchang, CHENG Guozhen, HE Wezhen
National Digital Switching System Engineering and Technological R&D Center, Zhengzhou, Henan 450002, China

Abstract: With the prevalence of SaaS applications, the security of cloud services has received widespread attention. Because it is tough for cloud tenants to deploy security products in SaaS cloud model, the security of cloud services pose greater challenges to cloud providers. The n-variant technology provides a way for cloud providers to devise a secure service architecture, but upgrading an application in cloud to the architecture of *N-variant system* with unfailing service exposure is difficult. In this paper, we propose SecIngress, an API gateway framework to secure cloud applications based on *n-variant system*. We design a *two-stage timeout processing method* to lessening the latency of transactions and *Analytic Hierarchy Process Voting under Metadata Mechanism (AHPVM)* to enhance the exactness of voting. We design and implement a prototype and set up a testbed with a web application based on n-variant system to evaluate the security gains and performance degradation of the prototype. The evaluation reveals that the quantity of vulnerabilities reduced by 55. 10% and the number of threats reduced by 41. 70%, along with 24. 99% delay increase, 39. 61% throughput drop, and 32. 57% CPU utilization increase. Through these efforts, our work demonstrates the practicality of decreasing reachable vulnerabilities or threats with acceptable performance degradation.

Key words: n-variant system; cloud service discovery; cloud security; analytic hierarchy process

1 Introduction

The Software as a Services （SaaS)， as a software delivery model， enables software providers to expediently use various IT services without the need to build their physical machine （PM） and network infrastructure ［1］. Companies are increasingly deploying their web-based application in cloud for lower cost， earlier management， and higher scalability， etc. ［2］ However， software applications in the cloud also face many security threats. According to a report of Mcafee ［3］， in the first half of 2020， affected by COVID-19， the application of cloud services in several industries increased by an average of 80.4%. At the same time， cloud security incidents increased by an average of 826.5%， which seriously affected the security of cloud services.

The explosive growth of cloud service threats is due to the complexity of traditional software functions leads to unavoidable backdoor loopholes and vulnerabilities ［4］. Besides， in the cloud environment， it is difficult for cloud tenants to deploy traditional security products and technologies for the abstraction and centralized management of computing resources by cloud service providers ［2］. The stagnant nature of the executing environment in cloud， as the main threat of the applications in cloud， gives adversaries chances to discover security holes， find opportunities to exploit them， escalate privileges， and maintain persistent presence over time ［5］.

N-variant system ［6］ is an approach to protect cloud applications from the stagnant drawback. *N*-

variant system executes multiple diversified variants in lockstep on identical input and monitoring outputs to verify their consistency to justify threats. The diversity technology avoids the common vulnerabilities of variants， so the output of a replica with vulnerabilities exploited by attackers is different from other variants at a high probability. Unfortunately， the substantial runtime overheads prevent the deployment of N-variant system. However， cloud computing is the overprovisioning of resources for scalability， and such dormant or underutilized resources can be utilized， at least sometimes， to enhance the security of services ［7］［9］. Applying such resources to construct n-variant system is a subtle balance between security and resource. Besides， with the benefit of container technology and convenient management in cloud environment ［8］， software applications can be containerized as images， and it is portable to replace the variants with read-only images when abnormal behaviors are detected by monitor， which enhances the resiliency of n-variant system. However， transferring a normal application in cloud into the architecture of *N-variant system* with unfailing service exposure is difficult for popular API gateways， such as Kubernetes-Ingress ［10］， Haproxy ［11］， Kong ［12］， etc.

In this paper， we propose SecIngress， an API gateway framework to secure the cloud application based on *n-variant system* and expose the service of them to users. The function of SecIngress includes enabling normal service workload between users and servers， ensuring users receive correct responses， and detecting abnormal behaviors of services variants. Firstly， SecIngress receives HTTP requests from clients and redirects identical requests to multiple diversified variants. Secondly， SecIngress collects responses of variants， and selects the correct one to send to client after voting. On this step， we design a *two-stage timeout processing method* to lessening the latency of users' transactions and an analytic hierarchy process voting under metadata （AHPVM） to improve the accuracy of discriminating complex protocols at the application layer during the voting phase. Thirdly， SecIngress determines whether reporting abnormal software variants to the cloud manager according to further analysis detected during the voting phase. In this manner， detecting the abnormal server and discarding its output.

We implemented the prototype of SecIngress and built a testbed for it with a web application based on n-variant system to evaluate the security gains and performance degradation of the prototype. We develop the prototype system based on NGINX ［12］， a free， open-source， high-performance HTTP server and reverse proxy， which is famous for its high performance and low resource consumption. Based on the prototype， we firstly evaluated the security enhancements of SecIngress， by deploying simple variants on our micro-benchmarks by varying software running environment， e. g.， operating system （OS）， and web containers. We tested them through vulnerabilities scanner tool and penetrating test tools to evaluate the security gains by comparing the number of vulnerabilities or alerts between a variant with Kubernetes ingress ［9］ and variants with SecIngress. Secondly， we evaluated the performance influence after employing SecIngress for application variants by Apache Benchmark ［13］. Those evaluation experiments show that our prototype enhances the security of service with acceptable performance degradation. It is worth mentioning that SecIngress is a lightweight software that can be containerized so that the SecIngress prototype can be deployed and managed in cloud environment easily with container technology.

The contributions of this paper are as follows：

We propose SecIngress， *a secure service discovery framework*， to expose the service instance based on *n-variant system* in cloud， increasing the availability and resiliency of cloud services.

We propose *a two-stage timeout processing method* in the process of collecting response messages of multiple variants with different execution efficiency， reducing the delays of responses received by cli-

ents.

We propose *an Analytic Hierarchy Process Voting under Metadata Mechanism （AHPVM）* for identifying the consistency of responses， reducing the probabilities of false-positive voting results under many "reasonable" inconsistent elements.

We evaluation *the security enhancements and the performance reduced* by SecIngress through empirical tests. Our prototype of SecIngress has a reduction of 55.10% vulnerabilities exposed and 41.70% reachable attack threats with 39.61% throughput reduction， 24.99% response increasing， and 32.57% CPU usage rate increasing at the average of tests under different resource-size， access-concurrency， and variant number.

The remainder of this paper arranges as follows： Section 2 introduces related work. Section 3 overviews the concept and the main challenges of SecIngress. Section 4 describes the design of SecIngress and section 5 presents voting mechanisms， called AHPVM. Section 6 shortly introduces the implementation of our prototype. Section 7 evaluates the security enhancements and performance degradation of Secingress and Section 8 makes a conclusion.

2 Related Works

N-variant systems ［5］ have proven effective against many attack classes. Given an optimal combination of variants， a system running just two variants can defeat return to libraries， function pointer overwrite， and stack smashing attacks ［25］. *N*-variant systems have also thwarted information leakage ［26］， partial overwrite ［27］， code injection ［5］， and code reuse attacks ［28］. The attacker must devise a way to exploit vulnerabilities in each variant， and the exploit must compromise all the variants simultaneously or compromise them in a way that does not affect their behavior in order to evade detection. Furthermore， DMON ［34］ proposed a new distributed n-variable execution system， which uses the diversity of instruction set architecture and application binary interfaces to prevent memory corruption attacks. To bypass DMON， an attacker must provide a vulnerability that works on two or more different platforms at the same time. However， the drawback of the multivariable execution environment is the high loss of performance.

To make the *N*-variant systems practically usable， substrates based on the low-overhead virtualization technique are desired ［35］. In particular， as a maturing IT paradigm empowered by the virtualization， cloud computing is being increasingly utilized to enhance the competitiveness of numerous applications via cost-effective service provision and sharing ［36］. The cloud platform lends itself to implementing the N-version program （NVP） ［37］［38］［39］［40］［41］. In the cloud environment， there is numerous usage of software diversity to enhance the security of cloud service. MEERKATS ［6］ creates idle and changing target applications to confuse the attacker and to keep the redundancy in case of server failures. The Merkaats architecture integrates anomaly detection， data replication， and checkpointing to provide protection and restoration. DREME ［8］ defends against SQL injection attacks by using redundant database variants and diverse processes. Polinsky ［32］ proposed an n-m variant system to protect the media wikiphp applications serving dynamic content from external SQL persistent storage in the cloud.

Cloud providers offer a diversity of cloud services for customers including processing， storage， and other services ［14］. It is becoming gradually more difficult for a user to select a suitable service provider that will supply them with the required services ［15］. Cloud API gateway， which satisfies the cloud users’ demands， is considered a hot issue ［16］［17］［18］［19］. The majority of the studies， such as ［20］， ［21］， ［22］， and ［23］， have confirmed that the lack of cloud service secure accessing schema. It renders this process more complex for users， especially those with low cloud services’ experience ［24］.

However， securing applications based on N-

variant system is a challenge due to the multiple data stream and complex voting process under the protocol of application layer. Our work aims to provide a proper API gateway framework to make it better that deploying secure applications based on the schema of N-variant system.

3 Overview

In order to effectively expose the service based on n-variant system and ensure the security gain of it， we need to overcome the high delay effect of n-variant system on the one hand and ensure the voting accuracy of the monitor for multiple diversified responses on the other hand. To achieve these two main goals， we face the following challenges.

The first is the high delay problem of n-variant systems. Due to the distributed and independent operation of multiple variants， the execution time of each variant is varied. Monitors of multiple variants need to collect the output of all variants， which undoubtedly binds the output efficiency of the system to that of the slowest variant. Generally， n-variant technology will increase the response time of cloud services， thus reducing the QoS of cloud services.

The second is the voting problem of n-variant output. The protocol complexity of the application layer brings more challenges to the voting of N variant output content. Taking HTTP protocol as an example， long or short connection， chunked transmission， and 304-cache mechanism increase the diversity of output of variants. Toward "reasonable" inconsistencies， voting correctly is a problem we need to address.

The third problem is the oscillation caused by the rotation of voting abnormal executors. Due to the dynamic scalability of resources in cloud environment， the variant of voting exception， under attacks， can be cleaned offline， and new variants quickly deployed in the cloud environment. Therefore， cloud manager and service agent need to exchange variant updated information， which will affect the continuity of services.

We address the first problem by designing *a two-stage timeout processing method* in the process of collecting response messages of multiple variants with different execution efficiency. The actual work environment statistics show that only a few of the variants are significantly lagging behind the others. Note that receiving two identical means achieving the pass condition of the majority decision. We can output the response message to the client first to avoid the client waiting too long. After that， another voting will work until the remaining response returns to check whether existing abnormal response or not.

We address the second problem by extracting the metadata of application layer protocol and using the analytic hierarchy process （AHP） ［29］［30］［31］ to compare the consistency of response messages of variants from multiple dimensions. Through the statistical analysis of a large number of response messages of normal variants， we first analyze the common metadata （maximum intersection） of response messages of different variants. Based on these metadata and status code classifications， we can vote more accurately under many "reasonable" inconsistent elements.

We address the third problem by decoupling the control plane and the data plane of the SecIngress. We decouple the interaction function between the system and cloud manager to the control surface and designed a Message Broker （MB） to be responsible for the communication and negotiation with the cloud manager. The existence of MB avoids the dynamic changes of application variants disturbing the efficient data flow processing of the SecIngress， ensuring the high-speed processing ability of the data plane without state maintaining.

Fig. 1 depicts SecIngress architecture with a service instance based on n-variant system in a private cloud environment. When requests from clients terminate at SecIngress， the Dispatcher Proxy （DP） module of SecIngress redirects requests to n diversified application after copying packets. Then， the Voting Monitor module collects the responses of mul-

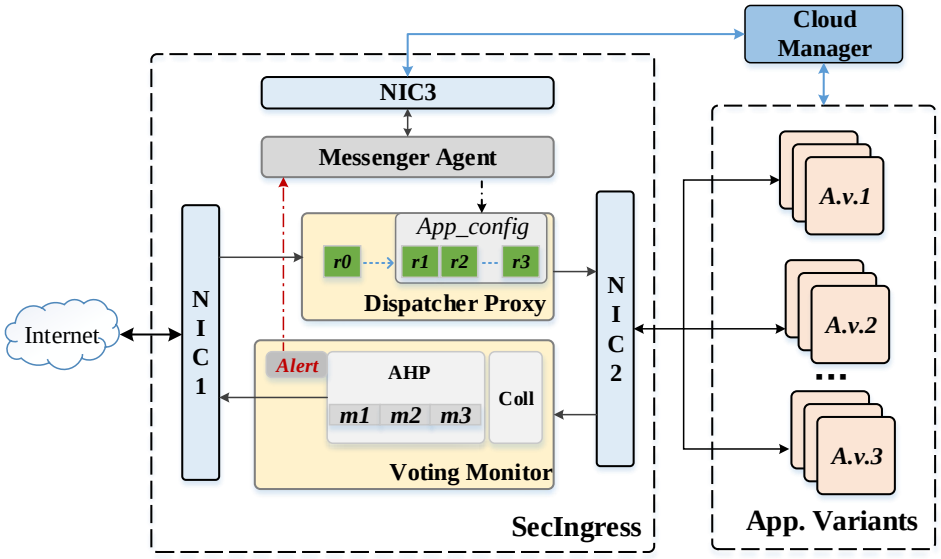


Fig. 1 SecIngress architecture.

tiple applications and classifies responses packets as "response groups" for each request. Further, it analyzes each "response group" according to the voting mechanism, selecting one of the normal responses sent to its client, while detecting abnormal responses and reporting compromised application variants. Finally, the Message Broker (MB) module as a feedback agent running on the control plane of SecIngress and communicating with cloud manager, receives deployed variants' IDs, including IP address and service port which inform the DP of the destination of requests, while sends compromised variants identities to cloud manager which will reset the abnormal variants.

4 SecIngress Design

In this section, we introduce the design of SecIngress. To simplify the description, web application environment is considered as an example. We firstly describe the main modules of SecIngress. Then, we introduce the abnormal process of SecIngress, especially in responses collecting phase.

4.1 Dispatcher Proxy

The Dispatcher Proxy (DP) is the only interface to access applications backend. It plays a central role in SecIngress architecture. Initially, DP receives the deployment information of application variants from Messenger Agent (MA) to configure its application set, A_s, i.e., $\{a_1, a_2, \cdots, a_n\}$, preparing for the work of data plane. When a request, r_q, arrives DP, it duplicates r_q into copies set, R_q, i.e., $\{r_{q1}, r_{q2}, \cdots, r_{qn}\}$, where r_{qi} will be relayed to backend application replica $a_i \in A_s$. The last work of DP is inserting a tag $T(r_q)$ into all request copies, which is a seed for application side verification, e.g., identifying code.

4.1.1 Connection Processing.

As shown in Fig. 2, client regards the Dispatcher Proxy, DP, as a server, and it needs to establish TCP connection with DP before sending HTTP requests. Then, DP, as a client of application variants, need to establish TCP connections with all of the application variants.

For the client connecting phase: if the client wants to access the application through browsers, it sends TCP SYN packet to DP, and DP responds with TCP SYN-ACK. After the client sends TCP ACK to DP, the TCP connection, i.e., c_0, is established. Then, the HTTP request is sent to DP through c_0.

For the DP connecting phase: DP establishes TCP connections with each of application variants respectively, i.e., $\{c_1, c_2, \cdots, c_n\}$ after receiving r_q and duplicating it into requests copies, R_q, i.e.,

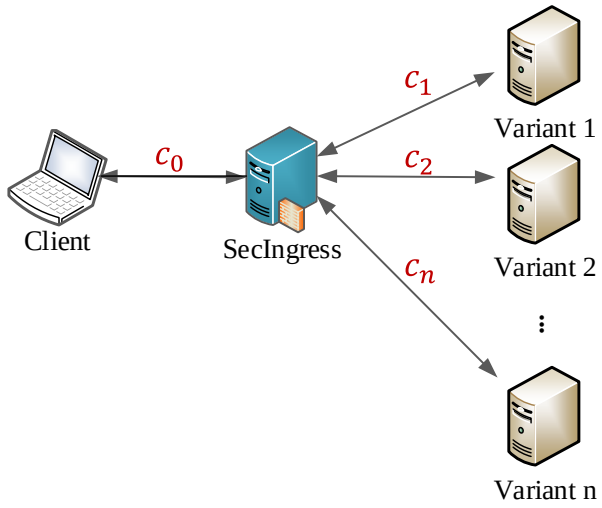


Fig. 2 Connections among Client, SecIngress, and Variants.

$\{r_{q1}, r_{q2}, \cdots, r_{qn}\}$, which then are sent to application variants based on $\{c_1, c_2, \cdots, c_n\}$ respectively. DP keeps established TCP connections with application variants, until collecting responses set R_s, i.e., $\{r_{s1}, r_{s2}, \cdots, r_{sn}\}$, of application variants. Then, a response, r_s, selected by Voting Monitor, will be sent to client through the TCP connection c_0.

4.1.2 Request Processing.

When receiving a request, r_q, from a client, the Distributor Proxy, DP, needs to duplicate it to copies, R_q i.e., $\{r_{q1}, r_{q2}, \cdots, r_{qn}\}$. However, to redirect correctly, DP need to modify several HTTP header fields which play important roles in communication with application variants, DP modifies the *Host* fields of all the copies according to the application set, A_s.

Furthermore, the identifier code is a popular solution of sever-side validation, usually presenting as random combinations of numbers and letters under irregular background, or other forms. Majority of identifier codes are obtained by program function on the input of random number seed. With this mechanism, it is impossible for application variants produce the same identifier, which will induce failed validation. To solve this problem, we insert a tag $\Gamma(r_q)$, a random number, to the HTTP header with a new field for $\forall r_{qi} \in R_q$, as the seed to produce the same dynamic identifier code. Besides, if the application variants return responses with inserting $\Gamma(r_q)$ into response header, DP could validate this tag for detecting whether there have packet hijack attack between DP and diversified variants set.

4.2 Voting Monitor

We mainly introduce this module with three components, responses collecting, timeout processing, and voting pretreatment. The detail of the voting mechanism is introduced at section 5.

4.2.1 Responses collecting.

Consider two users accessing SecIngress at concurrency, there are two requests, r_q^1 and r_q^2 need to be duplicated to R_q^1, i.e., $\{r_{q1}^1, r_{q2}^1, \cdots, r_{qn}^1\}$ and R_q^2, i.e., $\{r_{q1}^2, r_{q2}^2, \cdots, r_{qn}^2\}$, which will be distributed to application variants at the same time. After the distribution operation described at 4.1, two response sets, R_s^1, i.e., $\{r_{s1}^1, r_{s2}^1, \cdots, r_{sn}^1\}$ and R_s^2, i.e., $\{r_{s1}^2, r_{s2}^2, \cdots, r_{sn}^2\}$ responding R_q^1 and R_q^2, will return to SecIngress at the same time. However, the responses of application variants terminate SecIngress through NIC with the mixed and disorderly order, e.g., $\{r_{s1}^1, r_{s2}^2, r_{s1}^2, r_{s2}^1, \cdots\}$.

It is necessary to classify them into R_s^1 and R_s^2, providing the prerequisite for voting mechanism. It needs a distinguish tag in this process, and we notice that $\Gamma(r_q)$, inserted in processing request and returning back with responses, is a proper tag for this requirement. When we receive a response, we first extract the tag $\Gamma(r_q)$ and cache the response of each user according to the tag.

4.2.2 Timeout processing

As described before, responses do not return at the same time due to servers' processing delays or network latency. Reasonable timeout processing is the basis for collecting responses. We consider one request r_q received by SecIngress. SecIngress needs to collect all the responses before voting, i.e., $R_s = \{r_{s1}, r_{s2}, \cdots, r_{sn}\}$. As the situation shown in Fig. 3, SecIngress needs to wait for three responses until t_3 shown in Fig. 4. However, the waiting time may be too long if any application failed, causing a large delay in the transaction. For example, if application

variant 3 is blocked, t_3 may be very large and the client has to wait for the web page responding no less than t_3.

For the majority voting schema, at least three responses are needed before voting. But the condition that the first two responses are consistent is the passing condition of voting among the three responses. And this condition can be amplified to the voting of more than three responses under the assumption that only a small number of application variants are compromised. According to this premise, we design a *two-stage timeout processing method*. Specifically, when two responses return, SecIngress vote on them in the first stage. If they are consistent, one of them will be output to the client first without waiting for other responses; if not, SecIngress continue to wait for the next response until the condition that the majority of responses are consistent is met, that is, one of the most consistent responses can be output to the client. When all responses return, SecIngress votes based on them in the second stage to select the abnormal response message.

In this way, the client can receive the response message as soon as possible. In the scenario shown in Fig. 4, it only needs to wait for t_2 under optimal conditions. Note that a few of the application variants that are attacked have a high response delay in the actual working environment. Therefore, as depicted in Fig. 3, the client receives r_s' with the *two-stage timeout processing method* rather than r_s with the traditional timeout processing, which can improve the performance of N-variant system.

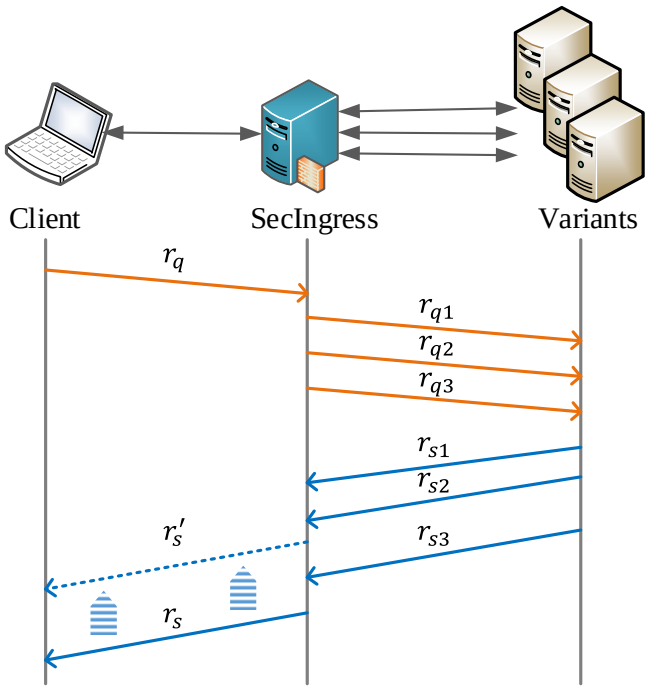


Fig. 3 L7 communicating flows.

4.2.3 Voting pretreatment

After the responses collection phase, the last work before voting executing is to extract metadata from responses, which can be regarded as the standards of voting. A correct decision need to be made under "reasonable divergences" of application variants. Because software diversification is a prerequisite to enhance the security of software in n-variant system, it is ineluctable that the normal applications return responses differing from others, for several divergences in packets, such as additional blank spaces, different protocol regulars, and negligible marks. We regard them as "reasonable divergences". For example, as shown in Fig. 5, it is different between the header fields of a simple web page running on Apache Tomcat and JBoss respectively.

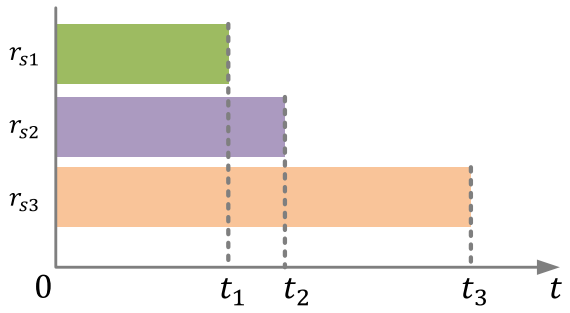


Fig. 4 Responses return time vary.

Consider "reasonable divergences" of diversified applications, the results of comparing strings simply that indicates there has abnormal applications are unbelievable, but both of them are believable actually.

To overcome this challenge, we extract metadata from response packets to compare the responses in multiple dimensions during voting. By analyzing the collected HTTP responses of diversified applications, we get the valid metadata in the HTTP response message, through semantic partition and pruning, including self-adaptive segment partition such as alignment, truncation, complement, etc.

Based on the metadata obtained, we first divide the type of response according to the status code, to avoid the interference of exceptional cases, such as the 304-cache mechanism and abnormal status code responses. All known special cases are dealt with in corresponding ways, which is not the focus of this paper. Then, the Analytic Hierarchy Process Voting under Metadata (AHPVM), which will be introduced in section 5, process all regular responses during the voting phase.

4.3 Message Broker

The Message Broker works on the control plane of SecIngress, interfacing with cloud manager through a self-defined protocol based on TCP. It receives deployment information of application variants that changes dynamic according to backend nodes heathy. And it sends malicious variants detected by the Voting Monitor to the cloud manager for further scheduling. By this feedback mechanism, the n-variant application service system achieves resilience.

5 Analytic Hierarchy Process Voting under Metadata Mechanism (AHPVM)

The voting mechanism works based on the assumption that compromised variants are a minor number of all variants. Therefore, the majority voting scheme can be utilized for selecting creditable responses. Because the metadata of response reveals the features of responses in multiple dimensions, a multi-objective comprehensive evaluation is necessary for balancing the vote results of all the metadata. In this section, the multiple-objective comprehensive evaluation based on the majority voting scheme is introduced.

5.1 AHP model setup

The voting is a multiple-objective comprehensive evaluation problem based on multiple criteria, i.e., the similarity of several metadata, and multiple options, i. e., responses of application variants. The analytic hierarchy process (AHP) [29][30][31] is a theory of measurement to translate the multiple-objective comprehensive evaluation to multi-criteria ranking. The voting problem can be expressed in Fig. 6 under the AHP schema. Selecting a response

JBoss AS 7.1.1.Final	Apache Tomcat 9.0.34
HTTP/1.1 200 OK *Server: Apache-Coyote/1.1* *Accept-Ranges: bytes* *ETag: W/"22606-1586251006000"* *Last-Modified: Tue, 07 Apr 2020 09:16:46 GMT* *Content-Type: text/html* *Content-Length: 22606* *Date: Tue, 07 Apr 2020 09:19:56 GMT*	*HTTP/1.1 200* *Accept-Ranges: bytes* *ETag: W/"22606-1586241573000"* *Last-Modified: Tue, 07 Apr 2020 06:39:33 GMT* *Content-Type: text/html* *Content-Length: 22606* *Date: Tue, 07 Apr 2020 08:16:20 GMT* *Keep-Alive: timeout=20* *Connection: keep-alive*

Fig. 5 header fields differences of JBoss and Apache Tomcat under a same web page.

sent back to the client locates at the goal layer. The criteria layer includes *m*-metadata of responses. Responses of *n*-variants are alternatives to the decision.

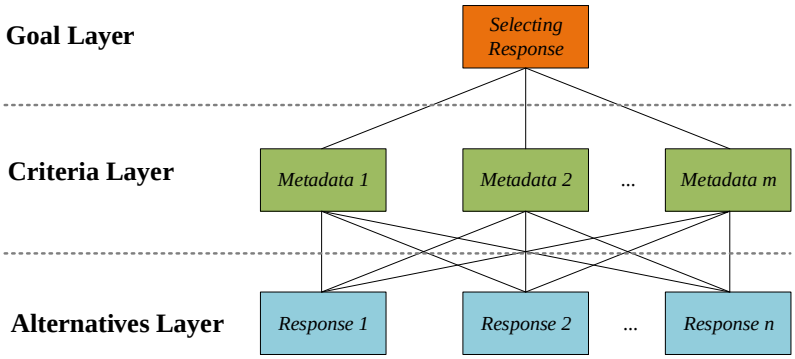


Fig. 6 Structure of the AHP method for voting.

5.1.1 Determine the weights vector of metadata

To calculate the weights for multiple criteria, a pairwise comparison matrix $\mathbf{P}_{m\times m}$ is created as:

$$\mathbf{P}_{m\times m}=\begin{bmatrix} p_{11} & \cdots & p_{1m} \\ \vdots & \ddots & \vdots \\ p_{m1} & \cdots & p_{mm} \end{bmatrix} \#(1)$$

where $p_{ii}=1$, let $p_{ji}=1/p_{ij}$ and $p_{ij}=p_{ik}/p_{jk}$, m denotes the number of evaluation metadata considered.

Additionally, each p_{ij}, the relative importance of two elements, can obtained from packet analysis and malicious packets recording at task, and quantified according to a numerical scale from 1 to 9 as described in Table I.

Table I Relative scores [29]

Value	Description
1	equally important
3	slightly more important
5	more important
7	strongly more important
9	absolutely more important

Table II values of the Random Index(RI) [29]

Size of matrix	2	3	4	5	6	7	8	9	10
RI	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.51

Then, we check the consistency of $\mathbf{P}_{m\times m}$. The first step is calculating the maximum eigenvalue λ_{max} of $\mathbf{A}_{m\times m}$ by the function that:

$$\mathbf{Aa}=\lambda\mathbf{a}\#(2)$$

where $\mathbf{a}$ is the eigenvector of $\mathbf{P}_{m\times m}$. And the Consistency Index (CI) is obtained by:

$$\mathrm{CI}=\frac{\lambda_{max}-m}{m-1}\#(3)$$

and the consistency is reached if $\frac{CI}{RI}<0.1$, where Random Index (RI) are given in Table 3. Note that $\mathbf{P}_{m\times m}$ needs to be modified until reaches consistency. After checking consistency, the metadata weight vector $\mathbf{w}$, i.e. $\mathbf{w}=[w_1,w_2,\cdots,w_m]^T$ is obtained by normalizing the eigenvector $\mathbf{a}$ corresponding to the maximum eigenvalue λ_{max} by:

$$\mathbf{w}=\frac{\mathbf{a}}{\|\mathbf{a}\|}\#(4)$$

5.1.2 Calculating responses scores matrix under the majority voting schema

Responses scores, reflecting the credibility of the responses on multiple dimension is defined as:

$$\mathbf{S}_{n\times m}=\begin{bmatrix} s_{11} & \cdots & s_{1m} \\ \vdots & \ddots & \vdots \\ s_{n1} & \cdots & s_{nm} \end{bmatrix} \#(5)$$

where s_{ij} denotes the score of *i*-th response under the *j*-th criterion. However, calculating the scores directly is difficult. Before that, we first obtain the pairwise comparison matrix $\mathbf{B}^j_{n\times n}$ for j-th criteria which represent the scores of all responses under *j*-th criterion. For example, the entry b^j_{ir} represents the score of the i-th response compared to the *r*-th response under *j*-th

criterion. Like the description in

Table II， the larger number of b_{ir}^{j} means that i-th response is more believable than r-th response under j-th criterion.

To avoid the inexactitude of personal experience， matrix $\mathbf{B}_{n\times n}^{j}$ is derived from the majority voting schema. In practice， the *j*-th metadata of variants are compared with each other， and classified as major set $\boldsymbol{M}^{j}$ containing α variants with consistent *j*-th metadata and minor set $\boldsymbol{m}^{j}$ containing other β variant. The numerical value of α and β satisfies：

$$\begin{cases} \alpha+\beta=n \\ \dfrac{n+1}{2}\leqslant\alpha\leqslant n \\ 0\leqslant\beta\leqslant\dfrac{n+1}{2} \end{cases} \#(6)$$

A response belonging to $\boldsymbol{M}^{j}$ is believable than that belonging to $\boldsymbol{m}^{j}$， and the ratio of α to β can denote the credibility. Furthermore， responses of $\boldsymbol{M}^{j}$ or $\boldsymbol{m}^{j}$ can be regard as equal credible. Therefore， an entry b_{ir}^{j} of $\mathbf{B}_{n\times n}^{j}$ is defined as：

$$b_{ir}^{j}=\begin{cases} 9\cdot\left\lfloor\dfrac{\alpha-\beta}{n}\right\rfloor & i\in\boldsymbol{M}^{j}\cap r\in\boldsymbol{m}^{j} \\ \dfrac{1}{9}\cdot\left\lfloor\dfrac{n}{\alpha-\beta}\right\rfloor & i\in\boldsymbol{m}^{j}\cap r\in\boldsymbol{M}^{j} \\ 1 & others \end{cases} \#(7)$$

Note that it is not need to check the consistency of matrix $\mathbf{B}_{n\times n}^{j}$ due to each entry b_{ir}^{j} obtained at the same time under the majority voting schema which eliminates the disadvantages of AHP's subjective judgment.

Then， to obtain the score matrix $\mathbf{S}_{n\times m}$， each matrix $\mathbf{B}_{n\times n}^{j}$ is firstly normalized to matrix $\overline{\mathbf{B}_{n\times n}^{j}}$ through：

$$\overline{b_{ir}^{j}}=\frac{b_{ir}^{j}}{\sum_{k=1}^{n}b_{kr}^{j}} \#(8)$$

and the entities of the score vector $\boldsymbol{s}^{j}$， i.e. $\boldsymbol{s}^{j}=[\boldsymbol{s}_{1}^{j},\boldsymbol{s}_{2}^{j},\cdots,\boldsymbol{s}_{n}^{j}]^{T}$， are obtained by normalizing the entries of each row of $\overline{\mathbf{B}_{n\times n}^{j}}$：

$$\boldsymbol{s}_{i}^{j}=\sum_{k=1}^{n}\frac{\overline{b_{ik}^{j}}}{n} \#(9)$$

Finally， the score matrix $\mathbf{S}_{n\times m}$ is obtained as $\mathbf{S}_{n\times m}=[\boldsymbol{s}^{1},\boldsymbol{s}^{2},\cdots,\boldsymbol{s}^{m}]$.

5.1.3 Ranking global scores of responses and making decisions

We obtain a vector **F** of global scores by：

$$\mathbf{F}=\mathbf{S}_{n\times m}\cdot\mathbf{w} \#(10)$$

F consists of *n* entities， i. e.， $\mathbf{F}=[f_{1},f_{2},\cdots,f_{n}]^{T}$， reflecting the finally score of *n* responses. Then we rank the global scores in decreasing order， and obtain final scores **F′**， The forehead responses of which are regard as believable and one of them will selected randomly to return to client， while other responses are marked as abnormal， and sent to cloud manager through Message Broker to replace its correlate variants.

5.2 Abnormal processing

It is rare but possible that collecting less than three responses， causing abnormal processing of majority voting. When there are two responses and they are consistent， they could satisfy the voting condition and both of them are creditable， and one of them can be selected then sent to the client. However， if there are two different responses or only a single response， the voting mechanism discussed at 5.1 will be invalid. Consider that safety tolerance varies in different scenarios， we provide the solution for this problem at two different levels as follows.

5.2.1 Optimistic voting

Several application scenarios may have rarely attack or high safety tolerance， such as none of the information leaking risk. When there are two different responses collected， a response will be randomly selected from two different responses to the client， and when there is only one response， the SecIngress simply redirects the response to the client. Meanwhile， all of the variants will be regarded as abnormal variants and report them to the cloud manager to replace them to avoid the next abnormal transaction.

5.2.2 Pessimistic voting

Like optimistic voting， pessimistic voting also triggers the replacement of all variants in the cloud manager to terminate abnormal transactions. However， such scenarios take high being attacked risk，

and the abnormal response need to be terminated at SecIngress. Therefore, the response is discarded and the SecIngress will interrupt the session of the current user.

6 Implement

We have implemented the SecIngress prototype based on NGINX [12], a free, open-source, high-performance HTTP server and reverse proxy, which is famous for its high performance and low resource consumption. We now describe the key function of SecIngress: (1) dispatcher module, (2) monitor module, and (3) messenger broker module, as shown in Fig. 1.

6.1 Dispatcher module

The Dispatcher module aims to duplicate the request to multiple copies and sends those copies to multiple application variants, which can be completed by mirror module, i.e., *ngx_http_mirror_module*, of NGINX. The mirror module, mirroring of an origin request by creating sub-requests background, usually is used for coping HTTP request traffic from users for, maybe, detecting malicious traffic. For SecIngress, the main process of proxy module transfers requests to one application replica while mirror module duplicate request of user to *n-1* sub-requests for application variants, ensuring that *n* application variants receive same input of request.

6.2 Monitor module

Responses to mirror sub-request usually are ignored. Therefore, we slightly modified responses receiving module and filter module of original NGINX, while developing a self-filter module, inserted to filter module chain, to collect responses of all the application variants. What the self-filter module preforms is that blocking-up response backs to the client and collecting them into memory until all the responses back or the timeout happening introduced at 4.2.2. About 5000 lines of C code are appended to the Nginx source code to accomplish the AHPVM along with the self-filter module.

6.3 Messenger broker module

Messenger Agent, developed in nearly 2000 lines of C code, plays the role of the control plane for SecIngress, which runs at the same machine but a different process with NGINX. It controls SecIngress instance by modifying the NGINX configure file, i.e. nginx.conf, and reloading NGINX instance. Meanwhile, it read abnormal replica from shared memory and sends alerts about them to the cloud manager.

7 Evaluation

7.1 Testbed setup

As shown in Fig. 7, we setup a simple work environment of API gateway. Our testbed setup uses a topology that consists of a test host, a internal switch and four nodes of our private cloud, equipped with 2.50GHz 64-bit Intel (R) Xeon (R) CPU E5-2680 v3 processor with 12-cores, 32GB RAM, 2T disks, and four network interfaces with 1Gbps network speed. A node as proxy server runs SecIngress and Kubernetes ingress [9] separately while other nodes run an open-source web application Jpress [42] with diversified web servers and operating systems.

7.2 Security Gains Evaluation

7.2.1 Vulnerabilities Scanning

We deployed Nmap with Vulscan [46], a module to enhance Nmap to a vulnerability scanner, on the test host, *Tester* and scanned web application vulnerabilities through it. The result, shown in Fig. 8, depicted that the vulnerabilities number of the service instance discovered by SecIngress is less than the service instances discovered by Kubernetes ingress by 70.08%, 41.13%, and 54.09% respectively, and at an average of 55.10%.

Meanwhile, we noted that for different vulnerability libraries, the vulnerability information obtained by scanning is different. For example, the number of vulnerabilities of service instance discovered by SecIngress corresponding to the vulnerability libraries of Exploit-DB and OpenVAS (Nessus) is

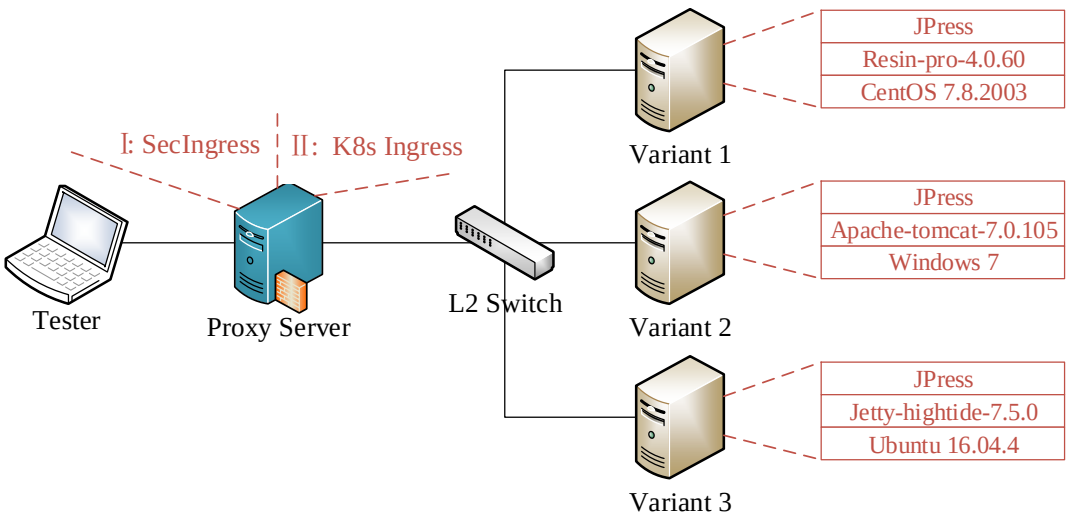


Fig. 7 The testbed of Experiment.

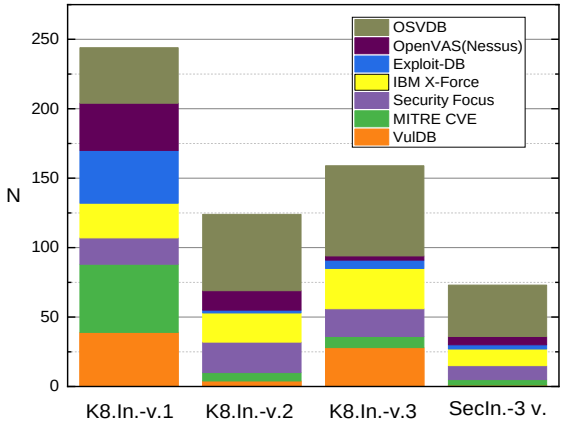


Fig. 8 The vulnerabilities of scanning.

more than that discovered by Kubernetes ingress. This is due to the inevitable accessorial vulnerabilities of SecIngress. However, the number of vulnerabilities decreases ultimately, which reveals the security gains of SecIngress exposing N-variant application system.

7.2.2 Penetration Testing

We deployed the OWASP ZAP [47] penetration test tool on the test host, *Tester* and ran a *Quick Start Automated Scans* based on it. The result of alerts found during the scans is shown in Fig. 9, which reveals that the SecIngress reduced the number of threats at an average of 41.70%. In detail, the number of the alert of services instance discovered by SecIngress is minimum and the types of alerts of it is the least among other tests. For example, only the service instance with Jetty discovered by Kubernetes ingress has the SQL Injection alert compared with Tomcat and Resin, and this SQL Injection alert cannot be exposed pass through SecIngress due to the majority voting.

Note that we only analyzed the SecIngress security gain through web applications with diversity technologies including heterogeneous web servers and operating systems. However, the security gain will be no less than the improvement of this experiment through other diversity technologies such as rearranging memory [45], randomizing system calls

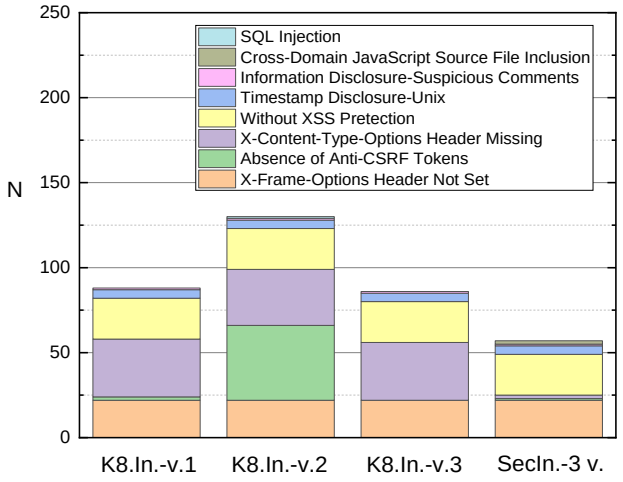


Fig. 9 The threats analysis result of SecIngress toward Kubernetes Ingress.

[46], randomizing the instruction set [47], N-version program [37], or others.

7.3 Performance Degradation Evaluation

The purpose of this part is to evaluate the performance degradation of our prototype system. We mainly verify the performance degradation of SecIngress exposing service of variants when the size of resource is different and the amount of concurrent HTTP requests is different. Through this evaluation way, the performance in the actual work environment of multi-user and multi-resource types is simulated.

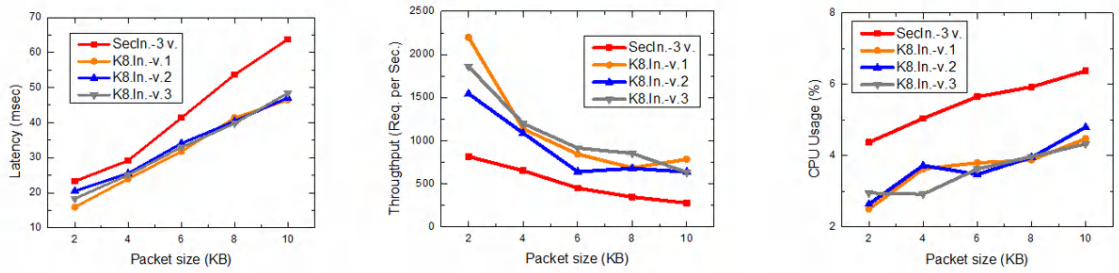
In the process of performance testing, we also evaluated the resource occupancy in the process of running secingress and Kubernetes ingress on the same proxy server under the same conditions. Generally, it is mainly CPU occupancy and memory occupancy. Due to Nginx's excellent memory management mechanism, we observed in actual tests that the memory usage of the proxy server does not surpass 2%, and the memory occupancy increasing is not obvious. Therefore, here, we only use the CPU usage to analyze the resource usage of SecIngress.

7.3.1 Performance impact of resource size

The prototype of SecIngress worked on a proxy server with three web applications on the backend, each of which contains web pages with resource sizes range from 2KB to10KB. In this way, we simulated the existence of various resource pages of different sizes in the general server to illustrate the performance of SecIngress's service exposure to multiple variants.

During the actual experiment, for each resource size web page, we set the concurrency to 100 and the total number of access requests to 1000, sent request packets to SecIngress, and recorded the response delay and the throughput of the requests using Apache Benchmark. Then, we tested the response latency and throughput of the Kubernetes ingress to expose each variant respectively. The experimental condition was identical as SecIngress, including Kubernetes ingress running on the same proxy server and exposing variant introduced before.

In the process of each group of testing, we record the proxy node CPU usage rate to evaluate the resource occupancy. To compare the performance degradation of SecIngress intuitively, both SecIngress and Kubernetes ingress are set with a single-core processor. The experimental results in Fig. 10 depicted that the latency increases, the throughput decreases, and the CPU utilization rate increases as the size of the resource increases. Compared with Kubernetes ingress exposing the service of a single variant, the

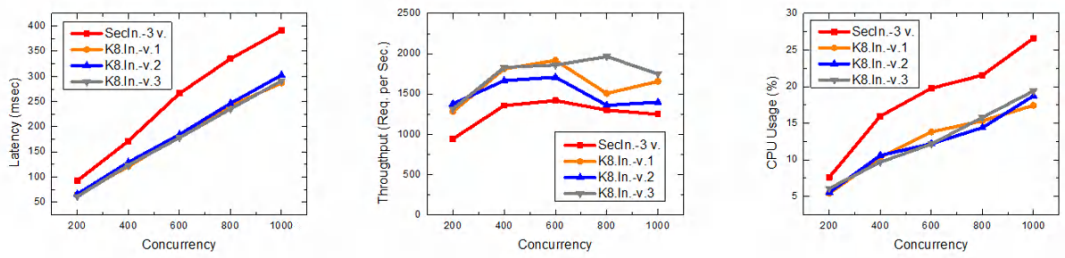


(a) (b) (c)

Fig. 10 The performance comparison between SecIngress and K8s ingress with different packet size. (a) Latency, (b)Throughput, (c) CPU usage.

response delay of SecIngress exposing the service of multiple variants increased by 21.45%， throughput decreased by 48.77% and CPU utilization increased by 33.72% on average.

7.3.2 Performance impact of request concurrency



(a) (b) (c)

Fig. 11 The performance comparison between SecIngress and K8s ingress with different request concurrency. (a) Latency, (b)Throughput, (c) CPU usage.

Similar to the aforementioned experimental method， we configured three back-end web applications that contain 2KB of web pages， and accessed the web services through SecIngress with concurrent value range from 200 to 1000 with 200 gradient increments， while recorded the delay and throughput. To guarantee the stability of the test results， in each experiment， we set the total number of requests in Apache Benchmark to ten times the corresponding concurrency to eliminate errors. Besides， we tested the performance of Kubernetes ingress that exposes a single variant under the corresponding conditions. In each experiment， we recorded the CPU usage.

The experimental result in Fig. 11 shows that the delay increases， and the CPU usage increases as the amount of concurrent user requests rises. Yet for throughput， on the one hand， as the number of concurrency rises， the number of requests processed per second increases， and on the other hand， as the amount of concurrency increases， thus processing pressure intensifies. They restrict each other， so there is a phenomenon of throughput increasing first and then slowly decreasing with concurrency increasing， which exists in both SecIngress and Kubernetes ingress. Compared with the use of Kubernetes for service exposure， the latency of using SecIngress increased by 28.53% on average， throughput decreased by 30.45%， and CPU usage increased by 31.41%.

7.3.3 Performance impact of variant number N

In practical applications， the number of variants should be no fewer than three. The greater the number of variants generated using different diversification technologies， the smaller the risk of shared vulnerability exposure. Considering this reason， we need to evaluate the impact of different variant numbers on performance. In this part， we merely deployed multiple isomorphic "variants" to simulate this

application scenario.

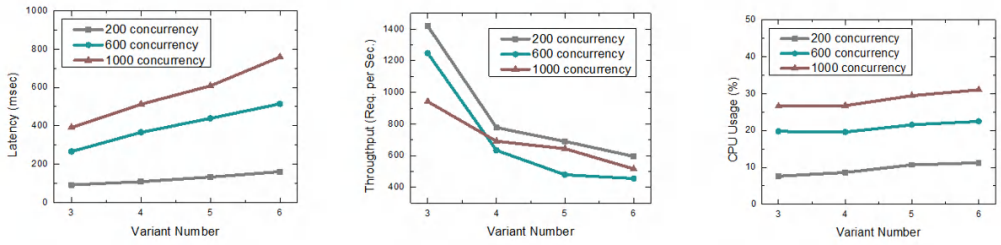


Fig. 12 The performance comparison between SecIngress and K8s ingress with different variant number. (a) Latency, (b) Throughput, (c) CPU usage.

As in the previous experiment process， we utilized the Apache Benchmark to perform appended inspections on scenarios with different variant number. The experimental results in Fig. 12 shows that latency of responses increases， throughput decreases， and CPU usage increases as the number of variants increases. For each variant appended， latency increases by 16.92% and throughput decreases by 33.57% on average， and CPU usage increases by 12.04% on average.

8 Conclusion

This work introduced SecIngress， an API gateway framework to secure the cloud application based on *n-variant system*. We first reduce the performance degradation of waiting time for multiple responses by designing a two-stage timeout processing method. We then improved the voting mechanism of the major voting schema by designing an analytic hierarchy process voting under metadata mechanism. Finally， we implemented a prototype of SecIngress. Through security gain evaluation and performance degradation evaluation of our prototype， SecIngress demonstrated the practicality of decreasing the rate of reachable vulnerabilities or threats with acceptable performance degradation.

Reference:

[1] VidhyalakshmiR. and KumarV.，“Design comparison of traditional application and SaaS，” 2014 Int. Conf. Comput. Sustain. Glob. Dev. INDIACom 2014，pp. 541 – 544，2014，doi: 10. 1109/IndiaCom. 2014. 6828017.

[2] AleemS.， AhmedF.， BatoolR.， and KhattakA.，“Empirical Investigation of Key Factors for SaaS Architecture Dimension，” IEEE Trans. Cloud Comput.，pp. 1 – 14，2019，doi: 10. 1109/TCC. 2019. 2906299.

[3] McAfee. Security Issues in Cloud Computing. https://www. mcafee. com/enterprise/en-us/assets/reports/restricted/rp-carr-wfh. pdf，2020.

[4] GirardA. and RommelC.，“Safety in Embedded Software Challenges in Growing from Rapid Rising of Third-Party Code，” ATZelektronik Worldw.，vol. 11，no. 1，pp. 18 – 23，2016，doi: 10. 1007/s38314-015-0572-3.

[5] Alexandru G. Bardas1(B), Sathya Chandran Sundaramurthy2, Xinming Ou3，and S. A. D.（2017）. MTD CBITS: Moving Target Defense for Cloud-Based IT Systems. 2（Cuhk 439713），370 – 388. https：//doi. org/10. 1007/978-3-319-66402-6

[6] B. Cox et al.，“N-Variant Systems A Secretless Framework for Security through Diversity," Proc. 15th USENIX Secur. Symp. Vancouver，BC，August 2006，no. August，pp. 1 – 16，2006.

[7] A. D. Keromytis et al.，“The MEERKATS cloud security architecture," Proc. - 32nd IEEE Int. Conf. Distrib. Comput. Syst. Work. ICDCSW 2012，pp. 446 – 450，2012，doi: 10. 1109/ICDCSW. 2012. 42.

[8] UngureanuO. M.， VlădeanuC.， and KooijR.，"Kubernetes cluster optimization using hybrid shared-state scheduling framework，” IntACM. Conf. Proceeding Ser.， 2019， doi: 10. 1145/3341325. 3341992.

[9] BenameurA.， EvansN. S.， and ElderM. C.，"Cloud resiliency and security via diversified replica execution and monitoring，” Proc. - 2013 6th Int. Symp. Resilient Control Syst. ISRCS 2013，pp. 150 – 155，2013，doi: 10. 1109/ISRCS. 2013. 6623768.

[10] Kubernetes ingress. https：//kubernetes. io/docs/concepts/services-networking/ingress/.

[11] HAproxy. http：//www. haproxy. org/.

[12] Kong. www. konghq. com/.

[13] Nginx. www. nginx. com/.

[14] Apache Benchmark. http：//httpd. apache. org/docs/2. 4/programs/ab. html/.

[15] Chang J. M.， Chao H. C.， Chen J. L.， Lai C. F. : An efficient service

discovery system for dual-stack cloud file service. IEEE Syst. J. 6（4），584－592 (2012）

[16] Zhao L.，et al.：Flexible service selection with user-specific QoS support in service-oriented architecture. J. Netw. Comput. Appl. 35（3），962－973 (2012）

[17] ZhangM.，RanjanR.，HallerA.，GeorgakopoulosD.，MenzelM.，NepalS.，An ontology-based system for Cloud infrastructure services' discovery, in：8th International Conference on Collaborative Computing: Networking，Applications and Worksharing (CollaborateCom)，Pittsburgh，PA，USA，2013，pp. 524－530.

[18] Saravana BalajiB.，KarthikeyanN. K.，RajkumarR. S.，Fuzzy service conceptual ontology system for cloud service recommendation, Comput. Electr. Eng. 69（2018) 435－446.

[19] AliA.，ShamsuddinS. M.，EassaF. E.，MohammedF.，Cloud service discovery and extraction：a critical review and direction for future research，in：International Conference of Reliable Information and Communication Technology (IRICT)，Kuala Lumpur，Malaysia，2018，pp. 291－301.

[20] NoorT. H.，ShengQ. Z.，NguA. H. H.，DustdarS.，Analysis of web-scale cloud services, IEEE Internet Comput. 18 (4) (2014）55－61.

[21] NoorT. H.，ShengQ. Z.，AlfaziA.，NguA. H. H.，LawJ.，CSCE：A crawler engine for cloud services discovery on the world wide web，in：20th IEEE International Conference on Web Services，Santa Clara，CA，USA，2013，pp. 443－450.

[22] AlkalbaniA.，ShenoyA.，HussainF. K.，HussainO. K.，XiangY.，Design and implementation of the hadoop-based crawler for saas service discovery, in: 29th IEEE International Conference on Advanced Information Networking and Applications, Gwangiu, KoreaSouth，2015，pp. 785－790.

[23] ParhiM.，PattanayakB. K.，PatraM. R.，A multi-agent-based framework for cloud service discovery and selection using ontology, Serv. Orient. Comput. Appl. 12（2) (2018）137－154.

[24] NganL. D.，KanagasabaiR.，OWL-S based semantic cloud service broker, in: 19th IEEE International Conference on Web Services (ICWS), Honolulu, HI, USA，2012，pp. 560－567.

[25] RekikM.，BoukadiK.，Ben-AbdallahH.，Cloud description ontology for service discovery and selection, in: 10th International Joint Conference on Software Technologies (ICSOFT), Colmar, France，2015，pp. 1－11.

[26] JacksonT.，SalamatB.，WagnerG.，WimmerC.，and FranzM.，"On the effectiveness of multi-variant program execution for vulnerability detection and prevention，”
in Proceedings of the 6th International Workshop on Security Measurements and Metrics, ser. MetriSec '10. New York，NY，USA: ACM，2010，pp. 7:1－7:8.

[27] KoningK.，BosH.，and GiuffridaC.，“Secure and efficient multi-variant execution using hardware-assisted process virtualization，”in 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)，June 2016，pp. 431－442.

[28] BruschiD.，CavallaroL.，and LanziA.，"Diversified process replicæ for defeating memory error exploits，”in 2007 IEEE International Performance, Computing, and Communications Conference, April 2007，pp. 434－441.

[29] VolckaertS.，CoppensB.，De SutterB.，De BosschereK.，LarsenP.，and FranzM.，"Taming parallelism in a multi-variant execution environment，”
in Proceedings of the Twelfth European Conference on Computer Systems, ser. EuroSys '17. New York，NY，USA: ACM，2017，pp. 270－285.

[30] Saaty T. Decision making with the analytical hierarchy process. International Journal of Services Sciences. 2008；1(1)：83-98.

[31] Alexander M. Decision making using analytical hierarchy process (AHP); 2012 SAS/IML

[32] Nataliya DP, Nadezhda IN. Sensitivity analysis of a decision - Making problem using the analytical hierarchy process. International Journal "Information Theories and Application. 2016；23(3)

[33] Polinsky, Isaac, et al. "N-m-Variant Systems: Adversarial-Resistant Software Rejuvenation for Cloud-Based Web Applications.” Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy，2020，pp. 235－246.

[34] Levitin, Gregory, et al. "Optimal Early Warning Defense of N-Version Programming Service against Co-Resident Attacks in Cloud System.”Reliability Engineering & System Safety，vol. 201，2020，p. 106969.

[35] Voulimeneas, Alexios, et al. "DMON: A Distributed Heterogeneous N-Variant System. " ArXiv Preprint ArXiv：1903. 03643，2019.

[36] Franz M. Making multivariant programming practical and inexpensive. IEEE Secur Privacy 2018；16(3)：90－4.

[37] Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, et al. A view of cloud computing. Commun ACM 2009；53(4)：50－8.

[38] Pramila S, Poonkuzhali S, Mythili S. Improvising reliability through N-version programming in cloud environment. Int J Adv Tech Eng Sci April 2015；3(1)：204－8.

[39] Liu J, Yang N. Proc. of 2017 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC). Optimal fault tolerant service provisioning for cloud application. Macau 2017：189－94.

[40] F. Khomh, "On improving the dependability of cloud applications with fault-tolerance，" Proc WICSA，Article No. 2，pp. 1－3，https：//doi. org/10. 1145/2578128. 2578228，April 2014.

[41] Wagner B, Sood A. Economics of Resilient Cloud Services. Proc 2016 IEEE Int Conf Softw Qual Reliab Secur Compan (QRS-C) 2016：368－74. https：//doi. org/10. 1109/ QRS-C. 2016. 56.

[42] R. WhiteD.，"Cloud Computing and SBSE，" In：RuheG.，ZhangY. (eds) Search Based Software Engineering. SSBSE 2013. Lecture Notes in Computer Science，vol 8084. Springer, Berlin, Heidelberg，2013.

[43] Jpress. https：//github. com/JpressProjects/jpress/.

[44] Vulscan. https：//github. com/scipag/vulscan/.

[45] OWASP ZAP. https：//owasp. org/www-project-zap/.

[46] BhatkarSandeep, DuVarneyDaniel, and SekarR.. Address Ofuscation: an Efficient Approach to Combat a Broad Range of Memory Error Exploits. USENIX Security 2003.

[47] Monica Chew and Dawn Song. Mitigating Buffer Overflows by

Operating System Randomization. Tech Report CMU-CS-02-197. December 2002.

[48] BarrantesElena, AckleyD. , ForrestS. , PalmerT. , StefanovicD. , ZoviD. . Intrusion Detection: Ran- domized Instruction Set Emulation to Disrupt Bi- nary Code Injection Attacks. CCS 2003.

Which Features Matter in Recognizing Phishing Emails?

WANG Xiujuan[1], ZHENG Qianqian[1], HONG Weijie[1], ZHANG Yuyang[1], YANG Rundong[2], WANG Zhe[2]

1.Country Faculty of Information Technology, Beijing University of Technology, Beijing, 100124;

2.School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing,100876

Key words: Phishing emails; feature analysis; MIC

Abstract. Phishing emails attack remains an active way to obtain victims' sensitive information. Therefore, phishing detection is still an important research focus. Researchers have extracted many kinds of attributes from emails so as to facilitate phishing emails detection with machine learning algorithms. In this paper, we propose to analyze the different effects of attributes for the email category, phishing or normal. Six metrics are adopted in this paper including some new statistical correlation method such as distance correlation and Maximal Information Coefficient. Finally, 4 classifiers are introduced to verify the validation of selected features. Experiments over 10417 emails show that classification with 4 features out of 39 original ones can achieve 99.34% accuracy rate and 99.35% F1 score.

1 Introduction

Phishing is a criminal mechanism employing both social engineering and technical subterfuge to steal consumers' personal identity data and financial account credentials. According to the global phishing survey published by APWG [1], 'The number of unique phishing e-mail reports received by APWG from consumers during Q1 was 557, 964. This was an notable increase from the 173, 262 received in Q4 of 2015.' Phishing attackers can obtain victims' sensitive information such as logon credential, credit card details and so on. Furthermore, malicious code can be planted in victims' computer during phishing attack for further damage.

Phishing email is one of the main forms of phishing attack. Especially, spear phishing often acts as the first and key step of Advanced Persistent Threats (APT) attack, which is the most complicated form of attack by far. Due to the high risk and world-wide influence, anti-phishing researches have been developed during the past decades. Despite the growth of prevention techniques, phishing remains an important threat since the principal countermeasures in use are still based on reactive URL black listing [2]. Furthermore, phishing emails have developed new trends. Attackers usually target at specific goal to implement phishing. To this end, attackers may pretend as the acquaintances to send phishing email which helps eliminating the dubiety and improve succeeds. This new characteristic brings challenge to researchers. Machine learning is combined in distinguishing phishing emails from legitimate ones. An important step is to find 'good' features used in classification.

To this end, the paper proposes to analyze fea-

tures extracted from emails or mined from related information around them to pick up the most discriminable ones. The paper adopted several measures in evaluating features to find the 'best' features including information gain, mutual information, Pearson correlation maximal information coefficient, etc. Finally, a series of familiar classifiers are adopted to verify the validity of selected features, as illustrated in Fig.1

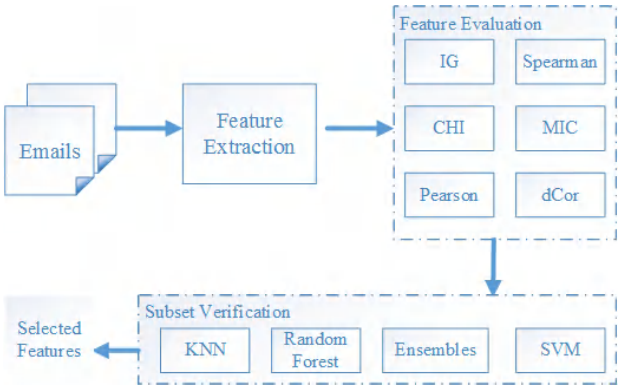


Fig. 1 Framework of this work

This paper is organized as follows: section II summarizes related work. Details of our work are provided in section III. In section IV and V, we present the experimental results and a conclusion is made in the end.

2 Related Work

Many anti-phishing researches have been developed in recent years in order to combat phishing attacks. The work can be summarized into 2 classes according to the actor in consideration the research focus.

1) Considering human related factors. Many researchers conducted study to find the connection between human-related factors and phishing success. Literature [3] examined how users' attention to "visual triggers" and "phishing deception indicators" influenced their decision-making processed and consequently their decisions. They used a real phishing email as a stimulus and surveys 321 members of a public university community in the Northeast US. The results implied that attention to visceral triggers, attention to phishing deception indicators and phishing knowledge played critical roles in phishing detection. Literature [4] conducted a user study to assess whether improved browser security indicators and increased awareness of phishing have led to users' improved ability to protect themselves against such attacks. Literature [5] explored the effectiveness of embedded training and found that many participants did not read the training, and clicked either all links or none across three trials. Literature [6] evaluated the anti-phishing knowledge retention for users and found that users retained their anti-Phishing knowledge after 16 days from their first training. Users' decision in the time of training were slightly better than that after about 16 days. Therefore, anti-phishing techniques are the first defense.

2) Phishing detection. Recognizing phishing emails before users open them is an important way to avoid further hurt. Many researches have investigated on improving the accuracy of said reorganization. Two kinds of methods are adopted here, filtering based on rules [7-8] and statistical methods [9-17]. Statistical methods need to extract features from the phishing emails firstly. Then machine-learning algorithms are applied to these features to distinguish the phishing emails. Literature [9] used 9 URL features and WHOIS information to detect phishing emails. Experiments on 860 phishing emails and 6950 non-phishing emails got 96% classify accurate. Literature [10] used 12 keywords extracted from the email bodies and employed Multilayer Perceptron, Decision Trees, Support Vector Machine etc. to detect phishing emails. A dataset of 2500 emails were analyzed. Literature [11] combined automatic and transparent email signing with an email client plugin to detect unsigned spoofed messages. Literature [12] combined 7 behavior-based features and content-based features. Information Gain, Gain Ratio and Symmetrical Uncertainty were used to test the quality of said features. The detecting accuracy rate on 6923 emails was 93%. Literature [13] used social features extracted from LinkedIn. Various

machine learning algorithms were applied to identify spear phishing emails. However, it was revealed in [13] that social features extracted from LinkedIn did not help in phishing detecting. Literature [14] proposed profiling phishing email model and profiling of email-born phishing (ProEP) algorithm for filtering phishing emails. Literature [15] adopted dynamic evolving Neural Network based on reinforcement learning to detect online phishing email. Fang Y et al. proposed to detect phishing email using improved RCNN model with multilevel vectors combined with attention mechanism [16].

Statistical techniques show more advantage in identifying new phishing emails. However, features are the premise of statistical classification. Valid features can accompany better classification performance. Most researches put the extracted features into the classifiers directly. A few work conducted simple dimension reduction [16]. The validity of features needs to be analyzed and verified with experiments. Some study [17] selected features with Information Gain, wrapper, inconsistency and correlation-based feature subset evaluator. The selected feature subset in this study resulted in a classification model with an F1 score of 99.396% for 21 heuristic features and a single classifier. Literature [18] investigated the entropy and information gain over 40 features that have been used in recent literature. Information gain for these features are calculated over Ham, Spam and Phishing corpora.

However, these works developed feature selecting with limited methods. To this end, this paper aims at analyzing the discriminability of each feature with many measures to pick out the 'best' features for the followed classification. The difference between this work with others lies in the combination of 6 measurements including MIC and dCor etc. as well as cross validation with different classifiers.

3 Methodology

Our goal is to develop effective features that can help in classifying whether an email is likely to be malicious or not.

In this section, we first describe the variety kinds of phishing emails. After that, main features used in related studies are presented and popular measurements are described to excavate the relationships between features and phishing category.

Detailed, the aimed problem is described as followed. An email set D comprises M emails and N features a_i coming from a defined attribute set A are extracted for each email. They can be classified into L categories C including legitimate email and L-1 kinds of phishing emails. Then, $D = \{\boldsymbol{d}_i\}\ i = 1, 2, \cdots, M$, $\boldsymbol{d}_i = (a_{i,1}, a_{i,2}, \cdots, a_{i,N})$ and $C = \{c_j\}, j = 1, 2, \cdots, L$. Phishing detecting turns to find a best hypothesis $\boldsymbol{H}_\theta$ which can map D to C, i.e.

$$\boldsymbol{H}_\theta(\boldsymbol{d}_i) \rightarrow c_j$$

Actually, N features $(x_{i,1}, x_{i,2}, \cdots, x_{i,N})$ is not equally effective for said map. Therefore, some researchers focuses on finding a subset $D_s \subset D$ that can improve the detecting accuracy by evaluating the individual importance of $(x_{i,1}, x_{i,2}, \cdots, x_{i,N})$ based on some measurements.

3.1 Features

Varity of features have been extracted in detecting phishing emails [9, 19] including content-based features, structural features, external features [20] etc. However, phishing emails tend to be more similar to the normal ones today. Therefore, approach based on features representing frequency of "bag of words" becomes unreliable and these features are not considered here. At last, 39 features are analyzed here as followed with reference to [9, 19, 21].

1. ***Subject has keyword bank*** . A boolean feature indicating whether word "bank" exist in the subject field of a given email message.

2. ***Subject has keyword debit.*** A boolean feature indicating whether word "debit" exist in the subject field of a given email message.

3. ***Subject has keyword FW***. A boolean feature that returns 1 if the word "Fwd: " is found in the

subject field of a given email message， and 0 otherwise.

4. ***Subject has keyword RE***. A boolean feature representing the existence/absence of word "Re： " in the subject field of a given email message.

5.***Subject has keyword verify***. A boolean feature related to the presence/absence of word "verify" in the subject field of an email.

6**. *Subject word number***. The total number of words in the email subject.

7. ***Subject character number***. The total number of characters in the email subject.

8.***Word_num and Character_num***. A continuous feature that returns the division of the total number of words by the total number of characters in the email subject.

9. ***Sender and reply-to addresses***. A boolean feature signifying the difference between the sender email address and reply-to addresses.

10.***HTML emails.*** A binary feature that is set to 1 if the email contains a section that is denoted with a MIME type of text/html. message has HTML content， and 0 otherwise.

11.***Number of dots.*** An integer feature that returns the maximum number of dots （ '.' ） contained in any of the links present in the email， as defined in［9.］.

12 ***Sender domain is modal domain***. A binary feature that returns 1 in the case where the sender email address uses an unmodal domain name, and 0 otherwise

13.***Here links to non_modal domain***. A binary feature referenced in［9］.

14.***Age of linked_to domain***. As mentioned in［9］， the registered time of the domain name linked by the embedded URL in the email can be reported with a WHOIS query. The life period refers to the duration from the registered time to the received time of the email. If the email contains multiple URLs， than the maximum life period is stored. This is a continuous feature.

15.***Number of external links***. A continuous feature that records the total number of external links included in a given email. An external link is a link that points to a resource that is accessible out of the email（e.g. a website）.

16. ***The existence of port number***. A binary feature corresponding to the existence/absence of URLs with port numbers in the email body.

17.***Number of port***. A continuous feature that returns the total number of URLs with port numbers in their authority section in a given email.

18 ***Number of @.*** A continuous feature that gives the number of "@" signs within the URL in the email. [21]

19.***Number of IP address***. A continuous feature that measures the number of links containing IP addresses rather than fully qualified domain names in an email.

20.***IP_based URLs*** . A binary feature indicating whether or not a URL with an IP address is included in the email authority portion.

21. ***Number of domains***. A continuous feature that returns the total number of domains linked with URLs in an email.

22.***Number of links***. A continuous feature that returns the total number of links contained in the email body.

23.***Nonmatching URLs***. As referenced in［9］， all links are checked and this binary feature is set to 1 if the HREF of the link in the email is to a different host than the link in the text.

24 ***URL click***. A binary feature that returns 1 if a link with "click" word in its link text is found, and

0 otherwise.

25.***URL here***. A binary feature that returns 1 if a link with "here" word in its link text is found， and 0 otherwise.

26. ***Internal link number***. A continuous feature that presents the total number of internal links in an email. An internal link is a link that points to a re-

source that is accessible in the email （e.g. anchors within the email — ragments）.

27 ***Linked image number***. A continuous feature that returns the total number of image links contained in an email

28. ***onClick JavaScript events.*** A binary feature corresponding to the presence/absence of an "onClick" JavaScript event in the email body.

29. ***Javascript from an unmodal domain***. The feature represents the presence of external javascript forms that come from domains other than the modal domain.

30. ***Change status***. A binary feature that returns 1 if an email contains JavaScript code modifying the status bar.

31. ***Pop-up window***. A binary feature that returns 1 if an email contains JavaScript code opening a pop-up windows.

32. ***The existence of Javascript.*** A binary feature reporting the existence/absence of JavaScript in the email body.

33. ***Suspension***. A binary feature reporting the existence/absence of the word "suspension" in the email body.

34. ***Dear***. A binary feature reporting the existence/absence of the word "dear" in the email body.

35. ***Verify your account***. A binary feature reporting the existence/absence of the phrase "verify your account" in the email body.

36. ***The existence of body forms***. A binary feature that returns 1 if the email message contains a HTML form, and 0 otherwise.

37. ***Text Word_num and Character_num***. A continuous feature that returns the division of the total number of words by the total number of characters in the email body.

38. ***Number of account***. A continuous feature counting the total number of the word "account" in the email body.

39. ***Suspended number***. A continuous feature that returns the total number of word "suspended" in the email body.

3.2 Evaluating features

As known to the skilled in machine learning art, subset selection evaluates a subset of features as a group for suitability. In other words, evaluating features aims at finding the most informative and discriminative features. Subset selection techniques can be broken up into Wrappers, Filters and Embedded. In this paper, Filters are firstly adopted due to its less computations. Detailed, each feature/ attribute A_i in the feature set A is evaluated to get a score reflecting its discriminability among classes C according to certain metric. Wherein, A_i and C are quantified with a M-dimensional vectors $\boldsymbol{a}_i$ and $\boldsymbol{c}$, individually. $\boldsymbol{a}_i = \left(a_{1,i}, a_{2,i}, \cdots, a_{M,i}\right)$ and $\boldsymbol{c} = \left(C_1, C_2, \cdots, C_M\right)$, M is the number of emails in the training dataset. All of the features are sorted according to said score and the subset A_s consists of K best features are chosen to represent emails [22], i.e.

$$A = \{A_i\} i = 1,2,\cdots,N, A_{sub} \subset A \tag{1}$$

$$A_{sub} = \{A_i\} F(A_i) > F(A_j) \ \forall A_i \in A_s, A_j \in A - A_{sub}\} \tag{2}$$

Popular metrics F involves in statistical correlation methods, consistency-based methods, mutual information and so on. Detailed metrics adopted in our scheme are presented as followed.

1. Mutual information.

In probability theory and information theory, the mutual information (*MI*) measures the mutual dependence between two variables. Therefore, it can quantify the 'amount of information' obtained about the class variable C through attribute A_i. [23]

Formally, the mutual information between $\boldsymbol{a}_i$ and $\boldsymbol{c}$ can be defined as:

$$I(\boldsymbol{a}_i, \boldsymbol{c}) = \sum_k \sum_j p(a_{i,k}, c_j) log\left(\frac{p(a_{i,k}, c_j)}{p(a_{i,k})\, p(c_j)}\right) \tag{3}$$

where $p(a_{i,k}, c_j)$ is the joint probability when the target attribute a_i takes the value of $a_{i,k}$ and the corresponding email belongs to class j. And $p(a_{i,k})$ and $p(c_j)$ are the probability that attribute a_i takes the value of $a_{i,k}$ and the probability of class j, individually.

2 Statistical correlation metrics

Pearson correlation

Pearson correlation coefficient r measures the linear correlation between two variables.

Formally,

$$r(\boldsymbol{a}_i,\boldsymbol{c}) = \frac{\sum_k (a_{i,k} - \overline{a}_i)(c_k - \overline{C})}{\sqrt{\left(\sum_k (a_{i,k} - \overline{a}_i)\right)^2}\sqrt{\left(\sum_k (c_k - \overline{C})\right)^2}} \quad (4)$$

Pearson correlation coefficient usually reflects the extent of linear correlation between two variables and it may range from -1 to 1. Wherein, the closer to 1 $r(\boldsymbol{a}_i, \boldsymbol{c})$is, the stronger linear correlation exists between attribute $\boldsymbol{a}_i$ and class C and they are positive correlative. $r(\boldsymbol{a}_i, \boldsymbol{c})$ around -1 means negative correlation. However, if $r(\boldsymbol{a}_i, \boldsymbol{c})$ is close to zero, then attribute a_i and class C owe weak linear correlation. [24]

Spearman correlation coefficient

Spearman correlation coefficient ρ, also known as rank correlation coefficient, is a nonparametric measure of statistical dependence between two variables which assesses how well the relationship between them can be described using a monotonic function [25].

Formally, the Spearman correlation coefficient between attribute $\boldsymbol{a}_i$ and class $\boldsymbol{c}$ is defined as:

$$\rho = \frac{\sum_k (r_k - \overline{r})(s_k - \overline{s})}{\sqrt{\sum_k (r_k - \overline{r})^2}\sqrt{\sum_k (s_k - \overline{s})^2}} \quad (5)$$

wherein, r_k and s_k are the ranks of $a_{i,k}$ and c_k, individually. $\overline{r}$ is the mean of r_k. So does $\overline{s}$.The value of ρ ranges from -1 to 1. ρ will score 1 if $\boldsymbol{a}_i$ increases monotonically with $\boldsymbol{c}$. Otherwise it equals to -1.

CHI-square

CHI-square is a measure based on the theory of hypothesis testing in Statistics, denoted as χ^2. CHI-square test is a generally used non-parameter hypothesis test method in order to analyze the correlation between two or more samples. It has been introduced to the field of feature selection in text classification [26].

Formally, χ^2 between attribute $\boldsymbol{a}_i$ and class $\boldsymbol{c}$ is defined as:

$$\chi^2 = \frac{M(A*D - C*B)}{(A+C)(B+D)(A+B)(C+D)} \quad (6)$$

Wherein, M is the size of training corpus, i.e., the total number of training emails. A is the number of phishing emails with attribute $\boldsymbol{a}_i$ >0. B is the number of phishing emails with attribute $\boldsymbol{a}_i$ =0. C is the number of normal emails with attribute $\boldsymbol{a}_i$ >0. D is the number of normal emails with attribute $\boldsymbol{a}_i$ =0.

dCor (distance correlation)

dCor is a new statistics put forward by Szkely et. al in 2007 [27]. It can measure nonlinear or non-monotonic relationship between two random variables or random vectors of arbitrary dimension by computing the Euclidean distance between them. dCor is zero if and only if the two random variables are statistically independent.

Formally, for $\boldsymbol{a}_i = (a_{1,i}, a_{2,i}, \cdots, a_{M,i})$ and $\boldsymbol{c} = (C_1, C_2, \cdots, C_M)$, the dCor between them can be defined as [28]:

$$dCor(\boldsymbol{a}_i,\boldsymbol{c}) = \begin{cases} \sqrt{\dfrac{V^2(\boldsymbol{a}_i,\boldsymbol{c})}{\sqrt{V^2(\boldsymbol{a}_i)V^2(\boldsymbol{c})}}}, & V^2(\boldsymbol{a}_i)V^2(\boldsymbol{c}) > 0 \\ 0, & V^2(\boldsymbol{a}_i)V^2(\boldsymbol{c}) = 0 \end{cases} \quad (7)$$

wherein, $V(\boldsymbol{a}_i)$ and $V(\boldsymbol{c})$ are the distance variance of attribute $\boldsymbol{a}_i$ and class $\boldsymbol{c}$, respectively. $V(\boldsymbol{a}_i, \boldsymbol{c})$ is the distance covariance. Detailed,

$$V(\boldsymbol{a}_i,\boldsymbol{c}) = \sqrt{V^2(\boldsymbol{a}_i,\boldsymbol{c})}$$

$$= \sqrt{\left\| f_{a_i,c}(t,s) - f_{a_i}(t) f_c(s) \right\|^2}$$

$$= \sqrt{\frac{1}{p_M^2}\int_{\mathbb{R}_{2M}} \frac{\left| f_{a_i,c}(t,s) - f_{a_i}(t) f_c(s) \right|^2}{|t|_M^{1+M} |s|_M^{1+M}} dtds} \quad (8)$$

$$V(\boldsymbol{a}_i) = \sqrt{V^2(\boldsymbol{a}_i,\boldsymbol{a}_i)} \quad (9)$$

$f_{a_i,c}(t,s)$ is the joint characteristic function of and $f_{a_i}(t)$ and $f_c(s)$ are the characteristic functions of $\boldsymbol{a}_i$

and $\boldsymbol{c}$, respectively. $p_M = \frac{\pi^{1+M/2}}{\Gamma\left((1+M)/2\right)}$. $\|\cdot\|$ represents the Euclidean distance.

MIC（Maximal Information Coefficient）

MIC was introduced in ［29］ published on ‘Science’ in 2011 by Reshef et.al to identify the correlations between pairs of variables. The basic idea lies in that if a relationship exists between pairs of variables, then a grid can be drawn on their scatterplot that partitions the data to encapsulate that relationship. The author claimed that this new measurement had two properties of generality and equitability and it could identify the widely correlations between two variables.

To calculate the maximal information coefficient between attribute $\boldsymbol{a}_i$ and class $\boldsymbol{c}$, the training dataset $\boldsymbol{D}_i$ which are constituted by dual variables （$\boldsymbol{a}_i$, $\boldsymbol{c}$） will be divided into different x rows or y columns according to the value of $\boldsymbol{a}_i$ and $\boldsymbol{c}$. The division is named x-*by*-y grids G and $\boldsymbol{D}_i|_G$ is the probability distribution of on G. For a given $\boldsymbol{D}$, different division generates diverse $\boldsymbol{D}_i|_G$.

$$MIC(\boldsymbol{D}_i) = \max_{xy<B} \frac{I^*(\boldsymbol{D}_i,x,y)}{\log_2 min(x,y)} \tag{10}$$

$$I^*(D_i,x,y) = max_G I\left(\boldsymbol{D}_i|_G\right)$$

$$= max_G\left(\sum_{x,y} p(x,y)log\left(\frac{p(x,y)}{p(x)p(y)}\right)\right) \tag{11}$$

wherein B is a function of sample size and usually is set to $M^{0.6}$. M is the number of sampled emails in the training dataset.

MIC ranges from 0 to 1. For sufficient training emails, $MIC(\boldsymbol{D}_i)$ tends to zero if $\boldsymbol{a}_i$ and $\boldsymbol{c}$ are statistically independent. Otherwise, it will close to 1.

4 Experiments

Corpus in this paper comprises 4423 phishing emails come from the public monkey phishing corpus as used in literatures ［9］, ［10］, ［12］, ［14］, ［16，［17］, ［30］ and 5994 normal emails from Enron email dataset.

Measurements stated in preamble are calculated for each 39 features after feature extraction and normalization, wherein, the MIC between features and category are conducted by means of MINE ［31］. All of the measurements results are illustrated in Fig.2.

These 39 attributes behave different roles under distinct measurement. In order to verify the validity of selected subset, this paper carries out phishing detection with several classifiers on emails featured with said subset including kNN （k-nearest neighbor）, RF （Random Forest）, SVM （Support Vector Machine） ［32］ and Ensemble （Ensembles for Boosting, Bagging, or Random Subspace）. Accuracy （A） and F1-score（F1） ［16，30］ are introduced as the evaluation criteria. The former measures the classifying performance over both phishing emails and ham ones and the latter focuses on the detecting performance against phishing emails. Their definitions are listed in （1-2）, respectively.

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$F1 = \frac{2*P*R}{P + R} \tag{2}$$

Wherein,

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{FN + TP}$$

And,

TP （True Positive）: the number of phishing emails correctly classified as phishing.

TN （True Negative）: the number of normal emails correctly classified as normal.

FP （False Positive）: the number of normal emails falsely classified as phishing.

FN （False Negative）: the number of phishing emails falsely classified as normal.

Experiment 1 is developed to verify the validation of feature selection in phishing detection. In the experiment, both mean classification accuracy and F1 score are around 99% with all the 39 original features. Then, we pick out top different K ‘best’ features to construct different feature subset of size K. Here, ‘best’ means selected features owe higher metrics value as described above and the K ranges

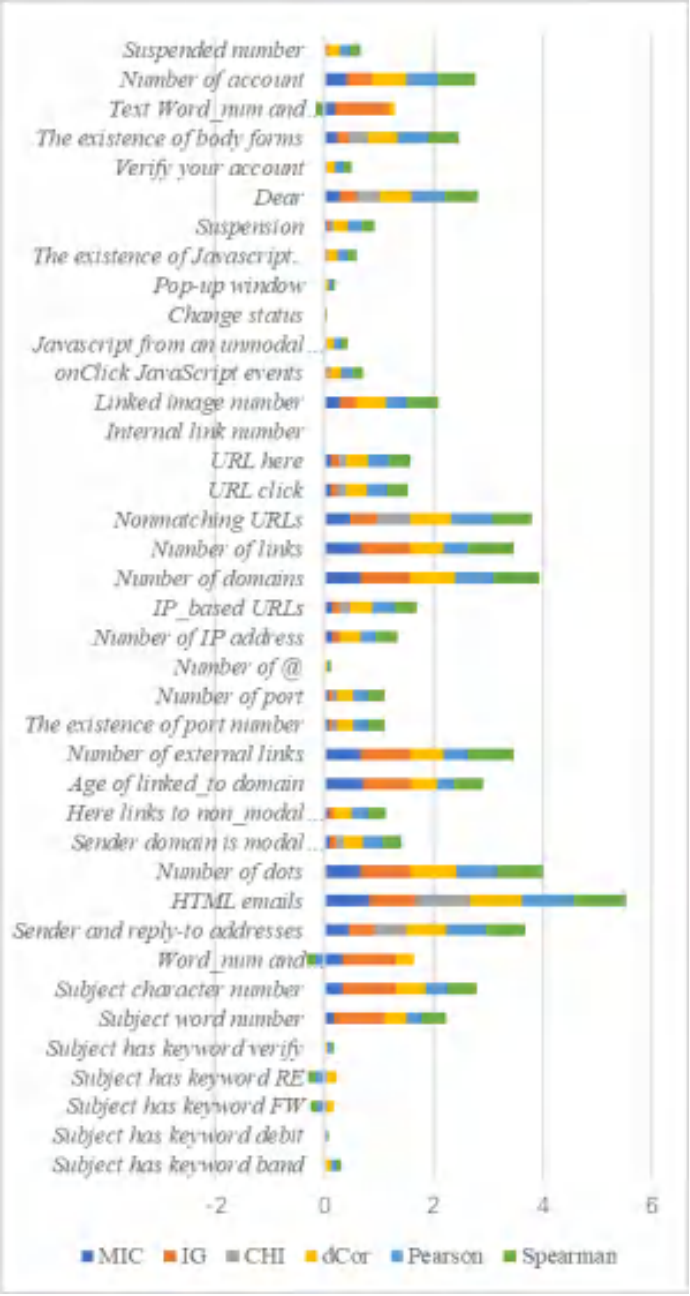


Fig. 2 Measurements results for each feature

from 2 to 20 in this experiment. Then 3-fold cross validation is involved to classify the emails into phishing or legitimate with kNN， RF， SVM and Ensemble. The average accuracy and F1-score are illustrated in Fig.3 and Fig.4， individually. As shown in Fig.3 and Fig. 4， the classification accuracy and F1 score ascend when K grows up no matter which classifier is adopted. When K is much less than 39， we can get relatively high accuracy and F1-score about 99% which is equivalent to the classification with original features. In detail， when K is set to 5/9/15/4/8/6 with Pearson/Spearman/IG/ CHI/MIC/dCor metric， the classification performance is cost-effective. Therefore， a few number of features are good enough to support phishing emails detection.

Experiment 2 is developed to find out the ’best’ features and verify their effectiveness. We list top 15 ‘best’ features under different metrics， as shown in Talbe1. It can be seen in Table 1， Pearson method， Spearman method， IG method， CHI

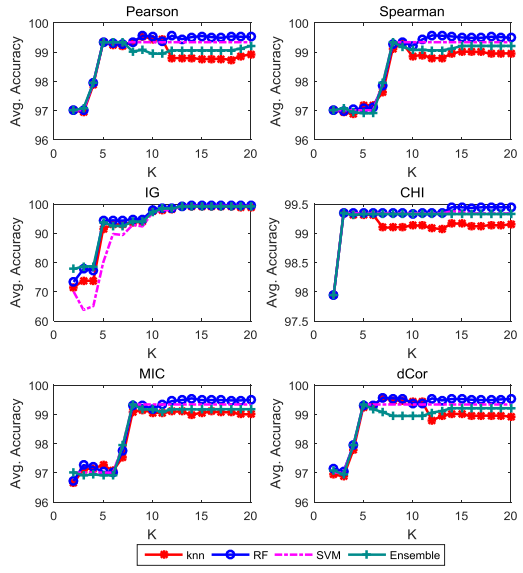


Fig. 3 Average accuracy

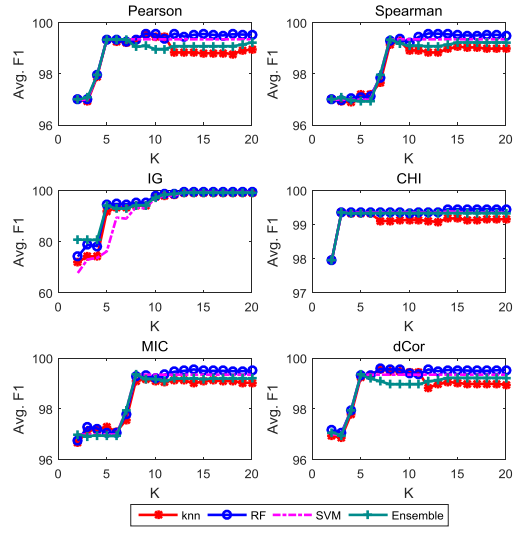


Fig. 4 Average F1

method， MIC method and dCor method share 4 common features in their top 15 selected ones， i.e. feature 9，10， 23，34. Then we carry out classification based on these 4 common features and compare the accuracy and F1 score with original classification and different feature selecting metric， as shown in Fig.5 and Fig.6. The detection with common features behaves considerable performance with the original one， i.e.， detection with all 39 features. Pearson， Spearman， MIC and dCor methods bring a little lower accuracy and F1 than the original one when K equals to 4. IG-based method behaves worst when K equals to 4. Furthermore， CHI method manifests stable and good performance no matter which classifier is adopted in the validation. It can be learned from Table 1 that the common features happen to be the selected features for CHI method when K is set to 4. However， the top 4 features for the other metrics are different to the common features more or less which can explain the classification performance difference between these 5 metrics and the CHI method. Therefore， we suggest that features 9， 10， 23， 34 are the cost-effective features in phishing emails detection.

TABLE I ‘Best’ features under different measurements

Metrics	TOP15 FEATURES IN ORDER
Pearson	26 **10** 11 **23** **9** 21 36 **34** 38 15 22 7 20 27 25
Spearman	26 **10** 11 21 15 22 **23** **9** 38 **7** 36 **34** 27 14 6
IG	7 37 8 6 15 21 22 11 14 **10** **23** 38 **9** **34** 27
CHI	**10** **23** **9** **34** 36 20 25 24 12 13 16 17 33 28 32
MIC	**10** 14 21 11 15 22 **23** **9** 38 8 7 **34** 27 36 37
dCor	**10** 11 21 **23** **9** 38 **34** 15 22 36 27 7 14 6 20

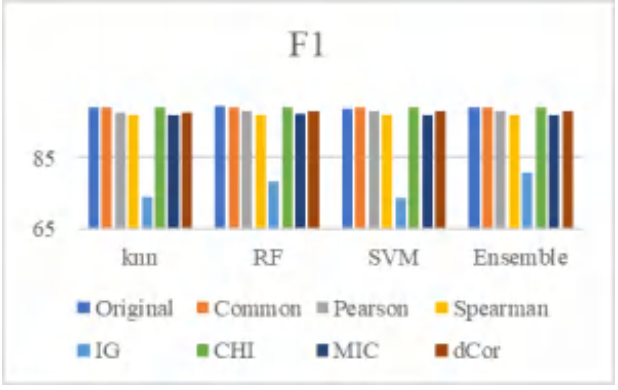


Fig. 5 F1 comparison when K=4

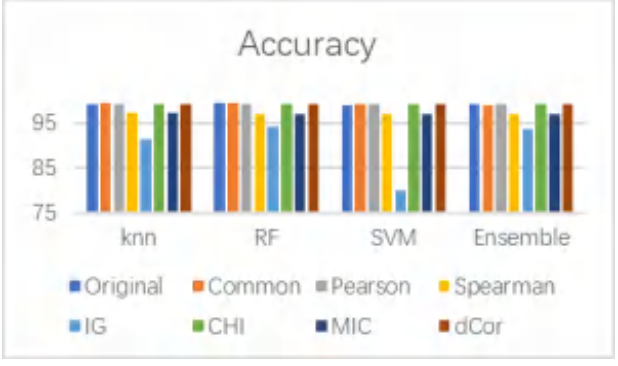


Fig. 6 Accuracy comparison when K=4

5 Conclusion

In this paper we propose to measure attributes of emails in order to conduct phishing detection with selected feature subset as few as possible. Totally 39 original features are analyzed with reference to existing research works and new metrics such as MIC and dCor are considered in our paper in addition to traditional IG and CHI. Finally， four classifiers are adopted to verify the validation of selected features. Experiments on 10417 emails show that feature selection can help in reducing the dimensions of vectors as well as the cost of classification. Meanwhile， the accuracy and F1 score are both considerable satisfied.

References:

[1] Anti Phishing Working Group: Phishing Activity Trends Report, Technical Report 1st Quarter 2016，[EB/OL].

[2] Marchal S , Francois J , State R , et al. PhishStorm: Detecting Phishing with Streaming Analytics [J]. IEEE Transactions on Network & Service Management，2017，11(4):458-471.

[3] Wang J , Herath T , Chen R , et al. Research Article Phishing Susceptibility: An Investigation Into the Processing of a Targeted Spear Phishing Email [J]. IEEE Transactions on Professional Communication，2012，55(4):345-362.

[4] Alsharnouby M , Alaca F , Chiasson S . Why phishing still works: User strategies for combating phishing attacks [J]. International Journal of Human - Computer Studies，2015，82:69-82.

[5] Caputo D D , Pfleeger S L , Freeman J D , et al. Going Spear Phishing: Exploring Embedded Training and Awareness [J]. IEEE Security & Privacy，2014，12(1):28-38.

[6] Alnajim, Abdullah, Munro, Malcolm. An Evaluation of Users' Anti-Phishing Knowledge Retention[M]. 2009.

[7] Sheng, Steve, et al. ：An Empirical Analysis of Phishing Blacklists [C]. Conference on Email & Anti-Spam 2009，2009，pp59－78.

[8] Marino J P , Fortin J . Apparatus and method for analyzing and filtering email and for providing web related services: US 2011.

[9] Fette I , Sadeh N M , Tomasic A . Learning to Detect Phishing Emails [C]. International Conference on World Wide Web. DBLP，2007，pp 649-656

[10] Pandey M , Ravi V . Detecting phishing e-mails using text and data mining [C]. Computational Intelligence & Computing Research (ICCIC)，2012 IEEE International Conference on. IEEE，2012，pp1-6.

[11] Crain J , Opyrchal L , Prakash A . Fighting Phishing with Trusted Email[C]. Ares 10 International Conference on Availability. IEEE，2010. pp 462-467.

[12] Hamid I R A , Abawajy J . Phishing Email Feature Selection Approach[C]// IEEE International Conference on Trust，Security & Privacy in Computing & Communications. IEEE，2011，pp916-921.

[13] Dewan P , Kashyap A , Kumaraguru P . Analyzing Social and Stylometric Features to Identify Spear phishing Emails[J]. Electronic Crime Research IEEE，2014，pp1－13.

[14] Hamid I R A , Abawajy J H . Profiling Phishing Email Based on Clustering Approach[C]. IEEE International Conference on Trust. 2013，pp628-635.

[15] Smadi S , Aslam N , Zhang L . Detection of Online Phishing Email using Dynamic Evolving Neural Network Based on Reinforcement Learning [J]. Decision Support Systems，2018，107(MAR.)：88-102.

[16] Fang Y , Zhang C , Huang C , et al. Phishing Email Detection Using Improved RCNN Model With Multilevel Vectors and Attention Mechanism[J]. IEEE Access，2019，7:56329-56340.

[17] Hamid I R A , Abawajy J , Kim T H . Using feature selection and classification scheme for automating phishing email detection [J]. Studies in Informatics & Control，2013，22(1)，pp 61-70.

[18] Khonji M , Jones A , Iraqi Y . A study of feature subset evaluators and feature subset searching methods for phishing classification[C]. Collaboration, MessagingElectronic, Anti-abuse & Spam Conference, Ceas, Perth, Australia, September. 2011，pp135-144.

[19] Toolan F , Carthy J . Feature selection for Spam and Phishing detection[C]，IEEE Ecrime Researchers Summit. 2011，pp1-12

[20] Abu-Nimeh S , Nappa D , Wang X , et al. A comparison of machine learning techniques for phishing detection [C] , Anti-phishing Working Groups Ecrime Researchers Summit. ACM，2007，pp. 60-69

[21] Gansterer W N , PölzDavid. E-Mail Classification for Phishing Defense[C]. European Conference on Ir Research on Advances in Information Retrieval. Springer-Verlag，2009，pp449-460.

[22] Khonji M , Iraqi Y , Jones A . Enhancing Phishing E-Mail Classifiers: A Lexical URL Analysis Approach [J]. International Journal for Information Security Research (IJISR)，2012，2(1)，pp236-245.

[23] Gomez J C , Boiy E , Moens M F . Highly discriminative statistical features for email classification [J]. Knowledge & Information Systems，2012，31(1)，pp23-53.

[24] Soofi E S . Principal Information Theoretic Approaches[J]. Journal of the American Statistical Association，2000，pp1349-1353.

[25] Rong F , Dazhi M , Dashun X , et al. Survey of Research Process on Statistical Correlation Analysis[J]. Mathematical Modeling and Its Applications，2014，3(1)，pp1-12.

[26] Kendall, M. G. Rank Correlation Methods [J]. British Journal of Psychology，1990，25(1):86－91.

[27] Chou, Chen Huei, SinhaA. P. , and ZhaoH. . A Hybrid Attribute Selection Approach for Text Classification [J]. Journal of the Association for Information Systems，2010，11(9)，pp491-519.

[28] Székely, Gábor J, Rizzo M L , Bakirov N K . Measuring and testing dependence by correlation of distances[J]. Annals of Stats，2007，35(6):2769-2794.

[29] Jingsi Zhang. High dimensional feature screening and model selection in time series[D] Shandong University，2013

[30] Reshef D N, Reshef Y A, Finucane H K, ale t. Detecting novel associations in large datasets[J]. Science. 2011，334，pp 1518-1524.

[31] Toolan, Fergus, Carthy, Joe. Phishing detection using classifier ensembles[J]. Ecrime Researchers Summit，2009(18)，pp1-9

[32] ReshefDavid, ReshefYakir, Pardis Sabeti and Michael Mitzenmacher. MINE:. Maximal Information-based Nonparametric Exploration. [OL].

[33] Chang, ChungChih, and LinC. J.. LIBSVM: A library for support vector machines [J]. Acm Transactions on Intelligent Systems & Technology，2011，2(3)，pp389-396.

About the authors

Xiujuan Wang received her PhD degree of information and signal processing from Beijing University of Posts and Telecommunications in July 2006. She is currently an associate professor at Faculty of Information Technology， Beijing University of Technology. Her research interests include machine learning， information security.（Email：xjwang@bjut. edu. cn）

Qianqian Zheng is currently pursuing the M. S. degree with the Information Technology， Institute， Beijing University of Technology， Beijing， China. Her research interests include information security， network information processing， and pattern recognition techniques.

Weijie Hong is currently working toward the B. S degree at Beijing University of Technology， Beijing， China. His interests include algorithm and machine learning.

Yuyang Zhang is currently working toward the B. S. degree at Beijing University of Technology， Beijing， China. His interests include image processing and machine learning.

Rundong Yang is currently pursuing the Ph. D. degree with the Information Security Center， School of Cyberspace Security， Beijing University of Posts and Telecommunications， Beijing， China. His research interests include information security and social engineering.

Zhe Wang is currently pursuing the Ph. D. degree with the School of Cyberspace Security， Beijing University of Posts and Telecommunications. Her main research interests include network information security and social engineering.

Research on Security Protection Mechanism of Mobile Edge Computing

ZHANG Weicheng, WEI Hongquan, LIU Shuxin, ZHANG Yiming

National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China

Key words: mobile edge computing; protocol and network security; access control; identity authentication; data security and privacy protection

Abstract. As an important starting point for 5G innovation and transformation， mobile edge computing needs to satisfy the application requirements of “low latency， large connections， high bandwidth， and localization”， while also facing the security challenges brought by maintaining the performance guarantee of mobile networks， the complexity of integrated virtualization technology， and the heterogeneity of IoT services， traditional security protection mechanisms cannot satisfy the requirements of mobile edge computing heterogeneity， resource constraints， distribution， real-time， large connections， and massive terminal data. First， the security threats faced by mobile edge computing and the existing security protection mechanisms are investigated. Second， it focuses on the development status of protocol and network security， access control， identity authentication， data security and privacy protection， and virtualization security， and proposes the next research direction. Finally， suggestions are made for the research of mobile edge computing security protection mechanisms from three aspects： the improvement of security protection mechanism， the need for customization of differentiated security， and the combination of new technologies.

1 Introduction

With the advent of the era of everything to do with the internet， various smart devices （e. g. smartphones， smart appliances， etc.）， industrial control sensors， and other terminals generate data all the time， massive amounts of data make the existing network a heavy burden， and cloud computing will not satisfy the need of terminals for intelligent， personalized services and real-time processing of massive data. Three main reasons as follow： First， the cloud computing model is difficult to deal with the problem of data flooding caused by rapid data growth ［1］. Second， the centralized ability of cloud computing has defects in responding to edge network delay requirements ［2］. And the third， the privacy protection and data security issues of cloud computing in remote transmission are prominent. To cope with the "low latency， large connection， high bandwidth， localization， and security" requirements that cloud computing cannot satisfy， mobile edge computing has entered people's field of vision.

In December 2014， the European Telecommunications Standards Institute （ETSI） created Mobile Edge Computing （MEC） Industry Specifica-

tion Group （ISG）， which defined MEC as "providing IT services and cloud computing capability at the network edge close to the user side". In 2016， ETSI expanded MEC to "Multi-access Edge Computing"， and included other networks beyond the 3GPP network into the mobile edge computing system. MEC has the characteristics of on-premises， proximity， lower latency， location awareness， and network context information ［3］. It can serve augmented reality service scenarios， assist in intensive computing， Internet of Vehicles， Internet of Things gateway services， and other scenarios ［4］. The use of key technologies such as computing offloading， wireless data caching， network function virtualization （NFV）， and software defined network （SDN） will enhance the MEC's effectiveness. ETSI emphasizes that MEC is close to mobile users， aiming to reduce the waiting delay of service delivery， reduce the number of network operations， improve service distribution and transmission capabilities， and improve the final user experience. MEC has been recognized by the European 5G PPP organization ［5］ as one of the key technologies of 5G networks， it can satisfy the stringent requirements of 5G in terms of integrity， scalability， delay， and automation. Meanwhile， MEC promotes the transformation of the mobile broadband network to programmable and represents the architectural concept and key technology of 5G evolution ［6］.

In the era of everything to do with the internet， MEC， as a new technology， can effectively alleviate the problems of limited data transmission speed and computing offloading in cloud computing ［7］. While， the expansion of cloud computing capabilities to the network edge is also facing new risks and challenges， mobile edge computing security protection must also keep pace with the times and adapt to new security development needs：

1） Massive characteristics. The security protection mechanism of mobile edge computing needs to deal with the fast and secure access of massive terminals， the secure and real-time processing of massive data， and automated and intelligent security protection at the edge.

2） The limited resources on the edge. Mobile edge computing security protection requires lightweight authentication protocols， intrusion detection technology， data encryption technology， privacy protection technology， and security collaboration with the cloud.

3） Ultra-low latency and real-time processing. The terminal needs ultra-low latency and lightweight communication protocol to access， the real-time data processing on the edge side must have a fault-tolerant mechanism， the edge side needs a lightweight data integrity check mechanism and an efficient backup redundancy mechanism. Besides， it also needs to ensure the robustness of the entire edge and have a dynamic trusted recovery mechanism.

4） Distributed and heterogeneous. Mobile edge computing security protection requires unified security management and scheduling strategy for heterogeneous and distributed edge assets （e.g. equipment， network， computing resources）， such as unified application programming interface management， unified access authentication， etc.

The remaining paper is organized as follows. Section 2 summarizes the security threats and challenges that MEC faced， and briefly introduces security threats such as a denial of service attack， intelligent attack， and edge-cloud collaboration. Section 3 gives to research on the MEC security protection mechanism by international standards organizations， operators， and researchers， focusing on the analysis of the development trend of five security protection mechanisms， and points out the next research direction based on the characteristics of MEC. Section 4 is the suggestions proposed for the future research of the MEC security protection mechanism from three aspects： improving security protection mechanism， combining with new technology， and differentiated security needs customization. Section 5 concludes the paper.

2 Security threats and challenges

The mobile edge computing technology introduction technical white paper [3] pointed out that the introduction of IT applications into the telecommunications field has brought more security challenges for MEC, and the expansion of cloud computing resources to the edge of the network also brings new risks. Figure 1 shown the risks and challenges faced by edge computing from multiple perspectives and for different application scenarios [8-16]. This section mainly introduces six common security threats, namely denial of service attacks, privacy disclosure and data security, rogue facilities, intelligent attacks, physical damage threats, malicious administrators, and edge-cloud collaborative threats.

1) Denial of Service attack

In the MEC-based large-scale MTC terminal connection scenario, the MEC servers are responsible for the data aggregation, calculation, storage, and other tasks of a large number of devices, and these devices are often difficult to handle due to their energy consumption and computing capabilities. Attackers can manipulate these devices to launch DOS attacks on MEC servers. Meanwhile, DOS attacks are also prone to emerge in MEC virtual resource management, attackers can request a large amount of computing, storage, network services, and other resources from the MEC host by controlling a malicious virtual machine (VM), causing normal users to be unable to normally request MEC host resources.

2) Privacy disclosure and data security

Professor Shi's team [7] pointed out that the data privacy protection faced by edge computing has the following challenges. First, the awareness of privacy protection is weak, the second is the lack of tools for data privacy and security, the third is that edge devices are both data generators and data collectors. Compared with cloud computing data centers, MEC data centers have relatively few desensitizations or encryption measures, and user information stored in edge data centers can be easily sniffed by hackers. For example, in a large shopping mall, it is extremely easy for an attacker to obtain user-related information by eavesdropping on the edge data center deployed in the mall, because the edge data center understands the context, the attacker can use technical means to analyze the information fragments to obtain more sensitive information related to the user.

MEC scenarios support massive and heterogeneous mobile devices or sensors to quickly connect to the MEC server. Communication encryption and authentication between mobile devices and the MEC server are facing huge challenges. Attackers can use insecure communication protocols to tamper or eavesdrop the information. In addition, mobile edge computing also faces security threats in terms of data disaster tolerance and security APIs. Subject to the disaster tolerance capability of the edge server itself, if user data is not backed up in the cloud, and the edge server does not have an effective data recovery mechanism, once user data loss or damage, the user will only accept data loss. MEC provides multiple service APIs, such as fixed access information API [17], UE identity API [18], location API [19], radio network information API [20], etc. In the MEC scenario, the MEC server needs to provide APIs to a large number of terminals, how to deal with the management of a large number of APIs and how to ensure user experience while protecting API security will be more complicated. For example, in a virtualized infrastructure, some APIs provide logical environments and physical information, the attacker can obtain the environmental sensitive information and execution environment of the mobile edge data center through API.

3) Rogue facilities

In the MEC scenario, there may be security threats to rogue data centers. In some specific scenarios, data exchanges between data centers are frequent, and attackers can monitor, intercept, and tamper with the information flow passing through the rogue data center. On the Internet of Vehicles scenar-

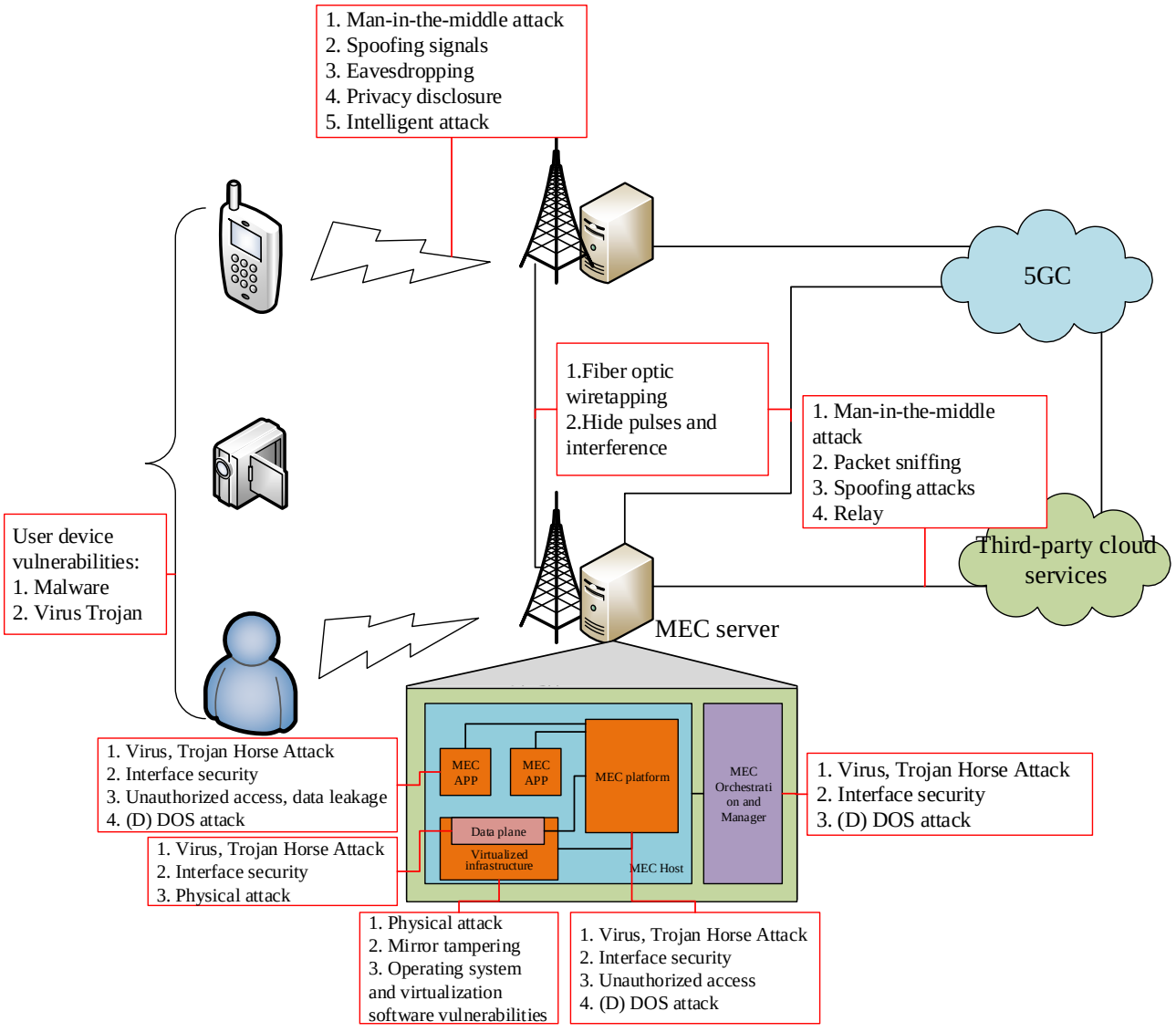


Fig. 1 Security threats faced by MEC

io， if a malicious data center transmits incorrect data information to a high-speed vehicle， it will cause serious consequences. In addition， because end-users often move and access between different edge nodes， attackers may pretend to be legitimate edge nodes or invade edge nodes with weak security protection mechanisms， and illegally obtain user account authentication information including service facilities and a large amount of behavior information. The attacker may obtain hidden user privacy from a large amount of behavior information through big data mining and deep learning.

4) Intelligent attack

Compared with traditional attacks， smart attacks are more harmful to MEC， attackers are no longer blind or opportunistic to attack but strive to strike at one hit. Attackers can use big data mining and deep learning techniques to investigate the weaknesses of the MEC system， for example， an attacker can use programmable smart radio equipment to investigate the MEC network status and find edge nodes with insufficient security protection to implement monitoring control. Xiao et al. ［21］ pointed out that the attacker can intelligently choose the attack mode according to the distance from the edge node.

5) Physical damage threats and malicious ad-

ministrators

Similar to cloud computing， MEC also faces physical and man-made threats. Generally， the MEC server is deployed close to the user， and it is not ruled out that malicious users may physically damage it. Meanwhile， attackers can also illegally access the MEC server to obtain sensitive data information stored in the server. In recent years， incidents of internal staff stealing and leaking user information have occurred frequently. In the MEC scenario， the trust of administrators will become more complicated. Data is processed at the edge， which means that a large amount of data related to users may be passively or actively leaked by the administrator. In addition， an attacker can invade the administrator account to illegally obtain the administrator's authority and control the entire edge node. A malicious administrator with super authority will pose a serious threat to the entire MEC system.

6) Edge-cloud collaborative threat

Only when MEC and cloud computing are complementary and synergistic can they better satisfy various types of application scenarios. Taking the Internet of Things as an example， data preprocessing is performed on the edge side， high-value data is transmitted to the cloud， big data analysis， algorithm model training upgrades are performed in the cloud， and the updated algorithm is pushed to the edge side in time to form a closed learning loop. The Collaboration between edge and cloud is also facing the challenge of security threats， how to ensure that data is safely uploaded from the edge to the cloud， and from the cloud to the edge， how does the cloud ensure secure access on the edge and how to prevent attacks from the edge. How to collaborate between heterogeneous edge and cloud is also a security point that needs to pay attention.

3 Security protection mechanism

Extending the function of cloud computing to the edge side will inevitably lead to the emergence of new security issues， such as collaboration issues between heterogeneous edge data centers， rapid access， and security authentication issues for a large number of mobile terminals anytime and anywhere. In addition， mobile edge computing also faces unique security threats in specific application scenarios. Such as the unique problems of data storage and processing on the Internet of Vehicles， it has been estimated that a self-driving car will generate at least 30TB of data every day［24］， massive data need to be processed locally， how to ensure the security of these data will be a huge challenge to the Internet of Vehicles scenario. A summary of this research on mobile edge computing security protection mechanisms and countermeasures by ETSI， edge computing industry alliance， mobile operators， and related researchers can be found in table 1.

3.1 Protocol and network security

MEC allows access to the network in a variety of forms. Due to the complex topology at the edge of the network， it is difficult to protect security. In order to ensure the high reliability and credibility of the edge network， an in-depth security protection mechanism should be established to ensure the security of the edge network in terms of secure communication protocols， isolation of heterogeneous network domains， intrusion detection， and flow control.

Li et al.［25］ proposed a transparent security design named MECsec to protect the MEC-integrated cellular network. It protects the MEC platform and its application servers against malicious cellular users while keeping the MEC service low latency. Naik et al.［26］ designed an intrusion detection method based on the weighted average of the functional link neural network responsible for the Internet of Things scenario， which was found to be more efficient in terms of the computational burden. Based on the idea of dynamic intrusion detection， Li et al.［27］ utilized the game theory in the field of edge computing network and suggested a data-driven mimicry intrusion detection game model-based technique called GLIDE. Hu et al.［28］ proposed a cluster-based protocol based on mobile edge computing framework

Table 1 Summary of security protection mechanisms and countermeasures

Source	For angle	Security protection mechanisms and countermeasures
[3]	Third party applications bring security risks	Trust management, authentication and authorization, virtualization technology, a combination of various isolation technologies to build a trusted third-party application
	Application isolation	Traffic isolation, virtual machine isolation, access control, integrity protection, security sandbox
	MEC platform security	Establish a reliable computing platform and fault tolerance mechanism with resilience to multiple attack media (including physical attack)
	Network interface	Mutual authentication, integrity and confidentiality inspection, security mechanism has been widely used in IT field for protection
[8]	Network infrastructure, Edge data center, Core infrastructures, Virtualization infrastructure, User devices	Identity and authentication, access control systems, protocol and network security, trust management, intrusion detection systems, privacy, virtualization, fault tolerance and resilience, forensics
[12]	Physical security	Physical environment security, safety management methods
	Platform security	Virtualization software security, virtual machine(container) security, data transmission security, UPF security protection, tenant and application isolation
	Application Security	Safety assessment, inspection and audit of third party applications
	Ability opening and security	Open capability hierarchical management, security audit, security capability virtualization and multi tenancy
[13]	Physical infrastructure security	Trusted computing, physical I/O trusted access, physical environment security
	Virtualized infrastructure security	Host OS security, Guest OS security, virtualization software security reinforcement, virtual network isolation, container security, data security
	MEC platform security	Security reinforcement, API security, sensitive data security protection, interface security, (D)DoS protection
	ME app security	APP life cycle security, user access control, security reinforcement, (D)DoS protection, sensitive data security protection
	Data plane gateway security	Security reinforcement, interface security, sensitive data security protection, physical contact attack protection
	MEC orchestration and management security	Interface security, API call security, data security, orchestration and management network element security reinforcement
	Management security	Password security, log security
	Network security	Security isolation between management, business and data planes, security isolation between security domains
[14]	Edge infrastructure security	Identity identification, virtualization security, OS security, and access authentication of edge nodes
	Edge network security	Security protocol, network domain isolation, network monitoring, network protection
	Edge data security	Data security, sensitive processing
	Edge application security	App reinforcement, authority and access control, application monitoring and application audit
[22]	Node security	ECN security, lightweight trusted computing, software hardening and security configuration, hardware safety
	Network security	Existing transport protocol security reuse, firewall, IPS / IDS, anti DDoS, VPN / TLS
	Data security	Data isolation (destruction), data tamper proof, privacy protection (desensitization), encryption (transmission, storage), data access control, data leakage prevention
	Application Security	Whitelist, application security audit, malware prevention, security detection and response, software reinforcement and patch, security configuration management, sandbox
	Security situation awareness	Big data security analysis (perception), advanced threat detection, correlation analysis, threat presentation and traceability, security compliance audit
	Arrangement of safety management	The whole network active protection management, application scheduling security, security service life cycle management, resource pool management, unified security policy and arrangement
	Safety operation and maintenance system	Operation and maintenance monitoring, emergency response
	Identity and authentication management	Decentralized and distributed authentication and certificate management
[23]	Vulnerabilities in user equipment (UE)	Intrusion detection system (IDS) capable of operating in a low-resource environment
	Attacks on communication channels within the access network	Physical unclonable functions (PUF), light-weight security protocols
	Vulnerabilities in the edge network	Trusted platform manager (TPM) and virtual machine introspection (VMI)
	Attacks on communication links among edge and core entities	Encrypted communication approaches as virtual private networks (VPNs)
	Vulnerabilities in MEC control elements	TPM, VMI
	5G core network entities	Auto-configurable security mechanism
	The Internet connectivity	Firewall and access control policies

for content distribution in vehicle networks， by integrating the Sub-6 GHz frequency band， IEEE 802.11p， and millimeter wave communication.

MEC allows access to 3GPP networks and networks other than 3GPP， and must involve multiple communication protocols， mature security protocols can be used in the MEC architecture to satisfy security issues such as data secure transmission and authentication.

3.2 Access control

The user or equipment connected to the MEC cannot be effectively managed， which may easily lead to the abuse of MEC resources. MEC includes characteristics such as distribution， heterogeneity， low latency， and high mobility， it requires that the MEC system should have a distributed and decentralized multi-domain access control strategy， and support dynamic authorization and rapid authentication. The access control strategy must take into account the issues of security and energy consumption， and the lightweight minimum authorization security model （e.g. whitelist technology， etc.） is favored by people.

Zhang et al. ［29］ gave a detailed introduction to edge computing access control from two aspects： role-based access control and attribute-based access control. Khalid et al. ［30］ provided a comprehensive and detailed classification and comparison introduction to fog computing access control schemes， which could bring inspiration to MEC access control solutions. Xu et al. ［31］ studied the problem of dynamic MEC access control with the help of energy

harvesting technology and proposed an intelligent and efficient MEC access control management algorithm based on long and short-term memory （LSTM） for IoT devices. Aiming at the IoT scenario， Wu et al. ［32］ combined SDN technology with deep learning technology to propose a flexible and secure software-defined edge computing system for flexible access management of IoT. Aiming at the industrial Internet of Things scenario， Ren et al. ［33］ designed blockchain-based identity management and access control strategy based on edge computing， which has the characteristics of irreversible modification and traceability. Yang et al. ［34］ pointed out that integrating blockchain and edge computing into one system can achieve reliable access and control to the network.

The mobile edge computing access control scheme must keep the unity between security level and performance. While considering the security level， it is necessary to take into account the characteristics of limited resources of terminal equipment and the requirements of energy consumption. Besides， access control solutions for fog computing and cloud computing can bring inspiration to mobile edge computing access control solutions， and deep learning， blockchain， and other technologies can also bring new ideas to mobile edge computing access control.

3.3 Identity authentication

Identity authentication is the process of verifying the legal identity of a user or device. In the MEC scenario， there are a large number of mobile devices that need to access the edge network to obtain the authorization. Meanwhile， the edge server and cloud data center also need to interact frequently， and they cannot do without an effective identity authentication mechanism. In addition， in certain specific scenarios， it is also necessary to solve the challenge of handover authentication in which highly mobile users frequently switch between multiple MEC servers.

Xiao et al. ［35］ proposed a lightweight authentication scheme based on reinforcement learning to cope with MEC systems with deception models and time-varying radio channel models， in this scheme， key authentication parameters are obtained through reinforcement learning technology. Wang et al. ［36］ proposed a handover authentication scheme （SHAS） for MEC in an SDN-based cyber-physical system. The target wireless access point and edge computing node perform a three-way handshake protocol through a session key to achieve mutual authentication and key confidentiality. Lu et al. ［37］ proposed a PHY authentication framework for vehicular ad hoc networks （VANETs） in the MEC scenario. The framework uses deep reinforcement learning to optimize the authentication strategy. the physical authentication scheme can effectively resist attacks on VANET from the rogue edge. Liao et al. ［38］ proposed a multi-user physical layer identity authentication scheme for industrial IoT scenarios by combining data enhancement methods with deep neural networks， even in the case of a small number of training samples， this solution can identify multiple terminals simultaneously at a lower cost. Jia et al. ［39］ designed a lightweight authentication protocol which can realize user non-traceability and anonymity for the MEC environment， and allow users to access multiple users with only one registration， the user's personal information is not stored on the MEC server， and the mutual authentication between the MEC server and the user only needs one round of message exchange. Chen et al. ［40］ proposed a location-based lightweight mutual identity authentication scheme to solve the problem that how to quickly achieve mutual authentication with MEC servers in different regions when patients wearing monitoring equipment in the smart hospital scene change regions.

Most terminals in the MEC scenario face limited resources （e. g. computation， storage， energy， communication and network services， etc.）， and it is difficult to carry traditional cryptographic-based identity authentication， and traditional centralized security authentication schemes will be difficult to adapt to the requirements of low latency， high bandwidth， and large connections in mobile edge com-

puting scenarios, distributed authentication methods and certificate management, as well as the combination of biometrics (e. g. fingerprint, iris scan, etc.) and physical features (e. g. radio frequency fingerprints) and deep learning authentication schemes will become new choices. In addition, the design of authentication schemes for specific application scenarios should be more diverse.

3.4 Data Security and Privacy Protection

The relatively centralized cloud computing data center facilitates unified security management, while the distributed and heterogeneous characteristics of the MEC data center increase the difficulty of unified security management of data, and the risk of information leakage. As an open ecosystem, mobile edge computing may store, analyze, and process user data in entities beyond the control of users, especially with the continuous development of big data mining technology in recent years, attackers may collect user-related data and use data mining technology to obtain user sensitive information. While MEC improves user service experience, it also faces more severe challenges in data security and privacy protection.

Zhang et al. [29] provided a systematic and detailed introduction to edge computing data security and privacy protection in terms of data privacy protection, location privacy protection, identity privacy protection, data confidentiality and secure sharing, integrity auditing, and searchable encryption. Khalid et al. [30] introduced the data privacy issues in fog computing from two categories: secure data storage and secure data sharing, and compared the existing privacy protection schemes, which can be used as a reference for mobile edge computing privacy protection schemes. Due to the limited resources on the edge side, some data needs to be outsourced to the remote cloud, and it will be difficult to control these data locally, the data integrity problem is particularly important, Tong et al. [41] proposed two data integrity check protocols, ICE-basic and ICE - batch, which can realize user data integrity check on a single edge or multiple edges. In the Internet of Things scenario, massive amounts of data will be generated at the edge, edge devices need to calculate, store, process, and analyze a large amount of data, and these data may contain pieces of user sensitive information, the attacker can analyze the user's behavior through technical means to launch a more serious attack. Aiming at the protection of pedestrian location privacy, Yuan et al. [42] proposed a scheme that applies differential privacy to protect the collected data. The data collection scheme uses a Gaussian mechanism before the data is uploaded to the edge node, sensitive information (e. g. high-resolution images, the user's identity, location, etc.) will be protected. Zhou et al. [43] designed a trajectory protection scheme based on fog computing and k-anonymity for real-time trajectory privacy protection in continuous query and offline trajectory data protection in trajectory publishing. MEC has location-aware features, location-based services (LBS) are more and more widely used, location-based privacy protection mechanism (LPPM) has attracted more and more attention, Tian et al. [44] classified LBS from the three dimensions of motion trajectory, ID information, and the type of information provided by LBS, and the existing LPPM research is summarized from the perspectives of k-anonymity, noise-added LPPM, obfuscation method, and POI model, and a new type of LBS is introduced for MEC scenarios named DLBS, it provides a new idea for the location privacy protection mechanism in MEC. Zhang et al. [45] proposed a mobility support system (MSS) enhanced with MEC to protect mobile users' network privacy without affecting performance and increasing operating costs. In reference to the problems of data security and privacy protection in edge computing, Ma et al. [46] proposed a model based on comprehensive trust to evaluate the credibility of edge nodes. Qin et al. [47] pointed out that security technologies such as zero-knowledge proofs that have made progress in the blockchain will provide new ideas for solving data

privacy security protection in edge computing scenarios.

Research on MEC data security and privacy protection does not have to start from scratch and can learn from other edge computing paradigms, and privacy protection requirements in different scenarios are also different. It will be a new idea to apply deep learning, AI technology, and security technology in blockchain to data security and privacy protection in mobile edge computing scenarios. Nawaz et al. [48] combined blockchain technology, artificial intelligence technology, and edge computing to open up a new paradigm in applications with strict privacy requirements and data-sensitive applications. Chen et al. [49] summarized the latest technology of deep learning at the edge of the network, which brings new inspiration for the security protection of mobile edge computing.

3.5 Virtualization Security

Network function virtualization and software defined network, as key technologies for mobile edge computing, enable MEC to cope with the needs of diverse service scenarios. NFV deployed in virtualized infrastructure blurs the boundaries of MEC's security protection and brings new security threats to MEC. Virtualization security refers to the realization of security enhancement and virtualization isolation for edge devices based on virtualization technology. In the MEC scenario, virtualization security faces some challenges such as virtual network isolation, container security, data security, virtualization software security reinforcement, Host OS security, Guest OS security, and VM migration, in addition, network slicing security based on virtualization technology is also an issue that needs to be considered in MEC security protection.

Mobile edge computing has relatively high requirements for service mobility, and services need to move with users. Wendland et al. [50] pointed out that the use of TOSCA service specification language to write security policies to enhance the NFV service orchestration function is helpful to satisfy the requirements of dynamic, flexible, and self-configuring service architecture in the 5G evolution process. Tao et al. [51] introduced three types of virtualization isolation: user data isolation, operating system isolation, resource isolation, and four security technology countermeasures against virtualization security threats: identity and authentication mechanisms, intrusion detection systems (IDS), trust management, fault tolerance mechanism. Tiburski et al. [52] proposed a security architecture based on trust mechanism and embedded virtualization, which is suitable for IoT edge devices. It can ensure the security of applications running on these devices, meanwhile, there are also good results in terms of communication delay between VMs.

Trusted platform manager (TPM) and virtual machine introspection (VMI) can be used in virtualization security protection. Lal et al. [53] pointed out that TPM can be used to store and verify system-sensitive components, VMI can detect malicious VMs. Mishra et al. [54] proposed a monitoring method based on VMI to detect malware in VM. In addition, the safe use of virtualized resources must also consider the hypervisor's own security protection, virtual machine failover mechanism, and other issues. Research on 5G core network endogenous security technology [55] can bring inspiration to mobile edge computing virtualization security.

4 Suggestions

Several key security protection mechanisms of MEC have been discussed above, summary analysis based on existing discussions, and the following suggestions are made for the research of MEC security protection mechanisms:

1) Improve the security protection mechanism of other edge computing paradigms. The research on the MEC security protection mechanism does not need to start from scratch, can learn from the research results of other edge computing paradigms (e.g. fog computing, mobile cloud computing, etc.), and pay attention to combining the character-

istics of MEC heterogeneity， mobility， location awareness， and distribution in the improvement process.

2） Combine new technologies for security protection. Combining AI， blockchain ［56］， endogenous security， federated learning ［57］， migration learning ［58］， and other technologies with MEC security protection will provide more possibilities for MEC security protection. For example， blockchain technology and physical unclonable functions can be applied to client authentication， and an edge-level AI-based IDS system can be developed.

3） Customize differentiated security needs. MEC security protection should unify security and efficiency for different application scenarios， different application scenarios have different requirements for security， differentiated security protection mechanisms should be customized according to different business needs. For example， in the mMTC scenario， a large number of terminal sensors may require a lightweight security mechanism to satisfy the device's own low energy consumption， low cost， and resource constrained requirements. However autonomous driving in the uRLLC scenario requires an efficient mobility security protection mechanism to cope with the low latency requirements in high-speed mobile scenarios.

5 Conclusion

The arrival of the new 5G infrastructure wave has accelerated the development of MEC， the security threats and protection issues faced by the development of MEC will be closely related to people's work and life， research on the MEC security protection mechanism must not only combine the characteristics of MEC but also take into account the security needs of different business scenarios. The paper briefly introduces the background of MEC， the threats and challenges that MEC faces， and the existing protection mechanisms， focusing on the development trend of key protection mechanisms such as protocol and network security， access control， identity authentication， data security and privacy protection， and virtualization security， and suggestions for the study of MEC security protection mechanisms are proposed from three aspects： improving the security protection mechanism， combining with new technology， and customizing different security needs.

On the whole， the research on MEC security protection does not have to start from scratch， it can learn from the related research of other edge computing paradigms， the combination of machine learning， endogenous security， blockchain technology， and MEC will give birth to more possibilities for mobile edge computing security protection.

References:

[1] BITTENCOURT L F, LOPES M M, PETRI I, et al. Towards Virtual Machine Migration in Fog Computing［C］// 2015 10th International Conference on P2P，Parallel，Grid，Cloud and Internet Computing（3PGCIC）. 2015：1-8.

[2] SHI W, DUSTDAR S. The Promise of Edge Computing ［J］. Computer，2016，49（5）：78-81.

[3] HU Y, PATE M, JOUBERT J，Mobile-Edge Computing — Introductory Technical White Paper［R］：Mobile-Edge Computing（MEC）Industry Initiative，2014.

[4] ETSI GS MEC-IEG 004, Mobile-Edge Computing (MEC); Service Scenarios [S],2015. 11.

[5] 5G Vision: The 5G Infrastructure Public Private Partnership: the next generation of communication networks and services［R］：The 5G Infrastructure Public Private Partnership，2015.

[6] YUN C H, PATEL M, SABELLA D, et al. ，Mobile Edge Computing A key technology towards 5G［R］. Sophia Antipolis，France：ETSI GS MEC-IEG，2015.

[7] SHI W, CAO J, ZHANG Q, et al. Edge Computing: Vision and Challenges［J］. IEEE Internet of Things Journal，2016，3（5）：637-646.

[8] ROMAN R, LOPEZ J, MAMBO M. Mobile edge computing, Fog et al. : A survey and analysis of security threats and challenges［J］. Future Generation Computer Systems，2018，78：680-698.

[9] LI H, SHOU G, HU Y, et al. Mobile Edge Computing: Progress and Challenges［C］// 2016 4th Ieee International Conference on Mobile Cloud Computing，Services，and Engineering. 2016：83-84.

[10] ABBAS N, ZHANG Y, TAHERKORDI A, et al. Mobile Edge Computing: A Survey［J］. IEEE Internet of Things Journal，2018，5（1）：450-465.

[11] YI S, QIN Z, LI Q. Security and Privacy Issues of Fog Computing: A Survey［C］// wireless algorithms systems and applications. 2015：685-695.

[12] China Mobile Edge Computing Technology System White Paper

[R]：China Mobile Edge Computing Open Lab.

[13] ZHUANG X J, YANG B, WANG X, et al. Approach on mobile edge computing security [J]. Telecom Engineering Technics and Standardization，2018，(12)：38-43.

[14] Edge Computing Consortium (ECC), Alliance of Industrial Internet (AII). Edge Computing Security White Paper [R]. 2019.

[15] MA L C, PEI Q Q, XIAO H Z. Security Requirements and Challenges in Edge Computing for Internet of Everything [J]. ZTE Technology Journal，2019，25(03)：37-42.

[16] HE D, CHAN S, GUIZANI M. Security in the Internet of Things Supported by Mobile Edge Computing [J]. IEEE Communications Magazine，2018，56(8)：56-61.

[17] ETSI GS MEC 029, Multi-access Edge Computing (MEC); Fixed Access Information API [S], 2019.

[18] ETSI GS MEC 014, Mobile Edge Computing (MEC); UE Identity API [S], 2018.

[19] ETSI GS MEC 013, Multi-access Edge Computing (MEC); Location API [S], 2019.

[20] ETSI GS MEC 012, Multi-access Edge Computing (MEC); Radio Network Information API [S], 2019.

[21] XIAO L, XIE C, CHEN T, et al. A Mobile Offloading Game Against Smart Attacks [J]. IEEE Access，2016，4：2281-2291.

[22] Edge Computing Consortium (ECC), Alliance of Industrial Internet (AII). Edge computing reference architectures 3. 0 [R]，2018.

[23] RANAWEERA P, JURCUT A D, LIYANAGE M. Realizing Multi-Access Edge Computing Feasibility: Security Perspective [C]// 2019 IEEE Conference on Standards for Communications and Networking (CSCN)，28-30 Oct. 2019. 2019：1-7.

[24] RIJMENAM M V. Self-driving Cars Will Create 2 Petabytes of Data，What Are The Big Data Opportunities For The Car Industry? [EB/OL]. https://datafloq. com/read/self-driving-cars-create-2-petabytes-data-annually/172#
！，2013.

[25] LI C Y, LIN Y D, LAI Y C, et al. Transparent AAA Security Design for Low-Latency MEC-Integrated Cellular Networks [J]. IEEE Transactions on Vehicular Technology，2020，69(3)：3231-3243.

[26] NAIK B, OBAIDAT M S, NAYAK J, et al. Intelligent Secure Ecosystem Based on Metaheuristic and Functional Link Neural Network for Edge of Things [J]. IEEE Transactions on Industrial Informatics，2020，16(3)：1947-1956.

[27] LI Q, HOU J, MENG S, et al. GLIDE: A Game Theory and Data-Driven Mimicking Linkage Intrusion Detection for Edge Computing Networks [J]. Complexity，2020，2020：1-18.

[28] HU Q, WU C, ZHAO X, et al. Vehicular Multi-access Edge Computing with licensed Sub-6 GHz, IEEE 802. 11p and mmWave [J]. IEEE Access，2018，6：1995-2004.

[29] ZHANG, J L, ZHAO Y C, CHEN B, et al. Survey on data security and privacy-preserving for the research of edge computing [J]. Journal on Communications，2018，39(03)：1-21.

[30] KHALID T, ABBASI M A K, ZURAIZ M, et al. A survey on privacy and access control schemes in fog computing [J]. International Journal of Communication Systems，2019.

[31] XU L, QIN M, YANG Q, et al. Deep Reinforcement Learning for Dynamic Access Control with Battery Prediction for Mobile-Edge Computing in Green IoT Networks [C]// 2019 11th International Conference on Wireless Communications and Signal Processing. 2019：1-6.

[32] WU D, HUANG X, XIE X, et al. LEDGE: Leveraging Edge Computing for Resilient Access Management of Mobile IoT [J]. IEEE Transactions on Mobile Computing，2019：1-1.

[33] REN Y, ZHU F, QI J, et al. Identity Management and Access Control Based on Blockchain under Edge Computing for the Industrial Internet of Things [J]. Applied Sciences，2019，9(10)：2058-2073.

[34] YANG R, YU F R, SI P, et al. Integrated Blockchain and Edge Computing Systems: A Survey, Some Research Issues and Challenges [J]. IEEE Communications Surveys & Tutorials，2019，21(2)：1508-1532.

[35] XIAO L, WAN X, DAI C, et al. Security in Mobile Edge Caching with Reinforcement Learning [J]. IEEE Wireless Communications，2018，25(3)：116-122.

[36] WANG C, ZHANG Y, CHEN X, et al. SDN-Based Handover Authentication Scheme for Mobile Edge Computing in Cyber-Physical Systems [J]. IEEE Internet of Things Journal，2019，6(5)：8692-8701.

[37] LU X, XIAO L, XU T, et al. Reinforcement Learning Based PHY Authentication for VANETs [J]. IEEE Transactions on Vehicular Technology，2020，69(3)：3068-3079.

[38] LIAO R-F, WEN H, CHEN S, et al. Multiuser Physical Layer Authentication in Internet of Things With Data Augmentation [J]. IEEE Internet of Things Journal，2020，7(3)：2077-2088.

[39] JIA X, HE D, KUMAR N, et al. A Provably Secure and Efficient Identity-Based Anonymous Authentication Scheme for Mobile Edge Computing [J]. IEEE Systems Journal，2020，14(1)：560-571.

[40] CHEN C L, CHIANG M L, HSIEH H C, et al. A Lightweight Mutual Authentication with Wearable Device in Location-Based Mobile Edge Computing [J]. Wireless Personal Communications，2020，113(1)：575-598.

[41] TONG W, JIANG B, XU F, et al. Privacy-Preserving Data Integrity Verification in Mobile Edge Computing [C]// 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). 2019：1007-1018.

[42] YUAN D, ZHU X, MAO Y, et al. Privacy-Preserving Pedestrian Detection for Smart City with Edge Computing [C]// 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)，23-25 Oct. 2019. 2019：1-6.

[43] ZHOU K, WANG J. Trajectory Protection Scheme Based on Fog Computing and K-anonymity in IoT [C]// 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)，18-20 Sept. 2019. 2019：1-6.

[44] TIAN Z, WANG Y, SUN Y, et al. Location Privacy Challenges in Mobile Edge Computing: Classification and Exploration [J]. IEEE Network，2020，34(2)：52-56.

[45] ZHANG P, DURRESI M, DURRESI A. Mobile Privacy Protection Enhanced with Multi-access Edge Computing [C]// 2018 IEEE 32nd

International Conference on Advanced Information Networking and Applications (AINA). 2018：724-731.

[46] MA X, LI X. Trust Evaluation Model in Edge Computing Based on Integrated Trust［C］// Proceedings of the 2018 International Conference on Algorithms，Computing and Artificial Intelligence - ACAI 2018. 2018：1-6.

[47] QIN Y B, HAN M, YANG Q L. Data-Driven Intelligent Application in Edge Computing: Prospects and Challenges［J］. ZTE Technology Journal，2019，25(03)：68-76.

[48] NAWAZ A, GIA T N, QUERALTA J P, et al. Edge AI and Blockchain for Privacy-Critical and Data-Sensitive Applications［C］// 2019 Twelfth International Conference on Mobile Computing and Ubiquitous Network (ICMU)，4-6 Nov. 2019. 2019：1-2.

[49] CHEN J, RAN X. Deep Learning with Edge Computing: A Review［J］. Proceedings of the IEEE，2019，107(8)：1655-1674.

[50] WENDLAND F, BANSE C. Enhancing NFV Orchestration with Security Policies［C］// Proceedings of the 13th International Conference on Availability，Reliability and Security - ARES 2018. 2018：1-6.

[51] TAO Z, XIA Q, HAO Z, et al. A Survey of Virtual Machine Management in Edge Computing［J］. Proceedings of the IEEE，2019，107(8)：1482-1499.

[52] TIBURSKI R T, MORATELLI C R, JOHANN S F, et al. Lightweight Security Architecture Based on Embedded Virtualization and Trust Mechanisms for IoT Edge Devices［J］. IEEE Communications Magazine，2019，57(2)：67-73.

[53] LAL S, TALEB T, DUTTA A. NFV: Security Threats and Best Practices［J］. IEEE Communications Magazine，2017，55（8）：211-217.

[54] MISHRA P, VERMA I, GUPTA S, et al. vProVal: Introspection based Process Validation for Detecting Malware in KVM-based Cloud Environment［C］// 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC). 2019：271-277.

[55] YOU W, LI Y L, BAI Y, et al. Research on Endogenous Safety and Security Technology of 5G Core Network［J］. Radio Communications Technology: 1-7.

[56] BHATTACHARYA P, TANWAR S, SHAH R, et al. Mobile Edge Computing-Enabled Blockchain Framework—A Survey［M］//. Proceedings of ICRIC 2019. Springer International Publishing. 2020：797-809.

[57] YANG J, DUAN Y, QIAO T, et al. Prototyping federated learning on edge computing systems［J］. Front Comput Sci，2020，14(6)：3.

[58] PAN S J, YANG Q. A Survey on Transfer Learning［J］. IEEE Transactions on Knowledge and Data Engineering，2010，22(10)：1345-1359.

About the authors

ZHANG Weicheng.［corresponding author］ was born in Datong， Shanxi. He is now a master student of the National Digital Switching System Engineering & Technological R&D Center. His main research direction is mobile edge computing security.（Email： Qwechily@ qq. com）

WEI Hongquan， from Tanghe， Henan， is an associate researcher of the National Digital Switching System Engineering & Technological R&D Center. His main research direction is the integration of network security， reconfigurable network theory and technology.（Email： whq@mail. ndsc. com. cn）

LIU Shuxin was born in Weifang， Shandong， and is an assistant researcher at the National Digital Switching System Engineering & Technological R&D Center. His main research direction is complex networks and network information mining.（Email： liushuxin11@ 126. com）

ZHANG Yiming was born in Handan， Hebei. He is now a master student of the National Digital Switching System Engineering & Technological R&D Center. His main research direction is mobile communication network security， cyberspace security.（Email： 913914944@ qq. com）

复杂网络中博弈论应用研究进展

石灏苒[1]，潘世东[2]，吉立新[1]，刘树新[1]

[1]中国人民解放军战略支援部队信息工程大学 河南省 郑州市 450001；

[2]江南计算技术研究所 江苏省 无锡市 214000

摘　要：博弈论作为研究个体间合作与竞争行为的理论，对于复杂网络中各主要研究领域具有重要指导作用。网络演化上，对提出的经典演化博弈模型进行了总结。链路预测上，根据对问题定义不同，将其区分为无监督与有监督的链路预测方法。社团发现上，根据社团发现研究网络对象不同，划分为基于经典博弈论以及演化博弈论的社团发现方法。综述了近年来博弈论与复杂网络相结合提出的有代表的方法模型，展望了博弈论在复杂网络中应用的前景与发展。

关键词：复杂网络、博弈论、网络演化、链路预测、社团发现

Game theory application research progress in complex networks

Shi Haoran[1], Pan Shidong[2], Ji Lixin[1], Liu Shuxin[1]

1.People's Liberation Army Strategic Support Force Information Engineering University, Zhengzhou 450001, China;

2.Jiangnan Institute of Computing Technology, Wuxi 214000, China

Abstract: Game theory, studying cooperation and competition among individuals, play an important guiding role in the main research fields of complex networks. In network evolution, the classic evolutionary game model proposed in prior studies is summarized. In link prediction, according to the different definitions of the problem, it can be categorized into unsupervised link prediction methods and supervised link prediction methods. In community detection, taking account of the different network objects of community detection research, it is divided into community detection methods based on classic game theory and community detection methods based on evolutionary game theory. This paper summarizes the representative method models proposed by the combination of game theory and complex networks in recent years, and looks forward to the prospects and development of game theory in complex networks.

Key words: Complex network; Game theory; Network evolution; Link prediction; Community detection

1　引言

现实世界中存在大量的复杂系统，复杂网络作为一个多学科交叉的研究领域，将这些承载着丰富信息的系统抽象为网络形式，如社交网络[1-3]、经济网络[4]、互联网[5]、科研合作网[6-7]、蛋白质网络等[8-12]，用节点代表真实系统中不同的实体，连边代表实体间的联系，以此对这类复杂网络上的信息进行挖掘与应用[13]。从最初的随机网络，到Watts与Strogatz在1998年提出的小世界网络模型[14]，以及1999年BARABÁSI与ALBERT提出网络具有无标度特性[15]。网络结构几经演变，基于复杂网络结构的研究也在近年来逐渐成为焦点。但由于数据具有隐私性、异质性等特性，实际抽象得到的网络连边信息往往是不完整的，如何获取网络的真实结构具有重要的理论研究和实际应用价值[16]。链路预测，社团发现以及网络演化作为复杂网络的重要研究方向，基于网络结构和节点特征近年来提出了一系列模型与方法以对网络结构进行预测与动力学研究，且都取得了较为不错的结果，但如何利用节点的特征属性与节点间的连接偏好来进一步提高准确性仍是目前的难点。

基金项目：国家自然科学基金(61803384)。

博弈论作为经济学研究的重要理论，随着复杂金融系统的出现[17]，为了刻画这种具有无标度与自组织特性网络上局部交互的博弈关系，与复杂网络相结合提出了网络演化博弈这一概念，用以研究网络演化的动力学过程。网络演化博弈通过观察网络上节点交互过程来研究网络上合作涌现机制，文献[18]认为个体间具有策略偏好会促进网络合作涌现，文献[19-20]实验证明引入惩罚者策略并不能很好的提高合作水平，但集体惩罚策略却可以很好的解决这一问题，此外，文献[21]还综述了网络中节点流动性、个体影响力与年龄对于网络演化的影响。根据以上研究，文献[22]通过为网络中节点设置收益函数与策略，在P2P网络中抑制了“搭便车”的现象，文献[23]把高分辨率像素看作图像网络中的博弈者，以此将高分辨率像素的估计问题变成了在进化博弈中寻找进化稳定策略的问题，文献[24]实现了VoIP流量的拥塞控制。

大部分基于网络演化提出的模型，如Co-authorship model与Influence game等[25]都基于外部激励来控制网络节点的行为，侧重于通过构造效用函数得到目标网络结构或通过效用函数来解释网络形成的原因，以此达到对网络动力学过程研究的目的。但不能直接用于复杂网络的链路预测与社团发现。如何将博弈论与复杂网络研究相结合，在对网络形成与演化合理解释的基础上，有效地利用节点行为之间的相互影响[26]，找到合适的节点属性特征来进行链路预测与社团发现，以对网络未来结构进行预测，发现网络中存在的潜在特征，近年来成为一个新的研究热点，并为复杂网络研究的理论基础完善起到了推动和借鉴作用。目前将博弈论应用于复杂网络研究的方法主要通过将网络演化博弈模型与传统方法相结合或是仅依据节点效用来进行连边与社团挖掘来实现，本文对于这些方法进行了介绍与总结，并对博弈论的应用前景进行了展望。

2 相关背景

自然界和人类社会中普遍存在着合作与竞争行为，而描述合作与竞争行为的理论则是博弈论。博弈论又叫对策论或赛局理论，是研究多个自主性个体在利益相关情形下的决策行为的理论[27]，在经典博弈模型中，假设每个个体都是理性的，即每个个体都掌握自己和对手的策略选择，根据博弈中个体间合作和竞争的利益关系，可将经典博弈分为合作博弈与非合作博弈。

2.1 非合作博弈

在群体博弈过程中，个体之间常常存在竞争关系，当一方获得利益时，通常会损害另一方的利益，这种博弈过程称作非合作博弈。在非合作博弈模型中，一个博弈模型可用$\Gamma=(V,\{\boldsymbol{S}_i \mid v_i\in V\},\{\boldsymbol{U}_i \mid v_i\in V\})$表示，其由3个基本要素组成：决策个体集合$\boldsymbol{V}$，每个决策者所能采取的策略集合S，以及每个决策者的收益函数$\boldsymbol{U}$[28]。个体根据收益函数做出最优决策从而得到个体最大收益，当群体中个体不能通过改变自身策略来获得更大收益时，该群体达到纳什均衡，即博弈的终止条件。但在非合作博弈中，可能存在多个纳什均衡，当每个个体做出了对自身最优的决策后达到某一纳什均衡，却往往在整体看来收益不是最优的，这一现象称为社会困境问题，由此出现了较为经典的博弈模型如囚徒困境博弈，雪堆博弈，猎鹿博弈，公共物品博弈等，如何摆脱这类社会困境，提升每个个体收益的同时整体也达到最优纳什均衡是非合作博弈研究的重点。

2.2 合作博弈

非合作博弈强调个体理性，是为了最大化个体收益。而合作博弈研究个体之间的合作关系，每个个体加入合作的收益高于个体的收益，因此强调集体理性，是为了最大化集体收益。与非合作博弈以达到纳什均衡为博弈终止条件不同，合作博弈中以个体集合为单位来研究博弈的终止条件。可以使用（N，v）表示一个合作博弈，其中N=｛1，2，3，…，n｝表示网络上的个体集合，N中每一个元素s代表一个联盟，v是这个合作博弈的特征函数，v（s）表示了这个联盟的效用。将满足帕累托标准的个体组合结构称为合作博弈的一个有效解，所有有效解的集合称为该博弈的解集。当个体相互合作形成的联盟结构能使得所有参与者都不能从联盟重组中获益，此时称这一有效解为合作博弈的核（core），即合作博弈的终止条件。但由于个体的复杂性，往往合作产生的收益是非线性的，衡量个体在合作中的实际价值是合作博弈的核心问题，目前较为经典的方法如Shapley值，Banzhaf权力指数等半值方法取得了较

好的效果[29]。但如何更好的衡量个体合作获得的收益仍是合作博弈研究的重点。

3 基于博弈论的网络演化模型

传统的网络演化模型通常基于WS小世界模型与BA模型对网络连边按一定比例重连以拟合真实网络结构，文献[30]将BA网络与ER网络相结合，生成了一种包含两种模型特性的演化网络以拟合真实雾计算网络。除此之外通过改变拓扑结构缩小网络平均路径并提高抗毁性也是网络演化的主要应用之一[31-33]，文献[34]利用节点异质性与空间结构相结合构建了具有较强通信能力的战术通信网络。

但目前网络演化研究仍主要基于网络拓扑性质以构建网络或拟合真实网络，较少考虑节点间行为对于网络演化的影响。网络演化博弈将博弈论与复杂网络结合，形成了一种刻画群体决策形成和演化的基本范式，摒弃了经典博弈中的绝对理性与完全信息的假设，从系统的动态角度考察个体决策到群体决策的机制[35]。博弈与网络结构具有密切关系，在具有噪声的网络环境中，每个个体收益仅仅与邻近个体的策略有关。文献[36]利用网络演化博弈思想，假设节点连边产生的效用与成本，并以此作为网络演化策略对网络结构进行优化，在缩小网络平均路径的同时拟合了真实网络的聚集系数。由于考虑了节点性质以及节点间行为对网络演化的影响，博弈论在网络演化中的应用对于网络优化以及拟合真实网络具有重大意义，以下对近年来的经典网络演化博弈模型进行介绍。

3.1 合著者关系模型

合著者关系模型（Co-authorship Model）利用合著者网络特性来描述网络演化博弈关系，将网络上节点看作研究者，节点间的连边则代表共同合作的项目，每个研究者都只拥有有限的精力，当一个研究者拥有n条连边时，在每一个项目上花费的精力则为1/n，因此可以用如下形式进行表示：

$$U_u(g)=\sum_{j:uj\in g}[\frac{1}{n_u}+\frac{1}{n_j}+\frac{1}{n_u n_j}]=1+(1+\frac{1}{n_u})\sum_{j:uj\in g}\frac{1}{n_j} \tag{1}$$

该模型虽然只说明了节点间连边带来的收益，没有表示维护连边带来的成本，但随着节点连边增多，每条边上分配的资源也相应减少，这也间接表示了增加连边的成本[25]。

3.2 线性最佳响应模型

线性最佳响应模型（Linear Best-Response Model）[37]通过构建向量U表示节点在网络上付出的成本，其中$\bar{u}$代表每个节点需要单独付出的成本，对于每个节点i，其成本可如下形式表示：

$$U_i(G)=\begin{cases}\bar{u}-\delta\sum_{e_{ij}\in E(G)}U_j(G) & \text{if } \bar{u}>\delta\sum_{e_{ij}\in E(G)}U_j(G)\\ 0 & \text{if } \bar{u}\le\delta\sum_{e_{ij}\in E(G)}U_j(G)\end{cases} \tag{2}$$

其中δ用来衡量节点i的邻居对它的努力的影响，当节点i的邻居付出了一部分成本后，节点i本身可以因此少付出一部分成本，因此$\bar{u}$也可表示为节点i自身付出成本与邻居付出成本联合，节点邻居付出的成本越多，则该节点付出成本越少，甚至可以不用付出成本，因此通过这种方式表示了节点维持连边的成本与收益，体现了网络演化博弈的局部交互以及博弈的思想。

3.3 影响力博弈

影响力博弈（Influence game）是在合作博弈基础上提出的一种网络演化博弈模型，其根据子集在网络中具有的影响范围的大小，用特征函数按比例评估每个节点子集。对于一个网络G=（V，E），影响力博弈通过（G，v_G）表示，子集C⊇V，则其影响力范围用影响力函数v_G表示为v_G=SF（C）。进一步可以变为k步影响力函数：

$$V_k(S)=|\{v\in V\mid d(S,v)<k\}\setminus S| \tag{3}$$

其中d（u，v）表示节点u与v之间的距离，S为节点集合，d（S，v）=$\min_{u\in S}$ d（u，v）通过个体之间一定范围之内的相互影响来体现了个体之间的相互合作与组合结构，体现了合作博弈的思想。

4 基于博弈论的链路预测方法

由于网络数据具有隐私性与天然缺失的特点，数据采集所获得的网络结构与真实网络结构之间必然存在差距，部分信息未获取导致所观察到的网络结构存在一定的缺失连边。这些缺失连边所带来噪声将对数据分析结果产生极大影响，如何在这类噪声环境中准确还原真实网络结构具有重要研究意义。针对这一问题近年来在链路预测研究中提出了一系列方法，大致可以分为基于相似性，基于最大似然估计，基于概率模型三类[38]。其中基于相似性链路预测方法是通过利用网络上节点的结构相似性进行预测；最大似然估计方法是通过抽样得到的样本网络的后验概率来计算网络最可能的结构，较为有名的模型有层次结构模型、随机分块模型[39]、闭路模型[40]与局部共现模型[41]；基于概率模型的方法则是通过构建一个具有一组可调参数的模型，通过机器学习或最优化等方法找到一组最优的参数用来再现网络的结构与特征。概率模型与最大似然估计方法虽然在设计上有所差异，但是核心思想是通过转化为参数估计问题来进行预测，属于有监督链路预测方法，因此链路预测方法根据聚类与分类的区别可以进一步分类为无监督方法与有监督方法。本文对博弈论的应用根据这一分类分别进行介绍。

4.1 基于博弈论的无监督链路预测方法

在没有明确的标签可以用于对网络进行分类的情况下，无监督的链接预测方法可以通过聚类的方法反映相关网络结构的多个方面，并且可以将它们视为比较分析的基准。传统方法一般是基于网络局部信息，如CN，AA，RA等指标；或是基于路径信息，如LP，Katz等指标；以及基于随机游走思想的ACT，SimRank，RWR等指标，这一类方法具有计算复杂度低，普适性较强的优点。但由于该类方法都是基于网络结构理论所提出，如同质、互惠、偏好链接、三重闭合等，所以该方法较依赖于网络结构的特征与规律性[42]，在不同网络上的预测精度波动较大，具有一定的局限性。而博弈论在此基础上，从节点间行为交互出发，通过对各类网络节点的基本行为进行刻画，在较为普适地适用于各类网络的同时，也保证了较稳定的预测精度。

文献[43]在基于相似性链路预测方法的基础上，提出了一种三阶段算法，该方法通过co-author和Linear Best Response两种网络演化博弈模型对节点的收益进行计算，后运用相似性方法进行链路预测得到预测边排序，此时得到每个节点收益并与初始收益进行差值分析。以此为评判标准将第二阶段中预测出可能存在的边再重新排序，以排除那些第二步中排序分高但是并没有提高节点收益的链接，得出最终的排序。这一方法将网络演化博弈模型与传统的基于结构相似性的链路预测方法相结合，应用范围广，可以根据不同的网络选择不同的网络演化博弈模型与链路预测算法，但在产生可能存在连边时也没有考虑收益信息，没有很好的将博弈论集成到链路预测模型中。文献[44]在考虑网络演化博弈外部激励作用的基础上，用节点重要性来衡量节点自身属性，综合考虑两种因素，以此提出了GAR（博弈推荐，game approach for recommendation）算法。该算法首先通过将节点的度中心性，节点的位置以及节点间距离等因素加权相加以此评估节点重要性，并计算一个节点与所有和它具有共同邻居的节点之间重要性差。根据重要性相似节点间更容易产生连接的思想，把差距最小的节点推荐给此节点，取得了不错的预测效果。该算法不同之处在于通过节点的效用差来表示节点间的相似度，不再通过以上介绍的指标计算相似度后预测，而是直接通过效用差来预测连边，该算法由于是以每个节点为中心进行计算，所以可以在网络中进行分布并行计算，适用于大型网络，但如何准确选取节点属性来评估节点重要性没有一个标准，这将在很大程度上影响预测的效果。同样，基于目标节点的分布并行计算思想，文献[45]利用GTG（图转导博弈，graph transduction game）构建链路预测问题，通过设置策略集｛“成为朋友”，”不成为朋友”｝，为每个节点随机分配相应策略进行博弈交互并计算收益，每一轮选择一个节点u，查看它与邻居的策略交互，进行GTG-Rank排序，即对比它

的邻居节点改变策略后收益的变化，并按收益增长从大到小排序。进而进行链路预测，该方法与GAR算法类似，都是独立预测每个节点的连边，具有并行化的优点，且直接通过节点效用进行推荐，不同之处在于GAR算法通过节点结构特性计算节点效用，考虑节点间交互行为对效用的影响较少；GTG算法则设置效用矩阵，通过节点策略交互得到节点效用，考虑结构相似性较少。

除了考虑将利用局部结构信息的预测方法与博弈论结合之外，文献[25]提出了一种基于博弈论与k壳分解方法的带权随机游走DeepWalk。该方法分别采用co-author网络演化博弈模型或Shapley值与K-核分解计算得出的权重相乘，作为链路预测随机游走方法的转移概率矩阵。通过有偏向性的随机游走，体现了博弈论中网络节点行为的相互影响以及节点具有连接偏好的思想，相比无权随机游走方法取得了更好的预测效果。但该方法采用的co-author网络演化博弈模型并不能准确描述大多数网络上的节点效用，特别在在社交网络中，连边的建立并不一定会同时提高节点双方的效用，且获得的网络结构信息一般情况下是稀疏且不完整的，针对这一问题，文献[46]提出了一种运用博弈论与SimRank结合的方法，首先确定网络的初始群集，计算节点特征向量集合。之后通过设置效用函数对网络的每个目标节点进行一个最近邻计算，用于识别节点之间的隐式和显式关系，并基于该相似度进行SimRank随机游走得到最终的连边预测。

4.2 基于博弈论的有监督链路预测方法

与无监督的链接预测方法不同，有监督的方法在具有先验知识的基础上，通过已知网络结构和节点属性对网络未知结构进行分类与预测。该方法大致可以分为基于最大似然估计及概率模型两类，最大似然估计通过对网络抽样得出出现概率最大的网络结构，具有高预测精度的优点，但计算复杂度太高不适用于大型网络。而概率模型通过构建一个由一组可调参数构成的模型，通过机器学习等方法进行参数估计，以得到能准确描述网络结构的模型，但同样具有计算复杂度高的缺点。而将网络节点的特征与行为交互抽象为效用函数进行表示，通过博弈论思想对网络结构信息进行预处理，排除参数估计过程中数据中的噪声，可极大的降低了计算复杂度并提高链路预测的准确性。

文献[47]通过将博弈论与EM算法结合，认为仅使用效用分析节点间连接意愿并不准确，因为有可能一些节点根本没有相遇，即没有决策的机会，所以应该将相遇这个因素考虑进效用分析中。因此将节点间的交互过程分解为相遇过程与决策过程，并将效用函数用U_i（G（N，L），$\boldsymbol{C}$；$\boldsymbol{\theta}$）表示，其中$\boldsymbol{C}$代表节点属性矩阵，定义为（$\boldsymbol{C}_1$，…，$\boldsymbol{C}_i$，…，$\boldsymbol{C}_n$），其中每一列$\boldsymbol{C}_i$表节点i的属性向量，可以由年龄、性别等组成，$\boldsymbol{\theta}$代表偏好参数向量。该算法将节点相遇看作一种未知潜在状态，假设节点的相遇概率为p_{ij}^{m}，而该概率与节点间的共同邻居数量有关系，可表示为如下形式：

该算法排除了节点不相遇这一情况产生的噪声，保留了极大似然估计方法的高预测精确度的同时降低了计算复杂度，使其可以用于规模更大的网络。在概率模型方法中，文献[43]考虑到了其中提出的三阶段方法中没有将博弈论集成到链路预测模型的缺点，在BPR（贝叶斯个性化排序法，Bayesian personalized ranking）[48]基础上，与网络演化博弈模型相结合，提出了一种基于博弈论的链接预测的贝叶斯个性化排序算法BPRLGT（Bayesian personalized ranking for link recommendation with game theory）[43]，该模型中用Pui来表示节点u与节点i之间产生链接的偏好值，在了解网络上部分偏好信息后，通过矩阵分解的方法，将真实网络中的链接偏好矩阵用分解矩阵和偏移项HuHiT+bi表示，通过梯度下降方法拟合真实网络矩阵进行连边偏好预测。此外，还通过考虑节点间的共同邻居特性来提高链路预测的准确性，该模型算法流程与以上提出的三阶段方法的思想一致，都是通过将连边加入后判断节点收益是否提升，以此去除那些排序分高但是并没有提高节点收益的链接。不同之处在于在选择训练集时BPRLGT运用了博弈论思想进行抽样，在抽取样本时随机抽取节点u后，考虑网络拓扑中的共同邻居影响。之后计算关于u的向量hu=HuHT+b，即u与其他节点间的链接偏好度向量，将hu中元素进行降序排列，从最大值开始扫描，当找到第一个值满足当其连边加入后节点u收益下降条件时，将其定为j，从而形成了一个四元组｛u，i，j，k｝。

$$p_{ij}^{m}=\frac{\exp(b\cdot\sum_{r\neq i,j}l_{ir}l_{jr})}{a+\exp(b\cdot\sum_{r\neq i,j}l_{ir}l_{jr})} \tag{4}$$

其中a与b都是非负数。a初始相遇的概率，b表示共同邻居的影响强度。相遇过程与决策过程中的概率相互独立，互不影响。因此可以使用极大似然估计方法，通过使用EM算法估计未知参数$\boldsymbol{\beta}$=（a，b，$\boldsymbol{\theta}$）的最优值，算法具体流程如图1所示。

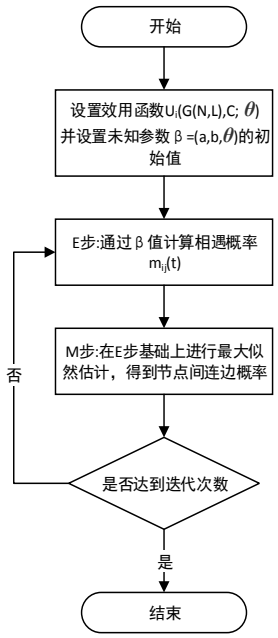


图1　基于EM算法流程图

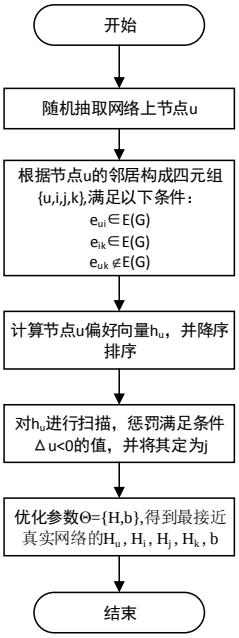


图2　BPRLGT算法流程图

作为梯度下降训练样本，来最大化AUC，得到最佳预测结果。该模型通过使用博弈论选择训练样本与基于机器学习的链路预测方法达到了将博弈论思想运用于链路预测方法的目的，该模型流程图如图2所示。

5　基于博弈论的社团发现方法

由于复杂网络具有自组织与无标度特性，部分节点因为拥有较多连边而成为网络中的枢纽节点，网络中大部分节点通过这些枢纽节点进行信息通信与交互，因此导致某些节点间相比其他节点连边较密集，这种由自组织特性形成的节点簇被称为社团。社团发现即通过对网络中蕴含的信息进行分析，达到挖掘网络中具有相似特性节点簇的目的[49]。由于网络中的社团往往是因为节点间行为影响或节点属性特征相似而形成的，因此通过博弈论构造效用函数可以准确的描述网络上这种自组织和模块化的特性。从博弈论角度看来，社团发现问题可以描述为节点之间进行的博弈，该过程直到网络处于平衡状态结束，在平衡状态下，每个节点不能通过单方面改变策略来提高收益，此时得到网络中的最终社团结构[50]。相比传统的社团发现算法，博弈论的应用将显著降低社

团发现的计算复杂度，从而可以适用于更多快速变化且规模庞大的网络。

5.1 基于经典博弈论的社团发现方法

文献[51]最早将博弈论与社团发现相结合，提出了一种在非合作博弈类型下的社团发现算法，与链路预测类似，将网络中节点行为定义为最大化自身效用以此来模拟网络演化行为。具体问题定义为：设置网络节点集合V=｛$\boldsymbol{C}_1$，$\boldsymbol{C}_2$，…，$\boldsymbol{C}_n$｝，其中$\boldsymbol{C}_i$，$i\in$［1，n］代表网络上社团的节点集合，$S_j\supseteq\boldsymbol{C}_i$代表第i个社团上第j个节点的策略，社团中的所有节点策略集合共同组成策略空间，因此社团的策略空间具有差异性，且当一个节点不属于网络上任何一个社团时其策略空间为空集，节点可以有加入社团，离开社团，转换社团三种行为。当无法通过改变自身策略来提高自身收益时，网络达到纳什均衡状态，此时网络达到稳定的社团结构该算法被应用于识别DBLP网络中的重复名称，并使用模块化函数来描述效用，结果证明对于重叠社团检测也具有较好的效果。在此基础上，文献[52]提出用节点间相似度构造效用函数，进一步降低了计算复杂度。

同样基于以上博弈论对社团发现问题定义，文献[53]从链路预测角度出发，同时考虑社团目前与未来的拓扑结构，在将社团看作一组连接紧密的节点簇的同时，还考虑了社团内部节点间将会产生的新连边。在博弈论基础上提出一种方法，使用链路预测AA指标对社团中节点连边能力进行衡量，并通过连边能力与各节点间间隔距离选择出中心LPN（局部主节点，local primary nodes），以LPN为中心运用博弈论框架扩展社团检测范围最终得到社团结构，其具体算法如图3所示。

该算法运用链路预测指标与社团发现相结合，运用连边能力强的节点在社团中也往往处于中心的思想计算出网络中的关键节点，并以此为初始社团进行社团间的博弈，对于各类真实无向网络上的互斥社团检测有较好的效果。

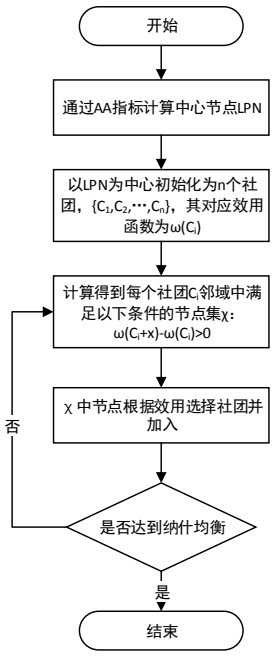


图3 基于经典博弈论的社团检测算法流程图

5.2 基于演化博弈论的社团发现方法

由于社交网络的快速发展，这类快速变化的网络已逐渐成为网络科学中的重点研究类型，针对这类动态变化的网络。基于经典博弈论提出的社团发现算法并不能很好的对其进行结构分析，因此为了解决这类问题，文献[54]最先提出了运用网络演化博弈的社团发现算法，该算法在动态有向网络上运用网络演化博弈模型，与经典博弈论不同之处在于，网络演化博弈将博弈论与自然界生物演化规律相结合，认为网络上节点并非完全理性且在网络演化达到稳态后大多数成员将采取一致的策略，这种策略被称作ESS（进化稳定策略，evolutionarily stable strategy）。具体问题定义为：定义网络上的博弈为Γ=（N，S，U），其中N代表节点集合；S_i，$i\in$N代表节点集中的节点i策略集合，社团的策略空间由成员节点策略集合共同组成；节点i的收益通过效用函数u_i计算，由于社交网络等动态网络具有方向性，所以效用函数可以通过以下形式表示：

$$u_i(S)=C_i(S)-C_{-i}(S) \tag{5}$$

$$C_i(S)=\frac{K_S^{in}}{(K_S^{in}+K_S^{out})^{\alpha}} \tag{6}$$

定义社团分数用于定义社团的整体效用，其中节点行为与经典博弈论上一致，若社团加入节点i将提升社团分数，则将节点加入社团，直到博弈过程达到稳态为止。该方法使用社团的整体出度与入度来衡量社团效用，既表现了有向网络中社团的模块化程度，也考虑了其方向性，并通过计算ESS代替纳什均衡设置为网络演化终止条件，在动态有向网络上取得了较好的社团检测效果。

文献[29]认为网络中的大多数节点都与某个节点相邻，不一定代表它们是相似的；但如果只有两个节点与某个节点相邻，则这是仅由这两个节点共享的独特特征，这表明它们是相似的，并认为在半径k内节点对目标节点具有等效影响力。在此基础上基于合作博弈中联盟博弈理论和模糊系统的概念构建了一种博弈论互动指数的节点间相似度的新度量方法，其通过网络演化博弈Influence game模型构建了一个k步影响力函数，是一种以k为半径的半局部链路预测算法，相对于全局链路预测算法如Katz等经典算法取得了较好的预测效果。但半径k内节点往往影响力并不是均匀的，文献[55]证明了距离越远的节点间的相互影响力越小，文献[56]基于这一理论在文献[29]基础上提出了一种改进的节点间相似度度量方法，具有在不是最佳半径k条件下也具有稳定的预测效果的优点，并在社团发现与链路预测问题中取得了较好的效果。

6 结束语

以上从网络演化，社团发现以及链路预测领域综述了近年来博弈论在其中的应用发展，各领域本质上都是对复杂网络结构演化进行预测与研究，进而挖掘有价值信息的方法。博弈论作为一种研究个体在利益相关情形下行为相互影响演化的理论，与网络科学同样都在各个学科研究中得到了广泛应用，渗透到社会科学、信息科学等多个领域[28]。本文按照博弈类型或实现方法对博弈论的应用进行了分类总结，其核心都是通过节点或网络结构的属性特征构造效用函数，将网络演化转变为节点间博弈过程，以此来描述复杂网络结构变化。相对基于结构的方法，可以达到降低计算复杂度并得到较好效果的目的。但由于博弈论描述的是节点间的相互作用，链路预测的连边建立或者社团的结构变化大多都是节点间相互影响的结果，因此目前大多数运用博弈论思想提出的方法都是基于无向网络，对于连边方向导致差异化信息的有向网络中，博弈论在该研究领域还具有巨大的研究价值。

参考文献:

[1] 王凯,刘树新,陈鸿昶,等. 一种基于节点间资源承载度的链路预测方法[J]. 电子与信息学报,2019,41(5):1225-1234.

[2] 白桦,马云龙,毕玉,等. 一种基于节点局部相似性的复杂网络链路预测算法[J]. 计算机应用与软件,2020,37(5):298-308.

[3] 孙士保,张亚楠,张京山,等. 基于复杂网络的协同舆情演化模型研究[J]. 计算机应用与软件,2017,34(6):52-68.

[4] 刘树新,季新生,刘彩霞,等. 一种基于节点集聚程度的相似性链路预测方法[J]. 信息工程大学学报,2017,18(6):62-66.

[5] 刘树新,季新生,刘彩霞,等. 局部拓扑信息耦合促进网络演化[J]. 电子与信息学报,2016,38(9):2180-2187.

[6] 代琼琳. 复杂网络上的演化博弈动力学研究[D]. 北京,北京邮电大学,2011.

[7] 严家萌,庞超逸,许立波. 复杂社会网络节点重要性可拓聚类动态分析方法[J]. 计算机应用与软件,2019,36(7):76-82.

[8] 刘树新,季新生,刘彩霞,等. 一种信息传播促进网络增长的网络演化模型[J]. 物理学报,2014,63(15):1-11.

[9] 潘永昊,于洪涛,刘树新. 基于神经网络的链路预测算法[J]. 网络与信息安全学报，2018,4(7):34-42.

[10] LI J, PENG J, LIU S, et al. Link Prediction in Directed Networks Utilizing the Role of Reciprocal Links[J]. IEEE Access，2020，8：28668-28680.

[11] LIU S ,JI X ,LIU C, et al. Similarity indices based on link weight assignment for link prediction of unweighted complex networks[J]. International Journal of Modern Physics B,2016,31(2):1650254.

[12] LIU S, JI X S, LIU C X, et al. Extended resource allocation index for link prediction of complex network [J]. Physica A: Statistical Mechanics and its Applications,2017，479：174-183.

[13] 刘思,刘海,陈启买,等. 基于网络表示学习与随机游走的链路预测算法[J]. 计算机应用,2017,37(8):2234-2239.

[14] WATTS D J, STROGATZ S H. Collective dynamics of small world networks[J]. Nature，1998,393：440-442.

[15] BARABÁSI A L, ALBERT R. Emergence of scaling in random networks[J]. Science，1999,286：509-512.

[16] 王润芳,陈增强,刘忠信. 融合朴素贝叶斯方法的复杂网络链路预测[J]. 智能系统学报,2019,14(1):99-107.

[17] 赵希文,王铁成. 复杂金融系统研究综述[J]. 前沿,2019,422：24-31.

[18] ZHANG S H, ZHANG Z P, WU Y E, et al. Strategy preference promotes cooperation in spatial evolutionary games[J]. Physica A: Statistical Mechanics and its Applications,2019,514:181-188.

[19] LI X L, Jusup M, WANG Z, et al. Punishment diminishes the benefits of network reciprocity in social dilemma experiments [J]. Proceedings of the National Academy of Sciences，2018，115(1)：

30-35.

[20] CUI P B, WU Z X, ZHOU T, et al. Cooperator-driven and defector-driven punishments: How do they influence cooperation? [J]. Physical Review E，2019，100(5)：052304.

[21] PERC M, SZOLNOKI A. Coevolutionary games—A mini review [J]. Biosystems，2010，99(2)：109-125.

[22] WU T Y, LEE W T, GUIZANI N, et al. Incentive mechanism for P2P file sharing based on social network and game theory[J]. Journal of Network and Computer Applications，2014，41：47-55.

[23] CHEN Y, GAO Y, Liu K. J. R. An evolutionary game-theoretic approach for image interpolation [C]. 2011 IEEE International Conference on Acoustics，Speech and Signal Processing（ICASSP），Prague，2011：989-992.

[24] WATANABE E H, DANIEL S M, SILVA E D S E, et al. Modeling Resource Sharing Dynamics of VoIP users over a WLAN Using a Game-Theoretic Approach[C]. 27th IEEE Conference on Computer Communications (INFOCOM 2008)，Phoenix，AZ ，2008：1588-1596.

[25] NASIRI E, BOUYER A, NOURANI E. A node representation learning approach for link prediction in social networks using game theory and K-core decomposition[J]. The European Physical Journal B，2019(92)：228.

[26] 代翔. 博弈论在社交网络中的应用[J]. 计算机与数字工程，2017，45(6)：1127-1132.

[27] 侯艳巧. 复杂网络的演化博弈及网络重构研究[D]. 西安，西安电子科技大学，2014.

[28] 吕金虎,谭少林. 复杂网络上的博弈及其演化动力学[M]. 北京：高等教育出版社，2019：2-5.

[29] SZCZEPANSKI P L, BARCZ A, MICHALAK T P, et al. The Game-Theoretic Interaction Index on Social Networks With Applications to Link Prediction and Community Detection [C]. 1st International Workshop on Social Influence Analysis / 24th International Joint Conference on Artificial Intelligence (IJCAI)，Buenos Aires，Argentina ，2015：638-644.

[30] 李治,柳妍珠,赵亚萌,等. 基于复杂网络的雾计算网络鲁棒性研究[J]. 中北大学学报，2017，38(2)：178-184.

[31] WU J J, ZENG J W, CHEN Z H, et al. Effects of traffic generation patterns on the robustness of complex networks [J]. Physica A: Statistical Mechanics and its Applications，2018，492：871-877.

[32] 周漩,张凤鸣,周卫平,等. 利用节点效率评估复杂网络鲁棒性[J]. 物理学报，2012，61(19)：1-7.

[33] ZHU L, LIU X C, YU L, et al. Model of Cascading Failures for Communication Networks[J]. International Journal of Computer and Communication Engineering，2016，5(5)：302-310.

[34] 李俊,吕欣,谭跃进,等. 基于空间结构的战术通信网络建模[J]. 系统工程与电子技术，2010，32(7)：1456-1461.

[35] 刘爱志. 基于演化博弈论的若干合作演化机制研究[D]. 北京，北京科技大学，2018.

[36] CAI Y J, ZHENG H, LIU J M, et al. Balancing the Pain and Gain of Hobnobbing [C]. 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)，Stockholm，SWEDEN 2018.

[37] BRAMOULLE Y, KRANTON R, DIAMOURS M. Strategic interaction and networks[J]. CIRPEE Working Paper，2014，104(3).

[38] LÜ L Y, ZHOU T. Link prediction in complex networks: a survey [J]. Physica A:statistical mechanics and its applications，2011，390(6)：1150-1170.

[39] GUIMERA R, SALES-PARDO M. Missing and spurious interactions and the reconstruction of complex networks[J]. Proceedings of the National Academy of Sciences of the United States of America，2009，106：22073-22078.

[40] PAN L , ZHOU T , LÜ L Y, et al. Predicting missing links and identifying spurious links via likelihood analysis [J]. scientific Reports，2016，6：22955.

[41] MARTÍNEZ V, BERZAL F, CUBERO J C. A Survey of Link Prediction in Complex Networks[J]. 2016，49(4)：1-33.

[42] LÜ L Y, PAN L, Zhou T, et al. Toward link predictability of complex networks. [J]. Proceedings of the National Academy of Sciences of the United States of America，2015，112(8)：2325-2330.

[43] ZHAO T, ZHAO H. VICKY K I. Exploiting Game Theoretic Analysis for Link Recommendation in Social Networks[C]. the 24th ACM International，Melbourne，Australia 2015.

[44] YANG A T, TANG Y, WANG J B, et al. Personalized Friends Recommendation System Based on Game Theory in Social Network [J]. Computer Science，2015，42(9)：191-194.

[45] ZAPPELLA G, KARATZOGLOU A, BALTRUNAS L. Games of Friends: a game-theoretical approach for link prediction in online social networks[C]. AAAI 2013 Workshop，Intelligent Techniques for Web Personalization and Recommender Systems（ITWP），Washington, USA，2013.

[46] YANG L, HONG T, GOPALAKRISHNAN A K. A Framework for Recommender System Based on Game Theory in Social Networks [C]. 2018 10th International Conference on Knowledge and Smart Technology (KST)，Chiang Mai，Thailand，2018：95-100.

[47] LI Y l, LUO P, FAN Z P, et al. A utility-based link prediction method in social networks[J]. European Journal of Operational Research，2017，260(2)：693-705.

[48] RENDLE S, FREUDENTHALER C, GANTNER Z, et al. BPR: Bayesian Personalized Ranking from Implicit Feedback [J]. Computer Science，2012，452-461.

[49] 刘唯一. 基于网络信息的社团发现关键技术研究[D]. 成都，电子科技大学，2019.

[50] JONNALAGADDA A, KUPPUSAMY L. A survey on game theoretic models for community detection in social networks [J]. Social Network Analysis and Mining，2016，6(1)：83.

[51] CHEN W, LIU Z M, SUN X R, et al. A game-theoretic framework to identify overlapping communities in social networks [J]. Data Mining and Knowledge Discovery，2010，21(2)：224-240.

[52] ALVARI H, HASHEMI S, HAMZEH A. Artificial Intelligence and Computational Intelligence [M]. Heidelberg：Springer Berlin Heidelberg，2011：620-630.

[53] HESAMIPOUR S, BALAFAR M A. A new method for detecting communities and their centers using the Adamic/Adar Index and game theory [J]. Physica A: Statistical Mechanics and its Applications，

2019，535：122354.
[54] LUNG R I, CHIRA C, ANDREICA A. Game theory and extremal optimization for community detection in complex dynamic networks. PLoS One，2014，9（2）：e86891.
[55] PIOTR L. S, TOMASZ P M, TALAL R, et al. An Extension of the Owen-Value Interaction Index and Its Application to Inter-Links Prediction［J］. Frontiers in Artificial Intelligence and Applications，2016：90-98.
[56] TARKOWSKI M, MICHALAK T, WOOLDRIDGE M. A Game-Theoretic Algorithm for Link Prediction［J］. arXiv: 1912. 12846［physics］，2019.

[作者简介]

石灏苒（1997—），男，在读硕士，主要研究方向：复杂网络链路预测，网络重构。

潘世东（1987—），男，研究生学历，工程师，主要研究方向：信息处理。

吉立新（1969—），男，博导，副总工程师，主要研究方向：电信网安全，数据挖掘。

刘树新（1987—），男，博士，助理研究员，主要研究方向：通信网络安全，复杂网络演化、链路预测。

有限异构资源下的拟态调度器与安全性评估

李合元，王延松，张汝云*，朱明星，骆汉光，李顺斌
之江实验室，杭州 310012

摘　要：网络空间拟态防御技术是应对信息系统未知漏洞后门攻击的有效手段，其安全性与执行体的异构化程度以及裁决调度策略具有紧密相关性。为了探究有限异构资源条件下的拟态信息系统安全性，本文提出了基于置信度与胜任系数的拟态调度方法与安全仿真评估模型。实验结果表明，所提出的用于三余度拟态信息系统的拟态调度器，可自适应根据环境特性选择合适的执行体上线。即使在高强度攻击环境下，依然能保持92.88%的高可用概率，给拟态系统设计的成本规划提供了新思路。

关键词：拟态控制器、内生安全、资源受控模型

Mimic scheduler and security evaluationfor limited heterogeneous resources conditions

Li Heyuan, Wang Yansong, Zhang Ruyun*, Zhu Mingxing, Luo Hanguang, Li Shunbin

Abstract: Cyberspace mimic defense technology is an effective method to deal with backdoor attacks on unknown vulnerabilities in information systems. Its security is closely related to the heterogeneity of the executors and the scheduling strategy. In order to explore the security of mimic information systems under limited heterogeneous resources condition, this paper proposes a mimic scheduler and a security evaluation model based on reputation and credibility coefficients. The experimental results show that the proposed mimic scheduler for triple-redundancy mimic information system can adaptively select a suitable executor to go online according to environmental characteristics. Even in a high-intensity attack environment, it can still maintain a high availability probability of 92. 88%, which provides a new idea for the cost planning of mimic system design.

Key words: Mimic controller; Endogenous security; Resource-constrained model

1　引言

当前网络空间安全存在四个本源性问题：一、受软硬件构件设计水平的局限，信息系统设计缺陷的漏洞难以避免；二、你中有我，我中有你的开放生态环境中，信息系统无法做到完全自主可控，软硬件后门的可能性长期存在；三、现阶段的技术尚无法做到彻查所有的漏洞后门；四、产品和系统的安全质量难以得到有效控制。

为了能在“有毒带菌”的环境下构筑网络安全，中国工程院院士邬江兴基于共识机制提出了拟态防御理论［1］，通过在“动态-异构-冗余”的DHR（Dynamic Heterogeneous Redundancy）架构上引入多模裁决机制，对攻击者形成测不准防御迷雾，有效地将网络空间的不确定威胁问题归一化为利用鲁棒控制理论能解决的工程技术问题。针对拟态防御理论的多模裁决机制和拟态调度策略，学术界展开了大量的研究。之江实验室吴春明等人提出了基于信誉度与相异度的自适应拟态控制器［2］，利用相异度指标衡量各执行体之间的差异，从而得到异构程度最高的执行体；利用信誉度指标衡量执行体的脆弱程度，从而决定执行体在多模裁决中的作用程度；为了解决现有裁决模块中超时机制适应性差的问题，国家数字交

基金项目：本论文受国家重点研发计划项目（2020YFB1804800），之江实验室开放课题资助（编号：2018FD0ZX01）。
通讯作者：张汝云，zhangry@zhejianglab.com

换系统工程技术研究中心刘勤让等人提出了一种基于回归样条的自适应超时机制［3］；数学工程与先进计算国家重点实验室马博林等人，建立了可描述分析DHR架构的核心特征的模型［4］，通过蒙特卡罗仿真验证了冗余度与拟态防御系统中防御攻击能力之间的关系。信息工程大学的马海龙等人在基于动态异构冗余机制的路由器拟态防御体系结构［5］一文中提出了基于执行体可信度的随机调度策略和基于执行体性能权重的随机调度策略；浙江大学陈利跃等人提出了基于K-means聚类算法的异构度调度策略［6］，可通过增强执行体集的异构度、优化调度算法，提升攻击难度、增强拟态防御系统性能。

伴随理论研究的不断深入，拟态防御技术已经广泛用于路由器、防火墙、数据中心、域名服务器等关键网元设备，赋予设备内生安全的功能。安全设计者的目光焦点从理论研究逐渐转移到实际场景中的设备拟态化改造上，更加关注工程可实现性、投入费效比的衡量上来。针对在工业控制领域的应用场景（如现场PLC、终端、RTU等控制设备），国家数字交换系统工程技术研究中心魏帅等人［7］认为，在处理机故障率小于1%的情况下，四核拟态处理机与三核拟态处理机相比整体的错误故障率降低幅度并不明显。三核心拟态处理机明显具有更高的费效比。

受当前应用生态局限性的影响，许多应用场景可能无法在较低的成本、较短的时间内生成多个异构化程度较大的执行体集合。在异构化执行体资源受限条件下，如何设计高效的多模裁决算法并进行可量化评估，是值得研究的一个问题。

本文针对当前应用最为广泛的三余度的DHR架构，提出了一种基于置信度和胜任系数的拟态调度方法，将执行体的当前状态与历史表现状态转化为可量化的指标来实施拟态调度，可以确保N-1模攻击只需要执行2次清洗，N模攻击只需要3次清洗即可杜绝逃逸。针对应用生态受限的客观条件，本文建立了基于攻击强度的仿真模型，可用于评价有限异构资源条件下拟态多模裁决机制的有效性。

2　资源受限条件下的安全调度模型介绍

2.1　拟态防御理论简介

根据拟态防御理论：□基于构造的内生安全信息系统包含由多个独立的异构执行体；□执行体根据时间（或环境）动态切换成上线或清洗备用状态；□系统的输入通过输入代理模块分发至在运行的多个执行体中并获得多个输出结果；□多模裁决模块通过比对执行体的输出结果，判定各执行体的安全状态，动态调控执行体的状态。由于系统架构采用严格的单向联系机制，攻击者基本无法直接触碰到由输入代理与输出裁决模块组成的拟态括号。实现攻击逃逸的唯一方式即为，对拟态括号内的由多个异构执行体实现非配合下的盲协同攻击。可见，基于拟态构造的内生安全架构，其安全性很大程度依赖于执行体相互间的异构化程度。倘若执行体完全异构，则采用简单的则多判决即可保证系统安全可信。

然而受当前应用生态发展现状的约束，许多应用场景中例化的异构执行体之间存在共模漏洞是不可避免的事实。因此，拟态防御理论并不强调绝对的静态安全，而是通过引入了多模裁决策略对受到攻击的执行体进行动态清洗切换。即使攻击者耗费巨大精力在非配合条件下的成功构造出盲协同攻击，也会被系统识别并清洗。指数量级提升攻击难度并且使得攻击效果不可持续，是拟态防御的两大显著效果。

2.2　安全调度控制模型与假设

区别于拟态防御理论中的一般化描述，我们做出如下假设：

1、 系统采用费效比最高的三余度DHR架构。因为同时在线运行的执行体数量必须大于等于3个，才能满足输出结果的比对与识别；

2、 系统的异构执行体数量设置为5。5个执行体是三余度DHR架构系统在遭受N-1模攻击后能通过清洗恢复到正常状态，实现攻击面转移的最低条件［1］；

3、 每个执行体均存在漏洞，且漏洞分为两大种类。第一类漏洞后门依赖于该执行体所独有运行环境，触发概率相对较高，容易造成差模攻击，概率值设为P_{diff}；第二类漏洞后门与执行体所独有运行环境无关，触发概率相对较低，一旦被挖掘

利用用以造成共模攻击，概率值设置为P_{comm}；

4、 在系统执行体启动前为每个执行体分配一个胜任系数值Tc，$Tc \in (0, 1)$。执行体胜任系数，可由用户根据经验信息结合执行体应用生态情况以及执行体相互间异构化程度综合评定。

a） 执行体应用生态越好，安全防护手段考虑越多，漏洞也越少，胜任系数值越高；

b） 执行体与其它异构执行体的差异度越大，在系统调度中就更偏向于选择其上线，以达到提升抗盲协同攻击难度的效果；执行体的差异度可借助相似性度量工具评估［8］。

c） 执行体胜任系数Tc是一个时变参数，根据执行体在实际工作环境中的表现自适应动态更新；

5、 假设系统内所有执行体均能通过下线清洗的方法消除漏洞或者使漏洞不再处于被激活状态。所有漏洞后门都需要经过触发加载恶意指令才能生效，下线后的执行体其存储数据被重置，因而该假设充分且合理。

3 基于置信度与胜任系数的拟态调度器

据前文分析可知，由于系统构造的复杂性、执行体的不完全异构性、攻击行为的不可预测性，资源受限条件下的拟态防御系统设计面临N-1模以及N模攻击时必然存在一定的逃逸概率，可能需要多次清洗才能彻底消除攻击带来的隐患。因此如何有效地降低异构冗余构造的逃逸概率和逃逸时间是动态异构冗余构造的关键难点。

本小节提出一种基于置信度与胜任系数的拟态调度方法，其中置信度V是执行体在运行环境中的综合表现评价（包括历史表现行为与当前风险抵御能力），用于判别是否需要进行清洗；胜任系数Tc是执行体原始属性（包括执行体应用生态，清洗成本，执行体间异构化程度等）与历史表现行为的综合评价，用作表征该执行体上线运行的胜任能力。

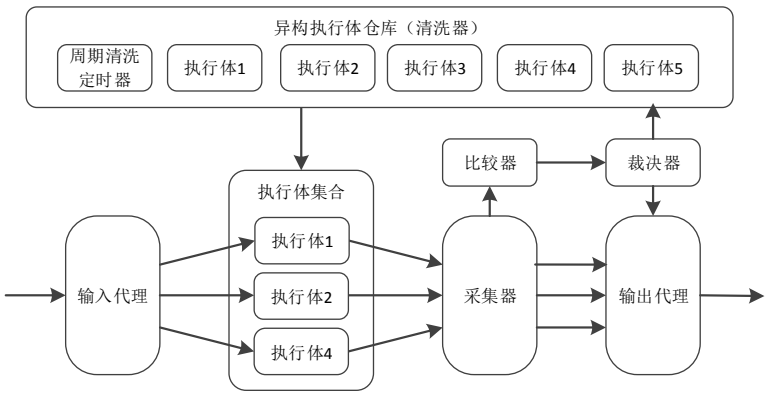


图 1 基于置信度与胜任系数的拟态调度器

基于置信度与胜任系数的拟态调度器如图 1 所示，包含采集器、比较器、裁决器、清洗器（含周期清洗定时器C）等。具体工作流程如下：

1、 系统启动态。从执行体仓库中选择 3 个胜任系数最高的执行体上线，并赋予每个刚上线的执行体初始置信度均为最大值$Vmax$。周期清洗定时器C复位为零。

2、 执行体采集。执行体采集器采集 3 个在线执行体的输出结果分别为R1， R2， R3。

3、 执行体比较。三个执行体输出结果两两比较（共 3 次）：

a） 当执行体输出结果一致时，两个执行体的置信度都增加D；

b） 当输出结果不一致时，两个执行体的置信度都减少D；

c） 比较结束后，若单个执行的置信度大于$Vmax$，则令其等于$Vmax$。

4、 执行体裁决。选择判断需要下线的执行体并选择置信度最高的执行体作为系统的输出。

a） 遍历所有在线执行体当前的置信度，选择置信度小于或等于 0 的执行体作为备选清洗对象；

b） 如果有 2 个执行体置信度小于或等于 0，选择则置信度最小的执行体作为清洗对象；

c） 如果有 2 个执行体置信度相等且小于或等

于0，选择胜任系数较小的作为清洗对象；

d） 如果有2个执行体置信度相等且小于或等于0，胜任系数亦相同，选择其一作为清洗对象；

e） 检查执行体上线保护寄存器C_{reg}。系统可能遭受N-1模攻击，保护上一次刚上线的执行体。

i. 清洗对象与上线保护寄存器C_{reg}记录的一致，随机清洗选择另外两个执行体中的一个；

ii. 清洗备选对象与上线保护寄存器C_{reg}记录的不一致，更新执行体上线保护寄存器的值；

iii. 若执行体上线保护寄存器经过3次裁决周期未发生变化，清空该寄存器的值。

5、 定时器更新。引入扰动，对抗基于0-day漏洞的共模攻击。

a） 如果所有执行体的置信度均为*Vmax*，周期清洗定时器加1；

b） 如果执行体发生过清洗行为，周期清洗定时器C复位为零；

c） 当周期清洗定时器C的值大于指定阈值，随机选择其中一个执行体进行清洗。

6、 执行体清洗。选择仓库内胜任系数最高的执行体上线运行，同时将下线清洗的执行体胜任系数乘以系数P（0<P<1）。

4 安全性仿真评估与分析

4.1 脆弱性仿真模型

本文采用数值仿真的方法，对所提出的基于置信度与胜任系数的拟态调度器安全性能进行评估。在评估拟态调度器的安全性之前，首先需要为执行体建立脆弱性仿真模型。

1、 本仿真模型采用差模漏洞触发概率P_{diff}表征该执行体在运行过程中遭受到攻击且表现形式与其他执行体不一致的概率。

2、 本仿真模型采用共模漏洞触发概率P_{comm}表征该执行体在运行过程中遭受到攻击且该攻击的表现形式与其他执行体的表现形式一致的概率。

3、 本仿真模型采用“0/1/2”三个值，分别表征单个执行体处于正常/差模漏洞生效/共模漏洞生效三种状态。仿真过程中的每次迭代，若执行体状态值为“0”，则概率进入状态“1”或者“2”；若执行体状态值已经为“1”或者“2”，则保持原值，直至被清洗（基于当执行体已经被渗透攻击，漏洞不会自行消失，即渗透继续进行，其表现形式应当维持原样）。

此刻，执行体的脆弱性仿真模型已经建立完毕，系统状态如下表所示：

表格 1 系统状态

情况	表现形式	系统状态	期望的预期操作
1	“0-0-0”	正常	定时器加1,超过阈值随机清洗
2	“1-0-0”“2-0-0”	差模攻击,可定位故障执行体	清洗输出结果不一致的执行体,恢复到正常状态
3	“1-1-0”“1-1-1”、“1-1-2”“0-1-2”	差模攻击,未能定位故障执行体	清洗输出胜任系数最低的执行体,逐次清洗其他受攻击的执行体
4	“0-2-2”“1-2-2”	N-1模攻击	清洗输出结果不一致的执行体,进入“清洗后仍未恢复状态”;然后清洗输出结果一致的其中一个执行体,退化到差模攻击状态
5	“2-2-2”	共模攻击	定时器加一,超过阈值随机清洗

4.2 实验设置

本文的实验仿真平台为matlab 2017.b。根据文献［9］基于美国国家脆弱性数据库（National Vulnerability Database， NVD）对11种操作系统的漏洞分析，来自于相同家族操作系统（如Windows 2003与Windows 2008）之间的共模漏洞数量会比较多，而来自于不同家族操作系统（如BSD-Windows）之间的共模漏洞很低，对于一般攻击而言，其异常输出矢量一致的比例可以设置为一个合理的较小值1e-4［10］。根据第二节所述的有限异构资源条件，为5个执行体的脆弱性仿真模型赋予高/中/低三档的仿真参数，如表格2所示。

在不同的攻击强度下，设置初始置信度4，检测到异常的减分差值D为1。随机模拟仿真1000次，每次模拟仿真包含10000次迭代。统计不同仿真参数条件下，拟态调度器干预下的运行结果。

4.3 实验结果分析

最多只需要三次清洗，即可将系统恢复到

表格 2　执行体脆弱性仿真模型(高/中/低强度攻击)

对象	差模漏洞触发概率 P_{diff}	共模漏洞触发概率 P_{comm}	执行体初始胜任系数
执行体1	0.005/0.002/0.001	1e−4/5e−5/1e−6	0.9
执行体2	0.01/0.005/0.002	5e−4/1e−4/5e−5	0.8
执行体3	0.02/0.01/0.004	5e−3/1e−3/1e−4	0.7
执行体4	0.03/0.01/0.008	5e−3/2e−3/2e−4	0.95
执行体5	0.03/0.015/0.01	1e−2/2e−3/5e−4	0.65

正常状态

图 3 给出了高强度攻击下的状态转移图细节。在第 5 到 6 个时钟周期间，执行体 2 被触发了 1 次漏洞。系统经过积分累计在 2 个时钟周期内将其置信度降低到 0 并清洗，系统输出正常，攻击无感移除；在第 56 个时钟周期，执行体 2 与执行体 3 同时被触发了共模漏洞，系统进入 N-1 模攻击状态。系统判定首先判定执行体 1 出现问题，经 56、57 两个时钟周期进行积分累计将执行体 1 清洗下线并更换另一个执行体；58、59 两个时钟周期积分累计发现清洗后状态仍未恢复，做出执行体 3 下线的决策，使得系统在第 60 个时钟周期退化到差模攻击状态。在时钟周期 80 至 110 则呈现了受定时机制保护下的 N 模攻击恢复状态转移。由于引入了执行体上线保护寄存器 C_{reg}，即使遭受 N 模攻击，系统也只需要三次清洗操作即可恢复正常。

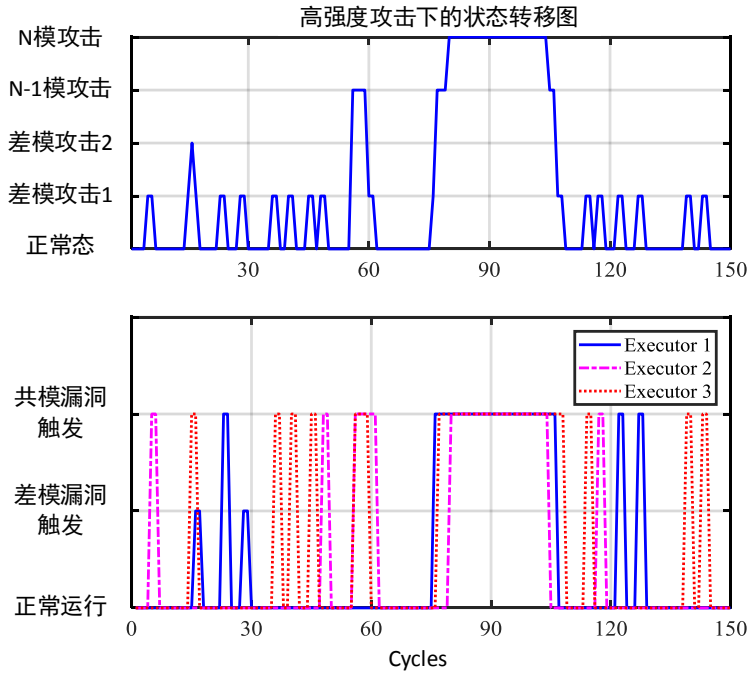


图 3　高强度攻击下的状态转移图

具备环境自适应特性的执行体上线替换机制

表格 4 给出了不同攻击强度下执行体上线概率情况。本文假设可用的执行体只有 5 个，且不同执行体在抗攻击性能上具有差异。在高攻击强度下，系统更倾向于根据执行体的实际表现情况选择对系统环境适应性更强（表现为漏洞分布更少，更难被触发）的执行体上线。如表格 4 所示，在高强度攻击模型下，执行体 1 上线的概率要比执行体 5 高 7.8%。在低强度攻击仿真条件下，由于各执行体表现差异不大，所以上线概率呈现平均分布的现象。值得注意的是，执行体的初始胜任系数是凭借人工经验设置的，该值可以让执行体在系统的初始阶段获得更高的上线优先级，但仍会在系统调度的迭代过程中不断更新。尽管在实验参数中将执行体 4 的初始胜任系数设置为 0.95，但在高强度攻击模型下，该执行体的上线概率依然与执行体实际漏洞后门的分布情况紧密相关。综上分析表明，所提出的基于置信度与胜任系数的拟态调度方法，其执行体上线替换机制具备环境的自适应特性。

在高强度攻击下，系统仍可保持 92.88% 的

表格 4　不同部攻击强度下执行体上线概率

攻击强度	执行体1	执行体2	执行体3	执行体4	执行体5
高强度攻击	63.05%	59.99%	62.29%	59.42%	55.25%
中强度攻击	60.86%	61.29%	60.03%	57.79%	60.03%
低强度攻击	59.81%	60.78%	59.63%	59.07%	60.71%

高可用概率

表格 3给出了不同攻击强度下系统的平均清洗概率与平均可用概率。其中，平均清洗概率由执行体清洗下线次数除以总迭代次数得出；平均可用概率为系统处于正常状态（case 1）与可定位故障执行体的差模攻击状态（case2），两种状态在系统运行过程中所有状态的比重。可见在高强度攻击下，尽管系统只有有限的5个可用执行体，系统平均可用概率依然可以达到92.88%的可用概率；而在低强度攻击下，系统可用概率高达99.48%。实验数据表明，所提出的基于置信度与胜任系数的拟态调度方法，能为有限异构资源条件下的拟态系统构造提供了高效的解决方案。

表格 3　不同攻击强度下系统的平均清洗概率与平均可用概率

攻击强度	平均清洗概率	平均可用概率
高强度攻击	10.42%	92.88%
中强度攻击	4.73%	98.38%
低强度攻击	2.98%	99.48%

5　结束语

针对当前应用生态局限性的约束条件，本文提出了有限异构资源条件下的拟态调度方法，并且基于数值仿真的方案建立了安全评估仿真模型。实验结果表明，所提出的拟态调度策略，最多只需要三次清洗，即可将拟态系统从N-1模攻击状态、N模攻击状态恢复到正常状态；执行体上线替换机制具备环境的自适应特性；即使在高强度攻击下，系统仍可保持92.88%的高可用概率。

拟态防御理论从架构上实现了设备的内生安全，其安全效果可量化设计，可验证度量。本文从节约成本、简化拟态调度策略的角度出发，评估了当前应用最广泛的三余度DHR架构在有限异构资源条件下系统可用概率，给拟态系统设计的成本规划提供了新思路。

参考文献:

[1] Hu H, Wu J, Wang Z, et al. Mimic defense: a designed-in cybersecurity defense framework [J]. IET Information Security, 2017, 12(3): 226-237.

[2] 沈丛麒, 陈双喜, 吴春明, 基于信誉度与相异度的自适应拟态控制器研究[J]. 通信学报, 2018, 039(0z2):173-180.

[3] Senjie L, Qinrang L, Yiteng W, et al. A self-adaptive timeout mechanism in Mimic Defense System [C]//2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS). IEEE, 2017: 588-591.

[4] Ma B , Zhang Z . Security research of redundancy in mimic defense system [C]// 2017 3rd IEEE International Conference on Computer and Communications (ICCC). IEEE, 2017.

[5] 陈利跃,孙歆,吴春明,等. 一种基于异构度的拟态执行体调度模型研究[C],第一届"先进计算与防御技术"会议,2018

[6] 马海龙,伊鹏,江逸茗,等. 基于动态异构冗余机制的路由器拟态防御体系结构[J], 信息安全学报,2017, 2(1).

[7] 魏帅, 于洪, 顾泽宇,等. 面向工控领域的拟态安全处理机架构[J]. 信息安全学报, 2017, 2(1).

[8] 赵长海,晏海华,金茂忠. 基于编译优化和反汇编的程序相似性检测方法[J]. 北京航空航天大学学报,2008(06):711-715.

[9] Garcia M, Bessani A, Gashi I, et al. Analysis of OS diversity for intrusion tolerance [J]. Software: Practice and Experience (accepted for publication), 2013.

[10] 邬江兴,网络空间拟态防御原理:广义鲁棒控制与内生安全[M]. 北京:科学出版社,2018. 11.

Secure-aware and QoS-guaranteed Heterogeneous Controller Placementfor Software-Defined Networking

Yi Peng[a,b], Hu Tao[a*], Hu Yuxiang[a], Lan Julong[a], Zhang Zhen[a], Li Ziyong[b]

1.aNational Digital Switching System Engineering and Technological Research Center, China;

2.bInformation Engineering University, China

Abstract: As the large-scale application of software-defined networking (SDN), one major research challenge for SDN is to reasonably place multi-controllers. The researchers have proposed many solutions for the controller placement problem (CPP). However, they usually deployed homogeneous controllers in the network and ignored the homogeneous controller common-mode fault caused by this. To fill this gap, this paper first defines a Heterogeneous Controller Placement Problem (HeCPP), considering the controller heterogeneity and QoS criteria (delay, controller utilization, and controller fault rate). Correspondingly, we propose a **S**ecure-aware and **Q**oS-guaranteed **H**eterogeneous **C**ontroller **P**lacement (**SQHCP**) approach to solve HeCPP effectively. Specifically, SQHCP consists of two steps: Step 1 computes the required heterogeneous controller types and numbers based on dynamic planning to maximize the security of the control plane. Then, adopting the "Divide and Conquer" strategy, step 2 partitions the network into several subnets based on the K-means algorithm and improves the genetic algorithm to optimize the heterogeneous controller placement for each subnet to guarantee QoS. The theoretical analysis demonstrates the feasibility of SQHCP, while simulation results show that SQHCP not only enhances the security of the control plane but also outperforms the existing approaches in terms of controller numbers, delays and load balance.

Key words: : Software-Defined Networking; heterogeneous controller; controller placement; heuristic algorithm

1 Introduction

Software-Defined Networking （SDN）， as a new network paradigm， abstracts the network control function from the distributed data plane， and all control functions are coupled into a logically centralized entity， which is called the controller ［1］. With the help of the above characteristics， SDN transforms the Internet into a more programmable， configurable， and manageable infrastructure. Nowadays， SDN has been widely used in many real-world networks such as Google B4 ［2］. With the scale expanding of the SDN， the single controller is difficult to meet the requirements of delay and flow requests in the network. In other words， the single controller is not enough to manage the large-scale network. Besides， the physically centralized controller may encounter a single point of failure， which degrades the reliability of SDN significantly. Once the single controller breaks down or suffers from the performance bottleneck， the SDN advantages will be lost ［3］.

In recent years， researchers have proposed several distributed multi-controller frameworks： Kandoo ［4］， Onix ［5］， DISCO ［6］， just to name a few. The multi-controllers manage the network in a logically centralized but physically distributed way， which improves the scalability and fault tolerance of the SDN effectively ［7］. Along with the increase of the number of controllers in the network， a new set of problems arises， the most important of which is how to place multiple controllers reasonably. Therefore， many researchers have introduced various solutions for the controller placement problem （CPP） ［8］. In fact， in terms of solving CPP， simply optimizing the number of control-

Foundation item: This work is supported by the National Natural Science Foundation of China (Grant No. 61802429, 61872382) and National Network Cyberspace Security Project (Grant No. 2017YFB0803201).

lers is inefficient， while both the controller

locations and mapping relationships also affect the network performance dramatically. As first defined in Heller et al. ［9］， the objectives of CPP are to determine the controller numbers and locations so as to minimize the average propagation latency and worst-case propagation latency. Yao et al. ［10］ introduced capacity-aware CPP. Several researchers consider multiple criteria （e.g.， delay， link utilization， controller loads） to address CPP more comprehensively ［11］［12］.

Nowadays， considerable research efforts have been made to address the CPP， and the existing studies generally place the homogeneous controllers in the SDN， which means that the entire network is managed by the controllers with the same type. However， there are great potential security threats in the SDN control plane composed of the homogeneous controllers （see Section 2.1 in detail）. Since all homogeneous controllers have the same functions， any possible design vulnerabilities will be reflected in the identical components of homogeneous controllers ［13］. If the attackers have mastered one vulnerability in the controller， they can exploit this vulnerability to attack all homogeneous controllers with the same type， which damages the entire control plane. Therefore， we define the homogeneous controller common-mode fault as follows.

Definition 1: Homogeneous controller common-mode fault

. In the multi-controller SDN network， if a controller fails due to one or more design vulnerabilities， all homogeneous controllers with the same type as this controller will be threatened in the same way.

Although the researchers have proposed good solutions to CPP ［8］， they may underestimate or even ignore the homogeneous controller common-mode fault. If the attackers exploit the same vulnerability to activate the common-mode fault， the control plane constructed by the homogeneous controllers will be destroyed no matter how to optimize controller placement. Besides， as a special network operation system ［14］， the controller is essentially a software product， and there are inevitable vulnerabilities in the process of its development and design ［15］.

With the basic axiom that there are usually multiple solutions to the same problem and multiple implementation structures for the same function ［16］， the heterogeneous controllers developed by different vendors and research institutions can effectively address the homogeneous controller common- mode fault （see Section 2.2 in detail）. In light of this observation， we first consider placing the heterogeneous controllers to cope with the above security threat faced by the SDN control plane. Specifically， we not only formally define a Heterogeneous Controller Placement Problem （HeCPP） but also propose a **S**ecure-aware and **Q**oS-guaranteed **H**eterogeneous **C**ontroller **P**lacement （**SQHCP**） approach to solve it. To the best of our knowledge， this paper is the first work for studying heterogeneous controller placement in the SDN. The main contributions of this paper are listed as follows：

- We take examples to analyze the homogeneous controller common-mode fault in the existing CPP solutions and elaborate the effectiveness of heterogeneous controllers for enhancing control plane security through testing the four mainstream controllers.
- We define the HeCPP， considering end-to-end delay， controller utilization and controller fault rate under the heterogeneous controller condition. We also formulate the objective functions of HeCPP.
- We propose SQHCP approach， including two steps， for solving HeCPP. Step 1 computes the required heterogeneous controller types and numbers based on dynamic planning. Step 2 partitions the network into several subnets based on the K-means algorithm and then improves genetic algorithm （GA） to optimize the heterogeneous controller placement for each subnet.
- We theoretically analyze the feasibility of SQHCP through several theorems. Meanwhile， the

simulation results show SQHCP outperforms the

representative controller placement approaches under various topologies， which greatly enhances the security of the control plane and guarantees QoS.

The rest of the paper is structured as follows. Section II gives the motivation of this paper. HeCPP is formulated in Section III. The proposed new approach is described in detail in Section IV. Section V evaluates the performance of our proposal. Finally， Section VI shows the related work. Section VII concludes this paper and provides some future research directions.

2 Motivation

In this section， we firstly introduce the homogeneous controller common-mode fault in the existing CPP solutions. Then， we illustrate the effectiveness of heterogeneous controllers for enhancing control plane security through testing the current mainstream controllers.

2.1 Homogeneous controller common-mode fault

Multi-controller ameliorates the resilience and fault tolerance of the SDN network， and the existing CPP solutions generally consider that the control plane is constructed by the homogeneous controllers. However， this homogeneous controller placement way poses serious security threats. For example， if the attackers have mastered the vulnerability of one controller， they can exploit this vulnerability to launch the malicious attacks， such as message leak， disconnection with switches， or controller crash ［15］. To make matter worse， due to the homogeneity of the controller placed， the other controllers will suffer from the same attack in the network.

For example， we set up an SDN network placing three controllers， as shown in Fig. 1. All controllers are homogeneous （e.g.， Floodlight）， each of which manages the corresponding SDN subdomain. When the attackers have detected that the controller type is Floodlight as well as mastered one vulnerability （CVE-2014-23041） in the Floodlight， they can exploit the vulnerability to crash Floodlight controller. No matter how many Floodlight controllers are placed as backups in the control plane， the attackers can still easily destroy them by exploiting this vulnerability. Finally， all Floodlight controllers fail in the same way， and the homogeneous controller common-mode fault damages the entire control plane.

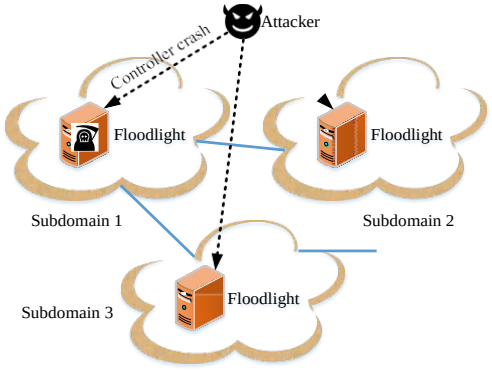


Fig. 1 Homogeneous controller common-mode fault

2.2 Heterogeneous controller security

As the core component of the SDN， the controller is a "brain" of the whole network. Nowadays， most controllers are static and have poor abilities to defend against probing attacks ［17］. As discussed earlier， the existing CPP solutions place homogeneous controllers in the SDN， all of which have the same

1 CVE-2014-2304 is an exposed vulnerability during OpenFlow protocol processing for Floodlight 0.9 version.

vulnerabilities and are incapable of coping with the common-mode fault threat. What calls for special attention is that all controllers are software products that inevitably have vulnerabilities in their designs ［15］. Actually， the common-mode fault is not unique to SDN controllers but widely exists in industrial productions and industrial control systems ［18］， while the most general way to solve this is the principle of the heterogeneity ［19］. The probability of the common-mode fault can be effectively reduced by adopting the multiple heterogeneous components with equivalent functions. Inspired by this， we consider that the key to solving the homogeneous control-

ler common-mode fault is heterogeneous controller placement. Fortunately, as SDN technology evolves, there are more than 30 types of controllers, which are developed by different vendors/research institutes and programmed in different languages [20]. Therefore, these controllers with diversified technical features are sufficient to meet the requirement of the heterogeneity in the controller placement.

In fact, each type of SDN controller (e.g., NOX, Ryu, Floodlight) can complete network control and management tasks independently, which means that they are equivalent in functions. It's undeniable that all heterogeneous controllers have various vulnerabilities. However, the heterogeneous environments and design modes (e.g., NOX, Ryu, and Floodlight are programmed in C++, Python, and Java, respectively) make the trigger mechanism for each type of controller vulnerability be different. Theoretically, heterogeneous controllers are independent of each other and have no dependencies during runtime. Thus, it is difficult for the attackers to make all heterogeneous controllers show abnormal behaviors at the same time with the same attack method.

In light of the above understanding, we propose for the first time to incorporate the heterogeneity into the controller placement to address the homogeneous controller common-mode fault in the existing CPP solutions. To this end, this subsection tests several controllers with different types and explains that heterogeneous controllers can indeed weaken the homogeneous controller common-mode fault. Further, the test results also provide a reference for the heterogeneous controller placement. Several researchers have evaluated various open-source controllers, but they only focus on controller performance [21] [22] (e.g., Packet_In response time, flow setup time, throughput). Distinguished with existing studies, we compare the security characteristics of four mainstream controllers. The details are described as follows.

Test objects. The tested controllers cover NOX (developed by C++, single thread), Ryu (developed by Python, single thread), Floodlight (developed by Java, multiple threads), ONOS (developed by Java, multiple threads).

Test environments. The testbed consists of two servers (Intel E3-1230, 8cores, 2.3GHz, 32GB RAM, Ubuntu 16.04), which are connected with one link (10Gbps). One server uploads controller and Mininet, while the other server generates traffic based on specific test scenarios. Two separate servers can ensure that traffic generation does not affect controller performance. We use hcprobe tools to test the SDN controller and record test results.

Test methods. To evaluate the security of the heterogeneous controllers more generically, we check their performances under incorrect OpenFlow messages. This is because that the attackers usually attack the controller by means of OpenFlow protocol. Concretely, we send the following incorrect OpenFlow messages to attack the controllers: 1) incorrect message length. Packet_In message with non-normative length field is generated in the OpenFlow header; 2) invalid OpenFlow protocol version. All tested controllers only support OpenFlow 1.2. OpenFlow v1.2 is declared when the switches and controller establish communication, but Packet_In message is set up an invalid OpenFlow protocol version; 3) incorrect OpenFlow message type. The value of OpenFlow header field does not correspond to the

encapsulated OpenFlow message type; 4) incorrect protocol type. We put the code of the ARP protocol (0x0806) in the Ethernet header.

Test results. The security of the controller can be categorized into four grades (A, B, C, D) according to its performance: A) the controller defenses the attack effectively and runs normally; B) the controller fails to identify the attack and there is a potential vulnerability, but it still works normally; C) the controller disconnects with switches; D) the controller crashes. We run experiments many times to avoid random errors. The results are shown

in Table 1.

Table 1 The security test results of NOX, Ryu, Floodlight, ONOS

	Incorrect OF message length	Invalid OF protocol version	Incorrect OF message type	Incorrect OF protocol type
NOX	D	A	B	B
Ryu	A	A	C	B
Floodlight	C	B	A	B
ONOS	A	B	B	D

Based on the above test results, we can conclude that the same malicious attack only threatens certain type of controller, while other types of controllers are unaffected or less vulnerable. For example, when attacked by incorrect OF message length, NOX crashes, while Ryu and Floodlight are not affected. Therefore, the reasonable combination of the heterogeneous controllers can effectively eliminate the homogeneous controller common-mode fault and improve the security of the SDN control plane.

3 Problem formulation

In this section, we will introduce a model of network environment and then formulate HeCPP.

3.1 Network environment

Here, we consider an SDN backbone network and characterize it based on graph theory [11]. SDN network represents an undirected graph $G = (V, L)$, where V is a set of N nodes and L is a set of bidirectional physical links between nodes. Each node vi corresponds to a switch si (vi and si are used interchangeably in this paper). S is a set of switches and $|S| = |V| = N$. $h(vi, vj)$ indicates the hop count between any two nodes vi and vj. Binary variable lij is the connecting relation between vi and vj, where $lij = 1$ represents there is a direct connection between vi and vj, as shown in Eq. (1). In this paper, we assume that that the network traffic tends to be stable in the SDN backbone network [23], and there is

little difference in the traffic change in each period. Besides, requests generated by every local switch si follow Poisson distribution at the rate of λi [24].

$$lij = \begin{cases} 1 & \text{vi directly connects withvj} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

SDN controllers can be placed on any network nodes. Unlike the traditional CPP solutions, we consider placing the heterogeneous controllers, which means that controllers placed in the network have

multiple types. C is a set of M controllers. Binary variable $ck(u)$ indicates the heterogeneous controller

placement, where $ck(u) = 1$ denotes the controller with type u is placed on node vk, as shown in Eq. (2). Parameter u is set according to the set of types of available heterogeneous controllers H. For example, $u = 1$ and $u = 2$ represents NOX and Ryu, respectively. The total number of the heterogeneous controller

types is $U = H$, and $NC(u)$ is the number of the heterogeneous controllers with type u in the network.

Since the controllers with different types have different flow request processing capabilities, $\omega k(u)$ is the

capacity of the controller $ck(u)$. Note that controllers with the same type have equal capacities. In order to be more consistent with the real network and reduce the placement costs, we set each switch to be

controlled by only one master controller [7]. So we introduce binary variable *xik* to represent the mapping relationships between switch and controller, where *xik* = 1 denotes the switch on node *vi* is managed by the controller on node *vk*, as shown in Eq. (3).

$$\begin{cases} c(u) = 1 \\ k \quad 0 \end{cases}$$

controller withtype u is placed onvk otherwise

(2)

xik

$$\begin{cases} 1 & switchon\ v_i\ is\ managed\ by\ controller\ onv_k \\ & otherwise \end{cases}$$

(3)

As discussed above, CPP is complex, especially for considering the heterogeneous controllers. In this paper, for the sake of QoS and security, we consider the following factors during formulating the heterogeneous controller placement.

1) End-to-end delay

In the SDN network, end-to-end delay is one of the important criteria for evaluating QoS. The lower the end-to-end delay, the better QoS performance. Specifically, the end-to-end delay *Dik* (*u*) between switch *vi* and controller *ck* (*u*) defines the time experienced from arriving a new flow to receiving the

Packet_Out message responded by the controller, as shown in Fig. 2.

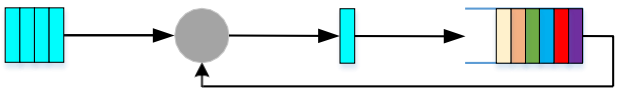


Fig. 2 new flow processing in SDN

New Flow Switch Flow Request ControllerPacket_Out

Therefore, the end-to-end delay generally consists of four parts: switch processing delay, packet transmission delay, packet propagation delay, and controller processing delay. As hardware technology develops, high-performance switches have been widely used (e. g., Pica8 [25] and Cisco [26]), producing negligible switch processing delays. Furthermore, in a high-speed network (e.g., backbone network), the

packet transmission latency is trivial [23]. The packet propagation delay *DPik* (*u*) defines the hops

between switch *vi* and controller *ck* (*u*), as shown in Eq. (3).

$$DPik(u) = h(vi, vk)$$

(3)

As shown in Fig. 2, the interactions between controller and switches can be regarded as an independent M/M/1 queuing model, where the controller and flow requests sent from switches are considered as the service counter and customers, respectively. Based on Little's Law, the average waiting time can be calculated as the inverse of the difference between the processing rate of service counter and

the arrival rate of customers. For switch *si*, the processing delay of controller *ck* (*u*) is the inverse of the

difference between controller capacity ωk (*u*) and flow request rate λi, as shown in Eq. (4),

DCik

$$(u) = \omega$$

1

$$(u) - \lambda$$

(4)

k i

In multi-controllers SDN, the controller not only processes the flow request sent from switches but also communicates with the other controllers to synchronize the global network view. In order to lower

communication overheads, only topology change events (e.g., switch on or off) should be synchronized between controllers. In fact, frequent topology change events are almost impossible in a backbone topology considered in this paper [5]. Therefore, the number of events related to topology changes is limited. To calculate the effect of state synchronization between controllers on network performance, we consider the worst-case scenario in which each controller communicates with all remaining controllers periodically. The controller communi-

cation overheads are proportional to M^2 (M is the total number of

heterogeneous controllers placed in the network). Concretely, controller synchronization cost $Qsyn(u)$ is

modeled as the product of M^2 and synchronization factor $\alpha(u)$, as shown in Eq. (5), where $\alpha(u)$ is related to the types of heterogeneous controllers.

$$Qsyn(u) = \alpha(u) \cdot M^2 \quad (5)$$

Since the controller synchronization also consumes a certain amount of controller capacity, the controller processing delay is recalculated in Eq. (6).

$$DCik(u) = \omega_1(u) - Q(u) - \lambda(u) \quad (6)$$

k syn i

Based on the above, the end-to-end delay $Dik(u)$ between switch vi and controller $ck(u)$ is shown in Eq. (7),

$$D(u) = DC(u) + 2 \cdot DP(u) = 1 + 2 \cdot h(v, v) \quad (7)$$

ik ik ik

$$\omega(u) - \lambda ik$$

k i

$\pi k(u)$ defines the sum of the end-to-end delay between the controller $ck(u)$ and switches managed, as shown in Eq. (8). Furthermore, the network average end-to-end delay π can be calculated in Eq. (9).

N

$$\pi k(u) = \sum_{i=1} D_{ik}(u) \cdot x_{ik} \quad (8)$$

$$\pi = 1/U \sum_{k=1}^{U} \pi k(u) \cdot c_k \quad (9)$$

(u) 2) Controller utilization rate

$u=1$ $k=1$

Controller utilization rate denotes the ratio of controller loads to controller capacity, where controller loads consist of two parts: processing flow requests and synchronizing states. Actually, the controllers with different types have different capacities, and the controller utilization rate has a significant effect on the performance of SDN. When the controller utilization rate is too high, the controller cannot handle the flow requests sent from the switches normally, resulting in packet loss, network congestion, and even controller crash. Instead, when the controller utilization rate is too low, the controller resources are idle and wasted. Ideally, QoS of the SDN network reaches optimal if and only if all heterogeneous controllers have the same utilizations. Specifically, for the controller with type u on

the node vk, the controller utilization rate $\eta k(u)$ is computed in Eq. (10).

$$\eta(u) = \frac{\sum_{i=1}^{N} x \cdot \lambda + \alpha(u) \cdot M^2}{\omega_{ik} k(u)} \quad (10)$$

Based on Eq. (10), the average value of all heterogeneous controller utilization rates is computed in Eq. (11). Further, CUV denotes the variance of heterogeneous controller utilization rates, as shown in Eq. (12).

$$\left(\frac{1}{M} \sum_{k=1} M \eta \right)^{k} \eta(u) \quad (11)$$

⎭

（11）

3） Controller fault rate

$$CUV = \frac{1}{M}\left(\sum_{k=1}^{M}(\eta(u) - \eta)^2\right)$$

（12）

In this paper， we pay close attention to the controller faults caused by controller vulnerabilities， while the controller faults caused by physical failures are beyond the scope of this paper. This is because physical failures are related to a variety of factors [7] （e.g.， device aging， natural environment）. It should be noted that the controller fault rate depends on not only the security of the controller itself but also the attackers. Specifically， the security of the controller itself is related to the number of vulnerabilities existed. Generally speaking， the lower the security of the controller itself， the higher the controller fault rate. Not only that， if the attackers prefer a controller that has certain type or is in critical location， and this controller will be attacked more frequently and more prone to fault [15]. Based on the above analysis，

for the controller with type u on the node vk， the controller fault rate is represented as $PCFk(u)$， as shown

in Eq. （13），

$$PCFk(u) \propto (CF(u), MAk)$$

（13）

where $CF(u)$ is the fragility coefficient of the controller with type u， and MAk is the malicious attack coefficient.

The value of $CF(u)$ depends on the number of controller vulnerabilities and the attackers' prior knowledge of controller with type u. In general， the more the controller vulnerabilities $VU(u)$， the higher $CF(u)$， as shown in Eq. （14），

$$CF(u) \propto VU(u)$$

（14）

Although we consider placing heterogeneous controllers in the network， they may still have some homogeneities in actual design. For example， both Floodlight and ONOS controllers are programmed in Java language. After attacking one type of controller successfully， the attackers can utilize the prior

knowledge obtained in the attack process to guide the subsequent attacks. Here， $\zeta(u)$ defines the prior

knowledge coefficient for attacking the controller with type u， as shown in Eq. （15），

$$\zeta(u) = \max\left\{\frac{VU(u, u')}{|VU(u')|}\right\}, \quad u' \in \Omega$$

（15）

where $VU(u')$ represents the total number of known vulnerabilities of the controller with type u'， $VU(u, u')$ represents the number of common vulnerabilities between the controllers with type u and type u'， and Ω is set of controllers compromised by the attackers. Specifically， $\zeta(u) \in [0, 1]$.

Based on Eq. （14） and Eq. （15）， $CF(u)$ can be computed in Eq. （16），

$$CF(u) = e^{\zeta(u)} \cdot VU(u)$$

（16）

Malicious attack coefficient MAk is related to the controller locations， because the attackers always regard the controllers placed on the critical nodes in the network as their attack targets to maximize attack gains. Referring to the concept of node degree of the complex network， we compute MAk in Eq. （17），

$$MAk = \frac{\sum_{i \in N} lik}{N-1}$$

（17）

Therefore， combining Eq. （13）， Eq. （16） and Eq. （17）， the controller fault rate $PCFk(u)$ is computed

in Eq. （18），

$_{k}PCF$ （u） = 1- e-CF （u）·MAk

= 1- e

-$e\zeta$ （u）

$$\left(\sum \cdot VU\ (u) \cdot \left| l_{ik} \right|\right)_{i\in N}$$

（18）

Accordingly， the control plane fault rate $PCPF$ is calculated in Eq. （19），

$$PCPF = \prod_{k\in M,\ u\in H} PCFk\ (u)$$

（19）

Since Eq. （19） is not linear， we should linearize it to facilitate calculation. From the perspective of the monotonicity， the control plane fault rate $PCPF$ is proportional to the single heterogeneous controller

fault rate PCF （u） ， and $PCPF \propto PCF$ （u） . Based on Eq. （18），we can get PCF （u） = 1- $e^{-CF\ (u)\ \cdot MA}k$ ， so

k k k

$PCPF \propto$ （CF （u）· MAk） ， as shown in Eq. （20），

$$PCPF \propto \prod_{k\in M,\ u\in H} \left(e^{\zeta\ (u)} \cdot VU\ (u) \cdot \left(\sum l \right)_{i\in N} \right)_{ik}$$

（20）

We implement the logarithm operation on Eq. （20）， and it can be further linearized to Eq. （21）.

$$\ln PCPF \propto \sum_{k\in M} \sum_{u\in H} \left(\varsigma\ (u) + \ln\ (VU\ (u)) + \ln \left(\sum_{i\in N} ik \right) \right)$$

（21）

The main notations used in this paper are summarized in table 2.

Table 2 Notations used in this paper

Notation	Definition
$V = \{vi\}$	Set of network nodes
$L = \{lij\}$	Set of network links
$S = \{si\}$	Set of switches
λi	Flow request rate of switch si
$h(vi, vj)$	Hop between node vi and vj
$C = \{ck\ (u)\}$	Set of heterogeneous controllers placed
$H = \{u\}$	Set of types of heterogeneous controllers
$NC\ (u)$	Number of placed controllers with type u
$\omega k\ (u)$	Controller capacity
xik	Mapping relationship between switch and controller
$Dik\ (u)$	End-to-end delay between controller and switch
$DPik\ (u)$	Packet propagation delay
$DCik\ (u)$	Controller processing delay
$Qsyn\ (u)$	Controller synchronization cost
$\alpha\ (u)$	Controller synchronization factor
π	Network average end-to-end delay
$\eta k\ (u)$	Controller utilization rate
$PCFk\ (u)$	Controller fault rate
$CF\ (u)$	Controller fragility coefficient
MAk	Malicious attack coefficient
$VU\ (u)$	Number of known vulnerabilities of controller with type u
$\zeta\ (u)$	Attacker prior knowledge coefficient for controller with type u
$VU\ (u, u')$	Number of mutual vulnerabilities of heterogeneous controllers
Ω	Set of controllers that the attackers have compromised

3.2 HeCPP formulation

In consideration of realistic requirements from network operators， we propose the HeCPP and define it as follows.

Definition 2: HeCPP

. For a given network topology and set of heterogeneous SDN controllers， how to identify the controller locations， controller types， and mapping relationships between switches and controllers to minimize network end-to-end delay， balance heterogeneous controller utilization， and lower control plane fault rate.

Obviously， HeCPP is a multi-objective optimization problem under multiple constrains， and the objective of HeCPP is formulated in Eq. （22）.

Objective

min

i， k， u

$\{\pi$， CUV， $\ln PCPF\}$

（22）

s.t.

$\sum x \cdot \lambda + \alpha$ （u） $\cdot M^2 \leqslant \beta \cdot \omega$

（u）

$\forall u \in H$

（23）

$\sum_{i=1}^{M} xik \, i \, k$

$k=1$

$\sum_{k=1}^{U} xik = 1$

$\forall i$， $k \in N$

（24）

$\sum c_k$ （u） $= 1$

$u=1$

$U \geqslant 2$

$\forall u \in H$

（25）

（26）

ck （u）， xik， $lij \in \{0, 1\}$

$\forall i$， $k \in N$， $u \in H$

（27）

Eq. （23） indicates that the controller loads cannot exceed controller capacity. Besides， in order to withstand unexpected traffic burst， we reserve a proportion of capacity for each heterogeneous controller， as determined by a traffic burst coefficient β. For example， in a network where the flow requests of switches vary significantly and rapidly， β is set to a higher value. Eq. （24） represents each switch only has one master controller. Eq. （25） shows that only one controller can be placed in the same location. Eq.

（26） indicates that there are no less than two types of optional controllers to guarantee the heterogeneity of the control plane. Eq. （27） denotes the binary variables in the network model.

3.3 Expression of solution

Based on the above formulation， the solution to HeCPP is to determine the locations of the controllers， the types of controllers at these locations， and the mapping relationships between switches and controllers. Therefore， the HeCPP solution is denoted by a two-dimensional matrix $A = [anm]$（$N+1$）$\times M$，

as shown in Eq. （28）.

$$
\begin{bmatrix}
a01 & a02 & a03 \\
a\,a\,a\,...\,a\,...\,a & & \\
11 & 12 & 13 \; 1m \; 1M \\
\vdots & \vdots & \ddots \; \vdots \; \vdots \; \vdots \; \vdots \\
a21 & A = & \\
a22 & & \\
a23 & & \\
a2m & & \\
a2M & & \\
a\,a\,a & & \\
\vdots & \vdots & \vdots \; \vdots \; \vdots \; \vdots \; \vdots \\
nm & nM & \\
a\,a\,a\,...\,a\,...\,a & & \\
N1 & N2 & N3 \; Nm \; NM
\end{bmatrix}
\quad (28)
$$

We explain the matrix A as follows. There are N nodes （$i=1$， 2，...， n，...， N） and M heterogeneous controllers （$k=1$， 2，...， m，...M） in the network. Matrix A has N+1 rows （from row 0 to row N） and M columns. For any element anm， if $n=0$， $a0m=u$（$u \in H$）， which represents the heterogeneous type of the

mth controller. If$1 \leqslant n \leqslant N$， $anm \in \{0, 1, 2\}$， where $anm=0$ and $anm=1$ represent whether there is a mapping

relationship between switch sn and the mth controller， respectively. $anm=2$ indicates that the mth controller

is placed on node vn and also forms a mapping relationship with vn.

The following is a simple illustration example. As shown in Fig. 3， there are 6 nodes in the network， and the heterogeneous controllers include NOX represented as $u = 1$ and Ryu represented as $u = 2$. The solution matrix A of HeCPP under this condition is shown in Eq. （29）. There are two columns in A， which indicates that two controllers are placed. The first column shows that the controller is Ryu， placed on node 2， and manages switches on node 1 and 2. The second column shows that the controller is NOX， placed on node 4， and manages switches on node 3， 4 and 5.

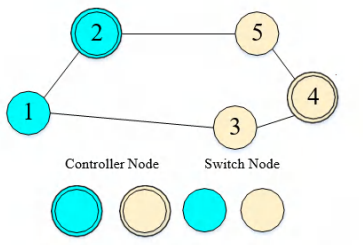


Fig. 3. Heterogeneous controller placement in the SDN

4 SQHCP approach

$$A = \begin{matrix} \lceil 2\ 1 \rceil \\ 1\ 0 \\ \vdots \\ 2\ 0 \\ 0\ 1 \\ \vdots \\ 0\ 2 \\ \vdots \\ \lfloor 0\ 1 \rfloor \end{matrix} \quad (29)$$

Based on the above problem formulation， we focus on the realistic network environment， where the number of heterogeneous controllers to be placed is unknown and different types of heterogeneous controllers have different capacities and security properties ［17］. Combining the objective function （Eq. （20））， we can find that the key to solving HeCPP is to identify the reasonable heterogeneous controller types， controller locations and mapping relationships between switches and controllers. To this end， we propose a Secure-aware and QoS-guaranteed Heterogeneous Controller Placement （SQHCP） approach， including two steps. In the first step， to maximize the security of the control plane， we calculate the required heterogeneous controller types and numbers based on dynamic planning. Further， in the second step， to reduce the complexity and guarantee QoS， we combine K-means and GA to solve HeCPP to determine the locations of the heterogeneous controllers and mapping relationships between switches and controllers.

4.1 Step 1: heterogeneous controller formulation based on dynamic planning

As discussed earlier （Section 2）， through placing the heterogeneous controllers in the network， the homogeneous controller common-mode fault can be avoided effectively. For this purpose， to maximize the security of the control plane， this step reasonably formulates the heterogeneous controllers， which determines the types and corresponding number of heterogeneous controllers to be placed. Note that it will be easier to solve HeCPP when the types and number of heterogeneous controllers are determined in advance.

Firstly， based on the network attack surface， we theoretically analyze the relationship between the types of heterogeneous controllers and security of the control plane， as shown in Theorem 1.

Theorem 1.

The security of the control plane increases with the types of heterogeneous controllers. ***Proof.*** We assume that a set of optional heterogeneous controller types is $H = \{1, 2, 3, ..., u, ...\}$ ， where each element of H represents a type of heterogeneous controller. For example， $u = 1$ represents NOX and $u = 2$ represents Ryu. The set of heterogeneous controller vulnerabilities is $VU = \{VU(1), ..., VU(u), ...\}$ ， where $VU(u)$ is a set of vulnerabilities of the controller whose type is u. Attackers can only launch the

probing attacks on one type of heterogeneous controller at a time. We consider a more practical case

that

different types of heterogeneous controllers cannot realize complete heterogeneity technically. This

means that $VU(u) \cap VU(u') \neq \Phi$ and $VU(u) \not\subset VU(u')$ for any two types (u and u') of the heterogeneous

controllers. Specifically, when placing one type of controllers (equaling to the homogeneous controllers), the attack surface that destroys the control plane completely is $VU(1)$. When placing two types of heterogeneous controllers, the attackers can destroy the entire control plane if and only if mastering their

common vulnerabilities. At this point, the attack surface that destroys the control plane completely is

$$VU(1) \cap VU(2)$$

. Due to $(VU(1) \cap VU(2)) < VU(1)$, the attack surface shrinks. Therefore, the following

conclusions can be drawn. When placing u types of heterogeneous controllers, the attack surface that

$$\bigcap_{u \in H} VU(u)$$

destroys the control plane completely is

$$\bigcap_{u \in H} VU(u)$$

, and

$\bigcap_{u \in H} VU(u) < VU(u)$. Hence, the attack surface shrinks as the heterogeneous controller type increases, and the security of the control plane increases accordingly. The proof ends.

Therefore, to maximize the security of the control plane, each type of heterogeneous controller requires to be placed on the network. Meanwhile, the total number of heterogeneous controllers to be placed needs to be minimized to reduce device overheads and placement costs. Here, we consider that identifying the number of heterogeneous controllers is a special case of the complete knapsack problem [27], where the heterogeneous controllers are regarded as goods and the network loads are regarded as a knapsack.

In virtue of the definition of the complete knapsack problem, the problem of identifying the number

of heterogeneous controllers is described as follows. For given U types of heterogeneous controllers

$\{c(1), c(2), ..., c(U)\}$, their capacities are $\{w(1), w(2), ..., w(U)\}$ and $w(u) = \omega(u) - \alpha(u)$ (considering the synchronization costs), and their values $V(1) = V(2) = ... = V(U) = 1$. The existing knapsack capacity is $W = \sum N\lambda$. The number of heterogeneous controllers of each type is $\{NC(1), NC(2), ..., NC(U)\}$.

Our objective is $\min M = \min \sum_{u=1}^{U=1} NC(u) \cdot V(u)$, which meets constraints $\sum U w(u) \cdot NC(u) \leqslant W$,

$NC(u) \in \{1, 2, ..., W w(u)\}$.

Since the complete knapsack problem is NP-hard, determining the number of heterogeneous controllers also belongs to NP-hardness. In this paper, we solve it with the help of dynamic planning [28]. Dynamic planning is based on the Behrman optimality principle, which is widely used to solve NP problems. The principle of dynamic planning is to divide the original problem into sub-problems, then solves a sub-problem by finding the iterative relationship between the original problem and the sub- problem, and finally achieves the result of solving the original problem. Compared with the brute force search, the complexity of dynamic planning is linear, which improves the solving efficiency.

Sub-problem $m(u, j)$ is described as follows: based on the knapsack capacity j, the minimum value

that u types of heterogeneous controllers can

reach. The state transfer equation is shown in Eq. (30),

$$m(u, j) = \min\{m(u-1, j - NC \cdot w(u)) + NC \cdot V(u), 1 \leqslant NC \cdot w(u) \leqslant j\} \quad (30)$$

The initial iteration condition is shown in Eq. (31),

$$m(1, j) = NC \cdot V(1), \quad 1 \leqslant NC \cdot w(1) \leqslant j \quad (31)$$

The algorithm process is shown in algorithm 1. Line 1-9 computes the required total number of the heterogeneous controllers. Line 10-17 counts the number of the heterogeneous controllers of each type.

The complexity of the algorithm is O (U · W).

Theorem 2.

Algorithm 1 can obtain the minimum number of the heterogeneous controllers.

Proof.

Here, we adopt the proof by contradiction. Firstly, we assume that the optimal solution based on dynamic planning is *Solve* (*NC* (1), ..., *NC* (*U*)). The sub-problem can be described as follows: based on the knapsack capacity *j*, min *M* that *u* types of heterogeneous controllers can reach Therefore, the

Solve (*NC* (1), ..., *NC* (*u*))

. We suppose that

Solve (*NC* (1), ..., *NC* (*u*)) is not optimal, and there is another solution *Solve* (*NC** (1), ..., *NC** (*u*)) for the

sub-problem, which makes *Solve* (*NC** (1), ..., *NC** (*u*)) > *Solve* (*NC* (1), ..., *NC* (*u*)). We can get

Solve (*NC** (1), ..., *NC** (*U*)) > *Solve* (*NC* (1), ..., *NC* (*U*)), and *Solve* (*NC* (1), ..., *NC* (*U*)) is not the

optimal solution, which contradicts the original assumption. Therefore, algorithm 1 can obtain the minimum number of heterogeneous controllers. The proof ends.

Algorithm 1 Heterogeneous controller formulation based on dynamic planning

Input Types of heterogeneous controllers *c* 1 , *c* 2 ,..., *c* *U* Heterogeneous controller capacities {*w* 1 , *w* 2 ,..., *w* *U* } Heterogeneous controller values *V* 1 *V* 2 ... *V* *U* 1
The existing knapsack capacity *W* *N* *i* 1 *i*
Total number of the heterogeneous controllers *M*

Output Number of controllers of each type {*NC* 1 , *NC* 2 ,..., *NC* *U* }

1. *m* 0, {0}
2. *m* , 0 {0}
3. ***for*** *u*=1 to *U*
4. ***do for*** *j*=1 to *W*
5. ***do for*** *NC*=1 to *j* *w*(*u*)
6. ***if*** *j* *w*(*u*)
7. ***then*** *m* *u*, *NC* min *m* *u*, *NC* , *m* *u* 1, *j* *NC* *w* *u* *NC* *V* *u*
8. ***endif***
9. ***endfor***
10. ***return*** *m* *U* , *W* and *M*
11. *u* *U* , *j* *W*
12. ***while*** (*u* 0 && *j* 0)
13. ***do if*** (*m* *u*, *j* *m* *u*, *j* *w* *u* *V* *u*)
14. ***then*** Output *NC* *u*
15. *j* *j* *w* *u*
16. ***else*** *u* *u* 1
17. ***endif***
18. ***endwhile***

4.2 Step 2: heterogeneous controller placement based on improved heuristic algorithms

In section 4.1, we have obtained the total number of heterogeneous controllers (*M*) and the number of heterogeneous controllers of each type *NC* (1), ..., *NC* (*u*), ..., *NC* (*U*). On this basis, we formally solve HeCPP to identify the locations of the heterogeneous controllers and the mapping relationships between

switches and controllers.

It is clearly seen that HeCPP is a typical multi-objective optimization problem from Eq. (22), and the objectives include minimizing the network end-to-end delay, controller utilization variance,

and control plane fault rate. In fact， HeCPP is a specific type of CPP. A lot of researchers have proved that CPP is NP-hard ［9］［10］， so HeCPP is also NP-hard. Because of high complexity and time consumption， traditional solutions （e. g.， Newton method， gradient descent method） and tools （CPLEX， Lingo） are hard to meet the requirements of solving HeCPP. In recent years， evolutionary computation has been applied to solve all kinds of NP problems， including CPP ［29］［30］. Among all evolutionary computation algorithms， we prefer genetic algorithm for two reasons： 1） plenty of previous researchers have demonstrated that GA is widely used to effectively solve multi-objective optimization problems in the controller placement； 2） the solution to HeCPP in the form of the binary array can be easily represented by chromosomes of GA. However， it is impractical to directly adopt GA to solve HeCPP in the large- scale SDN network. Because it requires enlarging the population sizes and iteration times of GA， which greatly increases the running time of the algorithm.

Therefore， in order to solve HeCPP efficiently， we combine the clustering algorithm and genetic algorithm and introduce the heterogeneous controller placement based on improved heuristic algorithms. In line with the “Divide and Conquer” strategy， we first partition the entire network into M subnets， where M is the number of the heterogeneous controllers obtained in Section 4.1. Then， we improve GA to solve the HeCPP of each subnet. Finally， the HeCPP of the whole SDN network is well solved. The benefit of

our proposal is elaborated as follows. For a network with N nodes， the complexity of the search space is O （$2N$）. After partitioning the network into several subnets based on the clustering algorithm， each subnet owns N/M nodes averagely， and the complexity of the search space reduces to O （$M \cdot 2N/M$）. Next，

we will introduce our proposal in detail below.

4.2.1 Network partition based on K-means clustering

Network partitioning is applied in large-scale networks in general， while the clustering algorithm is a common network partitioning method. Nowadays， several researchers have proposed applying the clustering algorithm for the SDN ［31］. Inspired by the existing research work， we adopt K-means clustering to partition SDN into M subnets， as shown in Algorithm 2. Algorithm 2 takes the hops between nodes as the criterion to implement clustering for nodes in the network， where the hops are calculated by Dijkstra algorithm. We describe the process of algorithm 2 as follows. Line 1-3 computes the sum of

$\cap$ $\cap$ hops from each node to all other nodes and selects the node $c1$ with the minimum sum of hops. Line 4 gets the node $c2$ that has the maximum hops to node $c1$. Line 5-11 selects nodes $c1$ and $c2$ as the initial centers and assigns relevant nodes to them. Specifically， for each node $v \in V$， we calculate the hops to the two initial centers respectively， labeled as h （v， $c1$） and h （v， $c2$）. If h （v， $c1$） $< h$ （v， $c2$）， nodes v and $c1$ are fall into the same clustering. If node v is assigned to one center， then $c1$ in this clustering will be recalculated based on Line 3. This process is repeated until the network with N nodes is divided into M subnets. The algorithm complexity of the whole process is O （$M \cdot N$）. Moreover， the effectiveness of K- means clustering has been widely demonstrated in ［40］.

Algorithm 2Network partitioning based on K−means clustering	
Input:	network topology G V, E Number of subnets M
Output:	Network partitioning results G M G M V , E k 1 k k 1 k k

4.2.2 Heterogeneous controller placement based on improved GA

After the network has been partitioned into M subnets， we improve the genetic algorithm to solve the HeCPP for each subnet. Based on the literature ［30］， we describe the implementation of GA in Fig.

1. Compute hops $h(vi, vj)$ between any two nodes, $\forall vi, vj \in V$

N

2. $\forall vi \in V$, $SUM = \sum h(vi, vj)$

j=1

3. $c1$ = arg min SUM, get $Cluster1$

4. $c2$ = arg max $h(c1, c2)$, get $Cluster2$

5. $v \in V$, compute $h(v, ci)$ and $h(v, cj)$

6. ***if*** $h(v, ci) < h(v, cj)$ $\forall i, j \in \{1, 2, ..., M\}$

7. $v \in Clusteri$, update $Clusteri$

8. ***endif***

9. ***if*** $h(vm, v) = \min SUM$, $\forall m \in size(Clusteri), v \in clusteri$

10. $c' = v$

i m

11. ***endif***

12. Repeat Line 4–11 until G is divided into M subnets

4， and the key elements include coding， computing fitness value， genetic operation （selection， crossover， mutation）. In addition， in order to find the optimal solution， we set the following three heuristic rules for

HeCPP： 1） each subnet Gk = （Vk， Ek） only places one heterogeneous controller， but the heterogeneous

controllers with different types can be placed on multiple subnets if necessary； 2） the controller should be preferentially placed in the node with the highest node degrees to facilitate management； 3） the subnet that has plenty of flow requests should place the heterogeneous controller with high capacity and low fragility coefficient.

● **Coding**

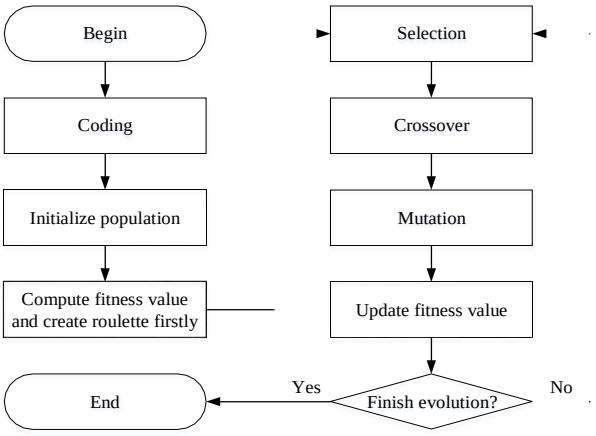


Fig. 4. The implementation of genetic algorithm

In this paper， we adopt the binary coding. This is because binary coding is easy to encode and decode， and the corresponding crossover and mutation can be realized by bit operation. Note that the improved GA is applied to each subnet. For each subnet， regardless of the type and location of heterogeneous controller， this heterogeneous controller certainly manages all switches of the subnet. Formally，

$xik = 1$， $\forall vi \in Vk$， $vk \in V$， where vi is the node in Vk and vk is the location of the heterogeneous controller. Based on the above， we just use the improved GA to determine the location and type of heterogeneous controller for each subnet. Fig. 5 shows the binary coding of the chromosome in the population， where

the green part represents the location of the heterogeneous controller and the blue part represents the type of heterogeneous controller.

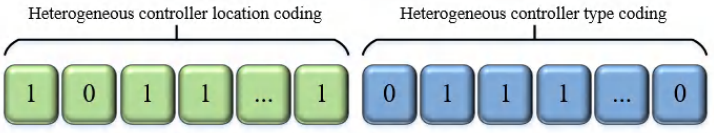


Fig. 5. Binary coding scheme of the chromosome

● **Fitness function**

The fitness function is used to determine the solution quality， usually the objective function. Therefore， we compute the fitness function based on Eq.（22）. Since HeCPP is a multi-objective optimization problem， we transform it into a single objective optimization problem based on the linear

weighted combination method. Specifically， the fitness function of subnet Gk = （Vk， Lk） is computed in

Eq.（32），

$Fk = \rho 1 \cdot \pi k + \rho 2 \cdot CUVk + \rho 3 \cdot \ln PCPFk$

（32）

s.t.

$\forall vi \in Vk$， $eij \in Lk$

（33）

where $\rho 1$， $\rho 2$， $\rho 3$ weigh the above three objectives， $0 \leqslant \rho 1$， $\rho 2$， $\rho 3 \leqslant 1$and $\rho 1 + \rho 2 + \rho 3 = 1$. Moreover， the weights can be adjusted dynamically according to the importance of the above three objectives in the

different network scenarios. For example， when more attentions are paid to the network delay， $\rho 3$ has the higher weights. Eq. （33） shows that all nodes and links are in subnet $Gk =$ （Vk， Ek） .

● **Genetic operation**

The genetic operations consist of selection， crossover， and mutation. In terms of selection operation， we adopt the elite selection way. Firstly， the selection operation is performed according to roulette. Then， in the current population， the individual with the highest fitness value is completely copied into the next generation population. In terms of crossover operation， we adopt the uniform crossover， as shown in Fig.

6. This means that the genes of two individuals are exchanged with the same crossover probability， thus forming two new individuals. In terms of mutation operation， we adopt the fundamental bit mutation， as shown in Fig. 7. This implicates that one or more genes of the individual are changed to the other values with the mutation probability.

Based on the above analysis， the heterogeneous controller placement based on improved GA is shown in Algorithm 3. Line 2-22 performs the improved GA， including initialization， comparison of fitness values， selection， crossover， and mutation， for each subnet obtained from Algorithm 2. Finally， we can identify the type and location of the heterogeneous controller placed in each subnet. The complexity of algorithm 3 is mainly attributed to the genetic process. The maximum number of nodes contained in each subnet is N， and the evolution iterates NE times. Therefore， the complexity of

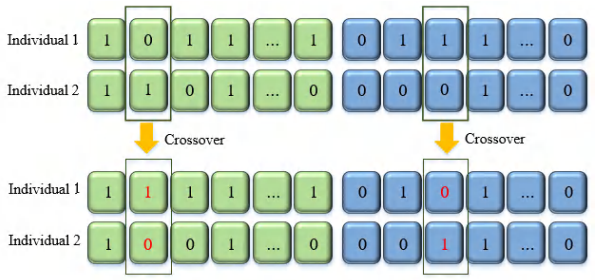


Fig. 6. Uniform crossover operation

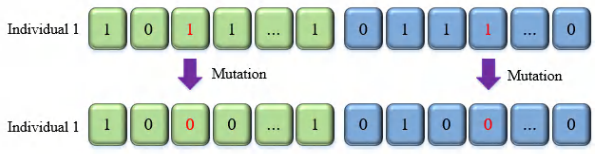


Fig. 7. bit mutation operation

computing each subnet is O （$NE \cdot \log N$） . Further， the genetic process requires to poll M （$0 < M \leqslant N$）

subnets. As a result， the complexity of algorithm 3 is O （$NE \cdot N \cdot \log N$） .

∩ ∩ **Theorem 3**. The probability that algorithm 3 converges to the optimal heterogeneous controller placement is 1.

Proof.

When algorithm 3 calls the genetic algorithm， it records the individual that has the maximum fitness value and implements the genetic operations for this individual to get the next generation. In the process of GA iteration， Z is the state space， and the population is transferred from state $i \in Z$ to state

$j \in Z$ by the genetic manipulation with the probability tij . Thus， the state transition probability matrix is $T =$ {tij} . PL （t） = {$\tau 1$ （t）， $\tau 2$ （t）， ...， τr （t）} shows the population in the tth generation， while the individual set is PL^* （t） = （MP （t）， PL （t）） and MP （t） is the individual with the maximum fitness value in the current population. PL （t） is transferred to PL （t +1） through the matrix T， so MP （t +1） = max {MP （t）， $MP0$} ，

and $MP0$ is the individual with the maximum fitness value in this population.

Actually， in algorithm 3， each generation of

Algorithm 3 Heterogeneous controller placement based on improved GA

Input: M subnets G M G M V , E k 1 k k 1 k k

Set of types of heterogeneous controllers H {1, 2,...,u,...,U}

Number of heterogeneous controllers with different types {NC 1 ,..., NC U }

Evolution times NE

Output: The heterogeneous controller placement results C {ck u }

1. ***for*** k=1 to M
2. ***procedure*** genetic algorithm
3. For subnet Gk , initialize *population*
4. *fits* Fk *population*
5. ne=0
6. ***while*** (ne<NE)
7. *fitbest* min *fits*
8. *populationnew* **Selection**(*population*)
9. *populationnew* **Crossover**(*populationnew*)
10. *populationnew* **Mutation**(*populationnew*)
11. *population* *populationnew*
12. *fits* Fk *population*
13. ***if*** *fitbest* min *fits*
14. ne ne 1
15. ***else***
16. ne 0
17. ***endif***
18. ***endwhile***
19. Decoding *population*
20. Get the type and location of the heterogeneous controller in Gk
21. Output ck u
22. ***endprocedure***
23. N u N u 1
24. ***endfor***

the population is only related to the previous generation

and independent of the initial generation. Therefore， the evolution of algorithm 3 can be regarded as a homogeneous Markov process， and its stochastic process {PL^* （t）， $t \geq 0$} belongs to a homogeneous Markov chain， as shown in Eq. （34），

$$PL^*（t）= PL^*（0）（R^*）t \quad （34）$$

The stable probability distribution of PL^* （t）is independent of the initial probability distribution， as shown in Eq. （35），

$$\lim_{t\to\infty} p \{ plt \in Z0 \} > 0 \quad （35）$$

Given $j \in Z$ ， j can take any values from {$Z0$ ， ...， Zr } . Hence， the probability of moving the optimal solution from any state is greater than 0， but the probability of going from the optimal solution to any

state is equal to 0. Finally， algorithm 3 can get the optimal solution with probability 1. The proof ends.

5 Simulation results

5.1 Simulation environment

(1) Simulation platform

In this paper， to avoid the performance interference， we adopt two physical servers （Intel E3-1230， 8 cores， 2.3GHz， 32GB RAM， Ubuntu 16.04）， and one for generating network traffic and the other for installing Mininet and heterogeneous controllers. We select four mainstream SDN controllers as the placement targets： NOX， Ryu， Floodlight， and ONOS， which means that there are four types of heterogeneous controllers and $U = 4$. Besides， we program all proposed algorithms based on Python.

(2) Topology environment

We implement our experiments on the existing topologies， all of which are from Internet Topology Zoo ［32］. Internet topology zoo is a publicly available data set， and a lot of researchers have used it to study CPP. In order to make our experiment more representative， we select three topologies with different sizes （e..， based on the number of nodes N）： OS3E （small-scale topology， $N < 50$ ）， Columbus （medium- scale topology， $50 \leqslant N < 100$ ）， and RF-II （large-scale topology， $N \geqslant 100$ ）， as shown in Table 3. We compute the hops between any two nodes in virtue of the Dijkstras algorithm. At the beginning of the

simulation， there are no packets in the network. After the simulation starts， each network node gen-

erates flow requests randomly and runs for 5 minutes. During the simulation， we have observed that the network delay and throughput reach a steady state after 20 seconds.

Table 3 Experimental topologies

	OS3E	Columbus	RF-II
Nodes	34	70	108
Links	42	85	306

(2) Parameters setting

In order to simulate the real network traffic， we hold that the flow requests of the switches follow the Poisson distribution， and $\lambda \in$（0， 40］ *kreq s* . Considering the heterogeneity of the controllers， each type of controller has different capacities and number of vulnerabilities ［15］. According to literatures ［17］， the heterogeneity parameters （controller capacity ω *kreq s* ， the number of vulnerabilities *VU* ， attacker prior knowledge coefficient ζ） set as NOX （80， 63， 0.1）， Ryu （130， 56， 0.1）， Floodlight （190，

55， 0.3）， ONOS （300， 51， 0.4）， respectively. Since state synchronization between controllers is not the focus of this paper， we consider the weak consistency between controllers to simplify the calculation， which means that the synchronization factors of all controllers are the same （ α = 10 ）. During the simulation， we try to set different parameters for the proposed algorithms. On the premise of not affecting the experimental results， the population size of the GA is set to be 20 times the number of nodes， the

maximum evolution times are 100， and the crossover and mutation probabilities are set to 0.3 and 0.2， respectively. To eliminate random error， all simulations run 100 times.

5.2 Results evaluation

In this subsection， we evaluate the performance of SQHCP from the perspectives of security and QoS. In terms of security， we evaluate SQHCP for enhancing the security of the control plane. As far as we're aware， we are the first to propose the heterogeneous controller placement， and there are no other approaches for comparison. Therefore， we change the types of heterogeneous controllers to evaluate the performance of SQHCP. In terms of QoS， we compare SQHCP with two representative controller placement approaches K-Center ［9］ and Survivor ［39］. K-Center clusters the nodes based on the propagation delay between nodes and places the controllers in the clustering center. Survivor considers the link utilization and load balance to place the controllers on the nodes with the most intersecting paths.

1) Control plane security

In this experiment， we evaluate the performance of SQHCP on the security of the control plane. Imitating the attackers， we attack the SDN control plane through exploiting the controller vulnerabilities and observe the performance of SQHCP. Specifically， we use the control plane fault rate to represent the security of the control plane. We suppose that the probability of successful vulnerability probing is P = 1- *e-t* ， where t is the number of probing attacks launched by the attackers. Attackers can exploit the vulnerability to attack the specific heterogeneous controllers after successfully probe it. Through varying the types of heterogeneous controllers placed， Fig. 8 to Fig. 10 shows the control plane fault rate changes with the number of probing attacks under different topologies， respectively. For example， U = 2 means that there are two types of heterogeneous controllers placed in the network. Obviously， as the topology scale enlarges， the control plane fault rate also increases. This is because increased network nodes expand the attack surface. However， regardless of any topologies， the types of heterogeneous controllers significantly affect the control plane fault rate. When U = 1， all controllers have the same type in the network， which are equal to the homogeneous controllers. In this case， the attackers only conduct fewer probing attacks to destroy the entire control plane. As U increases， the control plane fault rate decreases， and the attackers require to conduct more probing attacks to destroy the control plane. This is because the

heterogeneous controller placement has eliminated the homogeneous controller common-mode fault and increased the attack difficulty. Therefore， the proposed SQHCP can effectively degrade the control plane’s fault rate and enhance its security.

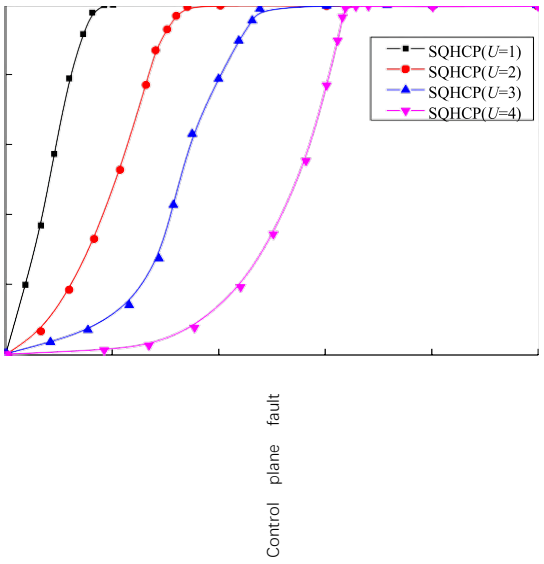


Fig. 8 Control plane fault rate in OS3E topology

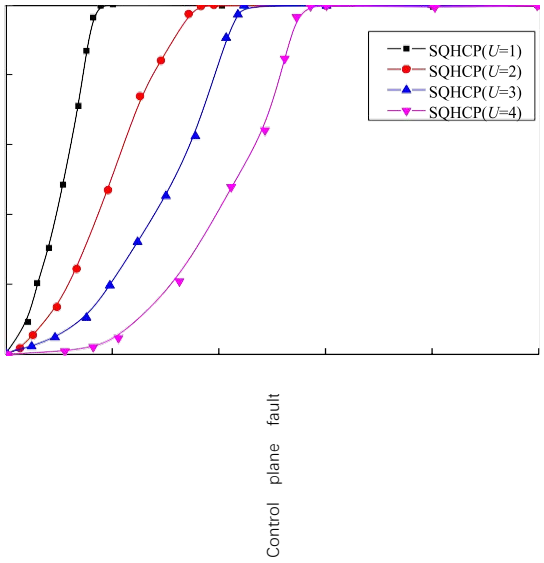


Fig. 9 Control plane fault rate in Columbus topology

2) The total number of controllers

Based on the network topologies of different sizes in table 3， we compare the total number of controllers required by three approaches. The results are

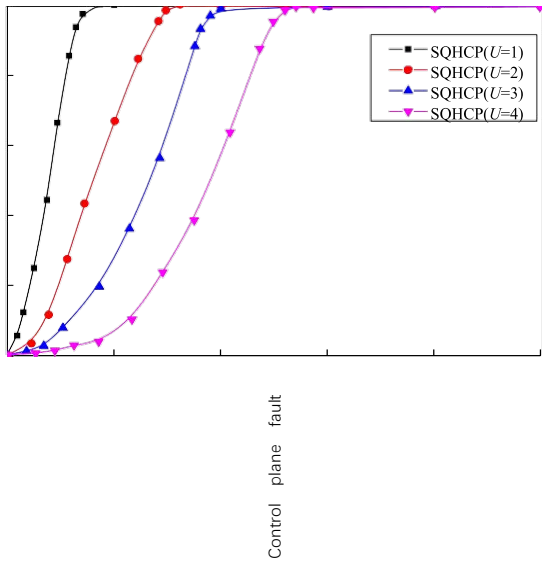


Fig. 10 Control plane fault rate in RF-II topology

shown in Fig. 11. It is clearly seen that the total number of controllers obtained from three approaches are almost identical under the small-scale topology. As the topology expands， the total number of controllers increases gradually. From OS3E （34 nodes） to RF-II （108 nodes）， K-Center requires the most controllers （from 6 to 21 controllers）， which is almost proportional to the increment of network nodes. Since Survivor places the controllers on the nodes with the most intersecting paths， as the number of links increases， so does the total number of controllers

required. Compared with the above two approaches， SQHCP requires the least controllers. The reasons can be explained as follows. SQHCP considers the heterogeneous controller placement as a complete knapsack problem and solves it by the dynamic planning. Meanwhile， based on the topology characteristics， it places controllers through improving the heuristic algorithms. Therefore， SQHCP can effectively calculate the minimum number of controllers that meet the network requirements， which is reduced by 23.5% on average compared to the other approaches.

3) Network end-to-end delay

In this experiment， we evaluate the perfor-

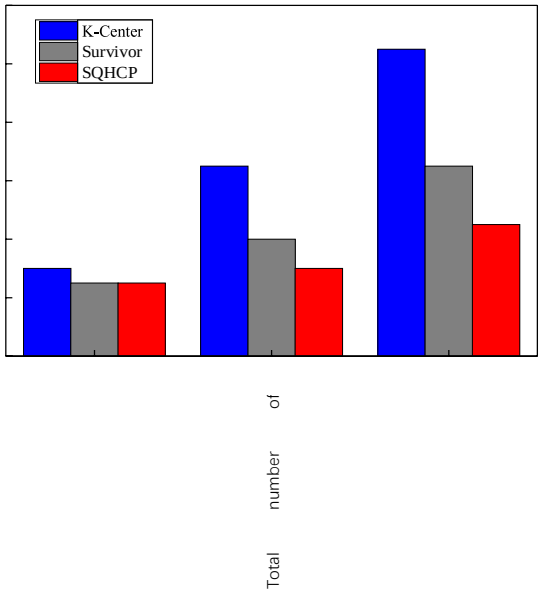


Fig. 11 The total number of controllers in different topologies

mance of our proposal from the end-to-end delay perspective. To make the simulation fair, we assume that all three approaches place four types of heterogeneous controllers. Fig. 12-14 shows the cumulative distribution function (CDF) of the maximum end-to-end delay of the three topologies, where the results of K-Center, Survivor, and SQHCP are shown in blue, black and red lines, respectively. It is observed that K-Center and SQHCP have a narrow gap but are superior to Survivor under small-scale topology OS3E. This is because Survivor places the controller in the network center, while the boundary node and the controller have long end-to-end delays. As the topology scale enlarges, the hops between nodes also increase. K-Center only considers the delay factor and cannot guarantee the optimal controller location. Therefore, in Columbus and RF-II topologies, the maximum end-to-end delay gaps between K-Center and SQHCP are bigger.

Fig. 15 further quantifies the end-to-end delay results and describes the average end-to-end delays of three approaches. For all topologies, SQHCP reduces end-to-end latency by 18.3% averagely, compared with K-Center and Survivor.

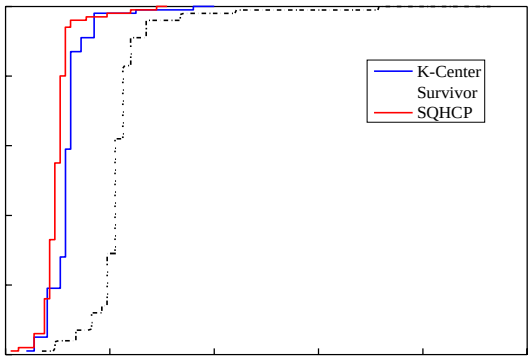


Fig. 12 CDF of the maximum end-to-end delay in OS3E topology

1.00.8 CDF 0.60.40.20.04 6 8 10 12 14Maximum end-to-end delay(ms)

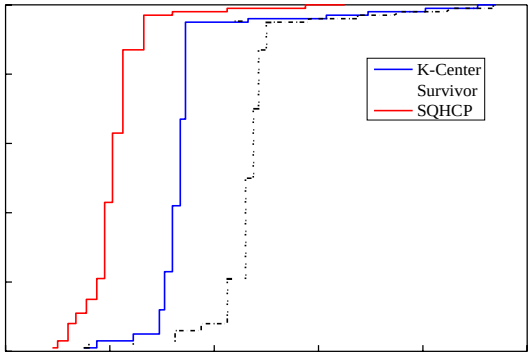


Fig. 13 CDF of the maximum end-to-end delay in Columbus topology

1.00.8 CDF 0.60.40.20.04 6 8 10 12 14Maximum end-to-end delay(ms)

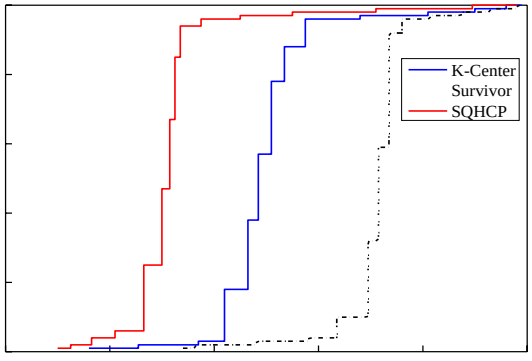


Fig. 14 CDF of the maximum end-to-end delay in RF-II topology

1.00.8 CDF 0.60.40.20.04 68 10 12 14Maximum end-to-end delay(ms)

4) Controller load balance

In this experiment, we compute the standard deviation of the controller utilization rate (standard deviation for short), which is the square root of *CUV* (*CUV* is shown in Eq. (12)), to evaluate the performance of the heterogeneous controller load balance. The smaller the standard deviation, the

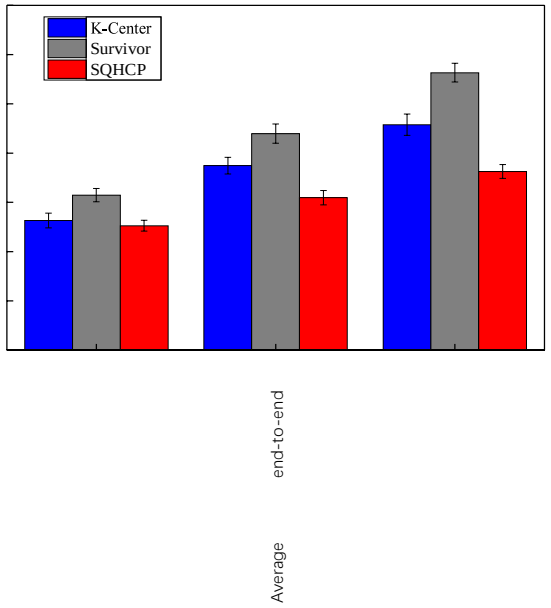


Fig. 15 The average end-to-end delay in different topologies
0OS3EColumbusNetwork TopologyRF-II

more balanced the heterogeneous controller loads. Similar to the above experiments， we also suppose that K- Center， Survivor and SQHCP place four types of heterogeneous controllers. Through changing the flow request rates of the switches， Fig. 16 to Fig. 18 shows the standard deviations under different topologies， respectively. On the whole， on the one hand， the larger the topology， the greater the standard deviation；

on the other hand， the more flow requests， the greater the standard deviation. However， no matter which kinds of topologies and how many flow requests， SQHCP always outperforms the other two approaches. The following reasons can account for the above results. Firstly， as the topology scale increases， the heterogeneous controllers need to manage to more network nodes with different flow request rates. The complexity of the distributions of both nodes and flow requests aggravates the difficulty of controller load balance， causing the increasement of standard deviation. Moreover， as the flow requests increase， so does the probability that the controller overloads， exacerbating the standard deviation. Compared with K-Center and Survivor， SQHCP determines the number of heterogeneous controllers based on the characteristics of topology and traffic， improves the heuristic algorithms to optimize the placement locations， and reserve part of the controller capacity to prevent overload. Therefore， SRCHP can always keep the standard deviation low and ensures the load balance of the heterogeneous controllers.

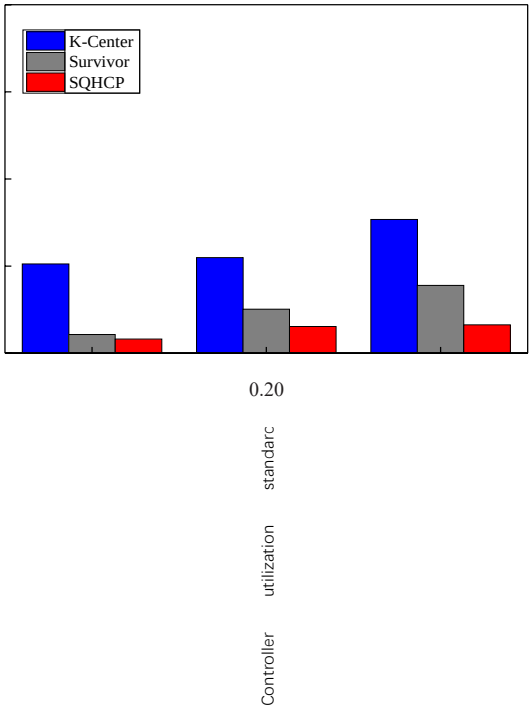


Fig. 16 Standard deviation of controller utilization in OS3E topology
0.150.100.050.00(0,10]kreq/s(10,20]kreq/sFlow requests(20,40]kreq/s

0.20
0.15
0.10
0.05
0.00（0，10］kreq/s
（10，20］kreq/s
Flow requests
（20，40］kreq/s

Fig. 18 Standard deviation of controller utilization in RF-II topology

6 Related work

In recent years， researchers have proposed a variety of solutions to CPP in the SDN and achieved rich results. Based on the optimization objectives，

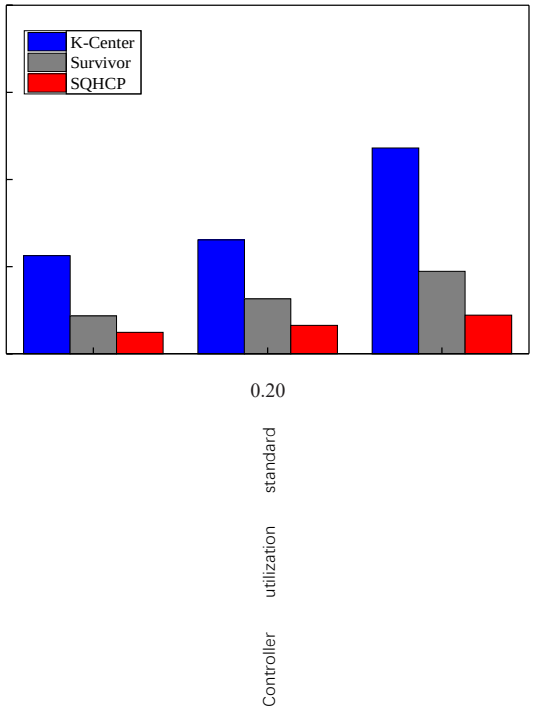
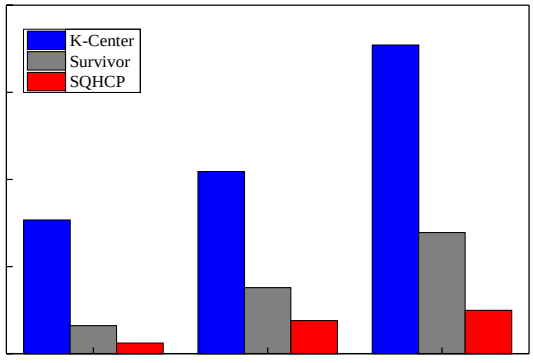


Fig. 17 Standard deviation of controller utilization in Columbus topology

0.150.100.050.00(0,10]kreq/s(10,20]kreq/sFlow requests(20,40]kreq/s



they can be divided into four categories: minimizing delay, minimizing deployment cost, minimizing fault rate, and multi-objective optimization. Each of these is described below.

Minimizing delay. Delay is crucial to SDN since switches and controllers interact with each other frequently. The existing CPP solutions usually focus on the propagation delay and controller processing delay. Considering the propagation delay, CPP is similar to the facility location problem. Heller et al. [9] first initiate the study on controller placement and give a clear definition for CPP: given a topology, how many controllers are needed, and where should they go? In the Internet 2 topology, they investigate the effects of controller placement on the average delay and worst-case propagation delay. Wang et al. [33] propose to transform CPP into a network partition problem and use the improved clustering method to solve it. Zhu et al. [34] consider minimizing the propagation delay between switches and controllers and between controllers. They formulate CPP as a control plane delay minimization problem and further propose a new algorithm based on clustering and Dijkstra to solve it. Yao et al. [10] define a capacitated CPP, taking into account propagation delay, and design an efficient algorithm to solve it.

Minimizing placement cost. With the large-scale application of SDN, how to reduce the controller placement cost has attracted the attention of researchers. Afrim et al. [35] establish a mathematical model to solve this problem, with the goal of reducing network cost. Through the more detailed analysis, this model determines the optimal controller numbers, locations and the connection relationships between network elements. Rath et al. [36] adopt game theory to solve CPP. Based on the non-zero-sum game, the optimization engine of each controller calculates a revenue function, compares its revenue value with the neighbor's revenue value, and takes appropriate decisions (e.g., add a controller, or delete a controller). Alejandro et al. [37] propose an energy-aware controller placement scheme, considering the maximum link utilization of each link when the flow routing is required. They formulate binary integer programming to reduce network energy consumption by closing as many links as possible.

Minimizing failure rate. Network node or link failure may lead to disconnections between switches and controllers, or even crash controller. Therefore, how to determine the reasonable locations to place controllers to minimize the failure rate is another hot potato. Hu et al. [38] research on reliability-optimized controller placement and prove it NP-hardness. Moreover, they define a new metric, the

expected percentage of control path loss， to characterize the reliability of the control plane in SDN. Müller et al. ［39］ introduce Survivor， an enhanced controller placement strategy， considering the path diversity， capacity， and failover mechanism. Zhong et al. ［40］ present a min-cover based controller placement approach to ensure the reliability of SDN. They first give the definitions of neighborhood domain and minimum coverage. The reliability and low latency of the control network are realized by placing as few controllers as possible. Hu et al. ［41］ propose a reliable and load balance-aware multi- controller deployment method， which exploring the reliable controller deployment through balance node efficiency and path quality. Rao et al. ［42］ design an optimization model to improve the resilience of the controller placement. The objective is to minimize the backup capacity while keeping the total number of controllers be constant.

Multi-objective optimization. Recently， several researchers have considered many factors （e.g， delay， load balancing， and reliability） for solving CPP and formulate it as a multi-objective hybrid optimization problem. Stanislav et al. ［11］ consider delay， isolated node， and controller loads during controller placement and propose a controller placement framework based on Pareto. Moreover， they analyze all possible placement scenarios. Jalili et al. ［12］ consider the propagation delay， hop， and link utilization in the controller placement and further analyze the impact of these metrics on QoS. They present GA based on the analytic hierarchy process to compute the optimal controller placement. Wang et al. ［43］ study the CPP in the SDWAN scenario and analyze the end-to-end delay and controller processing delay based on the queuing theory model.

Summary. Although there are plenty of research works on CPP， almost all solutions assume that the controllers are homogeneous and ignore the resulting common-mode fault. Therefore， we first propose to place heterogeneous controllers to solve the homogeneous controller common-mode fault and set the delay， controller utilization rate and fault rate as the optimization objectives to maximize the security of the control plane and guarantee QoS.

7 Conclusion

In this paper， we introduce a new controller placement idea that deploys the heterogeneous controllers in the network to solve the common-mode fault problem of the existing CPP solutions. Firstly， we formally define HeCPP， considering end-to-end delay， controller utilization and controller fault rate under the heterogeneous controller condition. HeCPP is mathematically described as a constrained multi-objective optimization problem aiming to minimize network end-to-end delay， balance heterogeneous controller utilization， and lower control plane fault rate. In order to solve HeCPP effectively， we propose SQHCP approach， which is Secure-aware and QoS-guaranteed. Specifically， SQHCP consists of two steps. In step 1， we compute the required heterogeneous controller types and numbers based on dynamic planning. In step 2， we first partition the network into several subnets based on the K-means algorithm and then improve the genetic algorithm to optimize the heterogeneous controller placement for each subnet. On the one hand， we prove the feasibility of SQHCP by several theorems. On the other hand， we conduct extensive simulations to evaluate the performance of SQHCP under real topologies from the Internet Topology Zoo. Simulation results verify that SQHCP can effectively enhance the security of the control plane and improve QoS compared with the existing representative algorithms. In future work， we will plan to extend SQHCP approach to a practical network environment with more real traffic to evaluate its performance.

References:

[1] KreutzD.，RamosF. M. V.，VeríssimoP. E.，RothenbergC. E.，AzodolmolkyS. and UhligS.，"Software-Defined Networking：A Comprehensive Survey," in Proceedings of the IEEE，vol. 103，no. 1，pp. 14-76，Jan. 2015.

[2] J, Sushant, et al. "B4: Experience with a Globally-Deployed Software Defined WAN. "in Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM ACM，2013.

[3] BannourF.，SouihiS. and MelloukA.，"Distributed SDN Control：Survey，Taxonomy，and Challenges，" in IEEE Communications Surveys & Tutorials，vol. 20，no. 1，pp. 333-354，Firstquarter 2018.

[4] Hassas YeganehS, GanjaliY. "Kandoo: a framework for efficient and scalable offloading of control applications，"in ACM HotSDN，2012.

[5] KoponenT, et al. "Onix: A Distributed Control Platform for Large-scale Production Networks. "In 9th USENIX Symposium on Operating Systems Design and Implementation，OSDI 2010，-6，2010，Vancouver，BC，Canada，Proceedings USENIX Association，2010.

[6] PhemiusK.，BouetM. and LeguayJ.，"DISCO：Distributed SDN controllers in a multi-domain environment，" in 2014 IEEE Network Operations and Management Symposium (NOMS)，Krakow，2014，pp. 1-2.

[7] HuT.，GuoZ.，YiP.，BakerT. and LanJ.，"Multi-controller Based Software-Defined Networking: A Survey，" in IEEE Access，vol. 6，pp. 15980-15996，2018.

[8] WangG.，ZhaoY.，HuangJ. and WangW.，"The Controller Placement Problem in Software Defined Networking：A Survey，" in IEEE Network，vol. 31，no. 5，pp. 21-27，2017.

[9] Heller, Brandon, SherwoodR.，and McKeownN.. "The controller placement problem. "in Workshop on Hot Topics in Software Defined Networks ACM，2012.

[10] YaoG.，BiJ.，LiY. and GuoL.，"On the Capacitated Controller Placement Problem in Software Defined Networks，" in IEEE Communications Letters，vol. 18，no. 8，pp. 1339-1342，Aug. 2014.

[11] LangeS. et al.，"Heuristic Approaches to the Controller Placement Problem in Large Scale SDN Networks，" in IEEE Transactions on Network and Service Management，vol. 12，no. 1，pp. 4-17, March 2015.

[12] Jalili, Ahmadet al. "Multi criteria analysis of Controller Placement Problem in Software Defined Networks. "In Computer Communications，2019.

[13] BenzekkiKamal, Abdeslam El Fergougui, and Abdelbaki Elbelrhiti Elalaoui. "Software-defined networking (SDN): A survey. "in Security & Communication Networks，2017.

[14] RPke, Christian, and HolzT.. "On network operating system security. "in International Journal of Network Management，2016.

[15] TatangD.，QuinkertF.，FrankJ.，Röpke and TC.. Holz, "SDN-Guard: Protecting SDN controllers against SDN rootkits，" in *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*，Berlin，2017，pp. 297-302.

[16] Xueming, Si, et al. "A Review of the Basic Theory of Mimic Defense. "in Strategic Study of CAE，2016.

[17] MazikuH.，ShettyS.，JinD.，KamhouaC.，NjillaL. and KwiatK.，"Diversity Modeling to Evaluate Security of Multiple SDN Controllers，" in 2018 International Conference on Computing，Networking and Communications (ICNC)，Maui，HI，2018，pp. 344-348.

[18] VoasJ.，GhoshA.，CharronF. and KassabL.，"Reducing uncertainty about common-mode failures，"
in Proceedings The Eighth International Symposium on Software Reliability Engineering, Albuquerque, NM, USA，1997，pp. 308-319.

[19] Avizienis and Kelly，"Fault Tolerance by Design Diversity: Concepts and Experiments，" in Computer，vol. 17，no. 8，pp. 67- 80，Aug. 1984.

[20] Oktian, EkoYustus, et al. "Distributed SDN controller system: A survey on design choice. "in Computer networks，2017，pp. 100-111.

[21] MamushianeL.，LyskoA. and DlaminiS.，"A comparative evaluation of the performance of popular SDN controllers，" in Wireless Days (WD)，Dubai，2018，pp. 54-59.

[22] KhondokerR.，ZaaloukA.，MarxR. and BayarouK.，"Feature-based comparison and selection of Software Defined Networking (SDN) controllers，"in 2014 World Congress on Computer Applications and Information Systems (WCCAIS)，Hammamet，2014，pp. 1-7.

[23] GuoZ.，ChenW.，LiuY.，XuY. and ZhangZ.，"Joint Switch Upgrade and Controller Deployment in Hybrid Software- Defined Networks，" in IEEE Journal on Selected Areas in Communications，vol. 37，no. 5，pp. 1012-1028, May 2019.

[24] HuT.，YiP.，ZhangJ. and LanJ. "A distributed decision mechanism for controller load balancing based on switch migration in SDN，" in China Communications，vol. 15，no. 10，pp. 129-142，Oct. 2018.

[25] BuK.，WenX.，YangB.，ChenY.，LiL. E. and ChenX.，"Is every flow on the right track?：Inspect SDN forwarding with RuleScope，" in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*，San Francisco，CA，2016，pp. 1-9.

[26] KasiA. A.，KhanF.，AhmedB. A.，RashidS. and WaseemS.，"Performance analysis of homogenous and heterogenous network core switches，" in 2017 International Symposium on Wireless Systems and Networks (ISWSN)，Lahore，2017，pp. 1-7.

[27] Borradaile, Glencora, HeeringaB.，and WilfongG.. "The knapsack problem with neighbour constraints. "in Journal of Discrete Algorithms，2012，pp. 224-235.

[28] Theil, Henri. "A Note on Certainty Equivalence in Dynamic Planning，" Henri Theil's Contributions to Economics and Econometrics. 1992.

[29] BabayiğitB.，UluB. and HasçokadarE. N.，"Solving Multi-Controller Placement Problem in Software Defined Networks with A Genetic Algorithm，" in 2019 4th International Conference on Computer Science and Engineering (UBMK)，Samsun，Turkey，2019，pp. 666-670.

[30] HuangV.，ChenG.，FuQ. and WenE.，"Optimizing Controller Placement for Software-Defined Networks，" 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)，Arlington，VA，USA，2019，pp. 224-232.

[31] XiaolanH.，MuqingW. and WeiyaoX.，"A Controller Placement Algorithm Based on Density Clustering in SDN，" in 2018 IEEE/CIC International Conference on Communications in China (ICCC)，

Beijing，China，2018，pp. 184-189.

[32] KnightS. , NguyenH. X. , FalknerN. , BowdenR. and RoughanM. ，"The Internet Topology Zoo，" in IEEE Journal on Selected Areas in Communications，vol. 29，no. 9，pp. 1765-1775, October 2011.

[33] WangG. , ZhaoY. , HuangJ. , DuanQ. and LiJ. "A K-means-based network partition algorithm for controller placement in software defined network，" in 2016 IEEE International Conference on Communications (ICC)，Kuala Lumpur，2016，pp. 1- 6.

[34] ZhuL. , ChaiR. and ChenQ. "Control plane delay minimization based SDN controller placement scheme，" in 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)，Nanjing，2017，pp. 1-6.

[35] Sallahi and M. St-HilaireA. "Optimal Model for the Controller Placement Problem in Software Defined Networks，" in IEEE Communications Letters, vol. 19, no. 1，pp. 30-33，Jan. 2015.

[36] RathH. K. , RevooriV. , NadafS. M. and SimhaA. ，"Optimal controller placement in Software Defined Networks (SDN) using a non-zero-sum game，"in Proceeding of IEEE International Symposium on a World of Wireless，Mobile and Multimedia Networks 2014，Sydney，NSW，2014，pp. 1-6.

[37] Ruiz-RiveraA. , ChinK. and SohS. ，"GreCo：An Energy Aware Controller Association Algorithm for Software Defined Networks," in IEEE Communications Letters，vol. 19，no. 4，pp. 541-544，April 2015.

[38] HuY. , WangW. , GongX. , QueX. and ChengS. ，"On reliability-optimized controller placement for Software-Defined Networks，" in China Communications，vol. 11，no. 2，pp. 38-54, Feb 2014.

[39] MüllerL. F. , OliveiraR. R. , LuizelliM. C. , GasparyL. P. and BarcellosM. P. ，"Survivor：An enhanced controller placement strategy for improving SDN survivability, " in 2014 IEEE Global Communications Conference，Austin，TX，2014，pp. 1909- 1915.

[40] ZhongQ. , WangY. , LiW. and QiuX. "A min-cover based controller placement approach to build reliable control network in SDN, " in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*，Istanbul，2016，pp. 481-487.

[41] HuT. , YiP. , ZhangJ. and LanJ. "Reliable and load balance-aware multi-controller deployment in SDN，" in China Communications，vol. 15，no. 11，pp. 184-198，Nov. 2018.

[42] Killi, Bala Prakasa Rao, and RaoS. V. . "Towards improving resilience of controller placement with minimum backup capacity in software defined networks. "in Computer Networks，2019.

[43] WangG. , ZhaoY. , HuangJ. and WuY. ，"An Effective Approach to Controller Placement in Software Defined Wide Area Networks，" in IEEE Transactions on Network and Service Management，vol. 15，no. 1，pp. 344-355, March 2018.

基于小样本学习的跨站脚本攻击检测技术研究

战略支援部队信息工程大学

摘　要：近年来，随着信息化、移动化的趋势不断加强，网络应用程序的使用非常普及，各类Web安全威胁层出不穷。跨站脚本攻击是最常见并且对用户危害甚重的一类Web攻击方式，大多数传统检测手段已经很难适应现有的各种XSS攻击的混淆变种。针对实际面临的新型类别攻击行为的小样本问题，引入虚拟样本生成技术，扩大样本数据集，是提高模型的泛化能力和分类精度的重要手段之一。本文通过改进现有的基于先验知识的虚拟样本生成方法，进一步提升机器学习模型的准确率。实验结果表明，在绝大多数情况下，本文算法性能更优，一定程度上提高了对未知攻击的检测发现能力。

关键词：跨站脚本攻击检测、小样本学习、机器学习、虚拟样本生成、卷积神经网络

Research on Cross-site Scripting Attack Detection Technology Based on Few-shot Learning

——(Lu Dongzhe. 906188591@qq.com, PLA Strategic Support Force Information Engineering University)

Abstract: With the intensification of informatization and mobility, the use of network applications is very popular, and various web security threats are emerging. Cross-site scripting attacks are the most common type of web attacks. Most traditional detection methods have been difficult to adapt to the existing con-fusion variants of various XSS attacks. Aiming at the few-shot problem of new types of attack bahaviors, introducing virtual sample generation technology and expanding the sample set is one of the important means to improve the generalization ability and classification accuracy of the model. This paper improves the accuracy of the model by improving the existing virtual sample generation method based on prior knowledge. Experimental results show that in most cases, the performance of the algorithm in this paper is bet-ter, and the ability to detect and discover unknown attacks is improved to a certain extent.

Key words: cross-site scripting attack; few-shot learning; machine learning; virtual sample generation; convolution neural network

1　引言

根据OWASP于2007至2018年间发布的Top10应用软件安全风险报告，XSS攻击一直位列其中，是极为普遍的安全问题。XSS攻击的实际影响通常取决于应用程序的性质、功能和数据以及受攻击用户的状态。在各类社交平台、开源的Web应用等场景中，可能造成的危害有：窃取用户cookie；将木马功能注入网站；发送广告和垃圾信息；传播蠕虫；导致DoS攻击；实施网络钓鱼；导致跨站点请求伪造；等等。如何对XSS进行准确的检测并制定有效的防御措施，是安全领域广大研究者亟需解决的问题。机器学习算法本质上是一种数据驱动的技术，将其运用到攻击检测领域要

解决的一个重要问题就是需要的大样本量和实际获得的小样本量之间的矛盾，因此本文改进了现有的VSG方法，从小样本中挖掘更多的信息，以丰富样本数据。

1.1 漏洞原因和利用方式

按照漏洞的特征和发生的位置作区分，XSS目前主要有三种类型：反射型XSS、存储型XSS、基于DOM的XSS。图1-3展示了这三种攻击流程。

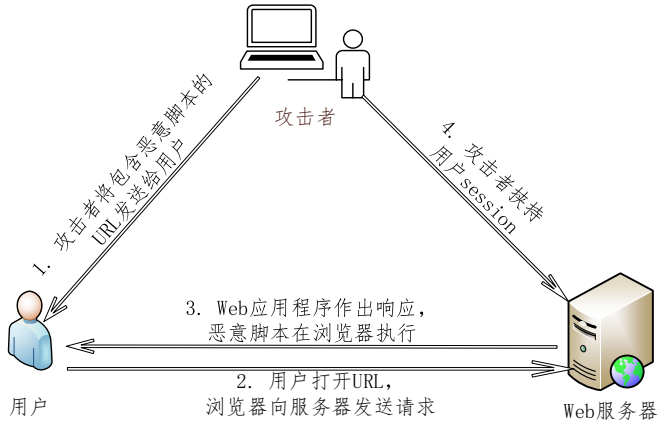


图1 反射型XSS

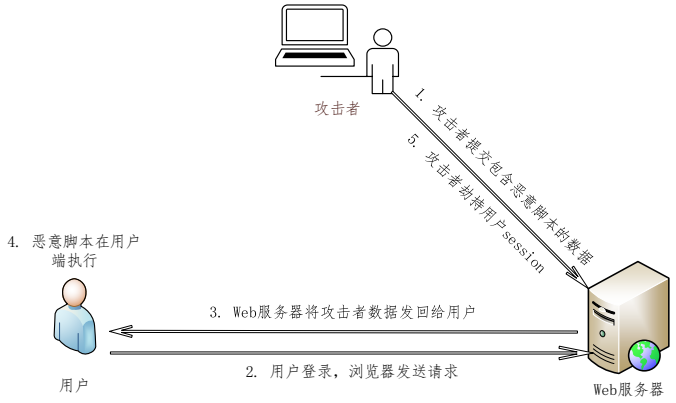


图2 存储型XSS

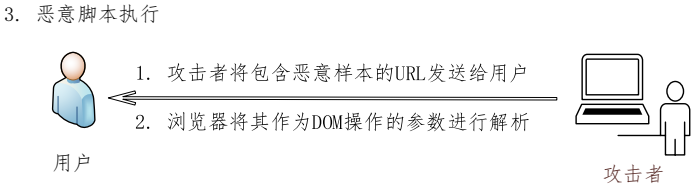


图3 基于DOM的XSS

然而，随着javascript和ajax技术的应用，随着客户端交互技术的进步，XSS攻击的形式变得更加灵活。通过分析多种攻击载荷发现，使用Base 64编码、HTML实体替换、URL编码等不同的编码方式，结合字符大小写转换、嵌套、替换等变异方式，甚至是凭借层叠样式表中的某些属性触发XSS，可以形成更加复杂多变且数目庞大的攻击脚本。

1.2 传统检测方法

静态分析方法是基于源代码的分析，即通过查看Web应用代码来发现其中可能存在的漏洞。Fortify SCA、Web XSS Test System、XSSDetect、saferXSS等工具都是根据静态分析的原理运行的。静态分析的检测精度较高，但是许多现实环境下

获取源码存在困难，并且大多数方法只是针对某种编程语言或动态网页语言进行检测，这是静态分析通用性不强的原因。

动态分析利用的是黑盒测试的思想，不需获取程序源码，而是通过模拟真实的攻击行为，注入恶意脚本，通过服务器的反馈数据来确定是否有XSS漏洞。考虑到测试向量查找的效率以及攻击向量集的大小，动态分析的效率是不如静态分析效率的，且动态方法使用的攻击向量集的质量会大大影响到检测效率。

机器学习算法有着强大的特征提取能力和自学习能力，但传统的浅层模型表达能力有限，且代价函数没有局部最优点，因此本文采用深层的机器学习模型。

1.3 虚拟样本生成

虚拟样本生成技术，目前是在小样本学习领域的重要技术手段，目的在于改善小数据集的学习性能，已经在诸多领域有了成功运用，如图像识别、工业建模、计数设备制造、机械信号建模等。该技术大体上可以分为三类［1］：基于研究领域具体的先验知识构造虚拟样本；基于扰动的思想构造虚拟样本；基于研究领域的分布函数构造虚拟样本。我们以两个典型的VSG技术为例，简单说明其主要特点。第一，Bootstrap起初来自统计学概念，就是一种获得训练样本的方式，一般指的是“有放回地全抽”。这个方法最早应用于图像识别领域，针对一个物体的三维视角，基于几何变换去生成虚拟样本，最大的特点就是“特征不变”，即只通过平移、对称变换、旋转角度等方法来生成虚拟样本。Bootstrap相当于重复采样，只是改变了种子样本的权重。但通过这种方式，取得的训练样本之间是线性相关的，这样会导致出现某些参数无法确定、协方差矩阵不稳定等问题。第二，添加扰动的方法就是在训练样本的各维数据上增加一个扰动，最早应用于解决类内协方差矩阵的奇异性问题。此方法并不完善，由于计算量极大，且扰动值不好确定，算法效率成为新问题。

假定有原始样本（x，y），若通过变换T得到的（Tx，y_Tf（x））是一个合理样本，则该样本就是由变换T生成的虚拟样本。变换T是依据先验知识得到的，由于不同领域先验知识以及分布特征不同，所以目前没有一种统一的虚拟样本生成方法。本文改进了传统的基于先验知识的VSG方法，依据新的孤立样本的特征向量与原始训练集的特征向量之间的差异来确定是否直接对模型进行再训练，或者是先转化为知识，再利用知识去指导虚拟样本的生成。

2 算法分析

2.1 定义

在基于机器学习模型的分类任务中，对知识、模型和数据之间的关系进行理论分析。我们首先做以下定义。

（1）定义S为一组样本，s为单个样本，v和t是特征值向量和对应的标签，V和T是一组样本的特征值向量和对应的标签。其中，特征值向量是指经过预处理和分词后通过word2vec得到的词向量；标签有两种，恶意攻击样本的payload记为1，正常的http请求的则记为0。那么就有s=（v，t）和S=（V，T）。

（2）定义M为V到T的映射，原像是V，像是T，即T=M（V）。可以理解为，在此二分类问题中，每组向量V对应唯一的一个标签T。

（3）定义K为一组真命题，代表知识。对于可解释的特征，可以对应其可解释性分析其中存在差别的知识，如果可解释性不强，这种差别本身也可以看作知识，认为以其为中心的邻域分布是区别于其他样本的新知识。

S、M、K三者都可以描述数据分布的特征。S是基于特征值空间分布的采样，M是由现有数据建立的模型，K是先验知识的规则集，通过谓词逻辑推理，可以根据样本特征获得分类结果。因此，S、M、K可以相互转换：样本数据集S可以经过训练得到模型M，模型M可以抽象出知识K，对知识K添加约束可以生成虚拟样本S’；反过来样本数据集S中可以推断出一定的逻辑知识K，利用先验知识K可以设计模型M，利用模型M可以对样本数据集S进行分类。

2.2 算法描述

本文的算法思想如图4。依据新的孤立样本的特征向量与原始训练集的特征向量之间的差异来确定是否直接对模型进行再训练，或者是先转化为知识，再利用知识去指导虚拟样本的生成。

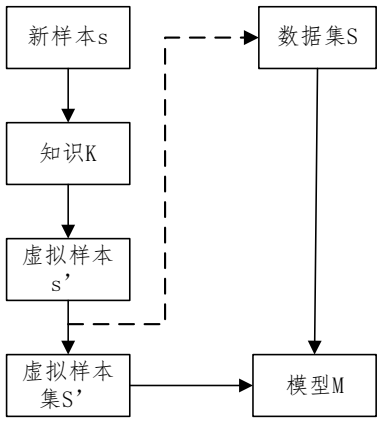


图4　算法思想

定义特征集合为F=｛F_1，F_2，…，F_n｝，虚拟样本集为D'，改进后的算法描述如下。

算法1　虚拟样本生成算法

Input：*s*: new sample
S: training dataset
F: feature set
x: threshold of distance

Output：*D'*: virtual samples set

Calculate the distance from the new sample *s* to the sample in *S*
Infer prior knowledge *K* based on distance
Generate virtual samples *s'* based on prior knowledge
for S_i in *S* do
  if target (S_i)=target (*s*):
    if the feature vector of *s* over F_i is larger than *x* :
      the feature is fixed and others are sampled in S_i
    else:
      break
  else:
    the feature vectors of *s* over F_i are the same as S_i
end for

3　实验和结果

本实验收集的正样本数据集经去重处理后，共30580条。其中28355条是从http：//xssed.com网站爬取的payload，该网站公开提供有关XSS漏洞的文章、新闻、教程等，是目前最大的XSS漏洞收集权威网站，这些数据截至2016年；另外1625条来自https：//github.com网站上他人的分享数据；其余600条来自平时的学习记录，均是从实际比赛或相关教程获得的。负样本数据是正常的http的GET请求，仅保留payload部分，共181012条。

样本集经预处理后，我们对样本数据进行分词处理，规则如表2所示。

表2　分词规则和示例

分词规则	示例
脚本标签	<script>
<>开头	<h1
参数名	name=
函数体	alert(
引号间的内容	‘XSS’
http(s)	http://xxxxx
属性名	type=
字符数字组成的单词	fu0

我们采用Word2vec中的Skip-gram模型，将其转化成嵌入式词向量。嵌入式词向量可以很好地表示词向量的特征和词汇之间的关联度。我们将训练集和测试集以7：3的比例分割，搭建SVM、LSTM、CNN三种模型进行训练，评估指标是精确率、召回率和F1分数。

实验开发语言为Python 3.7.6，机器学习模型基于tensorflow和keras搭建，本机配置为：操作系统Windows 10 x64、处理器Intel（R） Core（TM） i5-10210U @ 1.60GHz、内存16GB、显卡NVIDIA GeForce MX250。

实验结果如图5所示。

可见，CNN模型在各方面表现都不错，其结构图6所示。

为了验证改进算法的鲁棒性，本实验收集了40个2020年公布的CVE样本，这些样本的发现均在训练数据收集截止日期之后两年甚至更长的时间，每两个一组，共分为20组。使用综合性能最好的CNN模型进行再训练，准确率结果如图7所示。

4　结论

本文针对XSS攻击的检测，搭建机器学习模型，并改进了传统的虚拟样本生成方式。实验结果表明，CNN模型性能最优，这种改进的VSG技术也能在更大程度上扩充种子样本的变化范围，也使模型泛化性能更优。

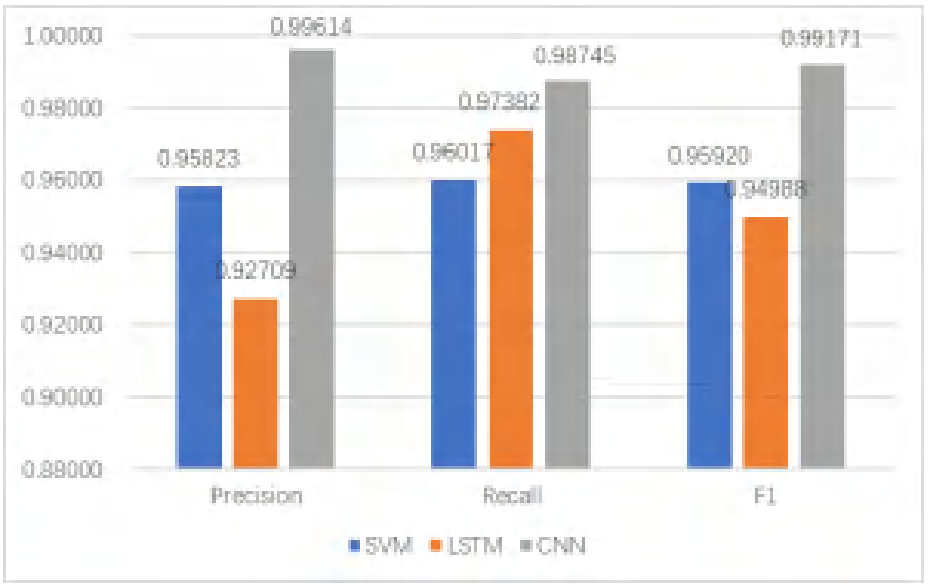


图5　三种模型的性能对比

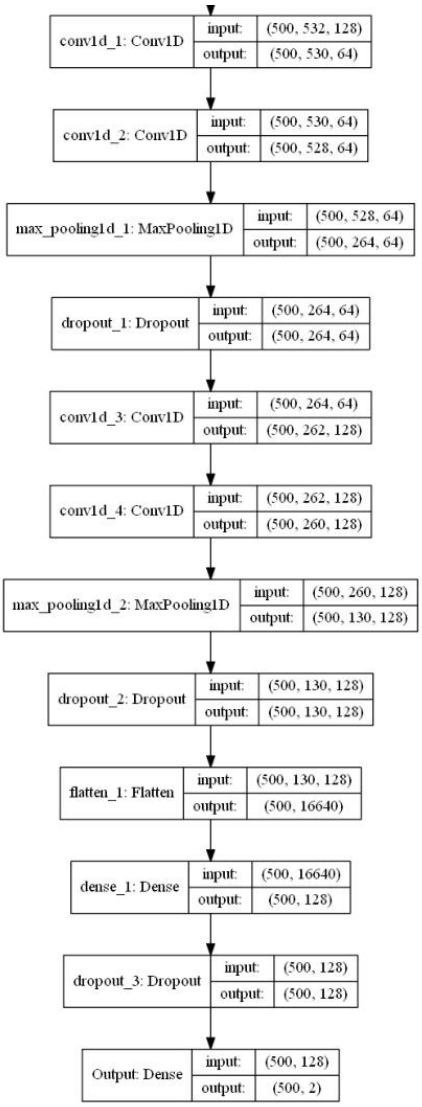

conv1d_1: Conv1D
input: (500, 532, 128)
output: (500, 530, 64)
conv1d_2: Conv1D
input: (500, 530, 64)
output: (500, 528, 64)
max_pooling1d_1: MaxPooling1D
input: (500, 528, 64)
output: (500, 264, 64)
dropout_1: Dropout
input: (500, 264, 64)
output: (500, 264, 64)
conv1d_3: Conv1D
input: (500, 264, 64)
output: (500, 262, 128)
conv1d_4: Conv1D
input: (500, 262, 128)
output: (500, 260, 128)
max_pooling1d_2: MaxPooling1D
input: (500, 260, 128)
output: (500, 130, 128)
dropout_2: Dropout
input: (500, 130, 128)
output: (500, 130, 128)
flatten_1: Flatten
input: (500, 130, 128)
output: (500, 16640)
dense_1: Dense
input: (500, 16640)
output: (500, 128)
dropout_3: Dropout
input: (500, 128)
output: (500, 128)
Output: Dense
input: (500, 128)
output: (500, 2)

图6　CNN结构

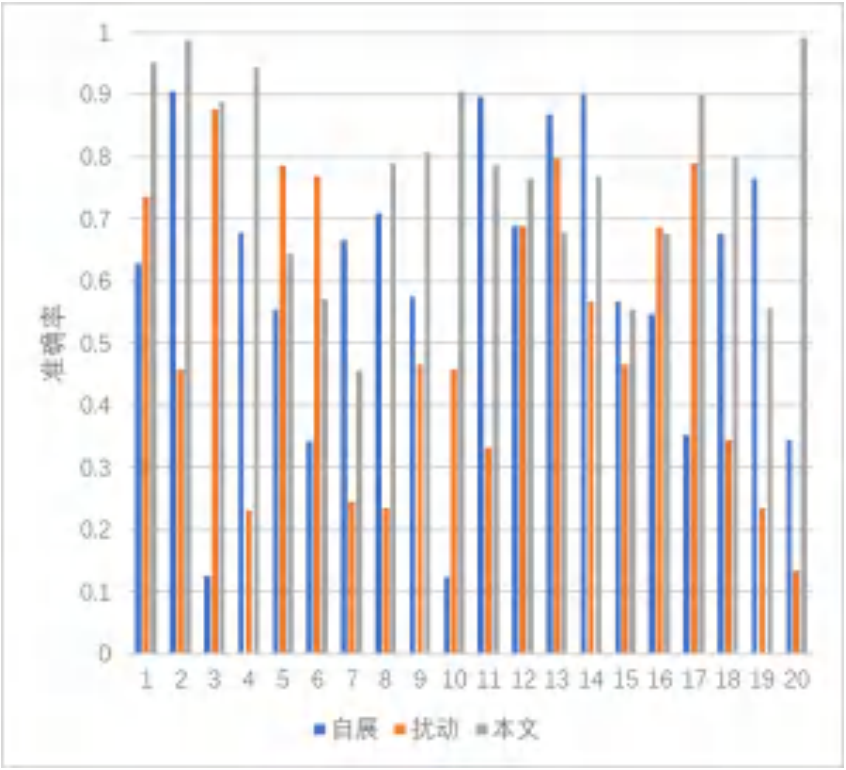


图7　三种VSG方法的准确率对比

参考文献:

[1] 储泽楠, 李世杨. 基于节点生长马氏距离K均值和HMM的网络入侵检测方法设计[J]. 计算机测量与控制，2014，22(10)：3-5.

[2] 陈君新. 基于机器学习的XSS攻击检测技术研究[D]. 浙江工业大学硕士学位论文. 2018:8-15.

[3] 洪镇宇. 基于机器学习的跨站脚本攻击检测研究[D]. 厦门大学硕士学位论文. 2018. 5：1-3.

[4] 张贵昌. 基于深度卷积网络的跨站脚本检测方法研究[D]. 南昌航空大学硕士学位论文. 2019. 6：1-3.

[5] 于旭, 杨静, 谢志强. 虚拟样本生成技术研究[J]. 计算机科学，2011，38(3):16-18.

[6] Zhong-Sheng Chen, Bao Zhu, Yan-Lin He, Le-An Yu. A PSO based virtual sample generation method for small sample sets: Applications to regression datasets [J]. Engineering Applications of Artificial Intelligence,2017(59)：236-243.

[7] Yimin Huang, Weiran Huang, Liang Li, Zhenguo Li. Meta-learning Pac-Bayes priors in model averaging[R]. AAAI，2019. 12：3-7.

[8] Aoxue Li, Tiange Luo, Tao Xiang, Weiran Huang, Liwei Wang. Few-Shot Learning With Global Class Representations[R]. ICCV，2019：2-4.

[9] 潘古兵, 周彦晖. 基于静态分析和动态检测的XSS漏洞发现[J]. 计算机科学. 2012(S1)：1-4.

[10] 李佩佩. 基于动态分析的XSS漏洞检测方法[D]. 华中科技大学硕士学位论文. 2016. 5：1-2.

[11] 谷家腾, 辛阳. 基于动态分析的XSS漏洞检测模型研究[J]. 计算机工程. 2018. 11：1-4.

[12] Bengio Y., Ducharme R., Vincent P., and Janvin C.. A neural probabilistic language model[J]. The Journal of Machine Learning Research. 2003:1137-1155.

[13] 王卫东, 杨静宇. 采用虚拟训练样本的二次判别分析方法[J]. 自动化学报，2007，34(4)：3-4.

[14] 周志华. 机器学习[M]. 清华大学出版社，2016：97-110.

[15] 斋藤康毅. 深度学习入门: 基于Python的理论和实现[M]. 陆宇杰译. 人民邮电出版社，2018:130-159.

[16] 孙伟, 张凯寓, 薛临风, 徐田华. XSS漏洞研究综述[J]. 信息安全研究. 2016(12)：3-5.

[作者简介]

路冬哲（1997—），女，计算机类工学学士学位，现就读战略支援部队信息工程大学硕士一年级，主要研究方向是基于机器学习和深度学习的漏洞攻击检测技术。

动态重构安全网络报文解析器结构

李翔宇*，周飞飞

清华大学微电子学研究所，北京 100084

摘　要：为了消除网络交换机芯片报文解析器模块的硬件木马隐患，本文设计了一种动态重构的报文解析器结构，实现了动态重构的查表关键字生成电路。该电路通过不断地配置冗余模块的功能，动态改变有效电路逻辑的物理位置，增大了硬件木马的植入难度，提高了交换机的安全性。电路功能通过了 FPGA 原型验证，采用 TSMC 32nm 工艺库的综合结果显示，关键字提取电路的关键路径延时为 0.8 ns。

关键词：网络安全；集成电路；报文解析器；硬件木马；动态可重构

Dynamically Reconfiguring Network Packet Parser Structure

LI Xiang-Yu, ZHOU Fei-Fei

Institute of Microelectronics, Tsinghua University, Beijing 100084, China

Abstract: To remove the hardware Trojan threads from the packet parser of network switch chip, a dynamically reconfiguring packet parser structure is proposed in this paper and the keywords extraction circuit was implemented. The circuit. Through dynamically reconfiguring the function of the redundant primary-elements (PEs), the physical positions of the active PEs are changed dynamically, it increases the difficulty of hardware Trojan's implanting and hence improves the security of the switch. The circuit has been verified using FPGA. Additionally, synthesized with the 32nm TSMC technology library, the delay of the critical path of the keywords extraction circuit is 0.8 ns.

Key words: network security; integrated circuit; parser; hardware Trojan; dynamic reconfiguration

1　引言

网络安全的重要性的不断提高要求网络交换机芯片对于硬件木马[1]具有防御能力。交换机芯片中的报文解析器根据报文头进行初步的报文识别和转发决策。作为入口模块，其更容易接收木马触发信息，因此是需要重点防御硬件木马的功能模块。由于硬件木马形态难以预知，硬件的详细、完备的行为“黄金模型”实际中难以建立，对于网络设备这类工作在复杂环境中重要基础设施，需要寻找内生安全的硬件木马防护方案。为此，本文提出了一种具备木马主动防御能力的以太网报文解析器结构：解析器由多个基本可重构处理单元和可重构互连组成。功能电路在芯片中的物理位置通过加工后的编程定义，这使得攻击者在木马植入时（加工和硬件设计阶段）的木马放置位置带有盲目性。而且这种结构中功能电路的位置在工作中随机地动态改变，这进一步大幅降低了木马成功激活和成功攻击的概率。上述方法是一种主动的防御策略，不依赖于硬件木马的类型、不需要宿主设计的“黄金模型”。

目前关于硬件木马防御策略，研究最多的是木马检测技术 [2]。但是这些检测技术通常都需要建立一个宿主设计的标准参考模型（黄金模

基金项目：国家核高基重大专项“高安全等级网络基础设施关键设备核心芯片及软件研发”
(2017ZX01030301)

型）。参考文献［3］建议 IP 供应商同时提供一个形式化证明用于自我证明无恶意后门。参考文献［4］建议使用移动目标的方法来防御木马植入攻击，但是它是在 FPGA 平台上实现， 利用的是 FPGA 芯片的可编程性，而高性能交换机普遍使用专用集成电路，FPGA 的性能难以满足要求。虽然近年研究中很多报文解析器或者交换机芯片采用了基于 FPGA 或者可重构硬件的实现方案［5-8］，但是主要面向软件定义网络所需的灵活性，而非安全性需求。因此，本文面向高安全交换机芯片需求，设计了一种可以动态改变功能电路物理位置的交换机芯片报文解析器专用集成电路。据我们所知，这项工作是第一个通过动态重构方式实现的抗硬件木马的报文解析器专用芯片方案。

2　动态可重构解析器结构

2.1　解析器总体结构

本文设计的解析器支持典型的三层（L3）以太网交换机的基本功能，支持 4 种基本以太网帧格式

（Ethernet II， Ethernet SNAP， Novel Ethernet 和 IEEE 802.3）、802.1Q、 QinQ、 IPv4 和 IPv6 协议［9］。从功能角度讲，它负责提取转发表查找所需的关键字，然后发送给交换机的查找引擎（多个查找表），通过查表确定接下来对数据包的处理。处理流程经历的步骤最多可以分为下面 5 步：

（1） 关键字提取：在三层交换机中，需要提取的关键字包括目的 MAC 地址、源 MAC 地址、第一层标签、目的 IP 地址、源 IP 地址，VLAN 标签；判断报文的以太网帧格式类型（共有四种类型），带有的 VLAN 标签层数，网络层的数据类型（IPv4、IPv6 或者其他）。

（2） 二层（L2）表学习（第一次查表）：通过源 MAC 和 VLAN 标签对第二层转发表进行搜索，如果没有找到对应表项，则将源 MAC、VLAN 标签和端口号的对应关系添加到第二层转发表中，方便以后进行转发。

（3） L2 表查找（第二次查表）：以目的 MAC 地址和 VLAN 进行查表，如果 L2 表没有找到对应表项，则根据交换机的设定进行广播或者丢弃。如果找到了对应的表项，而且发现对应的目的 MAC 地址不是交换机的虚拟接口 MAC 地址，依照表项中的目的端口进行转发；如果发现对应的目的 MAC 地址是交换机虚拟接口的 MAC 地址（这时在表项中 L3 位为 1），则表示该报文是发送端传送给网关（L3 交换机），希望通过网关对报文进行转发，这时通过第二层交换就无法完成，搜索引擎将这一消息传送回，该模块接收到消息后，就将关键字传送到下一级流水线。

（4） 三层（L3）表学习（第三次查表）：将源 MAC、VLAN 和源 IP 发送到搜索引擎，搜索引擎对第三层转发表（L3 转发表）进行学习，这个学习过程完成后，反馈给第三次查表关键字生成模块。

（5） L3 表查找（第四次查表）：查表关键字目的 IP，作为第三层转发表的查表索引，如果在 L3

转发表中找到了对应表项，则进行第三层转发，如果没有找到，则将报文发送给 CPU，做软件路由。

这些步骤以流水线的形式完成。如图 1 所示，整个入口解析硬件结构可以分为各步的关键字提取和表

查找/学习两个主要部分，其中的各个查找表的查找和学习由搜索引擎模块完成。在硬件结构上，上面 5

步对应 5 级流水线，每级流水生成不同的关键字输出给搜索引擎，取回搜索引擎的反馈结果给下一级流水， 下一级流水根据上一级的结果生成下一级提交给搜索引擎的关键字。本文主要设计了上述各个步骤中的查表关键字的提取电路——即图中方框内的部分。

2.2　动态重构的关键字提取硬件结构

动态重构解析器的关键字提取结构如图 2 所示。它的核心是一个 4×4 的可重构基本处理单元（processing element， PE）阵列。每个 PE 可以被配置成上述 5 步中的任一个功能或空闲模式。每个 PE 与相邻的 4 个 PE 之间都有连接（阵列边缘的 PE 除外）。此外，所有 PE 都和整个模块的输入数据总线和所有查找表的入口总线相连。这样，它们中的任意一个都可以被配置成第一级（接收来自输入数据总线的数据）； 也可以访问任意一个查找表。对于每个查找表的每个访问入口（查找

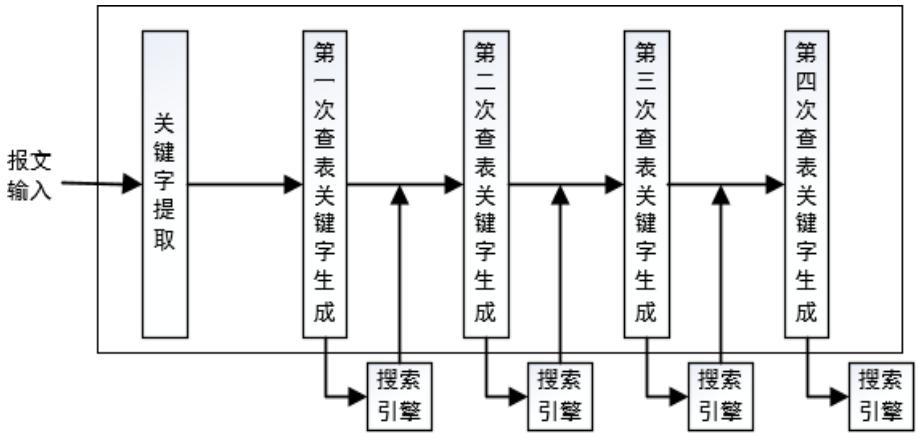


图1　报文解析流水线

和学习被看作不同的访问入口）在任意时刻总是有且仅有一个PE 可以访问。配置之后，上述每级流水线对应一个 PE。因为有四个阶段的查表

（包括学习），所以查表总线共有四组，分别连接搜索引擎的四个访问入口。对于同一个表同时来自查找和学习访问入口的访问请求，搜索引擎进行调度实现分时访问。PE 间的互连位宽根据可能提取的关键字的最大宽度设置，为 129 位（包括一个有效标志位），PE 间交换关键字时每次输出一个关键字。每一个PE 都与配置和控制模块相连，配置和控制模块不仅配置模块的功能，还会配置模块的互连。

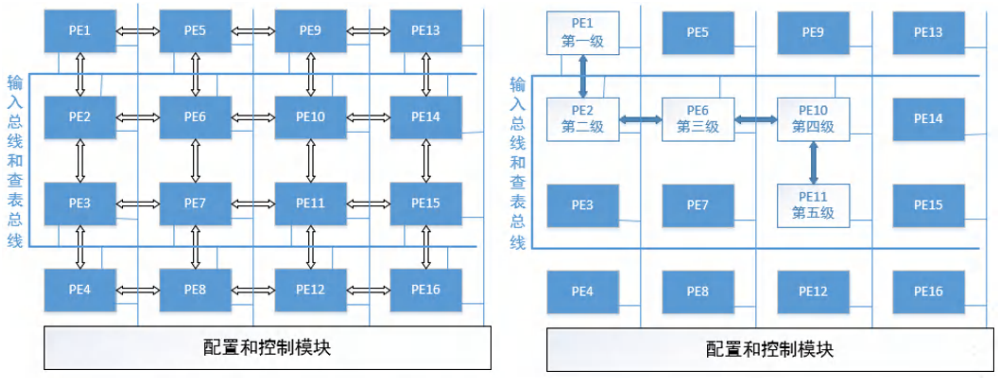


图2 动态重构报文解析器硬件结构（左：硬件结构示意图；右：配置效果示例）

例如图 2 右图所示：PE1，PE2，PE6，PE10，PE11 构成了流水线，其余模块被配置成空闲状态。PE1 与 PE2，PE2 与 PE6，PE6 与 PE10，PE10 与 PE11 之间的互连是有效的，其余模块与模块之间的互连是无效的。同时，只有 PE1 与输入总线的连接是有效地，其余模块与输入总线之间的连接均是无效的。PE2， PE6，PE10，PE11 分别连接到对应的查表总线上。

工作过程中，每隔一段时间，配置和控制模块会对阵列进行重新配置，改变各个模块的功能和互连， 从而实现在电路工作的过程中动态改变有效电路逻辑的目的，实现了电路的动态性。

可重构设计会动态改变电路的有效逻辑，只有结合硬件结构和配置信息才能最终确定各个 PE 的功能， 而在最终的设计之中，配置和控制模块将会引入随机数产生电路，进而增加电路的不可预测性。在后端设计和加工的环节中，将无法确定有效逻辑，从而增加了硬件木马的植入难度。

除此之外，芯片在工作中利用动态重构，有效逻辑在芯片的物理位置不断变化，而事先植入的硬件木马物理位置是固定的，在电路动态重构的情况下木马无法跟踪锁定电路的有效逻辑，也就无法进行恶意操作，大大提高了安全性。木马攻击往往是在特定时间对逻辑空间中的特定的位置实施的，采用这一方案， 即使在芯片中还存在着未被检测到的硬件木马，仍可破坏硬件木马生

效的时-空条件。

2.3 PE 设计

图 1 的每级流水线在电路中是分多个时钟周期完成，一步操作依次包括如下几个状态：

数据交换：前后 PE 之间进行数据传递，每个 PE 接收上级数据，同时向下级发送数据；

关键字生成：完成PE 的主要功能，从输入数据中生成查表关键字，发送给搜索引擎；

结果取回：接收搜索引擎的返回数据，生成发往下级的数据（关键字提取方案和查表结果）准备数据交换。

注意：由于上述 5 步并不是对每个报文都需要全部执行，如果上一级流水线通过查表得到了转发决策，那么它在下一个流水线周期将不需要把数据传送给下一级流水线和不再进行后续查表。因 PE 彼此之间的数据串行传输，故数据交换过程需要持续多个周期。关键字生成过程在数据接收的过程中同时进行。

当PE 被配置成空闲状态，则内部所有寄存器都清零，所有互连都失效。如果 PE 被配置成关键字提取，则根据网络协议格式，依次提取源和目的 IP，MAC，VLAN，报文类型等关键字信息传递给后级；如果 PE 被配置成查表关键字生成，则除了完成相应的报文数据传递外，还需要根据配置信息里的配置参数将查表需要的关键字依次传送到搜索引擎。并在搜索引擎完成搜索后，根据从搜索引擎返回的结果判断是否需要将数据传送给流水线的下一级。在 PE 单元中，不同功能模式下输入数据和输出数据缓存被复用，各自的关键字生成逻辑相对独立，通过配置码选择有效的生成逻辑，实现了功能和功能参数的可配置性。每个 PE 包括输入总线端口、4 个查找总线端口、上下左右四个相邻 PE 端口，其中各个端口是否有效通过配置信息控制，无效端口的控制和通信信号被钳位在无效值。

3 功能配置方法

3.1 配置文件

PE 阵列的配置信息包括位置信息，功能信息和互连信息三部分。其中每个 PE 的位置信息是在 Verilog 代码通过参数设置的静态信息，影响实际生成的硬件电路的端口连接情况。功能信息和互连信息由配置单元在加工后动态写入 PE 单元中的配置寄存器。功能配置信息一共 48 比特，每个 PE 3 比特；互连信息一共 64 比特，每个 PE 4 比特。在设计阶段会事先生成所有合理的配置文件，存储在芯片内的 ROM 存储器中，在工作时由配置与控制模块调用，配置文件都是在一个流水线周期结束的时候加载。具体地，每隔一定的配置周期，配置单元就会生成一个随机的 ROM 地址，使用该地址所对应的配置文件重新配置 PE 阵列。

3.2 配置文件生成算法

下面介绍生成所有满足条件的配置信息的算法：

采用穷举的方式来完成。将 4X4 阵列中的所有 PE 进行编号，第 1 行 1 列的 PE 为 PE1，第 2 行 1 列的PE 为PE2，依次编号到PE16。如果编号为x，y 的两个 PE 相邻，它们的编号一定满足以下条件：

x = y+1 或者 x = y-1 或者 x = y+4 或者 x = y-4 且（x，y）$\notin$ {（4，5），（5，4），（8，9），（9，8），（12，13），（13，12）}

配置信息的生成算法如图 3 所示

图中的“a1，a2，a3，a4，a5 满足相邻条件且各不相同”是指 a1 与 a2、a2 与 a3、a3 与 a4、a4 与 a5 都满足相邻条件，且这 5 个数不相同。由此就得到了流水线中的 PE（有效 PE）的 5 个编号分别为 a1，a2，a3，a4，a5。这 5 个PE 依次被配置为对应功能，其余编号的 PE 则被配置为空闲态。

互连配置信息可以由编号 a1~a5 推算得到。如果 a1-a2 为 1，说明编号为 a1 的PE 在编号为 a2 的 PE 下面，但编号为 a1 的PE 是流水线的第一级，按照数据流的方向，数据从编号为 a1 的PE 流向编号为 a2 的PE，所以编号为 a1 的PE 向上输出为有效，其他方向的输出为无效；同理，如果 a1-a2 = -1，编号为 a1 的PE 向下输出有效，其余方向输出无效；如果 a1-a2=4，编号为 a1 的 PE 向左输出有效，其余方向输出无效；如果a1-a2= -4，编号为 a1 的PE 向右输出有效，其余方向输出无效。同理可以计算出编号为 a2 到 a5 的互连信息。

利用 MATLAB 完成上述算法，得到功能配置信息 config_data 和互连配置信息 config_conncet。将得到的配置信息分别写入两个文件中，生成

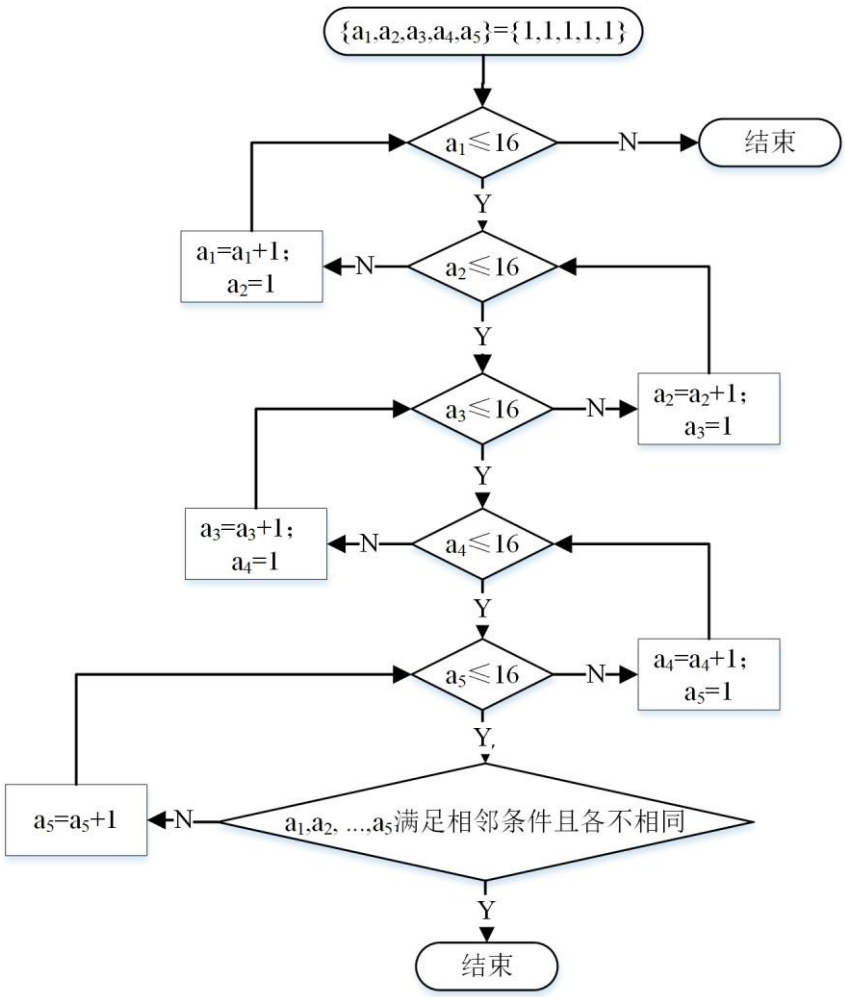


图3　报文解析器PE阵列配置文件生成算法

ROM 时作为数据文件。最终符合要求的配置信息共有 432 条。

3.3　配置切换

PE 阵列的配置需要动态切换。为避免在流水线功能切换后原流水线中的报文丢失。本设计采用“新路径配置——新旧路径同时工作——新路径工作”的三步骤方式进行。刚发生功能切换时，旧路径的各级流水线都在进行报文处理，而新路径的流水线因为还没有数据进入，所以虽然被配置成相应功能，但是并

没有对报文进行处理。这样，在新旧路径同时工作时，相应的模块和互连都是有效的，经过四个传输周期后，当旧模块中也有报文正在处理，而且新模块与旧模块处理的报文也是同一个报文，再撤销旧路径，就不会引起报文丢失的错误。

上述切换方案要求不能有同一个模块在新旧路径中被配置成不同的流水线级数。因此，配置模块在进行配置的时候会进行判断，如果新旧路径中有同一个模块，而且该模块在新旧路径中还处于不同的流水线级数，就将这种配置方案放弃，切换至下一个配置方案。图 4 是一个具体的配置切换示例，原来 PE1，PE2， PE6，PE10，PE11 被配置为流水线功能（级号用 1，2，3，4，5 表示），在新的配置信息中，PE9，PE5，PE6，PE7，PE8 被配置为流水线功能（级号用 A，B，C，D，E 表示），那么在新旧两次配置中，PE6 都处在流水线的第 3 级。完成新配置下载后，新旧路径同时工作时，PE6 的互连信息也如图所示，同时从 PE2，PE5 中接收数据，将两部分数据相或，作为真正的输入，同时将数据输出到 PE7 和 PE10 中，不会发生数据流冲突和错误，所以这样的配置信息是有效的。然后待工作 4 个流水线周期后，再将旧路径的配置信息改为空闲， 关闭 PE6 与 PE10 和 PE2 的连接。

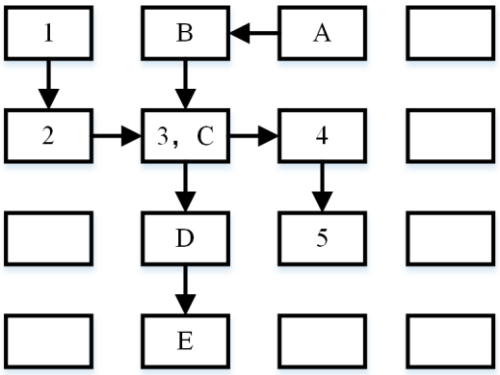


图4　配置切换示例

图 5 是各个 PE 的功能寄存器（Func_me）信号的仿真波形，波形图中的数字表示 PE 的功能（在流水线中的级数，0 表示空闲状态）。从波形图中可以看到一开始 PE1，PE2，PE3，PE4，PE8 被分别配置为流水线的第一级到第五级。经过新的配置后，PE14，PE13，PE9，PE10，PE11 被配置为流水线。从图中，我们可以看到当进行重新配置时，是按照“旧路径——新旧路径同时工作——新路径”的方案进行的。如图虚线框内所示，在新旧路径同时工作时，PE1，PE2，PE3，PE4，PE8 和 PE14，PE13，PE9，PE10，PE11 同时被配置为流水线功能，这与我们的要求是一致的。

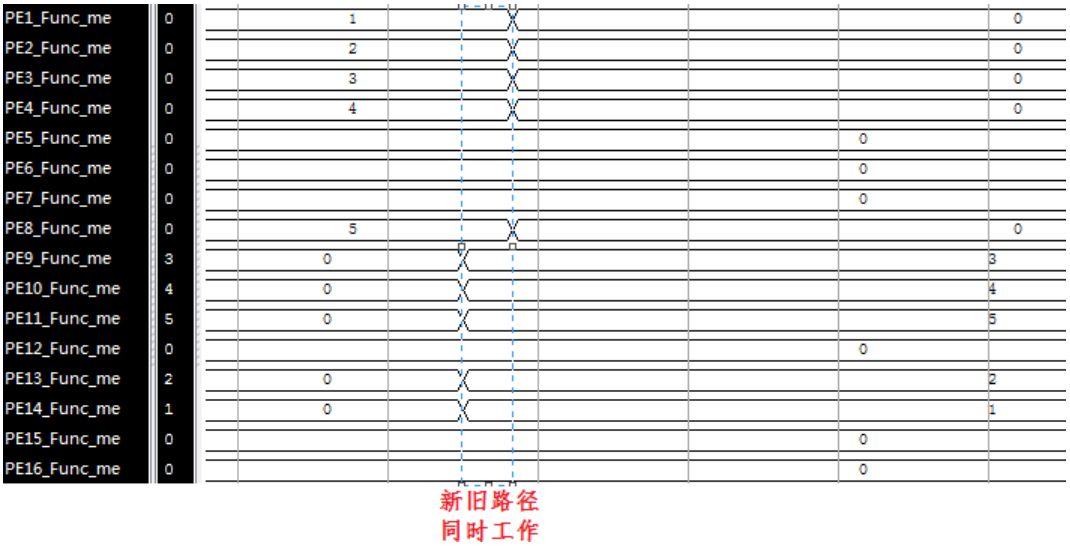


图5 PE配置切换过程

4　安全性分析

假设攻击者在加工阶段随机的在某一个 PE 中放置了针对某一步解析功能的木马，那么在运行时木马攻击成功需要满足攻击时该 PE 的配置与木马攻击功能一致。假设针对第 1 级流水线的木马植入在 PE6 中，根据统计，在上述 432 种可能的配置中 PE6 被配置为第 1 级流水线功能的情况共有 35 种，因此如果任一时刻 PE 阵列处于每种配置的概率相同，那么木马的攻击成功率为 35/432（≈8.1%），而其它边缘位置的 PE 对应于特定功能出现的概率则更低。上述数值可以随着冗余度的提高进一步降低。

对于那些分步实施的攻击，如果攻击持续时间超过了配置的更新周期，那么就可能在攻击过程中硬件发生了重构，攻击成功率则会进一步降低。

5　功能验证

我们编写了报文解析器的关键字生成流水线部分的 Verilog 代码，搜索引擎采用行为级模型搭建测试环境。我们首先使用网络嗅探器软件 WireShark 作为基准模型，通过仿真验证电路的报文解析功能，测试数据采用网络上抓取的真实数据包。实验表明解析器提取的各个关键字段与 WireShark 的输出一致，证明了设计的功能正确性。

我们将所完成的设计下载到 Xilinx 公司的 Artix-7-100T FPGA 开发板中进行 FPGA 验证，通过另一块FPGA 模拟搜索引擎和被测设计交互，驱动流水线正常工作，记录每次接收到的查表关键字，检查正确性。我们使用 chipscope 工具抓取查表关

键字信号，以查表使能信号作为触发条件，用 8 个窗口来观察，每个窗口采样两个数据。图 6 为一个测试报文的仿真结果。其中图 6（a）是对应的波形图的前 3 个窗口，图 6（b） 是 8 个窗口中的采样数据，lookup_en1 到 lookup4_en 分别表示四个阶段的查表关键字输出使能信号，data1 到 data4 是四个阶段的查表关键字。根据波形图和采样数据，可以知道第一个报文的源 MAC 地址为 0xFCAA1404A4F7， 目 的 MAC 地 址 为 0xFFFFFFFFFFFF，没有 VLAN 标签，源 IP 地址为 0xA66F4DA3， 即 166.111.77.163，目的 IP 地址为 0XA66F4DFF，即 166.111.77.255。这与我们利用 Wireshark 抓取到的数

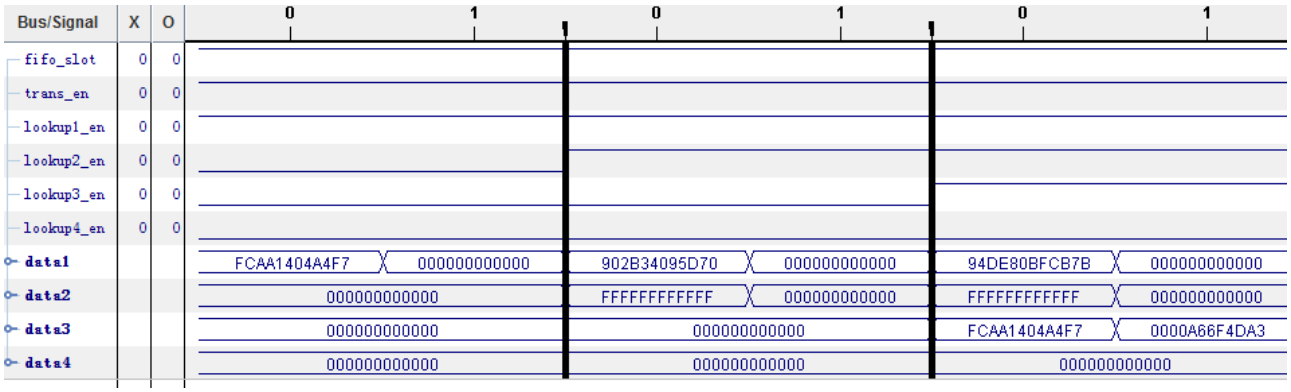
据（见图 7）是一样的。同理分析剩下的数据，发现都是与 Wireshark 中的结果是一致的。四次查表关键字依次是目的 MAC 地址+VLAN 标签，源 MAC 地址+VLAN 标签，源 IP 地址+源 MAC 地址，目的 IP 地址，由流水线的第二级到第五级按照流水线的模式依次生成。表明功能正确。

为了对设计的性能进行评估，我们使用 Design Compiler 对完成的设计进行综合，工艺库使用 TSMC

32nm 标准单元库。综合后的静态时序分析结果是：关键路径的最大延时等于 0.8 ns。由于最终报文解析的整体性能还与搜索引擎的响应速度和报文内容相关，仅由所实现的功能无法估计实际的报文解析延时，因此我们没有进行报文解析的延时估计，仅通过关键路径分析给出了所实现的模块对芯片整体性能的影响， 当芯片时钟频率低于 1.25 GHz 时，本文模块都不会成为限制时钟频率的瓶颈。

6 结束语

本文提出了一种基于动态重构技术防御硬件木马攻击的交换机报文解析器结构，并在一个支持基本常见以太网协议的报文解析器中采用该结构实现了各级查找表访问的关键字生成电路。基于 16：5 的冗余度的可重构 PE 阵列和随机配置技术实现了 432 种可能的配置变化，木马攻击的成功率降至 8.1% 以下。本文工作初步给出了动态重构报文解析器的实现方案，其创新性在于通过动态冗余的规则结构提高硬件的不确定性来防御硬件木马，不同于传统的被动的木马检测方式；而且本文基于专用集成电路给出了具体的实现方案，其实现效率大大优于 FPGA，使得该技术可以运用于高性能交换机芯片。当然，本文只是一个初期工作， 接下来还有一些工作需要做：包括设计搜索引擎，形成完整的交换机入口解析模块，进行准确的性能评估； 目前作为示范设计，其支持的协议类型有限，用于使用交换机芯片还需要进一步增加支持的协议类型和功能；降低可重构阵列的电路开销；优化配置更改的算法，减少产生非法配置，重新选取配置的情况。



（a） FPGA 仿真波形（查表关键字）

Sample	data1	data2	data3	data4
0	FCAA1404A4F7	000000000000	000000000000	000000000000
1	000000000000	000000000000	000000000000	000000000000
0	902B34095D70	FFFFFFFFFFFF	000000000000	000000000000
1	000000000000	000000000000	000000000000	000000000000
0	94DE80BFCB7B	FFFFFFFFFFFF	FCAA1404A4F7	000000000000
1	000000000000	000000000000	0000A66F4DA3	000000000000
0	94DE80BFCB7B	FFFFFFFFFFFF	902B34095D70	0000A66F4DFF
1	000000000000	000000000000	0000A66F4D6A	0000A66F4DFF
0	94DE80BFCB7B	01005E0000FC	94DE80BFCB7B	0000A66F4DFF
1	000000000000	000000000000	0000A66F4FB2	0000A66F4DFF
0	94DE80BFCB7B	333300010003	94DE80BFCB7B	0000A66F4FFF
1	000000000000	000000000000	0000A66F4FB2	0000A66F4FFF
0	94DE80BFCB7B	01005E0000FC	94DE80BFCB7B	0000E00000FC
1	000000000000	000000000000	000000000000	0000E00000FC
0	902B34095D70	333300010003	94DE80BFCB7B	0000D0A0DAF1
1	000000000000	000000000000	0000A66F4FB2	0000D0A0DAF1

（b） ChipScope 采样的查表关键字数据图6 FPGA 仿真结果

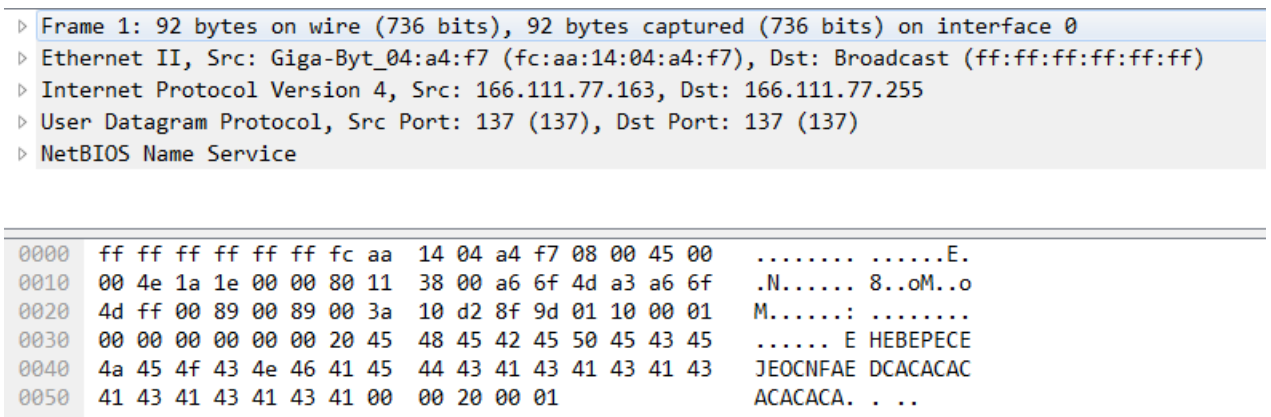


图7 图6报文由 Wireshark 解析的结果

参考文献:

[1] TEHRANIPOOR M, KOUSHANFAR F. A survey of hardware Trojan taxonomy and detection [J]. IEEE Design & Test of Computers, 2010, 27(1):10-25.

[2] HAIDER S K, JIN C, AHMAD C, et al. Advancing the state-of-the-art in hardware trojans detection [J] IEEE Transactions on Dependable Secure Computing. 2019, 16(1):18-32.

[3] LOVE E, JIN Y, MAKRIS Y. Proof-carrying hardware intellectual property: A pathway to trusted module acquisition [J]. IEEE Transactions on Information Forensics and Security, 2012, 7(1): 25-40.

[4] ZHANG Z, NJILLA L, KAMHOUA C A, et al. Thwarting security threats from malicious FPGA tools with novel FPGA-oriented moving target defense[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems，2019，27(3)：665-678.

[5] Yazdinejad A, Bohlooli A, Jamshidi K. P4 to SDNet: Automatic generation of an efficient protocol-independent packet parser on reconfigurable hardware[C]// International Conference on Computer and Knowledge Engineering. Mashhad：ICCKE，2018：159-164.

[6] Lee C, Chen R, Hsiao C. Reconfigurable parser architecture design with microprogrammed controller for multiple purposes[J]. Journal of Signal Processing Systems，2017，88：67-81.

[7] Chen Q, Mishra V, Zervas G. Reconfigurable computing for network function virtualization: A protocol independent switch [C]// International Conference on ReConFigurable Computing and FPGAs. Cancun：ReConFig，2016：1-6.

[8] Zhao Y, Li X Y, Yin S J. Reconfigurable parser for software defined network L4 Ethernet switch chip[C]//IEEE International Conference on Integrated Circuits，Technologies and Applications. Chengdu: ICTA，2019：81-82.

[9] 李聪聪. 以太网三层交换机硬件系统研究[D]. 长沙：湖南大学，2011.
LI C C. The research of layer 3 Ethernet switch hardware system [D]. Changsha：Hunan University，2011.

[作者简介]

李翔宇（1977—），男，博士，副研究员，主要研究方向：硬件安全，物联网芯片。

周飞飞（1993—），男，硕士，研究生，主要研究方向：网络交换机芯片，人工智能。

带结构约束描述的深度优先基因表达式编程

陈哲皓何

[1]广州大学计算机科学与网络工程学院，广东.广州 510006；

[2]北京大学计算语言研究所，北京市 100080；

[3]湖南工业职业技术学院信息工程学院，湖南.长沙 410208

摘　要：基因表达式编程（GEP）作为仿生进化算法的一种，因其编码简单却能解决复杂的问题而广泛运用于大数据分析、多目标优化等应用领域。现有 GEP 的基因型某种意义上多为随机产生，这使得种群中存在大量的无效个体，严重影响了进化的性能。本文初步将领域知识引入 GEP，提出了一种新型的 GEP 改进算法。它主要利用一个描述矩阵来约束和指导基因生成及种群的进化，因而使得 GEP 在收敛速度与解的精度方面有所改善。正如所设想的那样，这些优势与相关特质在与处理回归问题的部分方法进行比较实验和模型分析时得到印证，如与经典 GEP 相比，新方法在精度方面平均提升约 30%。此外运用本文的方法也可以挖掘基因关联关系，找寻特定问题的相关领域知识。

关键词：基因表达式编程、符号回归、解码方法、结构约束、关联关系

Depth-first gene expression programming with structural constraints

CHEN Zhehao

1.School of Computer Science and Cyber Engineering,Guangzhou University,Guangzhou 510006,China;

2.Institute of Computational Linguistics, Peking University, Peking 100080, China;

3.School of Information Engineering of Hunan Industry Polytechnic Changsha 410208,China

Abstract: As a kind of bionic evolutionary algorithm, gene expression programming (GEP) is widely used in big data analysis, multi-objective optimization, and other application fields because of its simple coding to solve complex problems. The genotypes of most existing GEPs in some sense were randomly generated, therefore resulting in a large number of invalid individuals in the population and seriously affecting the evolutionary performance. In this paper, domain knowledge is initially introduced into GEP, and a new improved GEP algorithm using a delineation matrix to constrain and guide both gene generations and population evolutions was proposed for improving its performance in convergence speed and solutions accuracy. These advantages and related properties, as imagined, are demonstrated through comparisons with some existing GEP methods over regression problems and further certified in model analysis, for example, the accuracy of the novel approach, compared with the canonical GEP, is averagely improved by about 30%. Besides, the method presented in this paper can also be used to mine the association relationship of genes and find the related domain knowledge of specific problems.

Key words: Gene Expression Programming; Symbolic Regression, Decoding Method, Structural Constraint, Association Relationship

1　引言

启发式算法是求解许多最优化问题的重要方法之一。它通常能在较短时间得到一个较优的解，并且所仰赖的计算设备算力越强，得到较优解的速度越快，因此在计算技术高度发展的今天，它

基金项目：国家自然科学基金资助项目（基金号 61977018, 60173005），湖南省自然科学基金项目科教联合（基金号 2020JJ7035），湖南省教育厅科学研究项目（基金号 19C0611）

可为复杂现实问题的求解提供极具参考价值的策略与思路。

遗传程序设计（Genetic Programming，GP）即为一类启发式算法。它由 Koza 于 1992 年提出[1]，起初旨在依据观测数据自动拟合建模，实现自动程序设计，后经多方改进陆续推出了包括基因表达式编程（Gene Expression Programming，GEP）[2-11]、多表达式程序设计（Multi Expression Programming，MEP）[3，13-14]、文法演化（Grammatical Evolution，GE）[15]、笛卡尔遗传程序设计（Cartesian Genetic Programming，CGP）[16]以及霍尔语义型遗传程序设计（Hoare Logic Based Genetic Programming，HGP）[17]等在内的众多变体，并广泛应用于时间序列预测、电路设计、大气环境分析、能源管理、大数据分析、多目标优化等诸多领域。

从算法框架看来，GP 家族均是遗传算法（Genetic Algorithm，GA）[18]的衍生品，在解决实际问题时会因搜索策略是否得当、评估困难费时这一瓶颈而消耗大量计算资源。众所周知，表示方法是 GP 最基本的要素，直接关系到解的刻画，遗传算子以及个体解码方法的设计，因此也是探寻提升性能的重要抓手。本文拟从表示方法和约束描述着手对 GEP 开展相应议题的探究。

GEP 是应用非常普遍的 GP 变体，由于源自 GA，自然兼具 GA 和 GP 之所长，既有 GA 固定长度的线性结构，方便进化操作；又可以较快地转化为 GP 的树状结构，方便语义表现型的表达。与其他 GP 变体相比，简明易用且不失复杂结构和问题的表达力是其主要特色。

围绕 GEP 性能提升并从表示结构入手的工作[10，12，14，18-23]大致聚焦于基因多样性改进、高效评估、语义复用等方面。总体而言，此类代表性工作尤其是利于部分领域知识引入的方法尚不多见。我们知道，GEP 的初始基因型是随机产生的，而现实应用场景中，很多问题又受领域知识的制约。若初始基因型完全随机，必然会花费大量的时间去淘汰那些不符合领域知识的个体。例如，将 GEP 用于逻辑电路设计，我们希望在电路进化中尽量避免非门连续这种意义不大的场景。若没有任何制约，必然会产生大量无效或者无意义的基因片段，即冗余而复杂的无用电路；若在 GEP 中能给出相应的制约，无效基因或者说意义不大的逻辑电路组合将不被创建，也就不会在基因型中出现。这样必然可以消除无效基因的影响从而有效提升GEP 的演化性能。文[19，23]所提出和进行的 ADF（Automatic Defined Function）实践主要专注于复用问题，一定意义上实现了高阶知识的刻画和引入但并未直接刻画结构要素间的关系。考察结构描述层面的工作，较为自然的做法是借助生成文法来实现类型一致性保障或语法合法性约束[24-25]。为此，综观 GEP 的简明

易用特点时，我们更期待解决的问题将是能否找到依旧简单易用但又不乏系统化的结构表达力的方法或刻画模型。

本文提出的带结构约束描述功能的深度优先基因表达式编程（Depth-first Gene Expression Programming with Constraints，DFC-GEP）是对标准 GEP 的改进。它一方面将一定的领域知识转化为限制表，继而阻止不合法的基因生成，另一方面又利用深度优先原则实施基因解码，构造表达式树，从而让基因间的联系更为紧密并有效地制导解的生成与搜索过程。在缺少领域知识的情况下，该方法也可以引入约束/限制表与个体种群协同进化的方式来挖掘约束模型和期待的解。经实验检证，该方法可以加速收敛进程，改善解的质量，是探寻 GEP 性能提升的新途径。

2 GEP 简介

GEP 是 Ferreira [2] 在 2001 年提出的一种 GP 变体。它把经典 GP 的树状个体分为基因型（具有固定长度的线性编码串）和表现型两个层面来认识和处理，因此在重新引入层间的解码转换关系后，既有简洁的线性表示形式又有复杂结构的刻画能力。形式上讲它是一个 7 元式<I，F，T，E，O，P，C>，其中 I 是函数集 F、终端符集 T 上定义的个体空间，E 是适应度评估函数，O 指遗传算子集，P、C 分别代表控制参数集和终止条件。介绍涉及的主要要素及相互关系前，我们先大致交代下 GEP 的总体算法框架：

步骤 1

创建初始种群。

步骤 2

对种群中的所有个体解码，并对当前种群的所有个体进行适应度评估。若满足进化结束的条件， 则转至步骤 5。

步骤 3

遗传进化产生新的种群。

步骤 3.1

选择最优的个体，直接复制到新的种群。

步骤 3.2

通过遗传算子产生新的个体步骤 4 转至步骤 2。

步骤 5

输出最优个体。

2.1 个体的基因型

标准GEP为线性遗传程序设计，其种群由若干个体组成，个体由多个染色体构成，而每个染色体又分为头部基因与尾部基因。根据具体问题可以设置头部基因长度为h，尾部基因长度为 $h*(n-1)+1$（这里的n表示所有函数的最大参数数目）。头部基因一般取自函数集与终端集，尾部基因只为终端集元素。若给定函数集为F= {+，-，*，/，S}（ S 代表sin 函数），终端集为 $T=\{x, y\}$ ，染色体的头部符号集 $H=F\cup T$ ，

1那么染色体 C 的正规式为（ $F\cup T$ ）$h \bullet Th*(n\text{-}1)+1$ ，例如 $C=S+-*/bababa$ 即为 h=5 时的一个符合约定的染色体，下文我们也会用该染色体进行说明与分析。

2.2 个体的表现型

GEP 借鉴了生物学开放解读框架的概念，即基因中前部会被解码为表现型，而最后的可能不会用到。GEP 中的解码分为了两个环节。首先，我们需要将线性的基因型通过广度优先或者深度优先构造成表达式树；然后，通过中序遍历得到相应的表达式；最后，我们得到的这个表达式即为个体的表现型。

2.3 个体的适应度评估

适应度评估是对个体的好坏进行评估，是GEP 模仿自然优胜劣汰的依据。令 T 为训练数据集合，其包含 m 组数据， Ti 表示训练集 T 中第i组的数据， yi 表示在 Ti 的观测值， $y\hat{}i$ 表示在 Ti 通过表达式得到的

估计值， M 是一个常数，则适应度有以下公式成立：

$$f=\sum_{i=1}^{m}(M-|yi-y\hat{}i|) \quad (1)$$

$$f=\sum_{i=1}^{yi}(M-|yi-y\hat{}i|) \quad (2)$$

上述公式是 Ferreira 提出的两种评估模型[26]，其中的式（1）是利用绝对误差来评估，式（2）是利用相对误差来评估。

2.4 复制与选择

在对所有个体评估完适应度之后，GEP 需要生成下一代种群，新种群的生成少不了复制与选择的操作。特别地，为了新的种群不弱于上一代种群，GEP 一般会采用精英保留策略，即把最好的个体直接复制进入下一代种群的操作，这样使得新一代种群都不会比上一代种群差，保证了进化不会向着不好的方向进行。为了生成更好的种群，通常需要一些选择策略，如轮盘选择策略和锦标赛选择策略。轮盘选择法是根据个体适应度得出不同的选中概率，再根据选中概率选择出进行遗传操作的个体；锦标赛选择法是随机地从种群中挑选一定数目的个体，然后将最好的个体选择出来进行遗传操作。实验表明，具有精英保留策略与轮盘选择方法的 GEP 在解决复杂问题时通常具有较好的性能。

2.5 遗传算子

GEP 的个体采用线性的基因结构，故只要保证基因长度不改变，头部由函数符与终端符构成，尾部只为终结符，那么该基因即为合法的基因。GEP 的遗传算子是作用于种群，在种群个体间进行遗传变化，以产生新种群的进化操作。文献[26] 给定了九种基础的遗传算子：选择、变异、倒串、基因变换、IS 插串、RIS 插串、单点重组、两点重组、基因重组。这些遗传算子为种群的进化与最优适应度的收敛取到了重要作用。

3 嵌入约束描述功能的深度优先解码型

GEP

上节介绍了标准 GEP 的预备知识，带约束描述的深度优先基因表达式编程（下文称 DFC-GEP）作为标准 GEP 的改进，和标准 GEP 一样都是线性遗传程序设计，因此都有相同的基因型表述方式，只是在基

因解码上采用的是深度优先解码法则。后续讨论涉及的其他要素一致约定为：采用绝对误差作为适应度评估函数，每代种群采取精英保留策略并采用轮盘选择法选出执行遗传操作的个体，遗传算子来自上述的九种方案。另外，为对比需要，本文涉及新旧架构下的 GEP 方法较多，这里也事先做个简要约定和梳理。

新方法本质为 GEP 的扩展，因为使用的解码法则通常有深度和广度优先之分，它又可以概括为三元式

＜｛DF-GEP，GEP｝，C，E>（标准 GEP 也可记作 BF-GEP，即广度优先模式），各变化形式的关系见表 1， 本文建议使用的是两个深度优先解码法则下的衍生物 DFC-GEP （Depth-first Gene Expression Programming with Structural Constraint） 和 EDFC-GEP （Depth-first Gene Expression Programming with Evolved Structural

Constraint）。

表 1　各类GEP的关系

	C E		
	不用	C	C E
GEP	GEP	BFC-GEP	EBFC-GEP
DF-GEP	DF-GEP	DFC-GEP	EDFC-GEP

* C、E 的三组取值分别代表不用、仅用约束表（对应 C）、限制表和进化都采用。

3.1　带约束描述的深度优先 GEP

标准 GEP 在个体生成上很随机，对于特定问题的处理效果有时不见得很理想，所以我们希望在标准 GEP 的基础上，能加入领域知识的约束。DFC-GEP 以二维数组这样的数据结构存放约束限制矩阵，通过 0 和 1 来约束基因的生成规则。例如，我们的函数集为｛+， -，*，/，*sin*，*cos*，*ln*｝，而我们需要得到的目标函数为

$f（x）= x2$ 。假设领域知识表明｛*sin*，*cos*，*ln*｝在本次符号回归问题中所起作用很小，甚至可能是多余的函数集。因此我们在创建种群前加入限制表的设计，是为了约束多余函数集对本次符号回归的影响。

表 2　限制表

	+	−	*	/	S	C	L	x
+	1	1	1	1	0	0	0	1
−	1	1	1	1	0	0	0	1
*	1	1	1	1	0	0	0	1
/	1	1	1	1	0	0	0	1
S	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0
x	1	1	1	1	0	0	0	1

针对上述例子，我们设计了限制表（表 2）来限制每个染色体中头部基因的生成。该限制表实际上是一个含有函数集与终端集数量之和阶数的矩阵，计算机的存储结构为线性结构，因此该矩阵以二维数组的

形式进行数据存放，我们把存放该矩阵的数组记为 arr。其中该数组的第一个元素（即矩阵中第一行第一列的元素 *arr*［0］［0］＝1）的数值为 1，表示在基因“+”的后面可以生成基因“+”；若数值为 0，则表示在基因“+”的后面不可以生成基因“+”。矩阵中第一行的元素（即 *arr*［0］［］）为｛1，1，1，1，0，0，0，1｝，故表示在此限制表约束下，每个染色体在随机生成头部基因序列时，除 S、C、L 外任何一个基因均允许生成在基因

“+”之后；矩阵中第五行的元素（即 *arr*［4］［］）为｛0，0，0，0，0，0，0，0｝则表示在此限制表约束下，每个个体的

染色体在随机生成头部基因序列时，任何一个基因都不允许生成在基因 S 之后。同理，我们可以对限制表的 L、C 所在行做类似理解。因此，在评估占据主导性的计算花费上，排除了冗余的复杂函数对符号回归的影响，很大程度地降低了运算时间，并加快了收敛速度。

根据上述原理，我们将有领域知识制导的 DFC-GEP 设计如下： 步骤 1 设计带有领域关联知识的限制表。

步骤 2

根据限制表创建初始种群。

步骤 3

对种群中的所有个体深度优先解码，并对当前种群的所有个体进行适应度评估。若满足进化结束的条件，则转至步骤 6。

步骤 4

遗传操作产生符合限制表的新种群

步骤 4.1

选择最优的个体，直接复制到新的种群

步骤 4.2

通过遗传算子产生符合限制表约束的新个体

步骤 5

转至步骤 3

步骤 6

输出最优个体

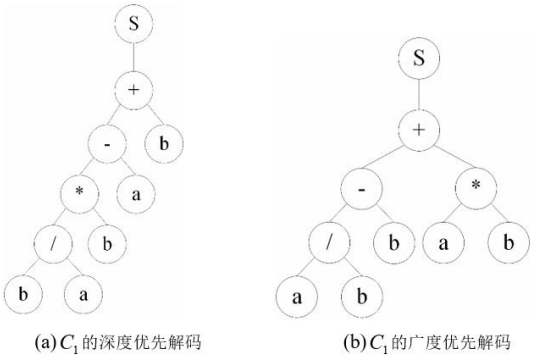
3.2 深度优先解码的优势

解码是 GEP 的一个重要环节，描述了个体的基因型向表现型转化的过程。GEP 的基因型一般是线性结构，其标准解码方法是将线性结构的基因通过广度优先法则转换生成为表达式树，或者具体地讲，它以层为序，从上到下、从左到右，以逐层构造方式完成表达式树的构造，因而易于采用队列技术予以实现。可见，这种方法是十分自然的表达式获取方法，比较适合整体意义上的基因型到表现型的转换，然而欲实施语义的碎片化处理与融合则较为困难，因为其基因序列上比邻基因的语义并非紧密关联的。相反，深度优先构造表达式树的解码方式则使得基因序列上的基因联系更加紧密，在线性序列上的前后基因，尤其是头部基因，经转化为表达式树后基本上能在树结构的同一个语法单位内。

接下来，我们以 $C1 = S + -* /bababa$ 为例并利用图 1 来阐述深度优先解码法则在表达式树构造逻辑上的

优越性。

图 1（b）的广度优先解码表明：基因序列与表达式树之间的对应关系并不紧密。例如个体中基因*后面跟着的是/，而在图 1（b）的表达式树中操作符*与/在相隔很远的树枝上，这使得基因型相邻而表现型相差甚远，在逻辑上显得不够紧密。反观图 1（a）的表达式树，操作符*与/为父子结点。深度优先构造表达式树能在大多数情况下，遵循基因序列的顺序，在位置上相邻的基因解码为表现型时，也尽可能逻辑上相邻。



(a) C_1 的深度优先解码　　(b) C_1 的广度优先解码

图 1 染色体的广度优先与深度优先解码对比

DFC-GEP 的主要思路是用限制表约束不利于种群进化的基因型生成，因此我们希望基因型和表现型之间的关联度越高越好，故深度优先构造表达式树相比起标准 GEP 的广度优先，在此条件下拥有得天独厚的优越性。

除此之外，深度优先构造表达式树还有很多其他的优势，文献［20］深入讨论深度优先解码法则在基因重用、分布并行评估方面的优势。

4 限制表协同进化的深度优先 GEP

从上一节我们知道：限制表的约束实际上是每单个基因对后序基因的约束。限制表一般设计为矩阵， 若某矩阵元素的值为 1，则表示允许在其后生成该基因；若值为 0，表示不允许在其后生成该基因。因此， 标准的 GEP 实际上是限制表中值均为 1 的特例。上节阐述了有领域知识情况下，我们可以通过设计限制表来约束头部基因生成的方式来制导 GEP 进化，从而加快其收敛速度并提高其精度。但是在很多时候，我们可能并不知道相关的领域知识，这样就无法对限制表进行设计。针对这种情况，我们提出了限制表协同进化的概念，带有这一概念的 DFC-GEP 称为限制/约束表协同进化的深度优先基因表达式编程（Depth-first gene expression programming with evolved structural constraint，下称 EDFC-GEP）。EDFC-GEP 的目的是为了给特定的问题或者一类问题找到一个合适的限制表。换句话说，也就是为现有问题找相关的领域知识或是寻找其基因关联规则，以便制导 DFC-GEP 的进化。

4.1 EDFC-GEP 的准备工作

EDFC-GEP 的目的在于搜索最优的限制表。在开始时我们需要设置好相关参数。EDFC-GEP 的大多数参数与标准 GEP 一样，不相同的是它增加了三个参数（协同进化的次数，重复进化次数和进化代数）来限制其进化的时间。

协同进化的次数主要用作判断协同进化结束的标准。刚开始进化的时候很容易找到更好的限制表，随着进化次数增加，限制表进化的阻碍增大，可能需要很多次的进化才得到一个更好的限制表。这与标准GEP 的收敛很相似，标准 GEP 一般会给出运行的最大代数来防止其无法继续收敛而无限循环，故EDFC-GEP 给出限制表协同进化次数这一参数来作为其进化结束的依据以防止其无限循环。在标准 GEP 中，进化代数越多，结果的精度会相对提高；适当的代数，可以达到最快的收敛效果。同理，在 EDFC-GEP 中，限制表进化的次数越多，其得到的限制表肯定会越好，但时间与计算成本会提高；设置适当合理的限制表进化次数，能在相对较短时间内得到一个相对较优的限制表。

重复进化次数是指在限制表进化过程中，每给定一个表需执行 DFC-GEP 的次数。EDFC-GEP 作为 EAs 的一种，每次的进化结果都可能存在一定的随机性。在 EDFC-GEP 中，设置重复进化次数这一参数，是为了多次执行 DFC-GEP，取最优适应度的平均值来评估这个限制表是否更优。这样防止了特殊情况对限制表进化的影响。

进化代数是指 EDFC-GEP 中每次 DFC-GEP 执行的代数。标准 GEP 的优化有两个大方向：一个是优化GEP 的收敛速度，另一个是优化 GEP 的精度。从逻辑上分析，若 EDFC-GEP 中进化的代数的值设置越大， 得到的限制表再用于 DFC-GEP 得到的结果精度越优；若 EDFC-GEP 中进化的代数设置在相对较小的值， 得到的限制表再用于 DFC-GEP 得到结果的收敛速度越快。

不同的参数设置会直接影响到执行 EDFC-GEP 的时间、计算成本与结果，具体的参数设置可以根据具体的问题，并通过实验与经验来确定参数，以便提高 EDFC-GEP 的性能。

4.2 EDFC-GEP 的设计

EDFC-GEP 的原理是不停地进化限制表以得到最优限制表，即找到基因的关联关系。从上述步骤我们可以看出 EDFC-GEP 的本质是在 DFC-GEP 的基础上套上了一层限制表进化，在限制表进化的过程中优胜劣汰：若进化出好的限制表，我们给予保留；若不优于当前的限制表，则选择淘汰。其具体操作是从表元素全为 1 的限制表（即标准 GEP）开始进化，记录初始限制表为当前最优限制表，相应约束下的最优适应度为当前最优适应度。随机修改限制矩阵的一个元素（若为 1 改为 0，若为 0 改为 1）得到新的限制表， 再根据这个限制表去执行 DFC-GEP 和得到本次最优适应度。若本次最优适应度优于既得的最优适应度， 则把当前最优限制表和最优适应度替换为本次进化所得的相应结果，若不优于当前最优适应度，则维持既有结果的现状，并再根据当前最优限制表随机改变一个元素的原则进化出新的限制表，反复上述操作直到达到我们所期待的效果停止，输出当前最优限制表为本次 EDFC-GEP 的结果。

根据以上设计得到 EDFC-GEP 算法的伪代码如表 3。

综上所述，EDFC-GEP 的目的是为了得到相对最优的限制表，这个限制表就是领域知识或者说是基因关联规则的表示。那么 EDFC-GEP 的实质就是一种关联关系的挖掘，这种以限制矩阵表示的限制表不仅能表达相应的关联关系，也可以利用这种关系去制导 DFC-GEP 的进化。

5 实验结果分析

本节针对符号回归问题做了 4 个实验，比较带限制表的深度优先 GEP（DFC-GEP）与带限制表的广度优先 GEP（BFC-GEP）、深度优先解码的 GEP（DF-GEP）、广度优先解码的 GEP（标准 GEP）的最优适应度曲线，并分析这四种 GEP 的进化性能。所选用例取自文献［15， 19］，均为较经典的案例，有一定代表性。实验的变量数据集均为 U［0，1，50］（U［a，b，c］表示从 a 到 b 间取 c 个均匀随机样本），并将四种 GEP 的参

数设置为一致，以免不同的测试数据与参数设置影响实验结果。BFC-GEP 的思想与 DFC-GEP 一致，只是在基因解码时，前者采用标准 GEP 的广度优先解码法则。我们称限制表协同进化的

表 3　EDFC-GEP 算法的伪代码

Algorithm EDFC-GEP 算 法

```
1:restrictionTable ← 初始化限制表 // 限制矩阵全为 1 的限制表,记为当前最优限制表
2:for i= 1 → NCTN do // NCTN 表示限制表协同进化的次数
3:    double bestFitness[NCTN] // 初始化存放每次进化最优适应度数组
4:    for j = 1 → NE do // NE 表示重复进化的次数
5:    种群初始化 // 根据当前最优限制表创建初始种群
6:    for k = 1 → EG do // EG 表示最大进化代数
7:    评估当前种群
8:    新种群 ← 精英个体保留 // 选择最优的个体,直接复制到新的种群
9:    新种群 ← 轮盘选择方法选出个体进行遗传操作 // 通过遗传算子产生符合限制表约束的新个体
10:    当前种群 ← 新种群
11:    end for // 若达到最大进化代数,退出循环
12:    Fitness ← 达到最大进化代数后的最优适应度
13:    bestFitness[j-1] ← Fitness // 得到每次进化的最优适应度
14:    end for // 若达到重复进化的次数,退出循环
15:    aveMinFitness ← sum(bestFitness) /NCTN // 得到此次进化限制表的最优适应度的平均值
16:    if i == 1 or aveMinFitness<bestAveminFitness then // 第一默认为最优,其他时候若优于当前最优则替换当前限制表
17:    bestAveminfitness ← aveMinFitness // 把当前的平均最优适应度记为最优平均适应度
18:    bestTable ← restrictionTable // 把该限制表记录为当前最优限制表
19:    end if // 若不优于当前最优则直接回到循环
20:    restrictionTable ← bestTable // 将限制表改为当前最优限制表
21:end for // 若达到限制表协同进化的次数,退出循环
22:return bestTable // 输出最优限制表
```

BFC-GEP、DFC-GEP 为 EBFC-GEP 和 EDFC-GEP。EDFC-GEP 与 EBFC-GEP 的实验结果分别是先由标准 GEP 进化 50 次得表 5

至表 8 与表 9 至表 12 的限制表，再利用此限制表来运行 DFC-GEP 与 BFC-GEP。实验选择的回归公式与四种 GEP 的参数设置如表 4。

$y = x^4 + x^3 + x^2 + x$ $y = x^5 - 2x^3 + x$

$y = x^6 + x^5 + x^4 + x^3 + x^2 + x$ $y = sinx^2cosx - 1$

（3）

（4）

（5）

（6）

表 11 式 5 的 EBFC-GEP 限制表

观察图 2 的实验结果，我们能明显看出 DFC-GEP 运行 100 代时的收敛速度与回归精度均优于其他方法。这四种方法中，DF-GEP 与标准 GEP 的比较与分析在文献［20］有详细的阐述。我们试着比较 EDFC-GEP 与 EBFC-GEP 的性能（即比较进化出的限制表的优劣）。实验约定参数相同、限制表协同进化的次数均为 50， 这时用 EDFC-

BFC-GEP DFC-GE DF-GEP GEP

函数集	+ − * / S C L	
终结符集	x	
种群大小	20	
染色体个数	3	
头基因长度	15	
最大进化代数	100	
变异率	0.3	
倒串率	0.1	
IS 插串率	0.1	
RIS 插串率	0.1	
基因变换率	0.1	
单点重组率	0.1	
两点重组率	0.1	
基因重组率	0.1	
运行的次数	10	
限制表进化次数	50	无

GEP 与 EBFC-GEP 产生的限制表分别制导 DFC-GEP 与 BFC-GEP 的进化。我们发现 DFC-GEP 的结果普遍偏好，而 BFC-GEP 的结果有时甚至不如标准的 GEP。我们试着分析实验现象，从 4 个协同进化的实验我们发现，EBFC-GEP 在 50 次的限

	+	−	*	/	S	C	L	x
+	1	1	1	1	1	1	1	1
−	1	0	1	1	1	1	1	1
*	1	1	1	1	1	1	1	1
/	1	1	1	0	1	1	1	1
S	1	1	1	1	1	1	1	1
C	1	1	1	1	1	1	0	1
L	0	1	0	1	1	0	1	1
x	1	1	1	1	1	1	1	1

表 5　表 6式 3 的 EDFC-GEP 限制表
式 4 的 EDFC-GEP 限制表

	+	−	*	/	S	C	L	x
+	1	1	1	1	1	1	1	1
−	1	1	0	1	1	0	1	1
*	1	1	0	1	1	1	0	1
/	1	1	1	1	1	1	1	1
S	1	1	1	1	1	1	1	1
C	1	1	1	1	1	1	1	1
L	1	1	1	1	1	1	1	1
x	1	1	1	1	1	1	1	1

表 7　式 5 的 EDFC-GEP 限制表

	+	−	*	/	S	C	L	x
+	1	1	1	1	1	1	1	1
−	1	0	0	1	1	1	1	1
*	1	1	1	1	1	1	1	1
/	1	1	0	1	1	1	1	1
S	1	1	1	1	1	1	1	1
C	0	1	0	1	0	1	1	1
L	1	1	0	1	0	0	1	1
x	1	1	0	1	1	1	1	1

表 9　式 3 的 EBFC-GEP 限制表

	+	−	*	/	S	C	L	x
+	1	1	1	1	1	1	1	1
−	1	1	1	1	1	1	1	1
*	1	1	1	1	1	1	1	1
/	1	1	1	1	1	1	1	1
S	1	1	1	1	1	1	1	1
C	1	1	1	1	0	1	1	1
L	1	1	1	1	1	1	1	1
x	1	1	1	1	1	1	1	1

制表进化中，有效的限制表进化次数（若产生的新限制表优于老限制表，即指有效进化）远低于EDFC-GEP。那么可能的原因如 2.2 小节所述：广度优先解码树的基因型与表现型关联较差（即相

表 8　式 6 的 EDFC-GEP 限制表

	+	−	*	/	S	C	L	x
+	1	1	1	1	1	1	1	1
−	1	1	1	1	1	0	1	1
*	0	1	1	1	1	1	1	1
/	1	1	1	1	1	1	1	1
S	1	1	1	1	1	1	1	1
C	0	1	1	1	1	1	1	1
L	1	0	1	1	1	1	1	1
x	1	1	1	1	1	1	1	1

表 10　式 4 的 EBFC-GEP 限制表

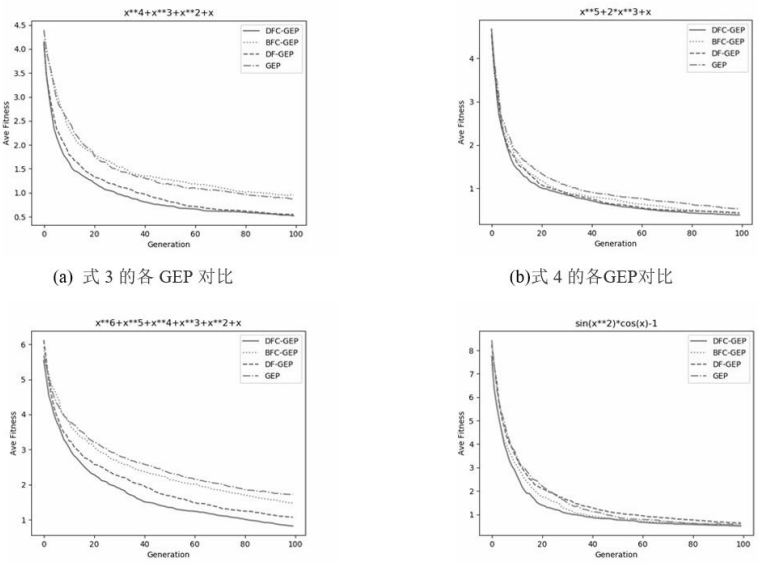
	+	−	*	/	S	C	L	x
+	0	1	1	1	1	1	1	1
−	1	1	1	1	1	1	1	0
*	1	1	1	1	1	1	1	1
/	1	1	1	1	0	1	1	1
S	1	1	1	1	1	1	1	1
C	1	1	1	1	0	1	1	1
L	1	1	1	1	0	1	1	1
x	1	1	1	1	1	1	1	1

表 12　式 6 的 EBFC-GEP 限制表

	+	−	*	/	S	C	L	x
+	0	1	1	1	1	1	1	1
−	1	1	1	1	1	1	1	1
*	1	1	1	1	1	1	1	1
/	1	1	1	1	1	1	1	1
S	1	1	1	0	1	1	1	1
C	1	1	1	1	1	1	1	1
L	1	0	1	1	1	1	1	1
x	1	0	1	1	1	1	1	1

	+	−	*	/	S	C	L	x
+	1	1	1	1	1	1	1	1
−	1	0	1	1	1	1	1	1
*	1	1	1	1	1	1	1	1
/	1	1	1	1	0	1	1	1
S	1	1	1	1	1	1	1	1
C	1	1	1	1	1	1	1	1
L	1	1	0	1	1	1	0	1
x	1	1	1	1	1	1	1	1

邻基因序列，在解码成 ET 树后，逻辑上不相连），而深度优先解码树的基因型与表现型关联较强，因此这种限制表约束的 GEP 更能体现关联规则。若从图像反映的各次实验第 100 代的适应值来看，DFC-GEP 在精度方面比经典 GEP 可平均提升约



(a) 式 3 的各 GEP 对比　(b)式 4 的各GEP对比

(c) 式 5 的各 GEP 对比　(d) 式 6 的各 GEP 对比

图 2 四种 GEP 的实验对比

30%。

6 约束描述模型与解码特色

GP 变体的约束研究尚不多见［27］，就个体描述而言主要分为性质和结构刻画两大类。在知识描述方面，文［17，27］采用谓词逻辑对需求予以性质刻画，并分别依据 Hoare 逻辑和反驳技术进行推演与验证演化所得的解。文［10，19，21-22，24，28-29］等则从个体的结构表示入手，抑或施加合法性限制、抑或引入全新编码/ 解码机制来约束 GEP、GE 等 GP 变体的搜索范围。例如 Quan［21］在处理 GEP 的表示时，除保留常规编码

原则外，还特地加入解码结构的 DAG（有向无环图）约束表式（类似 MEP），因而可以得到更具多样性的语义理解。为提高 GEP 的搜索性能，Ferreira C、Zhong［19，23］借鉴 Koza 对 GP 引入高阶知识的办法，让其表示再包含一些自动定义函数（ADF）和相应指针，相继提出包含高阶知识的 GEP-ADF 以及进一步的改进方案 C-GEP（允许描述参数化的 ADF）。文［24，28］也从结构上对个体的生成过程进行约束和限制，不同之处在于前者用文法来制约和审视GEP 个体的生成与合法性，而后者直接在 2 型文法或相应推演刻画模型（文法模型）上规范程序的生成。代表性的这类 GP 变体有文法演化，以文法模型刻画程序推演序列结构的方法则是有助高性能计算的重要改进，称谓文法演化的模型方法。由此看来，有关 GP 变体的系统化的约束研究与应用还较薄弱，本节拟从约束模型、部分 GP 变体的统一性以及解码特性三个方面来总结讨论所提出的 GEP 方法。

6.1 约束模型

由 2.1 节可知，GEP 的个体全体实为 （$F \cup T$）$h \bullet Th^*$（n-1）+1 所刻画的一个正规语言。每个形式句子的完整解读对应一个可翻译为合法表达式集合的表达式树（ET 树）集。当限定个体语义仅选取为表达式集合的第一元素时，就能获得常规 GEP 系统，而此限制条件下所有合法表达式或可以计算求值的 ET 树自然可由 2 型文法［30］予以形式描述。鉴于此，运用二型文法实施 GEP 的个体合法性约束是可行办法之一［24］，但如此而来还是太过间接，事实上此时直接升级选用基于文法的相应方法如文法演化（GE）等［15］或许更为直接有效。

可见，GEP 和 GE 在形式描述力方面分属不同层次，但简洁、易用且兼具较强的求解力特质委实为前者得以广泛应用加分不少。这些现象启发我们关注了正规式的图描述与支撑技术。本工作旨在继承 GEP 以上特性基础上，依据符号对象的关联关系初步探究其正规式刻画潜力和知识引入问题。

我们知道：个体全体形成了正规集，那么对

其某个子集的刻画即为约束。对此容易联想到的方法是谓词语言，不过这会涉及机器定理证明技术，应用起来会有些困难。所幸的是这样的子集也是正规集（可以证明），因而问题可以转化为正规式或有穷状态自动机模型来刻画，这便形成本文约束矩阵提出的基础。本方法具备的部分性质如下。

性质 1 设M 是 GEP 应用的约束矩阵，则在其有限自乘所刻画的定长符号序列上可以定义正规的个体搜索空间。

6.2 正规式作为统一描述框架

探究正规式上约束问题的另一好处还在于它也是 2 型语言生成过程的描述手段。诚然，2 型文法的描述力强于正规式，可是仔细考察语言的产生式推导序列集时，我们又会发现该集可为产生式集上的正规式所定义［28-29］。故此，2 型语言生成问题在这个意义上与 GEP 个体的正规描述具有形式统一性。基于文法的遗传程序设计如 GE 等 GP 变体尽管有更强表达能力，自然也可与 GEP 一道统一到正规式描述架构上予以一同研究。因此，整合的方法可为部分 GP 变体提供适于同时挖掘约束模型和解的协同进化途径。

6.3 解码特色

GEP 是个体编码满足固定长度准则的线性遗传程序设计方法，评估仰赖的基因型到表现型的转换一般建立在广度优先解码法则基础上。如前所述，本方法将采用深度优先法则来完成有关评估任务。前期工作表明，该法则很好保障了形式与语义的对应一致性（见性质 2），因而支持几乎任意粒度级基因片段的评估与融合，并为开展系统化的高性能研发工作奠定理论基础。

性质 2 设 F、T 分别为函数集和终端集，$\alpha \in (F \cup T)^*$，则 α 在深度优先解码意义下的语义由个体形式结构确定而与评估的划分方案无关，并且可据任意划分方案通过独立、并行和融合计算获得。

由此可见，为 GEP 额外附加一个限制表或约束描述矩阵可方便演化过程的知识引入，进而有效制导合法个体的高效生成；反之，没有相关描述的情况下，我们也可通过协同进化手段来挖掘面向领域的关联知识模型。这些工作虽是面向 GEP 的，但也可进一步泛化到更强表达力的文法描述系统，印证它们的彼此相通性，并最终在理论技术上直接与文法的模型表示理论有机统一。

7 结束语

为增强 GEP 求解复杂问题的能力，本文尝试引入部分领域知识，并借助约束矩阵描述手段提出了结构约束型 GEP。新方法有下述特点：

（1） 支持部分领域知识的刻画与运用：可利用给定的基因关联知识来预测、制导有效基因序列的生成，进而改善 GEP 性能。

（2） 支持部分基因关联关系的挖掘：在缺乏领域知识指导的情况下，可以借助约束的协同进化模式， 同时完成关联知识和解的挖掘与搜索。因此兼具演化制导和关联知识模型挖掘能力的 GEP 在实际应用上将有更广泛的应用价值和前景。

（3） 支持高性能计算与应用：选择深度优先解码法则来实施 GEP 的个体评估工作既能与形式刻画手段高度兼容又能支持几乎任意粒度级基因片段的并行评估、语义重用与融合，是天然的高性能并行计算模型。

未来工作中，我们深入探究和关注的目标包括限制表的协同进化（挖掘）技术、评估中的模式发现与运用、语义重用、高性能并行计算与应用以及与文法演化模型理论间的统一共性问题。当然为充分论证相关结论，我们还须开展更广泛的有效性对比实验。

参考文献:

[1] Koza JR. Genetic Programming: On the Programming of Computers by Means of Natural Selection［M］, MIT Press, Cambridge: MA. 1992.

[2] Ferreira C. Gene Expression Programming: a New Adaptive Algorithm for Solving Problems［J］. Complex Systems, 2001, 13(2): 87-129.

[3] Oltean M, Grosan C. A Comparison of Several Linear Genetic Programming Techniques［J］. Complex Systems, 2003, 14(1): 285-313.

[4] Mostafa M M, El-Masry A A. Oil price forecasting using gene expression programming and artificial neural networks［J］. Economic Modelling, 2016, 54: 40-53.

[5] Deng S, Yuan C, Yang L, et al. Distributed electricity load forecasting model mining based on hybrid gene expression programming and cloud computing［J］. Pattern Recognition Letters, 2018, 109: 72-80.

[6] Azamathulla H M, Rathnayake U, Shatnawi A. Gene expression

programming and artificial neural network to estimate atmospheric temperature in Tabuk, Saudi Arabia［J］. Applied Water Science，2018，8(6)：184.

[7] Hong T, Jeong K, Koo C. An optimized gene expression programming model for forecasting the nationalCO2 emissions in 2030 using the meta heuristic algorithms［J］. Applied Energy，2018，228：808-820.

[8] Yang L, Li K, Zhang W, et al. Optimization of classification algorithm based on gene expression programming［J］. Journal of Ambient Intelligence and Humanized Computing，2017.

[9] Huang Z, Li M, Chousidis C, et al. Schema Theory-Based Data Engineering in Gene Expression Programming for Big Data Analytics［J］. IEEE Transactions on Evolutionary Computation，2018，22(5)：792-804.

[10] Zhong J, Feng L, Ong Y. Gene Expression Programming: A Survey［J］，IEEE Computational Intelligence Magazine，2017，12(3)：54-72.

[11] Tonglin Liu, Hengzhe Zhang, Hu Zhang, Aimin Zhou. Information Fusion in Offspring Generation: A Case Study in Gene Expression Programming［J］，IEEE Access，2020，8(5)：74782-74792

[12] Kataria S, Sangal S, Tyagi T, Aggarwal S. Augmented Gene Expression Programming: A Population Diversifying Paradigm［C］，2018 IEEE Congress on Evolutionary Computation (CEC 2018)，2018

[13] Alavi A H, Gandomi A H, Sahab M G, et al. Multi expression programming: a new approach to formulation of soil classification［J］. Engineering with Computers，2010，26(2)：111-118.

[14] Oltean M, Grosan C, Diosan L, Mihaila C, Genetic Programming with Linear Representation：A Survey［J］，International Journal on Artificial Intelligence Tools，2009，18(2)：197-238

[15] O'Neill M, Ryan C. Grammatical Evolution［J］，IEEE Trans. on Evolutionary Computation，2001，5(4)：349-358

[16] Miller J F. Cartesian Genetic Programming: Its Status and Future［J］，Genetic Programming and Evolvable Machines，2020，21：129－168.

[17] He P, Kang L S, Johnson C G, et al. Hoare logic-based genetic programming［J］. Science China Information Sciences，2011，54(3)：623-637.

[18] Li Q, Cheng H B, Yao M. Adaptive Multi-phenotype Based Gene Expression Programming Algorithm
J］. Chinese Journal of Electronics，2016，25(5)：807-816.

[19] Zhong J, Ong Y, Cai W. Self-Learning Gene Express-ion Programming［J］. IEEE Transactions on Evolutionary Computation，2016，20(1)：65-80.

[20] 邓薇,何锫,钱俊彦．深度优先的多基因表达式程序设计[J]．模式识别与人工智能,2013，26(9)：819-828.

[21] Quan Hui-yun, Yang Guangyi. Gene Expression Programming with DAG Chromosome［C］，//Proceedings of the 2nd International Conference on Advances in Computation and Intelligence (ISICA'07)，LNCS 4683，2007，pp271-275，Berlin：Springer-Verlag.

[22] Chen Y, Li K, Chen Z, et al. Restricted gene expression programming: a new approach for parameter identification inverse problems of partial differential equation[J]. Soft Computing，2017，21(10)：2651-2663.

[23] Ferreira C. Automatically Defined Functions in Gene Expression Programming. In: Nedjah N., Mourelle L M, Abraham A(eds) Genetic Systems Programming. Studies in Computational Intelligence［M］，vol 13. Springer，Berlin，Heidelberg. 2006，pp. 21-56

[24] 唐菀, 杨喜敏, 谢夏, 曹阳. GEP 的网络入侵检测规则约束及演化策略[J]. 华中科技大学学报(自然科学版)，2008，36(11)：60-63

[25] Poli R, Koza J. Genetic Programming，in Burke E. K. Kendall G (eds). Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques［M］，2014，Springer，143-184

[26] Ferreira C. Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence (Studies in Computational Intelligence)［M］. Berlin，Heidelberg: Springer-Verlag，2006.

[27] Bladek I, Krawiec K, Swan J. Counterexample-Driven Genetic Programming: Heuristic Program Synthesis from Formal Specification［J］，Evolutionary Computation，2018，26(3)：441-469

[28] He P, Deng Z L, Gao C Z, Wang X N,, Li J. Model approach to grammatical evolution deep-structured analyzing of model and representation［J］，Soft Computing，2017，21：5413－5423

[29] He P, Colin G. J, Wang H F. Modeling Grammatical Evolution by Automaton［J］，Science China Information Science，2011，54(12)：2544－2553

[30] 左劫, 段磊, 唐良, 巩杰, 唐常杰. 求解复杂约束问题的基因表达式编程文法模型[J]，四川大学学报 (自然科学版)，2009，46(3)：577-582

[作者简介]

陈哲皓（1995 年—） 男，硕士研究生。主要研究方向：人工智能。

何锫（1963 年—）男，博士，教授。主要研究方向：人工智能，计算机软件与理论。

王厚峰（1965 年—）男，博士，教授，博士生导师。主要研究方向：人工智能，计算语言学。

肖卓宇（1979 年—）男，硕士，副教授。主要研究方向：软件工程。

多变体执行环境研究综述

陈玉枚，扈红超，王亚文，王庆丰，迟宇宁
战略支援部队信息工程大学 郑州 中国 450002

摘　要：软件内存漏洞利用已成为网络安全领域的重要威胁之一，并且随着漏洞挖掘技术的飞速发展，传统漏洞修复手段难以为用户提供可持续的安全保障。对此，多变体执行环境（Multi-Variant Execution Environment，MVEE）被提出，该技术利用软件多样化方法从单一软件来源产生出多个功能等价但内存布局、代码逻辑不同的变体，且通过锁步机制实现软件的安全执行。本文对现有多变体执行环境相关的研究工作进行了归纳总结，并从监视器权限、监视器部署模式、I/O操作模式、监视器-变体通信机制、同步点五个方面对现有研究工作进行分类对比，然后对多变体执行环境技术的安全性和性能进行了分析。最后，总结了该领域尚未解决的问题，并提出一种基于动态异构冗余架构的多变体执行环境（Dynamic Heterogeneous and Redundant Multi-Variant Execution Environment， DHRMVEE）设想。

关键词：多变体执行环境、软件多样化、软件内存漏洞

Research on Multi-Variant Execution Environment

CHEN Yumei, HU Hongchao, WANG Yawen, WANG Qingfeng, CHI Yuning
Information Engineering University, Zhengzhou 450002, China

Abstract: The exploitation of software memory vulnerabilities has become one of the important threats in network security. With the rapid development of vulnerability mining technology, traditional vulnerability defenses are difficult to provide users with sustainable security guarantees. In this regard, the Multi-Variant Execution Environment (MVEE) is proposed. This technology utilizes software diversification to generate multiple variants with equivalent functions but different memory layouts and code logics from a single software, and executes the software safely in lock-step. This paper summarizes the research work related to the existing MVEEs, and classifies them from five aspects: monitor privilege, monitor deployment mode, I/O operation mode, monitor-variant communication, and RendezVous Points (RVPs). And then the security and performance of the MVEE are analyzed. Finally, we conclude the limitations and future work in this field, and propose a Dynamic Heterogeneous and Redundant Multi-Variant Execution Environment (DHRMVEE).

Key words: multi-variant execution environment; software diversification; software memory vulnerabilities

1　引言

C/C++实现的软件中存在的内存破坏漏洞构成了漏洞利用者和安全防御者之间的军备竞赛。C/C++对于软件开发是必不可少的，然而，由于不合规范的编程带来了一些固有的局限性，从而导致内存信息泄漏，缓冲区溢出，代码注入［1］甚至系统崩溃等。软件内存安全需要重要的防御机制消除这些漏洞。

在现代操作系统中实现了地址空间布局随机化（Address Space Layout Randomization，ASLR）［2］和数据执行保护（Data Execution Prevention，DEP）［3］，在提供对软件概率保护的同时，以较低的性能开销减少了特定类型的攻击。但是信息泄露、暴力破解可以轻松绕开它们。此外，一些先进的技术，例如Address Sanitizer（ASan）［4］，SoftBound［5］和CETS［6］提供了强大的内存安全保护，控制流完整性（Control-flow Integrity，

基金项目：国家自然科学基金创新研究群体项目（61521003），国家重点研发计划课题（2018YFB0804004）

CFI）[7] 和代码点完整性（Code-point Integrity， CPI）[8] 有效减轻了控制流劫持攻击，Memory Sanitizer（MSan）[9] 可以缓解未初始化读取导致的信息泄漏，而 Undefined Behavior Sanitizer（UBSan）[10] 可以检测未定义行为的原因。但是，就运行开销而言，它们的成本高得令人望而却步。

在 N-version 编程 [11][12] 发展的基础上，由于多核处理器的广泛应用，多变体执行环境（Multi-Variant Execution Environment， MVEE）[13][14] 成为对软件内存漏洞进行有效和全面防御的一种更有吸引力的机制。该技术并行运行多个软件程序异构体，通过锁步机制检测行为差异。用于生成变体的多样化技术 [15] 确保异构变体对恶意输入做出不同的响应，同时使正常执行条件下的相同行为不受影响。MVEE 的监视器在变体锁步执行中同步变体行为，并在攻击者破坏软件之前发现变体执行差异并发出警报。

本文的其余部分介绍了 MVEE 的基本架构（§2），总结了 14 种有代表性的 MVEE 系统（§3），根据 5 个不同的特征对它们的架构进行了分类（§4），描述了产生良性差异的误报处理方法（§5），介绍了安全性分析（§6）和性能比较（§7），讨论了该领域目前存在的局限性和未来研究方向，并提出动态异构冗余多变体执行环境（Dynamic Heterogeneous and Redundant， DHRMVEE）安全架构设想（§8），得出结论（§9）。

2 MVEE 基本架构

MVEE 架构主要由监视器和变体组成，该架构将相同的输入分配给监视器监视的多个变体。这些变体通常在系统调用级别以锁步的方式同步执行同一程序，并在运行良性程序时产生一致的输出，而在遇到攻击时会产生明显的行为差异。当监视器检测到分歧时，会终止进程来阻止攻击。

图 1 显示了 MVEE 的基本架构以及工作流程。

步骤□：监视器启动变体并对其进行异构化改造。

步骤□：监控器监视变体行为，其中包括系统调用号以及相关参数。

步骤□：当监视器确认变体行为一致时，控制流才会返回到变体以执行下一个操作。当检测到不一致时，监视器将终止进程。

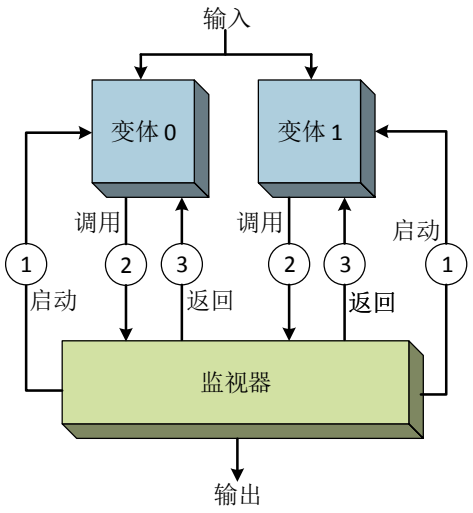


图 1　MVEE 的基本架构

2.1 监视器

监视器在某个特定级别监视和拦截变体的行为：指令/函数级别 [16]、系统调用和显式 I/O 操作，其中，系统调用级别最为普遍。在 Linux 中，每个不同的系统调用序列和参数都可能影响输出，因为系统调用是应用程序进入操作系统内核的入口点，即应用程序必须使用系统调用与系统进行交互。通常，MVEE 只允许执行一次 I/O 操作，以使变体执行过程对终端透明，而在变体锁步执行系统调用时监视其行为以检测分歧。

监视器提供监视功能，在变体执行实际系统调用之前，检查所有变体是否触发同一系统调用并且使用相同的参数；此外，监视器还检测从内核返回给变体的执行结果是否一致。从中检测到任何一个差异时，监视器首先检查是否为误报，如果是，则允许继续执行；如果不是，则触发警报并进行异常处理。

在某些 MVEE 中，监视器负责执行系统调用 [17][18]，例如那些改变系统状态、返回不可变结果且不改变系统状态的系统调用，监视器将执行结果复制给所有变体使得系统对用户透明执行。

2.2 变体

MVEE 的关键组件——变体（Variant）或副本（Replica），它们是同一程序的多个多样化实例。多个变体在监视器的监视下并行执行，对良性输入产生一致的响应，对异常操作产生明显的行为差异。生成异构变体的多样化技术和并行执行的

变体数量为系统提供了重要的安全保证。

多样化技术主要从代码/数据空间布局、调用约定、编译器特性、程序加载地址，甚至指令集本身等方面产生异构变体。变体在这些方面略有不同，对攻击的响应就会有所差异。因此，虽然所有变体都可能易受攻击，但任何漏洞利用都只能针对某一个特定的变体，从而系统能检测到变体受到攻击时产生的行为差异。

变体多样化技术对MVEE缓解内存攻击有很大影响。Larsen等人［15］总结了从机器级［19］、指令级、函数级、程序级和系统级［20］的多样化技术，如表1所示。在MVEE兴起之前，便出现了应用于操作系统的多样化思想。例如，对于缓冲区溢出漏洞，文章［19］［20］分别提出了多样化机器描述（Diversified Machine Descriptions）和系统调用映射随机化（System Call Mapping Randomization）。一些多样化技术也集中在代码多样性上，例如程序级别的指令集随机化（Instruction Set Randomization， ISR）［21］［22］。随着N-Variant系统［13］的出现，更完善的多样化技术应用越来越广泛。反向堆栈（Reverse-stack）［17］［23］［24］使用不同的堆栈增长方向来减轻内存错误，从而在函数级别上产生了堆栈对象寻址多样化。不相交代码布局（Disjoint Code Layout，DCL）［25］确保小工具不会同时出现在多个变体中的相同地址上，全面防御面向返回的编程（Returning-oriented Programming， ROP）攻击。

针对不同类型的漏洞，各种不相交的多样化技术（例如反向堆栈和系统调用映射随机化）可以互相组合［26］，全方位防御不同种类的攻击并增强系统鲁棒性。Double Helix［27］利用不同粒度的多样化技术覆盖更大的攻击面。它利用了三种多样化技术生成变体，包括细粒度（机器级别）、中粒度（编程语言/应用程序级别）和粗粒度（算法级别）。

3 MVEE安全系统

由于多样化技术的广泛应用，普遍提出了不同架构的MVEE安全系统。这些MVEE系统不依赖任何加密技术就足以缓解攻击并消除漏洞利用。本节总结了14种常见的MVEE安全系统。

Cox等人［13］提出了第一个N-Variant系统，该系统同时执行应用地址空间分区（Address Space Partitioning， ASP）或指令集标记（Instruction Set Tagging， IST）的两个变体，并使用监视器检测变体执行时的行为差异。该系统是MVEE的开篇之作，它定义了理想MVEE系统安全特性的等效项和检测机制。它在系统调用级别监视变体，并根据所需的安全等级将267个系统调用概括为三类：共享系统调用，反射系统调用和危险系统调用。实验证明，它可以为系统安全性带来实质性效益，但仅能防御绝对内存地址访问攻击或代码注入攻击。此外，该系统无法正确处理多线程调度程序以及异步信号传递；它也限制使用execve和mmap等系统调用，使用这些调用的进程被异常终止。

DieHard［28］仅同步变体的I/O操作，使用表决机制裁决每个变体的输出内容。如果大多数变体内容一致，则将该内容发送到标准输出后恢复变体执行。它描述了一种随机化内存管理算法，用于在堆布局中随机存储对象，可以有效容忍内存错误并提供概率性内存安全保护。但是，DieHard不能防止基于堆栈损坏的安全错误；无法运行产生非确定性输出（如时间，性能计数器，交互事件等）的应用程序，因为此类程序存在该系统无法处理的误报。

Cavallaro［29］利用监视器执行I/O复制操作，为两个变体提供一致的输出。该MVEE的多样化技术在ASP基础上将一个变体的地址空间“移位”k字节，使得两个变体的地址空间的相对距离被适当地多样化，从而阻止了部分地址覆盖攻击。该系统的局限性在于不能抵抗不涉及指针损坏的攻击，如相对寻址攻击。

Orchestra［17］并行运行堆栈反向生长的两个变体，以防御基于堆栈的攻击。其监视器在非特权用户空间中作为单独进程运行，代替变体执行I/O操作并将结果返回给它们。它正确处理了在多变体执行中可能导致误报的不一致源，包括子进程和线程的调度、信号传递、文件描述符、进程ID、时间和随机数。但是，在变体受到攻击后，Orchestra必须终止损坏的变体，而无法对其进行修复。但它致力于建立这样一个系统，该系统将自动隔离、重新初始化和恢复已损坏的进程。

GHUMVEE［30］是主从模式MVEE的代表，

表1　不同粒度的多样化技术及MVEE广泛使用的多样化技术[15]

多样化粒度	多样化技术	常用多样化技术	MVEE
函数级	堆栈布局随机化		
		堆栈填充	
		堆栈变量序列随机化	
		非连续堆栈分配	KMVX
	函数参数随机化	堆栈反向生长	Orchestra
	内联,整合和拆分		
	控制流展平		
	函数序列随机化		
程序级	程序加密随机化	指令集标记	N-Variant System
		代码布局随机化	
	数据多样性	库入口点随机化	
		结构布局随机化	
		常量加密	
		静态数据随机化	
		堆布局随机化	DieHard, KMVX, MvArmor,
	基地址随机化	地址空间分区	N-Variant System, Cavallaro,
		地址空间布局随机化	REVEN, MvArmor, DMON,
		不相交代码布局	GHUMVEE, ReMon
系统级	系统调用号随机化		
基本块	分支函数插入		
	不透明谓词插入		
循环级	基本块序列随机化		
指令集	等效指令替换		
	等效指令序列		
	指令序列随机化		
	寄存器分配随机化		
	垃圾代码插入		

它只允许主变体执行I/O操作，主变体进而向其他从变体复制I/O结果。它使用基于ASP的不相交代码布局（DCL）来有效地使程序免受控制流劫持漏洞的攻击——面向返回的编程（ROP）攻击，被证明在检测0-day漏洞方面也是高效和灵活的。此外，GHUMVEE能正确处理多线程复制和异步信号传递。但是，该系统被证明可以被just-in-time面向返回的编程（JIT-ROP）[31]攻击绕过，此外，它也无法达到修复受损变体的容错能力，这几乎是现有MVEE共有的挑战。

TACHYON[32]同时运行原程序和补丁后的程序，以进行补丁程序测试。它记录系统调用的语义输入和输出，以便重放它们进行串联执行。当用户通知允许语义差异继续自动补丁测试时，系统重写语义差异，例如忽略或重新排序系统调用。TACHYON还采用了确定性多线程（Deterministic Multi-Threading， DMT）技术，支持多线程处理程序。但是，由于通过虚拟动态链接的共享对象（Virtual Dynamically-linked Shared Objects， vDSO）进行的某些系统调用不会触发ptrace调试工具，因此TACHYON不支持vDSO。

Mx[18]提出了一种可靠的软件更新方法，协调新版本和旧版本程序的并行执行，选择更可靠版本的行为来同步它们的执行结果。Mx在运行过程中实现了崩溃程序容错：复制非崩溃版本的代码，替换崩溃版本中的错误代码，使得崩溃版本在崩溃错误（发SIGSEGV信号的错误）中幸存。

VARAN[33]是一个分布式架构，在每个变体的地址空间中都部署了一个监视器。通过二进制重写（Binary Rewriting， BR）和共享内存环形缓冲区（Shared Memory Ring Buffer）的进程间通

信机制，实现了系统调用的拦截和数据复制。VARAN使用二进制重写处理虚拟系统调用，避免了额外的系统调用开销。此外，该系统将主变体的文件描述符复制到从变体的进程中，使主变体在崩溃时能够透明地被替换。然而，VARAN致力于提高软件可靠性和性能，而不是安全防御。由于VARAN与应用程序驻留在相同的地址空间中，ROP攻击可以轻松绕过它的跟踪机制从而逃避检测。

ReMon［34］利用进程间监视器仅对安全关键的敏感系统调用实施交叉检查，同时支持进程内监视器对非安全关键的系统调用实施更宽松的监视策略。它通过将监视器和复制逻辑拆分成一个跨进程组件（GHUMVEE）和进程内组件（IP-MON）来实现这一点。IP-MON可以在无需切换上下文的情况下高效复制非敏感系统调用。它还部署了一个小的内核代理，该代理定制IP-MON处理的系统调用和宽松监视策略。它比GHUMVEE更具非侵入性、安全性和高效性。

REVEN［27］包括可加载内核模块（Loadable Kernel Module， LKM）、用户级监视器和恢复管理器等组件，类似于N-Variant系统。LKM拦截系统调用以检测变体的行为，并统一变体的输出。监视器充当中间组件，向内核报告变体的状态，并在变体崩溃或检测到变体行为差异时执行恢复管理器指示的异常处理。恢复管理器实现变体恢复协议。该系统在N-Variant系统基础上使用更细粒度的变体多样化技术，并新增了变体异常处理的容错功能。但其涉及大量修改内核的工作，可移植性较低。

MvArmor［35］基于硬件支持的进程虚拟化Dune，将监视器直接部署在每个运行变体的系统调用路径中。监视器被赋予特权级权限以高效而安全的方式监控变体的行为。它会尽可能在Dune模式下执行内存管理，而无需将系统调用转发到内核，因为Dune被授予了对CPU功能的特权访问权限。MvArmor依靠进程虚拟化来有效、安全地将监视器与不受信任的执行操作隔离开来；即使Dune限制了每个进程的最大虚拟内存大小，也可以利用lib OS-style优化来进一步减轻传统MVEE的性能影响。

BUNSHIN［36］联合使用了多种sanitizer多样化技术，如ASAN、MSAN和UBSAN，最大限度地增加了系统防御面且避免了安全机制导致的性能下降。BUNSHIN在LKM中构建临时系统调用表来拦截系统调用，并将结果从内核同步缓冲区中直接返回给变体。其创新之处在于，它将sanitizer安全工具分布在多个程序变体中，将功能不同甚至冲突的安全机制组合在一起，使得目标系统的安全防御面更加广泛。

DMON［37］提出了一种分布式异构MVEE，该系统在多个异构主机间编译和执行变体，每个变体都存在一个单独地址空间的监视器。监视器通过网络在系统调用边界对变体进行交叉检查，监视不同平台间的变体行为差异。它利用了跨平台自然存在的多样性，提高了对内存利用的抵御能力。但是，不同主机上监视器之间的通信成为MVEE系统性能开销的新来源。

KMVX［38］在同一主机上同时运行两个不同的内核变体来防御内核中的信息泄漏。它引入了两个同步点：I/O同步和系统调用同步。I/O同步为共享的硬件提供统一接口，在系统调用同步中检测用户空间边界处内核变体的不同行为。通常，现有MVEE将内核视为安全可信的环境，致力于解决用户空间中运行软件的安全问题。然而，KMVX说明内核仍然存在待解决的漏洞，从而提供了更广阔的研究视野。

4 MVEE分类

在上一节中，介绍了14种主要的MVEE，以便在本节中对其按架构进行分类。现有的MVEE主要分为两类：面向可靠性的MVEE和面向安全的MVEE，分别具有较高性能和较高安全性的特征，如表2所示。面向可靠性的MVEE一方面致力于提高系统的性能，另一方面注重提高系统的稳定性和可用性。它们可以自动生成变体，也可以同时运行同一应用程序的多个版本或者具有同一接口的不同程序。例如，如果软件补丁引入了新的错误和安全漏洞，则在系统升级或打补丁时需要确保系统的稳定性和可靠性。此类MVEE并行运行相同软件的不同版本来避免更新错误。在安全性方面，安全性的提高通常会伴随性能的下降。这些MVEE以适度且可接受的性能开销为代价，同时运行同一程序的不同结构或多样化变体，旨

在消除软件内存漏洞攻击。

表2显示了面向可靠性和安全性的MVEE分类。在此分类的基础上，我们将从监视器权限——特权/非特权级别、监视器部署模式——进程内/进程间、I/O操作模式——主从/代理模式、监视器-变体通信、同步点（RendezVous Points，RVPs）等进一步讨论，对每一类MVEE进行深入分析。

表2　面向可靠性和安全性的MVEE分类

分类	MVEE(时间)				
面向可靠性	TACHYON(2012)	Mx(2013)	VARAN(2013)		
	N-Variant System(2006)	DieHard(2006)	Cavallaro(2007)	Orchestra(2009)	GHUMVEE(2012)
面向安全性	ReMon(2016)	REVEN(2016)	MvArmor(2016)	BUNSHIN(2017)	DMON(2019)
	KMVX(2019)				

4.1　监视器权限

区分MVEE的关键特征之一是监视器运行在特权级（Privileged Level， PL）的内核空间或非特权级（Unprivileged Level， UL）的用户空间中。如表3所示，我们根据非特权级别（UL）和特权级别（PL）给出了MVEE的分类。

内核中实现的监视器具有特权级权限，效率较高，但具有侵入性。该监视器运行在内核中，跟踪用户空间的变体时避免了频繁的上下文切换。但是攻击者一旦攻破运行在内核中的监控器，便可以获取特权级权限，这将威胁到整个系统的安全。MvArmor是一个特例，它利用基于Dune［39］的进程虚拟化技术，通过硬件支持构造与变体完全隔离的监视器，并授予监视器对特权CPU功能的访问权限。所以它也属于这个类别，但监视器被隔离在沙箱环境中，更加安全可靠。

其他MVEE选择在非特权级别（UL）的用户空间部署监视器。这种方法完全依赖于操作系统的标准化调试接口，因此具有可移植性和易于实现的特点。用户空间应用程序的监视器简化了操作系统内核补丁的维护和实现。此外，监视器本身存在的错误不会对系统造成严重的危害，因为监视器是常规的非特权进程，而不是在特权模式下运行的内核补丁或模块。如果监视器被破坏，攻击者在不进行提权的情况下只具备用户级权限。

表3　基于UL/PL的MVEE分类

监视器权限	特征	MVEE
非特权级 UL	用户空间中实现 易于维护和实现,减少可信计算基PCB 频繁的上下文切换	DieHard, Cavallaro, Orchestra, VARAN, Mx, TACHYON, GHUMVEE, DMON, ReMon
特权级 PL	内核/沙箱中实现 避免频繁的上下文切换 增加可信计算基PCB	N-Variant System, REVEN, Mvarmor, BUNSHIN, KMVX, ReMon

4.2　监视器部署模式

为了最大程度地提升性能，监视器与变体的交互应精心设计，因为这种交互频繁发生。根据MVEE的监视器部署模式，将其进一步分为进程内（In-process， IP）和进程间（Cross-process，CP）两类，如表4所示。IP表示监视器在变体进程上下文中运行，或与变体共享相同的地址空间，CP表示监视器作为单独的外部进程运行。

部署在内核中的监视器，与变体共享相同的地址空间。由于大量减少了监视器-变体上下文切换的开销，因此IP监视是有效的，但安全方面需要加强。一方面，IP MVEE注重性能，与应用程序驻留在同一地址空间中，监视器没有硬件支持的边界保护，导致某些攻击（例如ROP）逃避跟踪检测机制。另一方面，一旦变体中的漏洞被成功利用，攻击者可能会获得监视器的控制权。这些MVEE牺牲了变体与监视器间的隔离性，但极大地降低了变体与监视器的通信开销，在消除上下文切换的需求中优于CP监视器。

在CP MVEE中，监视器作为一个单独的进程

运行。即使攻击者破坏了变体，攻击者也很难直接操纵监视器的内存空间，所以，在安全需求中CP监视器优于IP监视器。大多数MVEE都使用集中式的监视程序来交叉检查所有变体，这种设计侧重于简单性和安全性，而以牺牲性能为代价，导致监视器变体交互的延迟较高，因为监视器使用ptrace将大量的数据存储在缓冲区并频繁刷新高速缓存。除了同步开销外，监视器必须等待执行速度最慢的变体。因此，系统安全性和性能之间的平衡是至关重要的。

表4　基于CP/IP的MVEE分类

监视器部署模式	特征	MVEE
跨进程CP	监视器作为单独的进程运行 攻击者直接操纵监视器内存空间较困难 ptrace调试机制，监视器变体交互延迟较大	DieHard, Cavallaro, Orchestra, GHUMVEE, Mx, TACHYON, DMON, ReMon
进程内IP	监视器与变体共享相同的地址空间 降低监视器与变体的通信开销 攻击者易获得监视器控制权	N-Variant System, VARAN, REVEN, MvArmor, KMVX, BUNSHIN, ReMon

4.3　I/O操作模式

MVEE确保同时执行的程序变体将一致的结果呈现给客户端，这主要通过主从模式（Master-slave Mode）和代理模式（Proxy Mode）两种I/O操作模式实现，如表5所示。在主从模式下，主变体负责与外部交互；而在代理模式下，监视器充当I/O代理的角色。

通常，主从模式下的MVEE只在主变体中执行一次I/O操作。主变体将执行结果复制给从变体，保证了所有变体的一致性，该过程对客户端完全透明。例如：N-Variant系统中对于像I/O这样的共享系统调用，它只允许到达该调用的最后一个变体进行实际的系统调用，该变体检查系统调用号及其参数是否与前一个变体相匹配，然后监视器将执行结果从内核缓冲区复制给变体使其继续执行。主从模式的其他MVEE选用同一个变体执行I/O操作，即指定其中之一为主变体。TAYCHON是一种串联执行方法，完全封装了重放所需的系统调用流，也被认为是主从模式。

在代理模式下，监视器代替变体执行I/O操作。这些MVEE不允许主变体执行I/O操作，而是监视器在拦截系统调用后获得执行控制权，其执行结果直接复制给变体。该过程中，当监视器恢复变体的执行时，变体不得不继续执行原系统调用。这时，监视器将系统调用替换为不影响系统状态的调用（通常为sys_getpid），使变体继续执行并立即返回。然后，监视器执行原始的系统调用并替换所有变体的返回结果。

根据表5所示，主从模式是当前MVEE的主流模式，便于对客户端提供一致的I/O流。其主要挑战是，当主变体受到威胁时，仍然没有完善的解决方案来选择新的主变体，且恶意的I/O数据流将被复制给其他从变体，导致劫持I/O数据流攻击。而对于代理模式，克服上述缺陷并实现平等变体的容错能力具有更大的潜力。

表5　基于主从/代理模式的MVEE分类

I/O操作模式	特征	MVEE
主从模式	主变体与客户端交互，I/O操作仅主变体执行 提供系统执行过程透明性 选取新的主变体较困难 面临劫持I/O数据流威胁	N-Variant System, VARAN, REVEN, MvArmor, ReMon, TACHYON, KMVX, DMON, BUNSHIN, GHUMVEE, Cavallaro
代理模式	监视器充当I/O代理，I/O代理与客户端交互 克服主从模式缺陷，易实现系统的容错功能	DieHard, Mx, Orchestra

4.4　同步点（RVPs）

现有MVEE的同步点主要为指令集、I/O操作以及系统调用，如表6所示。监视器将暂停进入或退出系统调用的变体，直到所有变体都到达相同

的入口或返回点。在这种情况下，这些变体被称为已经到达同步点（RVPs）。

I/O同步操作通常用来管理变体的输出或统一与硬件交互的接口。如：DiaHard仅同步I/O write操作，选择多数变体达成共识的输出缓冲区同步点，根据裁决机制将其发送到标准输出。

指令集在同步管理上过于繁琐，而粗粒度的I/O同步缺乏有效的安全保障。大多数MVEE将系统调用作为同步点，因为应用程序必须使用系统调用与系统进行交互。这些MVEE可以拦截变体的系统调用，并在需要时更改或延迟它们。例如，GHUMVEE的同步点［30］是系统调用入口点和返回点，变体在调用前后暂停，直到它们都到达同步点才被恢复执行。监视器在同步点检测每个变体的行为，以防止变体访问其进程空间之外的数据或资源。

表6　基于RVPs的MVEE分类

同步点	特征	MVEE
系统调用	应用程序必须使用系统调用与系统进行交互	N-Variant System, Cavallaro, Orchestra, GHUMVEE, Mx
	监视器在系统调用入口点、返回点检测并同步变体的行为	DMON, ReMon, Orchestra, TACHYON, REVEN, MvArmor, BUNDHIN, KMVX
I/O	管理变体的输入输出，统一的硬件交互接口	
	缺乏有效的安全保障	DieHard, KMVX
指令集	同步管理上较繁琐	–

4.5　监视器-变体通信

在性能、安全性、灵活性和调试简易性之间达到平衡一直是MVEE面临的挑战，在性能方面其执行开销主要集中于监视器-变体通信，即系统调用拦截与同步。MVEE拦截和仿真了大量操作系统功能，以便所有变体真正在一致的输入状态下运行。目前有三种主流机制实现监视器-变体通信，包括可加载内核模块（LKM）、调试工具（Ptrace）和二进制重写（BR），如表7所示，表中还显示了两个特例MvArmor和KMVX的通信机制。

图2展示了MvArmor［35］给出的getpid在转发（将原始系统调用转发到操作系统内核）和返回（将结果返回给应用程序）模式下的性能开销。如图所示，基于可加载内核模块（LKM）［13］［27］［36］的监视器在两种操作模式下都实现了最有效的系统调用拦截策略，因为它不会引入额外的上下文切换，并且可以直接访问进程状态。但是，LKM在特权级通过内核补丁实现，导致可信计算基（Trusted Computing Base， TCB）的大幅增加：监视器出现任何程序错误或者安全隐患就会对整个系统的安全造成危害。基于Dune的硬件支持进程虚拟化的监视器具有对特权CPU功能访问的权限，并且与变体、底层内核隔离开来，满足了安全需求，但与LKM相比开销更高（getpid的开销高达7倍）。

大多数MVEE都使用ptrace_PEEKDATA调用ptrace调试工具，以便在拦截系统调用时将EAX中的系统调用号和其他寄存器中的参数写入监视器的内存，并在内核返回执行结果后将其写入变体。然而，ptrace机制一次仅传输四个字节，并且每次调用都进行上下文切换，导致执行开销随着数据缓冲区大小的增加而线性增加（getpid的开销高达217倍）。此外，虚拟系统调用（lock_gettime、getcpu和time）也成为基于ptrace的监视器的主要限制之一。由于这些系统调用完全在用户空间实现，因此不能通过ptrace进行拦截。

静态二进制工具（Static Binary Instrumentation， SBI）重写二进制文件并拦截系统调用，使用固定大小的共享内存环形缓冲区共享数据。它在进程内使用二进制重写（BR）消除了上下文切换以及额外系统调用的执行，所以该机制效率更高（getpid约5倍的开销），并且当数据缓冲区大小增加时，开销趋于平稳。如：VARAN［33］采用二进制重写机制，使用中断（INT 0x0）替换系统调用将控制流重定向到系统调用入口点。但这种机制通常不适用于安全应用程序，因为攻击者可以篡改进程中的监视器状态，或者运行未对齐的系统调用指令避开检测。

4.6　分类总结

结合4.1到4.5小节，该节总结了上述MVEE分类标准［40］，如表8所示。此外，我们根据MVEE的架构特点刻画了5种主流架构：IP/UL

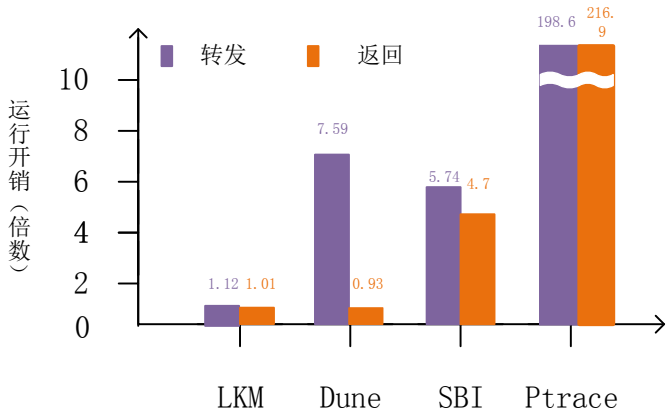


图2　getpid 系统调用拦截机制（转发和返回模式下）带来的开销。（MvArmor［35］）

表7　基于监视器-变体通信的MVEE 分类

监视器-变体通信	特征	MVEE
Ptrace	用户空间调试机制 拦截系统调用寄存器参数 易于实现，开销较大	DieHard, Cavallaro, DMON, Orchestra, GHUMVEE, ReMon, Mx, TACHYON
LKM	内核中实现 避免额外的上下文切换，直接访问进程状态 需要大量修改内核	N-Variant System, REVEN, BUNSHIN
Dune	沙箱中实现，具有对特权CPU 访问权限 监视器隔离性更强，安全性较高	MvArmor
Binary Rewriting	重写系统调用二进制文件 共享内存环形缓冲区 效率较高，无可靠的安全保障	VARAN
In-kernel Communication Channel	内核间共享缓冲区	KMVX

MVEE （a）、 CP/UL MVEE （b）、 IP/PL MVEE （c）、 IP/UL+CP/UL+IK-B /PL MVEE （d） 和 IK/PLMVEE（e）。图3 显示了5 种MVEE 的安全架构以及工作流程。

（a）协调器启动变体（步骤□ ）。监控器监视变体行为（步骤□ ）。监视器交叉检查系统调用号和参数（步骤□ ）。当监视器确认变体行为一致时，内核执行主变体的系统调用（步骤□ ），并将结果返回给主变体（步骤□ ）。最后，主变体将结果复制给从变体（步骤□ ）。

（b）步骤□ -□ 同（a）。内核分别执行两个变体的系统调用（步骤□ ）。监视器拦截并交叉检查内核返回的结果（步骤□ ）。监视器将结果返回给变体（步骤□ ）。I/O 操作只在主变体执行一次，即不执行虚线部分。主变体将I/O 结果复制给从变体（步骤□ ）。

（c）监视器也称为协调器，负责启动变体（步骤□ ），同时协调变体与LKM 的通信。步骤□ -□ 同（b）。值得注意的是，I/O 操作由LKM 直接返回给所有变体。

（d）变体发出系统调用（步骤□ ）。IK Broker（IK-B）拦截器拦截系统调用，并使用随机64 位令牌使用环形缓冲区（Ring Buffer， RB）将其转发到IP-MON 或CP-MON （步骤□ ）。IP-MON 执行一系列安全检查，完成系统调用转发（步骤□ ）。如果令牌在重新进入内核时完好无损，IK-B 验证器将允许完成系统调用的执行，并将结果返回给IP-MON；如果令牌不完整，IK-B 验证器撤销令牌并强制将调用转发到CP-MON （步骤□ ）。

（e）用户程序在两个内核变体上运行，通过系统调用同步保持一致性。内核变体运行在同一硬件上，该硬件只与主变体交互。主变体通过I/O 同

步复制I/O结果。

表8 MVEE的总体分类

分类	软件堆栈位置	MVEE	主从模式	监视器-变体通信	同步点	支持多线程	信号处理
可靠	IP/UL	VARAN	是	BR	系统调用	是	
	CP/UL	Mx	否	Ptrace	系统调用		
		TACHYON	是(串行)	Ptrace	系统调用	是(DMT)	
	IP/PL	MvArmor	是	Dune	系统调用		
		REVEN	是	LKM	系统调用		
		BUNSHIN	是	LKM	系统调用	是(DMT)	
		N-Variant	是	LKM	系统调用		
安全	CP/UL	Cavallaro	是	Ptrace	系统调用		
		Orchestra	否	Ptrace	系统调用	是	是
		GHUMVEE	是	Ptrace	系统调用	是	是
		DMON	是	Ptrace	系统调用	是	
		DieHard	否	Ptrace	I/O		
	IP/UL+CP/UL+IK-B/PL	ReMon	是	Ptrace	系统调用	是	是
	IK/PL	KMVX	是	内核通信通道	系统调用I/O	是	是

5 误报

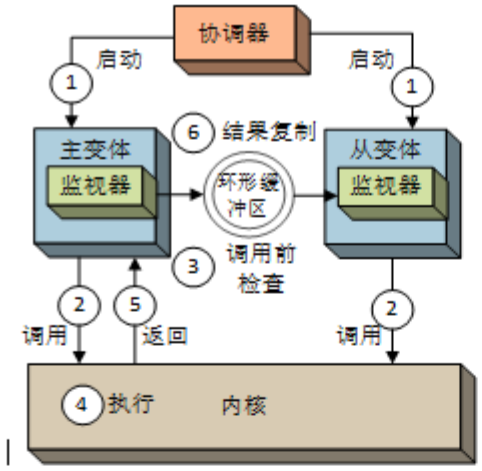
MVEE必须为所有变体提供相同的输入，并保证一致的系统调用执行顺序和参数，以确保它们在正常操作条件下行为相同。然而，某些输入源可以直接访问，而无需触发任何系统调用，如文件操作、时间、随机数、共享内存、多线程和异步信号处理等。从这些输入源直接读取输入后，变体的行为通常不同，这可能会导致监视器检测到误报。在本节中，我们将讨论可能导致监视器检测到误报的两个主要来源：多线程和异步信号处理，并描述MVEE如何为这些来源的变体提供一致的输入。表8还显示了这些MVEE是否支持多线程同步和异步信号传递。

5.1 多线程

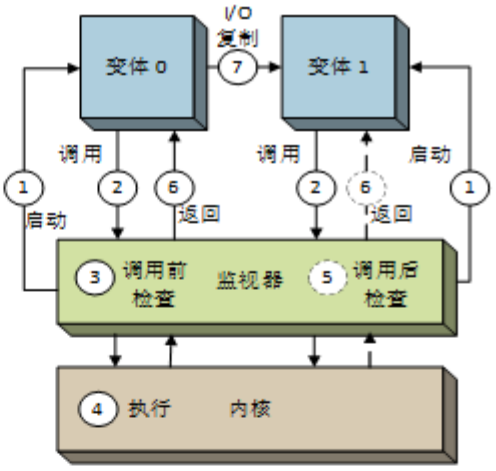
MVEE误报来源之一：非确定性多线程应用程序的安全复制［41］。多线程可以直接通过共享内存进行通信，而无需使用系统调用。由于系统调用的锁步执行，非确定性多线程调度中的任何细微差别（包括线程执行顺序和参数）都将被仲裁为行为差异。

为解决这一问题，主要有两种方法：一种是确定性多线程（Deterministic Multi-Threading，DMT）［42］，另一种使用记录/重放（Record/Replay，R/R）系统处理非确定性多线程［41］［43］。DMT强制所有线程调度顺序相同，确保具有多线程的并行程序每次都产生相同的显式输出。强确定性（Strong determinism，SD）［42］是DMT的一种，它对线程中涉及的所有指令强制执行完全确定的顺序，但性能开销通常很高。弱确定性（Weak Determinism，WD）［44］［45］，另一种更常用的DMT，仅强制执行锁的确定顺序，从而降低了开销。MVEE上使用DMT，非确定性多线程系统调用的顺序和参数也必须保持一致，以防止监视器检测到差异。但其仍可能由于变体代码布局改变时产生不同的线程调度顺序，从而导致不同的行为。此外，DMT不能保证使用临时同步和无锁算法应用程序的确定性。

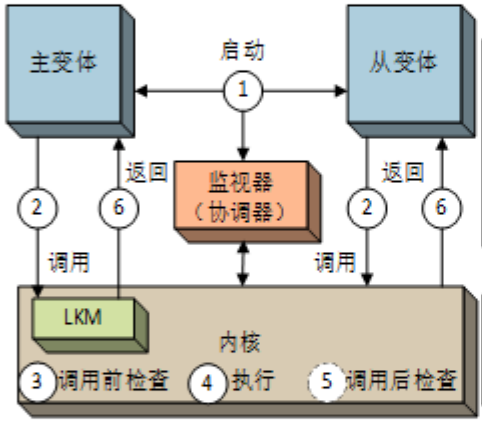
记录/重放系统捕获主变体线程的操作顺序，并在从变体中重放完全一致的同步操作。Stijn Volckaert等人手动识别和排序原子操作，提出了多线程的可复制确定性（Replicatable Determinism，RD）［46］。该策略通过嵌入到所有变体中的同步代理来识别并封装临界区的原子操作以及临时同步事件，提供与WD同等强大的原子操作排序，并具有比SD更高的性能。在这种情况下，每当主变体创建新线程时，监视器将产生一个新的监视线程来监视主变体产生的线程。由于在各变体间等效地重放同步操作，多线程调度顺序及其参数不太可能有所不同，所以记录/重放系统在MVEE中



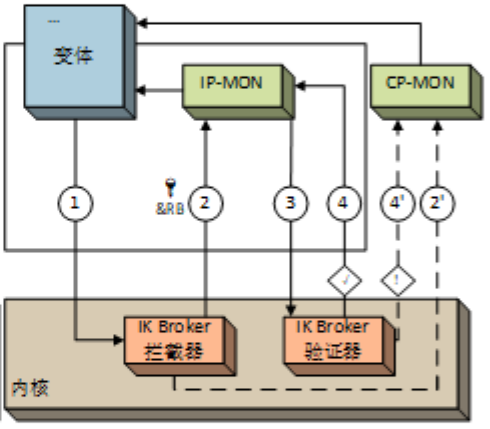
(a)IP/UL MVEE(VARAN[33])



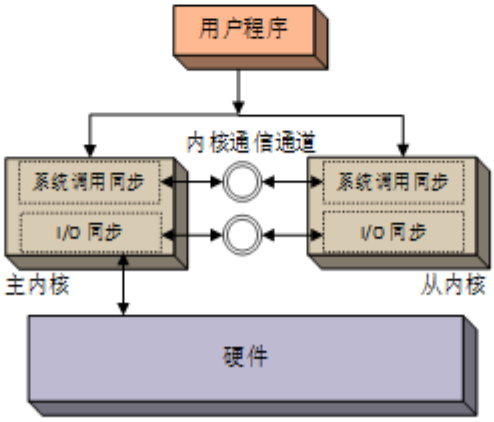
(b)CP/UL MVEE(GHUMVEE[30])



(c)IP/PL MVEE(REVEN[27])



(d)IP/UL+CP/UL+IK-B/PL MVEE(ReMon[34])



(e)IK/PL MVEE(KMVX[38])

图 3　IP/UL MVEE(a), CP/UL MVEE(b), IP/PL MVEE(c), IP/UL+CP/UL+IK-B/PL MVEE(d)和 IK/PL MVEE(e)安全架构

比DMT更适合。

5.2 异步信号传递

另一个由外部输入源导致的高误报率为异步信号传递。即使向所有变体发送相同的信号，微小的时序差异和调度延迟通常也会导致信号在不同变体执行流中的不同点进行传递。如果这些信号的传递没有受到监视器的严格控制，那么它们很容易在变体中引发行为差异。

Orchestra利用多数裁决机制［47］来决定在同步点之前或者之后传递信号，以确保所有变体的信号同步执行。无论何时向变体发送信号，监视器都暂停该变体，后续将信号继续传递给变体、保存或忽略该信号。例如，如果大多数进程在系统调用之前已经接收到信号，而其余变体在该调用后才会收到，则监视器使这些变体跳过系统调用并等待信号。当这些变体接收到信号时，监视器同步变体信号执行。该方法注重性能而不是正确性，而且不支持中断系统调用的信号。

GHUMVEE使用另一种方法进行异步信号处理。它延迟异步信号的传递，直到所有变体都到达下一个系统调用同步点。首先丢弃处于signal-delivery-stop状态的任何异步信号，并在所有变体处于相同的同步点时重新发送该信号。几乎所有与信号处理相关的程序都可以在GHUMVEE中正确处理。然而，该方法的缺点是它将导致某些信号传递延迟较长，这对于某些类型的信号（例如定时器信号）来说是不可接受的。

由于VARAN是IP / UL架构并且不使用ptrace API，因此VARAN在处理信号方面具有较高的性能和正确性。在VARAN中，主变体不加延迟地接收和处理信号，并将与该信号关联的元数据记录到事件流缓冲区中，以供从变体重放。

6 安全分析

MVEE提供的安全保证建立在三个属性之上：1）监视器与变体隔离性；2）监视器锁步拦截系统调用； 3）变体的多样化。

如前文所述，在用户空间中，单独地址空间上实现的MVEE比那些与变体共享地址空间的MVEE更安全。由于运行在非特权模式下，以及监视器和变体的隔离，所以监视器和变体本身的错误将不会对系统造成严重的危害。IP/UL VARAN可靠且效率高，但不太适合防止内存攻击，因为它只允许主变体执行系统调用，并且不区分敏感和不敏感系统调用。CP/UL MVEE的监视器是一个独立的进程，有它自己的地址空间，系统中的任何其他进程（包括变体）都不能直接操作其内存空间。在内核空间中，假设内核不会受到攻击，在特权级别实现的MVEE满足监视器的安全需求：完全隔离监视器，没有进程内状态、不可绕过的系统调用拦截机制。例如，MvArmor依赖于基于Dune的硬件进程虚拟化来使监视器与变体隔离。

几乎所有的MVEE都锁步监视系统调用的执行。对于所有依赖于系统调用的攻击，监视器可以在系统受损之前拦截调用并检测到变体的行为差异。DieHard虽只对I/O操作进行裁决，但由于变体的多样化技术也能防止堆错误。ReMon通过仅对安全关键系统调用实施严格的交叉检查，同时对非安全关键系统调用使用更宽松的监控策略，使得系统更加安全可靠、高效灵活。

此外，致力于避免系统内存攻击的安全性还取决于变体的多样化。表9根据第3节分析的MVEE系统给出了面向安全性的MVEE防御的攻击［48］，或者面向可靠性的MVEE的功能。

7 性能分析

MVEE的性能主要取决于监视器系统调用拦截时间以及等待所有变体到达同步点的时间。因此，MVEE的整体运行时间可以分解为两部分：1）用于变体同步和拦截的时间；2）执行最慢的变体所需的时间。表10总结了第3节MVEE安全系统（2个变体）在不同平台测试的开销。

VARAN作为面向可靠性MVEE中IP/UL的唯一代表，在所有基准测试中的开销都低于14.2%，特别是在像thttpd这样的Web服务器基准测试中开销几乎为零，因为它结合了灵活而高效的二进制重写和高性能的进程内读写机制。

另外两个面向可靠性的CP/UL MVEE：Mx和TAYCHON都致力于软件更新容错，以创建更安全可靠的应用程序。在lighttpd基准测试中的开销是VARAN的200多倍。而面向安全的CP/UL MVEE的开销是VARAN的数十倍到数百倍，这些MVEE依赖于UNIX进程ptrace API来实现可部署但效率

表9　MVEE提供的攻击防御/功能

分类	软件堆栈位置	MVEE	同步点	多样化技术	防御/功能
	IP/UL	VARAN	系统调用	地址空间布局随机化	提升性能
可靠	CP/UL	Mx	系统调用	新旧程序版本	软件更新容错
		TACHYON	系统调用	补丁前后程序	补丁程序测试
	IP/PL	MvArmor	系统调用	地址空间布局随机化	内存错误利用
				栈布局随机化	任意代码执行,信息泄露
		REVEN	系统调用	多粒度多样化技术	内存错误
		BUNSHIN	系统调用	Sanitizer内存检测器	心脏滴血,代码注入攻击
		N-Variant	系统调用	地址空间分区	绝对地址攻击
				指令集标记	代码注入攻击
安全	CP/UL	Cavallaro	系统调用	地址空间分区	缓冲区/栈溢出
		Orchestra	系统调用	堆栈反向生长	堆溢出,恶意代码执行
		GHUMVEE	系统调用	不相交代码布局	面向返回的编程攻击
		DMON	系统调用	代码执行保护	与位置无关的代码重用
		DieHard	I/O	栈布局随机化	堆内存错误利用
	IP/UL+CP/UL+IKB/PL	ReMon	系统调用	不相交代码布局	检测0-day漏洞
	IK/PL	KMVX	系统调用I/O	堆栈随机化	内核信息泄露

较低的系统调用拦截策略。此策略在每个系统调用的监视器和跟踪进程之间引入了多个上下文切换，从而产生了较高的开销。其中，DMON的高开销在于它在不同主机上运行变体从而引入了网络延迟以及通信开销。

面向安全的IP/PL MVEE使用基于可加载内核模块（LKM）的监视器，相对于CP/UL MVEE具备更有效的系统调用拦截策略，因为它不会引入额外的上下文切换，并且可以直接访问进程状态。随着系统调用同步机制更加成熟，BUNSHIN和MvArmor比N-Variant系统更高效；MvArmor中由于Dune的限制，它的效率微低于BUNSHIN，且不支持多线程应用程序。

ReMon通过利用CP监视器（GHUMVEE）对所有敏感系统调用强制锁步执行，安全强化的IP监视器（IP-MON）实现非敏感调用的高效复制而无需上下文切换。性能评估显示，在实际的服务器工作负载上，ReMon的开销仅在0-3.5%之间，和VARAN相当。

IK/PL KMVX通过在同一台计算机上同时运行多个多样化的内核变体，以可承受的性能开销（在最流行的服务器应用程序中约27%）来防御内核信息泄漏。其性能开销主要体现在与硬件进行I/O同步以及系统调用同步上，但它涉及拦截的系统调用同步仅限于copy_to_user、put_user和系统调用返回值。

8　局限性和未来研究方向

8.1　局限性

MVEE在内存漏洞利用方面具有较强的无需先验知识的入侵检测能力。研究主要集中于安全和性能之间的平衡，在保证系统安全前提下降低上下文切换开销。在第7节中，我们讨论了影响MVEE性能的因素，并指出跨进程MVEE存在较高的上下文切换开销。一种方法是放松监控策略，将监视器设计分为进程内和跨进程的组合，以减少上下文切换的次数。另一种实现是系统调用拦截组件部署在具有管理程序特权级别的内核或虚拟机中。

当MVEE检测到行为差异时，通常选择终止整个进程来阻止攻击。但在一些具有容错需求的应用场景中，进程的异常退出是不能接受的，在此类场景下，实现变体容错或异常处理非常重要。在主从模式中，主变体处理文件描述符并控制I/O数据流，一旦主变体发生异常，选择新的主变体是必要的，但当前大多数MVEE不能执行该操作。仅VARAN通过UNIX域套接字将主变体的文件描述符传递给新的变体来解决此问题。此外，该模式下一旦主变体被攻击者入侵则可以通过劫持I/O数据流来感染其他良性变体，进而实现恶意行为同步，从而躲避监视器的检测。

代理模式由于监视器控制I/O数据流并执行I/

表10　MVEE的性能分析

分类	软件堆栈位置	MVEE	监视器-变体通信	Apache	lighttpd	thpptd	SPEC CINT 2000/2006
可靠	IP/UL	VARAN	BR	2.40%	1.00%	0.00%	14.20%
	CP/UL	Mx	Ptrace		249%		17.90%
		TACHYON	Ptrace		272%	17%	
	IP/PL	MvArmor	Dune				9%
		REVEN	LKM				
		BUNSHIN	LKM				4.99%
		N-Variant	LKM	94%			
	CP/UL	Cavallaro	Ptrace			31%	
安全		Orchestra	Ptrace	50%			17%
		GHUMVEE	Ptrace				12.10%
		DMON	Ptrace		443%		
		DieHard	Ptrace				12%
	IP/UL+CP/UL+IKB/PL	ReMon	Ptrace	2.50%	0.00%	2.70%	3.50%
	IK/PL	KMVX	内核通信通道		27%		

O操作，更有利于实现变体容错或异常处理，即仅终止异常变体使系统无感知的继续运行。但减少同时执行的变体数量可能会降低MVEE的可靠性和安全保证。所以，每当MVEE终止异常变体时，它都应该生成一个新的变体，复制良性变体的状态，使得系统变体数量不减。这样的系统将自动隔离、重新初始化或恢复已损坏的进程。此外，从异常变体的状态中恢复到未被破坏的状态，可以根据两个变体状态之间的差异推断出攻击特征。然而，该解决方案无法创建变体的常规检查点并在新生成的检查点中恢复，因为它需要对内核进行侵入性修改。随着Linux内置的检查点/恢复（Checkpoint/Restore）功能的出现，该方案有望得到实现。

8.2　DHRMVEE安全架构

为解决MVEE的局限性，文献［49］提出了一个MVEE框架，该框架包括控制如何以及何时生成新变体的变体控制器，确定变体和基础结构如何进行调度的调度程序，监视程序决策模块等。它有助于充分利用软件多样性来增强系统的安全性。文献［50］提出一种基于多变体执行架构的CFI（MVX-CFI），提高了原架构的执行效率，同时保证了在安全防御方面的有效性。然而，这些架构都没有涉及实现变体容错或异常处理。

本节根据该领域尚未解决的问题，提出一种基于动态异构冗余架构的多变体执行环境（Dynamic Heterogeneous and Redundant， DHRMVEE）设想。在该架构中，监视器以代理模式执行I/O操作，消除变体地位差异，为实现异常变体的销毁奠定基础。该架构主要包含四个模块：进程监视模块，系统调用裁决模块，I/O模块，变体异构化模块，如图4所示。

进程监视模块负责变体的生成，并且当变体进行系统调用时，进程监视模块会让变体进入暂停状态并捕获变体触发的系统调用号以及参数；实现变体恢复协议，在变体受到攻击时指示监视器终止异常变体，并重启新的变体实例或者复制良性变体的状态恢复异常变体。系统调用裁决模块在同步点分析变体的系统调用号以及参数是否一致，以此来实现无需先验知识的威胁检测，判断变体是否受到攻击。I/O模块主要负责变体与内核空间以及用户空间的交互，该模块根据变体触发的系统调用类型采用不同的处理模式。针对普通系统调用，I/O模块处于透明状态，即不对变体的系统调用进行干涉；针对I/O系统调用，I/O模块会拦截变体的系统调用，并由I/O代理执行实际的系统调用。执行体异构化模块利用ASLR和DCL等技术构建多个功能等价但具有不同内存布局和代码逻辑的变体，以此来降低执行体间的共同漏洞。

网络空间拟态防御（Cyberspace Mimic Defense，CMD）以动态异构冗余（DHR）作为核心架构技术［51］。在现有MVEE系统中导入拟态防御DHR思想，对DHR的系统调用裁决、变体异构

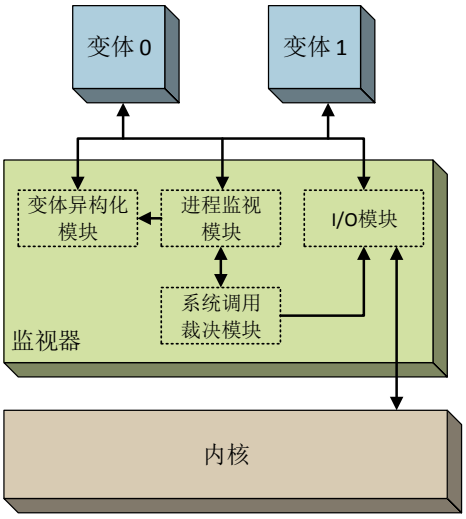


图4　DHRMVEE 安全架构

化调度、多维动态重构以及 I/O 代理等环节施以拟态伪装策略，使得系统获得安全性的同时不降低它的性能。

在某些安全关键系统中，软件的安全性超过了软件的可靠性，因此将在这些系统中应用 DHRMVEE 架构实施软件容错技术［52］，以免发生用户可感知的不良体验。由于攻击者很难通过破坏变体直接操纵监视器的内存地址空间，DHRMVEE 避免了主从模式架构的劫持 I/O 数据流攻击。此外，当前 MVEE 只能静态的构建执行体，在运行时很少体现动态过程。DHRMVEE 安全架构不仅能够实现高可靠、高可信、高可用的一体化服务，且能够根据威胁感知动态调整异构执行体，实现异构执行体的动态选择以及错误执行体的清洗替换功能，从而提高系统的防御增益，突出内生安全特性。

9　总结

本文梳理了近年来 14 种 MVEE 技术，并从监视器权限、监视器部署模式、I/O 操作模式、监视器-变体通信机制、同步点五个方面对其进行分类讨论并总结优缺点。然后本文列举了在 MVEE 框架下可能存在误报的几种情况以及相应的解决办法。此外，对现有 MVEE 技术进行了安全性和性能的定性、定量分析并给出对比结果。最后，本文探讨了当前 MVEE 面临的局限性和挑战，并提出了 DHRMVEE 来解决当前 MVEE 存在的问题。

参考文献:

[1] "https://www. cvedetails. com/vulnerabilities-by-types. php. "

[2] NiuB. and TanG. , “Modular Control-Flow Integrity, ” ACM SIGPLAN Notices, vol. 49, no. 6. pp. 577 – 587, 2014, doi: 10. 1145/2594291. 2594295.

[3] NiuB. and TanG. , "RockJIT: Securing Just-In-Time compilation using modular Control-Flow Integrity," Proc. ACM Conf. Comput. Commun. Secur. , pp. 1317 – 1328, 2014, doi: 10. 1145/2660267. 2660281.

[4] SerebryanyK. , BrueningD. , PotapenkoA. , and VyukovD. , "AddressSanitizer: A fast address sanity checker, ”
Proc. 2012 USENIX Annu. Tech. Conf. USENIX ATC 2012, pp. 309 – 318, 2019.

[5] NagarakatteS. , ZhaoJ. , MartinM. M. K. , and ZdancewicS. , “SoftBound: Highly Compatible and Complete Spatial Memory Safety for c," SIGPLAN Not. , vol. 44, no. 6, pp. 245 – 258, 2009, doi: 10. 1145/1543135. 1542504.

[6] NagarakatteS. , ZhaoJ. , MartinM. M. K. , and ZdancewicS. , “CETS: Compiler-enforced temporal safety for C," ACM SIGPLAN Not. , vol. 45, no. 8, pp. 31 – 40, 2010, doi: 10. 1145/1837855. 1806657.

[7] AbadiM. , BudiuM. , ErlingssonÚ. , and LigattiJ. , "Control-flow integrity principles, implementations, and applications," TransACM. Inf. Syst. Secur. , vol. 13, no. 1, 2009, doi: 10. 1145/1609956. 1609960.

[8] KornerT. W. “Code-Pointer Integrity Volodymyr, ” Scepticism Hero Villain, no. October, pp. 47 – 56, 2012.

[9] StepanovE. and SerebryanyK. , "MemorySanitizer: Fast detector of uninitialized memory use in C++ , Proc". 2015 IEEE/ACM Int. Symp. Code Gener. Optim. CGO 2015, pp. 46 – 55, 2015, doi: 10. 1109/CGO. 2015. 7054186.

[10] "LLVM. UndefinedBehaviorSanitizer (UBSan) is a fast undefined behavior detector, Feb. 2015. http://clang. llvm. org/docs/UndefinedBehaviorSanitizer. html. ”

[11] ChenL. and AvizienisA. , "N-Version Programming: a Fault-Tolerance Approach To Reliability of Software Operation. ," *Proc. - Annu. Int. Conf.* Fault-Tolerant Comput. , vol. 1, pp. 3 – 9, 1978.

[12] KovalevI. V. , KovalevD. I. , ChefonovN. S. , TestoedvovN. A. , and GolovenkinE. N. , "Implementation of multiversion software based on an object-oriented approach, ”
ConfIOP. Ser. Mater. Sci. Eng. , vol. 734, no. 1, 2020, doi: 10. 1088/1757-899X/734/1/012035.

[13] B. Cox
et al. , “N-variant systems a secretless framework for security through diversity," 15th USENIX Secur. Symp. , no. August, pp. 105 – 120, 2006.

[14] A. Voulimeneas, "Building the Next Generation of Security Focused NVX Systems: Overcoming Limitations of N-Variant Execution," pp. 1982 – 2004, 2020.

[15] LarsenP. , HomescuA. , BrunthalerS. , and FranzM. , "SoK: Automated software diversity, ”

Proc. - IEEE Symp. Secur. Priv.，pp. 276 – 291，2014，doi: 10. 1109/SP. 2014. 25.

[16] VeeraraghavanK. , ChenP. , FlinnJ. , and NarayanasamyS. , Detecting and Surviving Data Races using Complementary Schedules. 2011.

[17] SalamatB. , JacksonT. , GalA. , and FranzM. ，"Orchestra: Intrusion detection using parallel Execution and monitoring of program variants in user-space，"
Proc. 4th ACM Eur. Conf. Comput. Syst. EuroSys'09，pp. 33 – 46，2009，doi: 10. 1145/1519065. 1519071.

[18] HosekP. and CadarC. ，"Safe software updates via multi-version execution，"
Proc. - Int. Conf. Softw. Eng. ，no. May，pp. 612 – 621，2013，doi: 10. 1109/ICSE. 2013. 6606607.

[19] HollandD. A. , LimA. T. , and SeltzerM. I. ，"An architecture a day keeps the hacker away，"
SIGARCH ComputACM. Archit. News，vol. 33，no. 1，pp. 34 – 41，2005，doi: 10. 1145/1055626. 1055632.

[20] ChewM. and SongD. ，"Mitigating buffer over fl ows by operating system randomization，" C. Tech. Rep. ，pp. 02 – 197, 2002，[Online]. Available: http://citeseerx. ist. psu. edu/viewdoc/summary?doi=10. 1. 1. 116. 1093.

[21] BarrantesE. G. , PalmerT. S. , AckleyD. H. , StefanovićD. , ForrestS. , and ZoviD. D. "Randomized instruction set emulation to disrupt binary code injection attacks，" *Proc*. ACM Conf. Comput. Commun. Secur. ，pp. 281 – 289，2003，doi: 10. 1145/948143. 948147.

[22] KcG. S. , KeromytisA. D. , and PrevelakisV. ，"Countering code-injection attacks with instruction-set randomization，"
Proc. ACM Conf. Comput. Commun. Secur. ，pp. 272 – 280，2003，doi: 10. 1145/948143. 948146.

[23] SalamatB. , GalA. , Todd ManivannanJ. K. , WagnerG. , and FranzM. ，"Multi-variant program execution: Using multi-core systems to defuse buffer-overflow vulnerabilities，"
Proc. - CISIS 2008 2nd Int. Conf. Complex，Intell. Softw. Intensive Syst. ，pp. 843 – 848，2008，doi: 10. 1109/CISIS. 2008. 136.

[24] SalamatB. , GalA. , and FranzM. ，"Reverse stack execution in a multi-variant execution environment，"
Work. Compil. Archit. Tech. Appl. Reliab. Secur. ，pp. 1 – 7，2008.

[25] VolckaertS. , CoppensB. , and De SuttermemberB. ，"Cloning your gadgets: Complete ROP attack immunity with multi-variant execution，"
IEEE Trans. Dependable Secur. Comput. ，vol. 13，no. 4，pp. 437 – 450，2016，doi: 10. 1109/TDSC. 2015. 2411254.

[26] B. Salamat，"Multi-Variant Execution: Run-Time Defense against Malicious Code Injection Attacks," University of California at Irvine，USA，2009.

[27] M. Co
et al. ，"Double helix and RAVEN: A system for cyber fault tolerance and recovery," Proc. 11th Annu. Cyber Inf. Secur. Res. Conf. CISRC 2016，2016，doi: 10. 1145/2897795. 2897805.

[28] BergerE. D. and ZornB. G. "DieHard: Probabilistic memory safety for unsafe languages," *Proc. ACM SIGPLAN Conf. Program. Lang. Des*. Implement. ，vol. 2006，pp. 158 – 168，2006.

[29] MatematicheS. and NaturaliF. ，"Comprehensive Memory Error Protection via Diversity and Taint-Tracking Comprehensive Memory Error Protection via Diversity and Taint-Tracking，" 2007.

[30] VolckaertS. , De SutterB. , De BaetsT. , and De BosschereK. ，"GHUMVEE：Efficient，effective，and flexible replication，" Lect. Notes Comput. Sci.（including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)，vol. 7743 LNCS，no. c，pp. 261 – 277，2013，doi：10. 1007/978-3-642-37119-6_17.

[31] SpanogheJ. "JIT-ROP attack against a Multi-Variant Execution Environment，" 2017.

[32] MaurerM. and BrumleyD. ，"Tachyon: Tandem execution for efficient live patch testing，"
Proc. 21st USENIX Secur. Symp. ，pp. 617 – 630，2012.

[33] HosekP. and CadarC. ，"VARAN the Unbelievable：An Efficient N-Version Execution Framework，" SIGPLAN Not. ，vol. 50，no. 4，pp. 339 – 353，2015，doi：10. 1145/2775054. 2694390.

[34] S. Volckaert
et al. ，"Secure and efficient application monitoring and replication，"
Proc. 2016 USENIX Annu. Tech. Conf. USENIX ATC 2016，pp. 167 – 179，2016.

[35] KoningK. , BosH. , and GiuffridaC. ，"Secure and efficient multi-variant execution using hardware-assisted process virtualization，"
Proc. - 46th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Networks，DSN 2016，no. Mvx，pp. 431 – 442，2016，doi: 10. 1109/DSN. 2016. 46.

[36] XuM. , LuK. , KimT. , and LeeW. "Bunshin: Compositing security mechanisms through diversification，"
Proc. 2017 USENIX Annu. Tech. Conf. USENIX ATC 2017，pp. 271 – 283，2019.

[37] A. Voulimeneaset al. , "DMON: A Distributed Heterogeneous N-Variant System，" 2019，[Online]. Available: http://arxiv. org/abs/1903. 03643.

[38] ÖsterlundS. , KoningK. , OlivierP. , BarbalaceA. , BosH. , and GiuffridaC. ，"KMVX: Detecting Kernel Information Leaks with Multi-variant Execution，"
Int. Conf. Archit. Support Program. Lang. Oper. Syst. - ASPLOS，pp. 559 – 572，2019，doi: 10. 1145/3297858. 3304054.

[39] BelayA. , BittauA. , MashtizadehA. , TereiD. , MazièresD. , and KozyrakisC. ，"Dune: Safe User-level Access to Privileged CPU Features，"
Proc. 10th USENIX Symp. Oper. Syst. Des. Implementation，OSDI 2012，pp. 335 – 348，2012.

[40] S. Crane
et al. ，"The Continuing Arms Race，" pp. xi--xiii，2018，doi: 10. 1145/3129743. 3129752.

[41] VolckaertS. , CoppensB. , De SutterB. , De BosschereK. , LarsenP. , and FranzM. ，"Taming parallelism in a multi-variant execution environment，"
Proc. 12th Eur. Conf. Comput. Syst. EuroSys 2017，pp. 270 – 285，2017，doi: 10. 1145/3064176. 3064178.

[42] BerganT., AndersonO., DeviettiJ., CezeL., and GrossmanD.，“Coredet：A compiler and runtime system for deterministic multithreaded execution," ACM SIGPLAN Not.，vol. 45，no. 3，pp. 53－64，2010，doi：10. 1145/1735970. 1736029.

[43] VolckaertS., De SutterB., De BosschereK., and LarsenP.，“Multi-Variant Execution of Parallel Programs，”pp. 1－14，2014.

[44] OlszewskiM., AnselJ., and AmarasingheS.，“Kendo：Efficient deterministic multithreading in software," ACM SIGPLAN Not.，vol. 44，no. 3，pp. 97－108，2009，doi：10. 1145/1508284. 1508256.

[45] LuK., ZhouX., BerganT., and WangX. “Efficient deterministic multithreading without global barriers，”ACM SIGPLAN Not.，vol. 49，no. 8，pp. 287－300，2014，doi: 10. 1145/2555243. 2555252.

[46] VolckaertS., De SutterB., and De BosschereK.，“Replicatable Determinism for Parallel Programs，”pp. 1－14，2015.

[47] SalamatB., WimmerC., and FranzM.，“Synchronous Signal Delivery in a Multi-Variant Intrusion Detection System，”Library (Lond).，2009，[Online]. Available: http://www. babaks. com/files/TechReport-08-14. pdf.

[48] LiuZ., ZhangZ., ZhangJ., and LiuH. “Multi-Variant Execution Research of Software Diversity," *J. Phys*. Conf. Ser.，vol. 1325，no. 1，2019，doi: 10. 1088/1742-6596/1325/1/012107.

[49] JunchaoW., ZhenwuL., HaoL., and JianminP.，"A Framework for Multi-Variant Execution Environment，”
PhysJ.. Conf. Ser.，vol. 1325，no. 1，2019，doi: 10. 1088/1742-6596/1325/1/012005.

[50] DongY. A. O., ZhengZ., GaofeiZ., and JiangxingW. U., "MVX-CFI：一种实用的软件安全主动防御架构 MVX-CFI : a practical active defense framework for software security," vol. 5，no. 4，2020.

[51] HongchaoH. U., FucaiC., and ZhenpengW., "拟态防御 DHR 模型若干问题探讨和性能评估 Performance Evaluations on DHR for Cyberspace Mimic Defense,vol". 1, no. 4，2016.

[52] O. A. Abdulhameed, "Software Fault Tolerance: A Theoretical Overview," Int. J. Simul. Syst. Sci. Technol.，no. June，pp. 0－16，2019，doi: 10. 5013/ijssst. a. 20. 03. 07.

[作者简介]

陈玉枚 女，1997年2月出生，重庆奉节人，国家数字交换系统工程技术研究中心硕士研究生，主要研究方向为拟态防御、多变体执行环境。

扈红超 男，1982年3月出生，河南商丘人，国家数字交换系统工程技术研究中心副研究员，博士生导师，主要研究方向为网络空间安全，网络主动防御。

王亚文 男，1990年8月出生，河南郑州人，国家数字交换系统工程技术研究中心助理研究员，主要研究方向为拟态防御、多变体执行环境和软件多样化。

王庆丰 男，1995年8月出生，河南周口人，国家数字交换系统工程技术研究中心临聘人员，主要研究方向为多变体执行环境。

迟宇宁 女，1995年9月出生，云南楚雄人，国家数字交换系统工程技术研究中心硕士研究生，主要研究方向为拟态防御、多变体执行环境。

Research on Cross-site Scripting Attack Detection Technology Based on Few-shot Learning

LU Dongzhe[1], Liu Long[1,2]

1.PLA Strategic Support Force Information Engineering University, Zhengzhou, 450001, China;

2.State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, 450001, China

Key words: cross-site scripting attack; few-shot learning; machine learning; virtual sample generation; convolution neural network

Abstract. With the intensification of informatization and mobility， various web security threats are emerging. Cross-site scripting （XSS） attack is the most common type of web attack. Most traditional detection methods have been difficult to adapt to the existing confusion variants of XSS attacks. In this paper， we extract features based on big data collected from 2016 to the present. In order to improve the XSS detection effect of detection tools， we build machine learning models based on more than 210，000 positive and negative samples， among which CNN has the best performance. Furthermore， we propose a new algorithm that improves the traditional virtual sample generation technology based on prior knowledge in order to improve the generalization of the models. Experimental results show that in most cases， the performance of the algorithm in this paper is better than other VSG methods， and the ability to detect and discover unknown attacks is improved to a certain extent.

1 Introduction

According to the Top10 application software security risk report released by OWASP from 2007 to 2018， XSS attacks have always been among them and are extremely common security issues. The actual impact of XSS attacks usually depends on the nature， functionality， and data of the application and the state of the attacked users. In various social platforms， open-source web applications， and other scenarios， the possible harms are： stealing user cookies； injecting Trojan horse functions into websites； sending advertisements and spam； spreading worms； causing DoS attacks； implementing phishing； leading to Cross-Site Request Forgery and so on. How to accurately detect XSS and formulate effective defense measures is a problem urgently needed by researchers in the security field. The machine learning algorithm is essentially a data-driven technology. An important problem to be solved when applying it to the field of attack detection is the contradiction between the large sample size required and the small sample size obtained.

Therefore， this paper improves existing VSG technology. The new method could mine more information from small samples to enrich the sample data. The main contributions of this paper are as follows：

We summarized the causes and classification of XSS vulnerabilities and analyzed a variety of confusion and mutation patterns.

We collected more and updated positive and negative sample sets and used the word2vec model to convert embedded word vectors.

We have established three machine learning models and adjusted the parameters to achieve the best performance.

We have improved the way of generating virtual samples and generated more convincing virtual samples to retrain the model. It improves the generalization of the model to a higher level than traditional VSG technologies.

1.1 Causes of vulnerability

According to the characteristics of the vulnerability and the location where it occurs， there are currently three main types of XSS： Reflect XSS， Store XSS， and DOM-based XSS. Figure 1-Figure 3 shows the process of three kinds of attack.

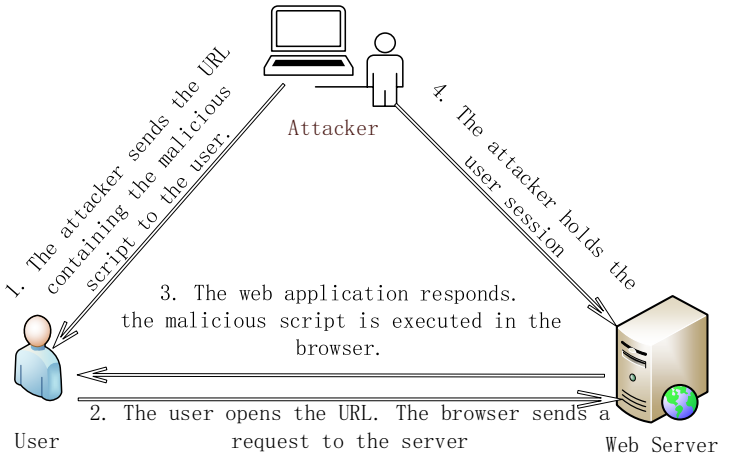


Fig. 1 Reflect XSS.

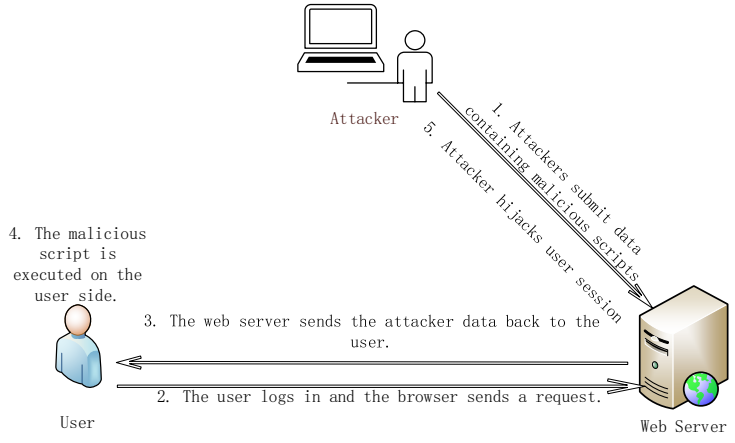


Fig. 2 Stored XSS.

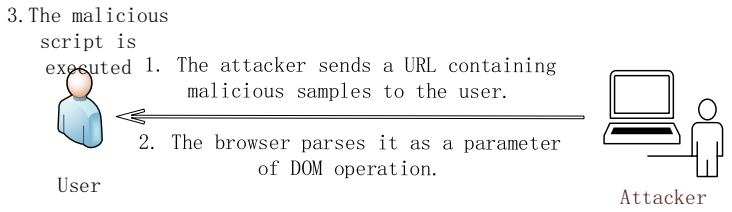


Fig. 3 DOM-based XSS.

However, with the application of javascript and ajax technology, the client interaction technology has been taken to a new level, and the form of XSS attack has become more flexible. Through analysis of various attack payloads, it is found that Base 64 encoding, HTML entity replacement, URL encoding, and other different encoding methods are used, combined with character case conversion, nesting, replacement, and other mutation methods, and even relying on certain attributes in the cascading style sheet, can form a more complicated and huge number of attack scripts.

1.2 Traditional detection methods

The static analysis method is based on the analysis of the source code, that is, by looking at the web application code to find possible vulnerabilities. Tools such as Fortify SCA, Web XSS Test System, XSSDetect, saferXSS, and others are all run according to the principles of static analysis. The detection accuracy of static analysis is high, but it is difficult to obtain the source code in many real environments, and most methods only detect a certain programming language or dynamic web page language, which is the reason why static analysis is not universal.

The dynamic analysis uses the idea of black-box testing, without the need to obtain the program source code. It will simulate real attack behaviors and inject malicious scripts to determine whether there is an XSS vulnerability through the server's feedback data. Considering the efficiency of searching the test vector and the size of the attack vector set, the efficiency of dynamic analysis is not as good as the efficiency of static analysis, and the quality of the attack vector set used by the dynamic method will greatly affect the detection efficiency.

Machine learning algorithms have powerful feature extraction capabilities and self-learning capabilities. Traditional shallow models have limited expressive capabilities, and cost functions do not have local best advantages. Therefore, this paper uses deep machine learning models.

1.3 Virtual sample generation

Virtual sample generation technology is currently an important technical means in the field of few-shot learning. The purpose is to improve the learning performance of few-shot data sets. It has been successfully used in many fields, such as image recognition, industrial modeling, counting equipment manufacturing, and mechanical signal modeling, etc. The technology can be roughly divided into three categories [1]: generating virtual samples based on specific prior knowledge in the research field; generating virtual samples based on the idea of disturbance; generating virtual samples based on the distribution function in the field of research. We take two typical VSG technologies as examples to explain their main features briefly. Firstly, Bootstrap originally came from the concept of statistics, which is a way to obtain training samples, which generally refers to "full sampling with replacement" [2]. This method was first used in the field of image recognition. For a three-dimensional perspective of an object, a virtual sample is generated based on geometric transformation. The biggest feature is "feature unchanged", that is, only through translation, symmetric transformation, rotation angle, and other methods to generate virtual sample. Bootstrap is equivalent to repeated sampling, only changing the weight of seed samples. However, in this way, the obtained training samples are linearly related, which will lead to problems such as certain parameters that cannot be determined, and the covariance matrix is unstable. Secondly, the method of adding perturbation is to add a perturbation to each dimension of the training samples [3]. It was first used to solve the singularity problem of the intra-class covariance matrix. This method is not perfect, because the calculation is extremely large, and the disturbance value is not easy to determine. The efficiency of the algorithm becomes a new problem.

Assuming that there are original samples (x, y), if (Tx, y_Tf (x)) obtained by transforming T is a reasonable sample, then the sample is a virtu-

al sample generated by transforming T. The transformation T is obtained based on prior knowledge. Due to the different prior knowledge and distribution characteristics in different fields， there is currently no unified virtual sample generation method. This paper improves the traditional VSG method based on prior knowledge. It relies on the difference between the feature vectors of the new isolated sample and the feature vectors of the original training set to determine whether to directly retrain the model， or convert it to knowledge and then use knowledge to guide the generation of virtual samples.

2 Algorithm analysis

2.1 Definition

In classification tasks based on machine learning models， theoretically analyze the relationship between knowledge， models， and data. We first make the following definition.

1） Define S as a set of samples， s as a single sample， v and t are feature vectors and corresponding labels. V and T are feature vectors and corresponding labels of a set of samples. Among them， the feature value vector refers to the word embeddings obtained by word2vec after preprocessing and word segmentation； there are two kinds of tags， the payload of the malicious attack samples is recorded as 1， and the normal HTTP request is recorded as 0. Then there are $s = (v, t)$ and $S = (V, T)$.

2） Define M as the mapping from V to T， the original image is V， the image is T. $T = M(V)$. It can be understood that in this two-class classification problem， each set of vectors V corresponds to a unique label T.

3） Define K as a set of true propositions， representing knowledge. For interpretable features， you can analyze the knowledge of differences in their interpretability. If the interpretability is not enough， this difference can also be regarded as knowledge. The neighborhood distribution centered on it is considered to be new knowledge different from other samples.

S， M， and K can describe the characteristics of data distribution. S is a sample based on the spatial distribution of eigenvalues， M is a model built from existing data， and K is a rule set of a priori knowledge. Through predicate logic reasoning， classification results can be obtained based on sample characteristics. Therefore， S， M， and K can be converted to each other： the sample data set S can be trained to obtain the model M， and the model M can abstract the knowledge K， and adding constraints to the knowledge K can generate a virtual sample S'. Infer a certain logical knowledge K from S， use prior knowledge K to design model M， and use model M to classify the sample data set S.

2.2 Algorithm description

According to the difference between the feature vector of the new isolated sample and the feature vector of the original training set， it is determined whether to directly retrain the model， or firstly convert it into knowledge， and then use the knowledge to guide the generation of virtual samples.

The feature set is defined as $F = \{F_1, F_2, ..., F_n\}$， the virtual sample set is D'， and the improved algorithm is described as followed.

3 Experiments and results

After de-duplication processing， the positive sample data set collected in this experiment has a total of 30580 items. 28355 of them are crawled from thehttp：//xssed.com website， which provides articles， news， and tutorials on XSS vulnerabilities. 1625 pieces of shared data from others onhttps：//github.com； the remaining 600 pieces of usual learning records are all obtained from actual competitions or related tutorials. The negative sample data is a normal HTTP GET request， which only retains the payload part， a total of 181012.

After the sample set is preprocessed， we perform word segmentation processing on the sample data. The rules are shown in Table 2.

We use the Skip-gram model in Word2vec to

Tab. 1 Virtual sample generation algorithm.

Algorithm	
Input:	s: new sample S: training dataset F: feature set x: threshold of distance
Output:	D': virtual samples set
1.	Calculate the distance from the new sample s to the sample in S
2.	Infer prior knowledge K based on distance
3.	Generate virtual samples s' based on prior knowledge
4.	for S_i in S do
5.	if target (S_i)=target(s):
6.	if the feature vector of s over F_i is larger than x :
7.	the feature is fixed and others are sampled in S_i
8.	else:
9.	break ;
10.	else:
11.	the feature vectors of s over F_i are the same as S_i
12.	end for

convert it into embedded word vectors. The embedded word vectors can well represent the characteristics of the word vectors and the correlation among the

Tab. 2 Word segmentation rules and examples.

Rules	Examples
script tag	<script>
<>beginning	<h1
parameter name	name=
function body	alert(
contents between quotation marks	'XSS'
HTTP(s)	http://xxxxx
attribute name	type=
characters and numbers	fu0

vocabularies. We split the training set and the test set in a 7: 3 ratio and built three models of SVM, LSTM, and CNN for training. The evaluation indicators are precision rate, recall rate, and F1 score.

The experimental development language is Python 3.7.6. The machine learning model is based on Tensorflow and Keras. The machine configuration is: Windows 10 x64 operating system, Intel (R) Core (TM) i5-10210U @ 1.60GHz, 16GB memory, NVIDIA GeForce MX250 graphics card.

The experimental results are shown in Figure 4.

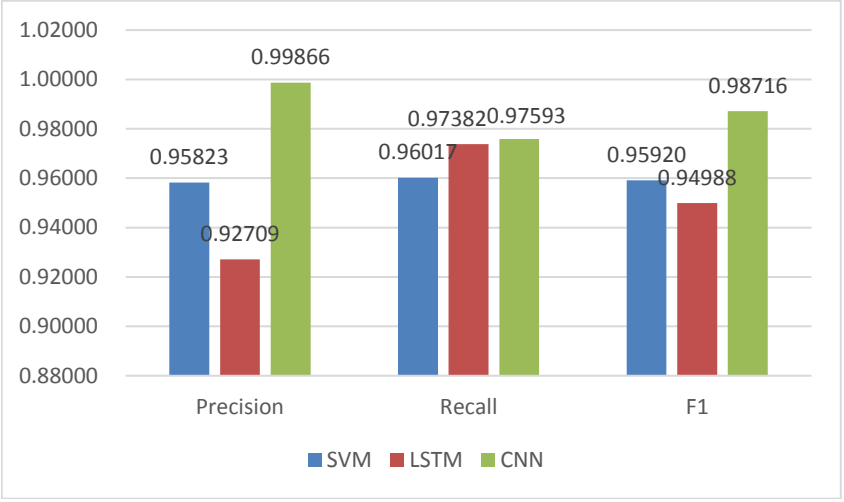


Fig. 4 Experimental results.

It can be seen that the CNN model performs well in all aspects, and its structure is shown in Figure 5.

To verify the robustness of the improved algorithm, we collected 40 CVE samples published in 2020. The findings of these samples were two years or even longer after the training data collection deadline. Every two of them are divided into 20 groups. The CNN model with the best performance is used for retraining. The accuracy results are shown in Figure 6.

The average accuracy of our algorithm is 78.6%, which is higher than the 61.3% and 50.1% of the other two methods. It can more effectively detect malicious samples using the same CVE represented by a single sample.

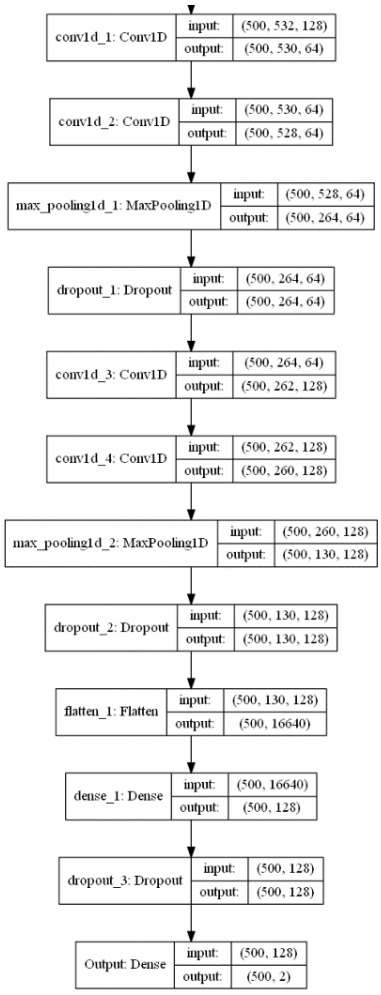


Fig. 5 CNN model.

4 Conclusion

In this paper， we propose a new method of virtual sample generation technology， which makes the new samples quality-higher and more reasonable， based on traditional VSG technology. We formulate a relatively complete detection rule and build machine learning models for a large number of XSS attack samples， among which the CNN model is the best. And this model is applied to the new few-shot problem emerged in 2020. The experimental results show that the accuracy of the samples generated by the method in this paper is far better than the bootstrap and perturbation methods. Moreover， it also expands the variation range of the seed samples to a greater extent， and make the model generalization performance better. Especially， some virtual samples can recurrence vulnerabilities.

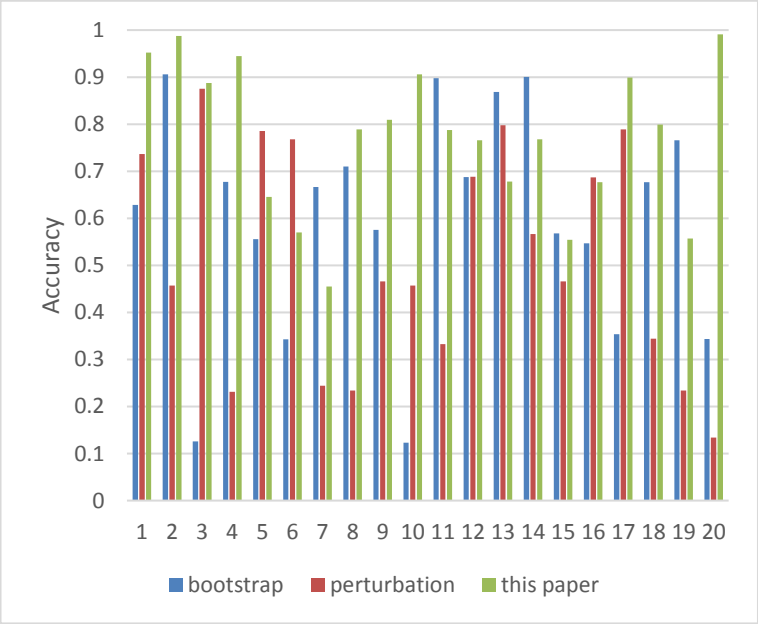


Fig. 6 Comparison of accuracy results.

References:

[1] Yu X, Yang J, Xie Z-Q. Research on Virtual Sample Generation Technology[J]. Computer Science，2011，38：2-5.

[2] Chen Z-S, Zhu B, He Y-L, and Yu L-A. A PSO based virtual sample generation method for small sample sets: Applications to regression datasets [J]. Engineering Applications of Artificial Intelligence，2017，59：236－243.

[3] Zhu B, Chen Z, and Yu L. A novel mega-trend-diffusion for small sample[J]. CIESC Journal，2016，67：820－826.

[4] Lu C and Shen H. Virtual sample generation approach for imbalanced classification [J]. 2018 9th International Symposium on Parallel Architectures，Algorithms and Programming (PAAP) (IEEE)，2018：177－182.

[5] Li A-X, Luo T-G, Xiang T, Huang W-R and Wang L-W. Few-shot learning with global class representations[J]. ICCV，2019：2-4.

[6] Huang Y-M Huang W-R Li L and Li Z-G. Meta-learning Pac-Bayes priors in model averaging[J]. AAAI，2019，12：3-7.

[7] Chu Z-N, Li S-Y. Design of network intrusion detection method based on node growth mahalanobis distance K Mean and HMM[J]. Computer Measurement and Control，2014，22：3-5.

[8] Chen J-X. Research on XSS attack detection technology based on machine learning [D]. Zhejiang University of Technology Master Thesis，2018：8-15.

[9] Zhang G-C. Research on Cross-site Script detection method based on deep convolutional network [D]. Nanchang Hangkong University Master Thesis. 2019：1-3.

[10] Chen Z-S, Zhu B, He Y-L and Yu L-A. A PSO based virtual sample generation method for small sample sets [J]. Applications to regression datasets Engineering Applications of Artificial Intelligence，2017：236-243.

[11] Pan G-B, Zhou Y-H. XSS vulnerability discovery based on static analysis and dynamic detection[J]. Computer Science，2012：1-4.

[12] Gu J-T, Xin Y. Research on XSS vulnerability detection model based on dynamicanalysis[J]. Computer Engineering，2018：1-4.

[13] Sun W, Zhang K-Y, Xue L-F and Xu T-H. XSS vulnerability research review[J]. Information Security Research，2016：3-5.

[14] Pan C-Y, Huang J, Hao J-G, Gong J-X, Zhang Z-J. Survey of weakly supervised learning integrating zero-shot and few-shot learning [J]. Systems Engineering and Electronics，2020，10：2249-2253.

[15] Ji Z, Wang H-R, Yu Y-L, et al. A decadal survey of zero-shot image classification [J]. Scientia Sinica (Informations)，2019，10（49）：1299-1320.

[16] Ji Z, Wang J, Yu Y, et al. Class-specific synthesized dictionary model for zero-shot learning[J]. Neurocomputing，2019，329：339-347.

[17] Qiu J. Research and application of neural network in few shot learning [D]. University of Electronic Science and Technology of China Master Thesis. 2020：6-16.

[18] Zheng Y. Research on few-shot learning method based on deep feature metric[D]. Hefei University of Technology，2019：17-22.

About the authors

LU Dongzhe was born in Xingtai City， Hebei Province. She received a bachelor's degree in Information Engineering University of the Chinese People's Liberation Army Strategic Support Force. She is now in the first year of a master's degree. Her main research direction is machine learning for vulnerability detection. （Email：906188591@qq. com）

基于RLWE的主动安全认证密钥交换协议

王超[1，2]，韩益亮[1，2]，段晓巍[1，2]，李鱼[1，2]

[1]武警工程大学密码工程学院，陕西 西安 710086；

[2]武警部队密码与信息安全保密重点实验室，西安 710086

摘　要：基于格理论构造密钥交换协议成为密钥交换领域的研究前沿，其安全性、高效性、可并行性体现了其重要的理论研究和实用价值。基于紧凑型RLWE公钥加密方案与NewHope-Simple中的密文压缩和NTT转换技术，结合FO转换机制，提出了一种新的主动安全的KEM方案，采用隐性认证和身份标识认证的方式，构造出一种在标准eCK模型下可证明安全的认证密钥交换协议。在协议安全性方面，所提出的协议与NewHope-Simple协议相比，安全性由被动安全提升为主动安全。在密文尺寸方面，协议中的密钥封装机制与CRYSTALS-Kyber在相同参数设置下密文尺寸减小59%，是一个紧凑高效、主动安全的基于加密机制的抗量子认证密钥交换协议。

关键词：RLWE、FO转换、加密机制、认证密钥交换、标准eCK模型

Active Secure Authentication Key Exchange Protocol based on RLWE

WANG Chao[1,2], HAN Yiliang[1,2], DUAN Xiaowei[1,2], LI Yu[1,2]

1.College of Cryptographic Engineering, Engineering University of PAP, Xi'an 710086,China;

2.Key Laboratory of PAP for Cryptology and Information Security, Xi'an 710086, China

Abstract: The construction of key exchange protocol based on lattice theory has become a research frontier in the field of key exchange. Its security, efficiency and parallelism reflect its important theoretical research and practical value. Based on the compact RLWE public key encryption scheme and the ciphertext compression and NTT conversion technology in NewHope-Simple, and combined with FO conversion mechanism, a new active secure KEM scheme is proposed. Using the implicit authentication and identity authentication methods, an authenticated key exchange protocol which can prove secure under the standard eCK model is constructed. In terms of protocol security, the proposed protocol improves from passive security to active security compared with NewHope-Simple protocol. In terms of cipher size, the key encapsulation mechanism in the protocoldecreases the cipher size by 59% with the same parameter settings as CRYSTALS-Kyber, which is a compact, efficient and active secure anti-quantum authentication key exchange protocol based on encryption mechanism.

Key words: RLWE; FO conversion; encryption mechanism; authentication key exchange; standard eCK model

1　引言

认证密钥交换（Authenticated Key Exchange）协议使通信双方或多方在不安全的信道上协商得到共享会话密钥，且为通信参与者提供对彼此的身份认证，以保证在后期的数据加密中实现数据的保密性和完整性。

随着量子计算的发展，基于传统数论的密码

基金项目：国家自然科学基金资助项目（No.61572521）；全军军事类研究生资助课（No.JY2019C241）； 武警工程大学科研创新团队科学基金(No.KYTD201805)。

Foundation item: The National Natural Science Foundation of China (No.61572521); Military Graduate Student Fund Project of Whole Army (No. JY2019C241); The Scientific Foundation of the Scientific Research and Innovation Team of Engineering University of PAP (No.KYTD201805).

问题，如大整数分解和离散对数问题所构造的密钥交换协议，多项式时间内容易受到Shor[1]等量子攻击算法的攻击。1997年，Ajtai[2]首次提出的基于格的加密技术以来，随着格密码学者们十几年的探讨和发掘，提出了大量的实用抗量子密钥交换方案，NIST在2016年启动量子安全密码算法的标准化[3]流程以来，经过三轮的评估筛选，胜出的4个KEM算法中，有3个算法是基于格上的困难问题构造的，这样的发展趋势不仅因为格上发展出的困难问题可以抵抗量子攻击，而且源于基于格的密码系统[4]相比基于编码、基于多项式构造的密码系统[5]，通常在算法上简单、高效可并行化，具有很强的可实施性和应用性。

格上本身的困难问题如最短向量问题（SVP）、最近向量问题（CVP）[6]等构造出的密码方案虽然具有很强的安全性，但是实现效率较低， 2005年Regev等[7]基于格理论提出错误学习问题（Learning with Errors，LWE），其困难性可归约到一般格上CVP的最坏情形，基于LWE问题设计的加密[4]、签名[8]、密钥交换[9]、全同态加密[10]等在后量子密码算法中具有很强的竞争力，2010年Lyubashevsky等[11]提出基于环上的错误学习问题（Ring Learning with Errors，RLWE），其困难性可归约到理想格上的SVP的最坏情形。RLWE独特的环结构结合NTT转换[12]，使基于RLWE的密码方案在算法性能、密钥尺寸上具有很大的优势。

近年来，基于LWE、RLWE、MLWE[13]问题构造密钥交换协议的研究上，出现了许多新型、高效的密钥交换方案，2012年Ding等[14]首次提出了错误协调机制，基于SLWE和该机制构造了密钥交换协议，并将其扩展到RLWE上。2014年，Peikert[15]提出一种新的错误协调机制构造了一种被动安全的密钥封装协议，且给出了提升为主动安全密钥交换协议的框架，这种新的错误协调机制将密文长度减少为50%。2015年，Bos等[16]基于RLWE问题和Peikert提出的错误协调机制结合传统的签名方案将KE方案转换为AKE方案并集成到了TSL协议中。2016年Alkim等[17]提出基于RLWE的后量子密钥交换方案NewHope。同年，Alkim等[18]基于NewHope方案提出了一种使用加密机制的密钥交换协议NewHope-Simple，降低计算复杂度的同时只产生了很小的通信量代价，但是该协议在ROM模型中仅能达到IND-CPA安全。2018年，Bos等[12]基于MLWE提出了CRYSTALS-Kyber方案，将基于错误学习问题的密钥交换协议安全性提升到了IND-CCA级别。2019年，李子臣等[19]基于RLWE提出IND-CCA安全的密钥封装机制，并构造了认证密钥交换协议。同年，杨亚涛等[20]在eCK模型下基于RLWE提出了支持身份隐私保护的认证密钥交换协议。2020年，Hövelmanns[21]等在量子随机预言模型下提出将任何被动安全的公钥加密（PKE）转化为认证密钥交换协议（AKE）的通用结构FO-AKE，对NIST提交的方案进行实例化，并给出了严格的正确性和安全性证明。

本文对NewHope-Simple中的密钥封装机制进行改进，提出一种新型的基于加密机制的主动安全KEM方案和AKE协议。首先基于紧凑型RLWE公钥加密方案与NewHope-Simple中的密文压缩和NTT转换技术，提出一种IND-CPA安全的高效公钥加密方案，而后利用FO转换机制将其转换为IND-CCA安全的新型KEM方案，最后基于新型KEM方案，结合协议双方分别拥有长期、临时密钥对的隐性认证和身份标识认证两种方式，构造出IND-CCA安全的双方认证密钥交换协议。本文协议与相同参数设置下的NewHope-Simple方案相比，安全性由IND-CPA安全提升为IND-CCA安全。协议中的密钥封装机制与CRYSTALS-Kyber在相同参数设置下（$n = 256, d_u = 11, d_v = 3$），密文尺寸减少59%。与文献［19］中基于RLWE提出的IND-CCA安全的密钥封装机制相比，密文尺寸减少43.26%，通信量减少10.23%。新协议在eCK模型[20]下可证明安全，并满足未知密钥共享安全性和弱的完美前向安全性。本文的方案和数据安全性可以归约到格上RLWE问题的困难性，是一种紧凑高效、主动安全的抗量子双方认证密钥交换协议。

2 相关知识

2.1 RLWE问题

定义1[11].RLWE分布.定义n维整系数多项式环为$R^n = \mathbb{Z}(x)/x^n + 1$，其中$n \in \mathbb{Z}$为2的幂次方，设是$R_q^n = R/qR = \mathbb{Z}(x)/x^n + 1$模为正整数$q$的多项式商环。在$R_q^n$上随机均匀选取多项式向量$a$和秘密

值s，在χ_q^n上随机均匀选取错误向量e，χ_q^n在R_q^n上服从某种公开的特定分布。令$b = as + e$，则(a, b)为$R_q^n \times R_q^n$上的RLWE分布$A_{s,\chi}$。

定义2[11].RLWE判定问题给定样本分布(a, b)，能否以不可忽略的优势区分(a, b)为RLWE分布$A_{s,\chi}$和$R_q^n \times R_q^n$上的随机均匀分布。

2.2 扩展输出函数Sam()

定义3[12].假设$Sam(\,)$为一个扩展输出函数，它可以将一个字符串扩展为任意长度或对应产生一个特定分布中的值，且$Sam(\,)$的输出值分布是理想的。假设通过向$Sam(\,)$函数输入r产生一个满足中心二项分布[12] ψ_k^n的值r'，可记为$Sam(r) = r' \in \psi_k^n$。

2.3 密文压缩技术

定义5[12].压缩函数$Compress_q(x, d) = 2^d/q \cdot x \bmod^+ 2^d = y$，函数的输入为$x \in \mathbb{Z}_q$，输出记为$y \in \{0, \cdots, 2^d - 1\}$，其中$d < \lceil \log_2 q \rceil$。

定义6[12].解压函数，$Decompress_q(y, d) = q/2^d \cdot y = x'$，输入为$y \in \{0, \cdots, 2^d - 1\}$，输出记为$x' \in \mathbb{Z}_q$，其中$d < \lceil \log_2 q \rceil$，$x$与$x'$满足$|x' - x \bmod^{\pm} q| \leqslant B_q = q/2^d$。

2.4 NTT转换

定义7[12].设$R^n = \mathbb{Z}(x)/x^n + 1$，其中$n$为2的幂次方，$q$为素数且满足$q \equiv 1 \bmod 2n$，则定义在$R_q^n$上的NTT可有效实现，对于$a, b, c \in R_q, c = ab$，存在$c = NTT^{-1}(NTT(a) \cdot NTT(b))$。

定义8[12].设多项式$\mathrm{g} = \sum_{i=0}^{1023} g_i X^i \in R^q$，$\omega$是g的本原$n$次根，$\gamma = \sqrt{\omega}$，定义

$$NTT(\mathrm{g}) = \hat{\mathrm{g}} = \sum_{i=0}^{1023} \hat{g}_i X^i, \hat{g}_i = \sum_{i=0}^{1023} \gamma^j g_j \omega^{ij} \quad (1)$$

$$NTT^{-1}(\hat{\mathrm{g}}) = \mathrm{g} = \sum_{i=0}^{1023} g_i X^i, g_i = n^{-1}\gamma^{-i} \sum_{i=0}^{1023} \hat{g}_j \omega^{-ij} \quad (2)$$

3 带身份标识的双方认证密钥交换协议

本节分三步构造带身份标识的双方认证密钥交换协议。

3.1 IND-CPA安全的公钥加密方案

本文基于紧凑型RLWE公钥加密方案[4]和NewHope-Simple中的密文压缩技术和NTT转换技术，提出一种IND-CPA安全的新型公钥加密方案，*Algorithm*1为密钥生成算法，*Algorithm*2为加密算法，*Algorithm*3为解密算法，下面对3个算法作以具体介绍。

*Algorithm*1:*CPA.KeyGen*( ):*Keygeneration*( )

1. $seed \xleftarrow{\$} \{0,1\}^l$;
2. $\hat{a} \leftarrow Parse(SHAKE\text{-}128(seed))$;
3. $s, e \xleftarrow{\$} \psi_{16}^n$;
4. $\hat{s} \leftarrow NTT(s), \hat{b} \leftarrow \hat{a} \cdot \hat{s} + NTT(e)$;
5. $Output: pk = (\hat{b}, seed), sk = \hat{s}$;

*Algorithm*2:*Enc*($pk = (\hat{b}, seed), m, r$):*encryption*( )

1. $seed \xleftarrow{\$} \{0,1\}^l$;
2. $\hat{a} \leftarrow Parse(SHAKE\text{-}128(seed))$;
3. $r \leftarrow \{0,1\}^l, Sam(r) = r' \in \psi_{16}^n, \hat{r} \leftarrow NTT(r')$;
4. $m \leftarrow \{0,1\}^L, e', e'' \xleftarrow{\$} \psi_{16}^n$;
5. $c_1 = Compress_q(u = \hat{a} \cdot \hat{r} + NTT(e'), d_u)$;
6. $c_2 = Compress_q(v = NTT^{-1}(\hat{b} \cdot \hat{r}) + e'' + m \cdot q/2, d_v)$;
7. $Returnc = (c_1, c_2)$;

*Algorithm*3:*CPA.Dec*($sk = \hat{s}, c = (c_1, c_2)$):*decryption*( )

1. $u' = Decompress_q(c_1, d_u)$;
2. $v' = Decompress_q(c_2, d_v)$;
3. $m' = Compress_q(v' - NTT^{-1}(u' \cdot \hat{s}), 1)$;

3.2 IND-CCA安全的新型KEM方案

已知哈希函数$G: \{0,1\}^* \to \{0,1\}^L, H: \{0,1\}^* \to \{0,1\}^l$，利用FO转换将上节的IND-CPA安全的公钥加密方案转变为IND-CCA安全的KEM方案。本节设计的IND-CCA安全的KEM方案包括*Algorithm*1: *NNS.KeyGen*( )、*Algorithm*4: *NNS.Encaps*( )和*Algorithm*5: *NNS.Decaps*( )。

如果重加密后验证失败，则返回随机协商密

*Algorithm*4:*NNS.Encaps*($pk = \hat{b}$)

1. $m \leftarrow \{0,1\}^L, \sigma \leftarrow \{0,1\}^l$;
2. $K' = G(\sigma, m)$;
3. $(c_1, c_2) = CPA.Enc(pk = \hat{b}, \sigma, H(\sigma, m)); c_3 = G(\sigma) + m$;
4. $c = (c_1, c_2, c_3)$;
5. $K = H(K', c)$;
6. $Return(c, K)$;

Algorithm5:$NNS.Decaps\left(sk=\left(\hat{s},\hat{b},seed,x\right),c=\left(c_1,c_2,c_3\right)\right)$

1.$\sigma'=CPA.Dec\left(\hat{s},\left(c_1,c_2\right)\right)$;

2.$m'=c_3-G\left(\sigma'\right)$;

3.$K''=G\left(\sigma',m'\right)$;

4.$\left(c_1',c_2'\right)=CPA.Enc\left(pk=\hat{b},\sigma',H\left(\sigma',m'\right)\right)$;

5.*If*$\left(c_1',c_2',c_3'\right)=c,returnK=H\left(K'',c\right)$;

6. *else*,$returnK=H\left(x,c\right)$;

钥$K=H\left(x,c\right)$，x是一个随机秘密种子。

3.3 带身份标识的双方认证密钥交换协议

通信参与者*Alice*和*Bob*各自利用*Algorithm*1生成长期公私钥$\left(pk_A,sk_A\right),\left(pk_B,sk_B\right)$和临时公私钥$\left(pk_a,sk_a\right),\left(pk_b,sk_b\right)$，通信双方分别公布自己的公钥信息和身份标识信息$\left(pk_A,sk_a,ID_A\right),\left(pk_B,sk_b,ID_B\right)$，利用*Algorithm*2和*Algorithm*3中的密钥封装机制构造了一个基于RLWE问题的带身份标识的双方认证密钥交换协议，使得通信双方最终得到一个相同的会话密钥，相比文献［19］中构造的AKE协议，本文协议通过在会话密钥中增加身份标识信息可以有效的抵抗未知密钥共享攻击，下面是本文协议的具体构造过程。

Alice		*Bob*
$\left(pk_A,sk_A\right)\leftarrow NNS.KeyGen()$		$\left(pk_B,sk_B\right)\leftarrow NNS.KeyGen()$
$\left(pk_a,sk_a\right)\leftarrow NNS.KeyGen()$		$\left(pk_b,sk_b\right)\leftarrow NNS.KeyGen()$
$\left(c_B,K_B\right)\leftarrow NNS.Encaps\left(pk_B\right)$	$\underline{c_B,c_b}$	$K_B'\leftarrow NNS.Decaps\left(sk_B,c_B\right)$
$\left(c_b,K_b\right)\leftarrow NNS.Encaps\left(pk_b\right)$		$K_b'\leftarrow NNS.Decaps\left(sk_b,c_b\right)$
$K_A'\leftarrow NNS.Decaps\left(sk_A,c_A\right)$		$\left(c_A,K_A\right)\leftarrow NNS.Encaps\left(pk_A\right)$
$K_a'\leftarrow NNS.Decaps\left(sk_a,c_a\right)$	$\underline{c_A,c_a}$	$\left(c_a,K_a\right)\leftarrow NNS.Encaps\left(pk_a\right)$
$SK=H\left(K_B,K_b,K_A',K_a',ID_A,ID_B\right)$		$SK=H\left(K_A,K_a,K_B',K_b',ID_A,ID_B\right)$

图1 带身份标识的双方认证密钥交换协议

4 参数设置和正确性分析

4.1 参数设置

为了使本节公钥加密方案至少达到128位的抗量子安全性，采用与NewHope-Simple相同的参数设置，选取模数$q=12289<2^{14}$，维度$n=1024$的多项式环$R_n^q=\mathbb{Z}\left(x\right)/x^n+1$，$k=16$的中心二项分布$\psi_k^n$，$\omega=49$（多项式系数有效位为14位）的NTT算法。为了更好地平衡安全性、解密错误概率、密文尺寸，压缩和解压函数中的参数设为$d_u=d_v=3$。

4.2 正确性分析

定义δ为解密失败概率，$\delta=\Pr\left[\left\|er+e''+l_v-se-sl_u\right\|_\infty\geqslant q/4\right]$，那么NNS.CPA方案的解密成功概率为$1-\delta$，其中$\delta<2^{-60}$，$l_u\in R_d^n$，$l_v\in R_d^n$，$R_d^n$为上$R^n$的分布。下面证明$m=m'$，将*Algorithm*2中的$c_1=Compress_q\left(u=\hat{a}\cdot\hat{r}+NTT\left(e'\right),d_u\right)$代入$u'=Decompress_q\left(c_1,d_u\right)$得$u'=Decompress_q\left(c_1=Compress_q\left(u=\hat{a}\cdot\hat{r}+NTT\left(e'\right),d_u\right),d_u\right)$，计算得$u'=\hat{a}\cdot\hat{r}+NTT\left(e'\right)+l_u$。同理将*Algorithm*2中的$c_2$代入到*Algorithm*3中的$v'=Decompress_q\left(c_2,d_v\right)$，可得$v'=NTT^{-1}\left(\hat{b}\cdot\hat{r}\right)+e''+m\cdot q/2+l_v=asr+er+e''+m\cdot q/2+l_v$，进而可得$v'-NTT^{-1}\left(u'\cdot\hat{s}\right)=er+e''+m\cdot q/2+l_v-se'-sl_u$，令$\omega=er+e''+l_v-se'-sl_u$，$\left\|\omega\right\|_\infty<q/4$，得$v'-NTT^{-1}\left(u'\cdot\hat{s}\right)=\omega+m\cdot q/2$。根据*Algorithm*3可知$m'=Compress_q\left(v'-NTT^{-1}\left(u'\cdot\hat{s}\right),1\right)$，可得$\left\|v'-NTT^{-1}\left(u'\cdot\hat{s}\right)-m'\cdot q/2\right\|_\infty=\left\|\omega+m\cdot q/2-m'\cdot q/2\right\|_\infty\leqslant q/4$，又因为$\left\|\omega\right\|_\infty<q/4$，根据三角不等式，有$\left\|q/2\cdot\left(m-m'\right)\right\|_\infty<q/4\cdot2$，又因为$q$为奇数，所以有$m=m'$，由此可得本文公钥加密方案的正确性，解密正确率为$1-\delta$，又因为$H$和$G$为随机预言模型下的哈希函数，根据文献［21］中提供的FO转换技术的具体安全界限，由FO转换产生的IND-CCA

安全的KEM方案的解密正确性也为$1-\delta$，因此根据IND-CCA安全的新型KEM方案构造的密钥交换方案，通信双方可以协商出正确的会话密钥。

5 安全性分析

本节分为两小节，5.1节对本文提出的KEM方案进行安全性证明，5.2节对本文提出的AKE协议的安全属性进行分析。

5.1 KEM方案安全性证明

证明基于RLWE困难问题的构造的公钥加密方案为IND-CPA安全。定义三个游戏G_0，G_1，G_2，G_0为IND-CPA攻击，在G_0中定义一个挑战者和一个敌手A，对手拥有加密查询能力，挑战者随机选取一个$t, t \leftarrow \{0,1\}$，加密m_t后得到密文c发送给敌手A，A通过加密查询训练（没有查询m_t的权限）给出t'，定义敌手赢得游戏的优势为$Adv_{PKE}^{CPA}(A)=\left|\Pr\left[t=t' in the game\ G_0\right]-1/2\right|$。在$G_1$中公钥$\hat{b} \leftarrow \hat{a}\cdot\hat{s}+NTT(e)$被一个从$R_q$中随机均匀选取的随机值代替，对于敌手$A$，$G_1$与$G_0$在计算上是不可区分的，这是基于判定RLWE问题的困难性，总存在一个攻击判定RLWE问题的敌手B，在相同的攻击时间内，有$\left|\Pr\left[t=t' in the game\ G_0\right]-\Pr\left[t=t' in the game\ G_1\right]\right| \leqslant Adv_{n,q,k}^{RLWE}(B)$。在$G_2$中，用于生成挑战密文的$u=\hat{a}\cdot\hat{r}+NTT(e')$和$v''=NTT^{-1}\left(\hat{b}\cdot\hat{r}\right)+e''$，被从$R_q$中随机均匀选取的随机值代替，同样基于判定RLWE问题的困难假设，对于敌手A，G_1与G_2在计算上是不可区分的且总存在敌手B，在相同的攻击时间内有$\left|\Pr\left[t=t' in the game\ G_1\right]-\Pr\left[t=t' in the game\ G_2\right]\right| \leqslant Adv_{n,q,k}^{RLWE}(B)$，因为$v''$与生成挑战密文的$t$的选取是独立的，所以有$\Pr\left[t=t' in the game\ G_2\right]=1/2$，代入$\left|\Pr\left[t=t' in the game\ G_1\right]-\Pr\left[t=t' in the game\ G_2\right]\right| \leqslant Adv_{n,q,k}^{RLWE}(B)$，得$\Pr\left[t=t' in the game\ G_1\right] \leqslant Adv_{n,q,k}^{RLWE}(B)+1/2$，再代入$\left|\Pr\left[t=t' in the game\ G_0\right]-\Pr\left[t=t' in the game\ G_1\right]\right| \leqslant Adv_{n,q,k}^{RLWE}(B)$可得$Adv_{PKE}^{CPA}(A)=\left|\Pr\left[t=t' in the game\ G_0\right]\right| \leqslant 2\cdot Adv_{n,q,k}^{RLWE}(B)$，表明敌手$A$通过IND-CPA赢得游戏的难度要大于破解RLWE问题的难度，因此A赢得游戏的优势是可忽略的，得证本节的公钥加密方案为IND-CPA安全。

因为FO转换机制为通用的安全性提升机制，关于本文KEM方案的IND-CCA安全性的具体转换证明参见文献［22］。

5.2 AKE协议安全属性分析

新协议可以抵抗未知密钥共享攻击，具有弱的完美前向安全性，并且在标准eCK模型下可证明安全，本节将分三小节详细分析新AKE协议的这些安全属性。

5.2.1 未知密钥共享安全性

如果会话密钥中不含有身份信息ID，则会导致如下的未知密钥共享攻击，通信参与者A和B建立*Session*1，敌手M与参与者B建立*Session*2。敌手M将A的长期公钥注册为自己的公钥，通过中间操作，使A认为与B建立了相同的会话密钥，而B认为与M建立了相同的会话密钥。建立会话密钥以后的通信过程中，敌手M可以截获*Session*1中A发送给B的用会话密钥加密的消息，在*Session*2中发送给B，由于B与A建立了相同的会话密钥SK，因此可以解密该消息并相信消息是M发送的，并将A想要的反馈信息发送给M，导致信息泄露。在eCK模型中，M可对*Session*2进行*SessionKeyReveal*查询获得SK，并宣布*Session*1为*Test*会话，进而获得*Test*会话中的密钥，赢得游戏。

B在同时参与*Session*1与*Session*2时，选取的临时公私钥对是不同的，且本文协议中会话密钥包含会话双方的身份标识信息ID_A和ID_B，由于身份信息是唯一标识的，敌手M无法与B在*Session*2中建立与*Session*1相同的会话密钥，从而无法构成未知密钥共享攻击。

5.2.2 弱的完美前向安全性

在本文提出的AKE协议中，会话密钥的协商必须包含通信双方的长期公私钥和此次会话的临时公私钥两个计算要素，敌手在获得了某个参与者的长期私钥后，如果长期私钥在泄露之前的会话没有被敌手破坏（没有对会话中的参与者进行临时私钥查询），那么敌手仍然无法求出参与者在长期私钥泄露之前协商所获得的会话密钥，因此本文协议具有弱的完美前向安全性。

5.2.3 AKE协议安全性证明

该协议安全性是指在敌手M存在且拥有eCK模型中允许的最强攻击能力的情况下，只有通信双方可以通过协议共享会话密钥SK，敌手M无法在多项式时间内区分真实会话密钥和随机生成值。

定理1

已知哈希函数H和G为随机预言机，且RLWE困难问题假设成立，则本文构造的认证密钥协议在标准eCK模型下可证明安全。如果存在敌手M在多项式时间内，通过查询至多n个诚实参与者和至多λ次会话以优势$Adv_{AKE}^{eCK}(M)$赢得游戏，则一定存在一个区分器D，可以在多项式时间内以优势$Adv_{AKE}^{eCK}(D)$解决RLWE问题，且

$$Adv_{AKE}^{eCK}(D) \geqslant 1/2\min\left(2/\lambda^2 Adv_{AKE}^{eCK}(M), 1/n\lambda Adv_{AKE}^{eCK}(M)\right) \tag{3}$$

证明

假设H和G为随机预言机，$SK = H(K_B, K_b, K_A', K_a', ID_A, ID_B)$，多项式时间内解决RLWE问题的优势是可忽略的，如果敌手M在多项式时间内以不可忽略的优势赢得游戏，那么M可以通过以下三种情形的攻击区分真实的会话密钥和随机生成值。

情形1

猜测攻击：敌手M通过猜测得到通信双方的会话密钥。

情形2

密钥复制攻击：敌手M迫使参与者发起另外一次会话t，并建立与*Test*会话相同的会话密钥SK。此时敌手M通过*SessionKeyReveal*查询得到会话t的会话密钥，进而获得*Test*会话的会话密钥。

情形3

伪造攻击：敌手M在某时刻成功计算出会话密钥的所需值$\left(K_B, K_b, K_A', K_a'\right)$，进而通过$H$查询得到$SK$。

针对情形1

由于H是输出为256比特的随机预言机，因此敌手M通过t次猜测猜到会话密钥的概率为$O\left(t/2^{256}\right)$，可知通过猜测攻击无法在有效的时间内低成本的得到真实的会话密钥。

针对情形2

根据假设，H和G为随机预言机，因此产生碰撞的概率是可以忽略的。由于eCK模型不允许对一个参与者同时进行*Long - termKeyReveal*查询和*EphemeralKeyReveal*查询，而且同一个参与者在不同会话中选取的临时密钥对是不同的，因此两次会话拥有相同的参与者和相同的临时密钥对的概率是可以忽略的。综上，敌手M无法进行密钥复制攻击。

针对情形3

如果敌手M通过伪造攻击，能以不可忽略的优势区分*Test*会话的会话密钥和一个等长的随机生成值赢得游戏，则可以构造出一个区分器D，它可利用M作为子程序以不可忽略的优势解决RLWE问题。为了便于讨论，情形3可分为以下两种情况且这两种情况中至少有一种发生的概率大于等于1/2。

情形3.1

*Test*会话存在匹配会话，由2个诚实的参与者拥有

情形3.2

所有的诚实参与者都不拥有*Test*会话的匹配会话

首先分析情形3.1，游戏开始由D随机选择两个会话，验证他们是否为匹配会话，若不是匹配会话，则D终止并重新选择，若为匹配会话且有两个诚实参与者A和B拥有该匹配会话，则D将匹配会话中的A和B的临时公钥pk_a和pk_b用U和V代替，D最多可选取λ次会话，因此本次选择的概率为$2/\lambda^2$。

攻击开始后，M以$2/\lambda^2$的概率选中其中一个会话为*Test*会话，另一个作为*Test*会话的匹配会话。会话双方最终得到会话密钥$SK = H(K_B, K_b, K_A', K_a', ID_A, ID_B)$，若$M$获得$SK$，那么$M$一定可以计算出$K_A' \leftarrow NNS.Decaps\left(sk_A, c_A\right)$，$K_a' \leftarrow NNS.Decaps\left(sk_a, c_a\right)$，并通过$B$的长期公钥和临时公钥可以计算出$K_B, K_b$，最终通过$H$查询得到$SK$. 由于在eCK模型中不允许敌手在*Test*会话中对同一个参与者同时进行长期私钥和临时私钥查询，因此敌手通过查询得到sk_A, sk_a并计算出K_A', K_a'的概率可以忽略不计。而且H查询发生碰撞的概率也可以忽略不计。因此若M可以在只拥有pk_A, pk_a的情况下成功进行伪造攻击获得SK，那

么根据假设，D一定可以解决RLWE问题.已知D选择一对匹配会话的概率为$2/\lambda^2$，因此D与M的优势关系为：$Adv_{AKE}^{eCK}(D) \geqslant 2/\lambda^2 Adv_{AKE}^{eCK}(M)$。

分析情形3.2

D随机选择一个参与者B，在此情形中假设D不知道B的临时私钥sk_b，为了解决RLWE问题，D对B发起的与参与者C的会话消息进行处理，在这里假设敌手M已经完全控制C.

首先D随机选取sk_b'作为B的临时私钥，在此情况下，模型允许D计算参与者B的会话密钥SK和临时密钥，所以D有向敌手M返回K_b'和H查询值的能力，返回值可能为真实值也可能为随机值。由于参与者C被敌手控制，敌手M可以进行*Long - termKeyReveal*查询和*EphemeralKeyReveal*查询来获得C的长期密钥和临时密钥，因此M可以解封K_C', K_c'，通过向D进行H查询得到会话密钥SK.如果敌手M赢得游戏，说明SK的返回值为真，D一定可以解决RLWE问题。然后D选取随机会话*Session*，会话*Session*中B作为应答者，设此会话的发起者为A，当会话发起后，D按照协议流程对会话进行处理，敌手M以$1/n\lambda$的概率选中随机会话*Session*作为*Test*会话（以$1/n$的概率选中B，以$1/\lambda$的概率选中随机会话*Session*）。若敌手M赢得游戏，则M一定进行了H查询，D可解决RLWE问题。此情形下D成功解决RLWE问题的优势为$Adv_{AKE}^{eCK}(D) \geqslant 1/n\lambda Adv_{AKE}^{eCK}(M)$。

综上可得：$Adv_{AKE}^{eCK}(D) \geqslant 1/2\min\left(2/\lambda^2 Adv_{AKE}^{eCK}(M), 1/n\lambda Adv_{AKE}^{eCK}(M)\right)$，在RLWE困难问题假设下，$D$解决RLWE问题的优势是可忽略的，进而由公式3可知敌手M赢得游戏的优势也是可忽略，因此本文提出的AKE协议在标准eCK模型下安全的，证毕。

6 效率分析

根据NIST第三轮后量子密码标准化的评选结果，本文选取目前最具代表性的基于加密机制的IND-CCA安全的CRYSTALS-Kyber协议，无认证的KE协议NewHope-Simple，以及文献［19］基于RLWE困难问题假设构造基于加密机制的AKE协议[19]。在密钥交换协议的安全性方面，通过对维度n、模数q、困难问题、安全性、安全模型五个方面进行对比分析，在相同参数设置的条件下，本文提出的认证密钥协议较NewHope-Simple安全性提升为IND-CCA安全。另外，本文协议在标准eCK模型下可证明安全，能够达到弱的完美前向安全性，而CRYSTALS-Kyber协议仅能达到CK安全。在文献［19］安全性分析的基础上，增加了通信双方的身份标识信息（ID_A，ID_B），可以有效抵抗未知密钥共享攻击。表1是本文协议与基于加密机制设计的抗量子密钥交换协议的安全性对比。

表1 基于加密机制设计的后量子密钥交换协议的安全性对比

协议	维度n	模数q	困难问题	安全性	安全模型
CRYSTALS-Kyber	256	7681	MLWE	CCA	CK
Ref[19]	1024	12289	RLWE	CCA	eCK
NewHope-Simple	1024	12289	RLWE	CPA	RO
本文协议	1024	12289	RLWE	CCA	eCK

在KEM方案的效率方面，通过对封装密钥、解封密钥和密文尺寸（Bytes）、通信量进行对比分析。本文提出的KEM方案在保证IND-CCA安全的前提下，在参数设置相同时（n=1024，q=12289）与文献［19］中的密钥封装机制相比密文量减少了43.26%。在利用密钥封装机制构造的密钥交换协议的通信量方面，较相同维度和模数下的NewHope-Simple和文献［19］有较大优势，分别降低了32%和10.23%。表2是本文KEM方案与NewHope-Simple、文献［19］中密钥封装机制的尺寸对比。

表2 维度为1024时密钥封装机制的尺寸对比

KEM方案	封装密钥 (size/B)	解装密钥 (size/B)	密文量 (size/B)	通信量 (size/B)
NewHope-Simple	1824	1792	2176	4000
Ref[19]	1451	3200	1579	3030
本文方案	1824	3776	896	2720

在与CRYSTALS-Kyber相同的参数设置下，本文方案的密文量减少了59.46%，如表3所示。

表3　与CRYSTALS-Kyber参数设置相同时密钥封装机制的密文尺寸对比

KEM方案	维度n	d_u	d_v	密文量(size/B)
CRYSTALS−Kyber	256	11	3	1184
本文方案				480

7　结束语

本文基于RLWE困难问题假设，结合NewHope-Simple中的密文压缩函数和NTT转换技术，首先提出了一个IND-CPA安全的PKE方案，采用通用的FO转换机制，将PKE方案转换为主动安全的KEM方案，与当前基于加密机制的KEM方案相比，新型KEM方案在密文尺寸和密钥协商过程中的通信量有较大优势。将隐性认证与身份标识认证结合构造的双方认证密钥交换协议能够更有效的抵抗未知密钥共享攻击。依据NIST第三轮公布的结果，虽然基于RLWE设计的NewHope算法落选，但是由于通过结构化程度较高的RLWE构造的密码算法在尺寸和执行效率方面的独特优势，NewHope算法对基于结构化程度较低的LWE和MLWE构造安全性与效率更加平衡的密钥交换协议仍有很好的借鉴意义。

参考文献:

[1] Shor P W . Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer[J]. Siam Review，1999，41(2):303-332.

[2] Ajtai M, Dwork C. A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence[C]//Proceedings of the 29th Annual ACM Symposium on the Theory of Computing. New York：ACM，1997：284-293.

[3] Chen L, Jordan S, Liu Y. -K，
Moody D, Peralta R, Perlner R and Smith-Tone D. Report on Post-Quantum Cryptography[R]. National Institute of Standards and Technology，2016：1-15.

[4] Peikert C . A Decade of Lattice Cryptography[J]. Foundations & Trends in Theoretical Computer Science，2016，10(4):283-424.

[5] 韩益亮, 王众. 基于多变量和LRPC码的抗量子密码方案研究[J]. 信息网络安全，2019，19(8):36-43.

[6] 毕蕾, 李帅钢, 刘亚敏, 等. LWE问题实际安全性分析综述[J]. 信息安全学报，2019，000(002):1-12.

[7] Regev O. LatticesOn, Learning with Errors, Random Linear Codes，and Cryptography[C]//Proc of the 37th Annual ACM Symp on the theory of computing. New York：ACM，2005：84-93.

[8] Ducas L, Lepoint T, Lyubashevsky V, et al. CRYSTALS – Dilithium: Digital Signatures from Module Lattices[J]. IACR Transactions on Symmetric Cryptology，2018：238-268.

[9] Gao X, Ding J, Saraswathy R V, et al. Comparison analysis and efficient implementation of reconciliation-based RLWE key exchange protocol[J]. International Journal of High Performance Computing and Networking，2019，13(2)：141-152.

[10] Chen H, Dai W, Kim M, et al. Efficient Multi-Key Homomorphic Encryption with Packed Ciphertexts with Application to Oblivious Neural Network Inference[C]//Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. New York：ACM，2019：395-412.

[11] Lyubashevsky V, Peikert C, Regev O. On Ideal Lattices and Learning with Errors over Rings[C]//LNCS 6110: Advance in EuroCrypt 2010. Berlin：Springer，2010：1-23.

[12] Bos J W, Ducas L, Kiltz E, et al. CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM[C]//Proc of IEEE European Symposium on Security and Privacy. Piscataway, NJ：IEEE Press，2018：353-367.

[13] Langlois A , StehléDamien. Worst-Case to Average-Case Reductions for Module Lattices[J]. Designs，Codes and Cryptography，2015，75(3)：565-599.

[14] Jintai Ding, Xiaodong Lin. A Simple Provably Secure Key Exchange Scheme Based on the Learning with Errors Problem, Report 2012/688[R]. [S. 1.]：IACR，2012：688.

[15] Peikert C . Lattice Cryptography for the Internet[C]//LNCS 8772: Proc of the Int Conf on Post-Quantum Cryptography. Berlin：Springer，2014：197-219.

[16] Bos J W, Costello C, Naehrig M, et al. Post-Quantum Key Exchange for the TLS Protocol from the Ring Learning with Errors Problem[C]//Proc of IEEE Symposium on Security and Privacy. Washington DC：IEEE Computer Society，2015：553-570.

[17] Alkim E, Ducas L, Poppelmann T, et al. Post-Quantum Key Exchange: A New Hope[C]//USENIX Security Symp. Berkeley, CA：USENIX Association，2016：327-343.

[18] Alkim E, Ducas L, Poppelmann T, et al. NewHope without Reconciliation, Report 2016/1157[R]. [S. 1.]：IACR，2016:1157.

[19] 李子臣, 谢婷, 张卷美, 等. 基于RLWE的后量子认证密钥交换协议[J]. 计算机研究与发展，2019，56(12):2694-2701.

[20] 杨亚涛, 韩新光, 黄洁润, 等. 基于RLWE支持身份隐私保护的双向认证密钥协商协议[J]. 通信学报，2019，40(11):180-186.

[21] Hövelmanns K, Kiltz E, Schäge S, et al. Generic Authenticated Key Exchange in the Quantum Random Oracle Model[C]//IACR International Conference on Public-Key Cryptography. Springer，Cham，2020：389-422.

[22] HofheinzDennis, HövelmannsKathrin, KiltzEike. A Modular Analysis of the Fujisaki-Okamoto Transformation[C]//Theory of Cryptography Conference，Springer，Cham，2017：341-371.

[作者简介]
王超（1997—），男，武警工程大学硕士生，主要研究方向：抗量子密码。

韩益亮（1977—），男，博士，武警工程大学教授，主要研究方向：信息安全、抗量子密码。E-mail：hanyil@163.com。

段晓巍（1997—），男，武警工程大学硕士生，主要研究方向：抗量子密码。

李鱼（1995—），男，武警工程大学硕士生，主要研究方向：抗量子密码。

Post-quantum Key Exchange Protocol Analysis based on Automatic Learning Structure

Zhu Shuaishuai[1], Han Yiliang[2], Yang Xiaoyuan[2], Li Yu[1,2]

1.College of Cryptography Engineering, Engineering University of the PAP, 710086, China;

2.Key Laboratory of Network and Information Security under the PAP, Xi'an, 710086, China

Abstract: We analyzed one of the NIST post-quantum cryptography candidates: NewHope-Key-Exchange, the post-quantum key exchange protocol, using an automatic analysis strategy to attack the security properties of the scheme. Our analysis approach mainly concentrates on the Number Theory Transform(NTT) as well as the RLWE assumption applied in NewHope. The influences of security and efficiency toward NewHope are analyzed based on specially designed attack models applying an automatic analysis oracle. Under the assumption of full security indistinguishability model and partial leakage security indistinguishability model, we configure the key exchange protocol respectively, and evaluate different security strength and efficiency in different scenarios to validate the influences of NTT structure. The quantitative results show that the NTT process performs an significant role in the key exchange protocol.

Key words: Post-Quantum Cryptography; Key Exchange Protocol; Indistinguishability Model; Security Model

1 Introduction

1.1 Background

For the new appealing hardness and designing characters， post-quantum cryptography is a major researching trend of post quantum data security. In NIST's final round of post-quantum algorithm candidates， lattice-based schemes became the most promising ones. With decades of worldwide research， lattice based schemes and their variants are now widely accepted as a fundamental basestone in public key encryption schemes [1-2, 19, 25]， key encapsulation mechanisms [5-6] and key exchange protocols [15-17, 26]. Although， the intractability on lattice， such as the shortest vector problem and the closest vector problem， is assumed to be concrete hard， but many primitives based on lattice are not so dependable because of designing problems. Varieties of vulnerable points can be applied in attacking an algorithm instantiation， such as weak parameters， sampling process， random seeds， and even the unexpected distributions.

NewHope [3-6] is one of the most promising post-quantum candidates in the final contest round that covers public key encryption （PKE）， key encapsulation mechanisms （KEM）[3-4] and key exchange protocols （KEP）. In the official documentation， the passively CPA security level of PKE， NewHope-CPA-PKE is the essential component of NewHope in the continual practical primitives that the subscribers claim semantically security. The NewHope-CPA-KEM is designed by calling the PKE key generation， encryption and decryption algorithms. By applying Fujisaki－Okamoto transform， the adaptively secure NewHope-CCA-KEM can be constructed with PKE algorithms. Also， the NewHope key ex-

change scheme is constructed by basic PKE algorithms， including the NewHope with reconciliation [5] and NewHope without reconciliation [6]. The former one is a complex two-pass key exchange protocol with key encapsulation direct relying on RLWE hardness. The later one is a much simple key exchange with the reuse of NewHope-CPA-PKE. Both of them are theoretically proved semantically secure under quantum oracle model. More specifically， the NewHope key exchange protocol is claimed to be semantically secure under CCA model， and now applied in online key negotiation [5] of Google Chrome browser. As in a key exchange protocol， a major security concern is the identity extraction from all the participators， which is specially considered in adaptively selective identity model that much weaker than selective ciphertext model in indistinguishability attack. As the NewHope without reconciliation is a CPA-KEM wrap， we only consider the security of the former key exchange protocol under adaptively secure model. The interactions of the two-pass messages is compact and quite effective， but it covers the phases of public seed sampling， an uniform Number Theoretic Transform （NTT）， and a negotiation round of reconciliation vectors which generated from the same noise distribution.

1.2 Inspiration

Traditional methodology of cryptographic analysis basically focuses on the semantically provable security under pre-designed theoretical attack models. A major drawback of the traditional routine is the lack of systematical evaluation of the schemes in practical implementation. Varieties of security weak points potentially exists in real instantiations， for example， the sampling algorithm， distribution parameters， NTT and NTT^{-1}. These weak points may have nonnegligible influences that is so far hard to detect and describe by theoretical methods.

With new computing technologies and tools emerging， such as automatic systems based on AI， deep learning systems and Big Data processing technology. A quite natural way to systematically analyze a cryptographic primitives is to apply these techniques in practical instantiations， especially those schemes leaking special features on unsafe channels such as NewHope key exchange protocol. For NewHope key exchange protocol under CCA security model， it is quite necessary to analyze and detect these weak points.

Focusing on the identity indistinguishability during key exchange implementation， we designed a framework based on automatic learning structure to practically simulate the attack that applied in semantical security model， and collected the results for each potential weak points. By realizing the automatic learning structure according to the features released by NewHope implementation， a systematic primitive analyzer that is universally functioning for all the post-quantum algorithms is built.

In this paper， three contributions are made surrounding the analysis of a post-quantum key exchange protocol candidate in the second round of NIST submissions. Firstly， compared with traditional theoretical provable analysis， we firstly explore the automatic analysis approach in the evaluation of security models， which will benefit the development of quantitative analysis of cryptographic protocols. Secondly， applying the automatic learning approach， we quantitatively analyzed the influences of NTT component toward security levels and system efficiency under different attack configuration. Thirdly， a standard methodology of cryptographic protocol is developed with practical steps to handle in real instantiations. By generalizing these steps， any theoretically secure primitives can be detected and evaluated in case of potential weak points. The rest of the paper is organized in five sections. In Section 2 and Section 3， we give some necessary preliminaries and related works. The attack structure and simulation oracle are explained in details in Section 4. Some results of the quantitative tests are demonstrated Section 5. The paper concludes in the final section.

2 Preliminaries

2.1 Notations

All our working space is in the polynomial ring noted as $R_q = \mathbb{Z}_q[x]/x^n + 1$, $s.\ t.\ n \in \mathbb{Z}$ and $q \equiv 1 \bmod 2n$. G is an intelligent learning system, namely, a guess system to yield a result given enough samples of specified set. One of the most important benchmarks of G is accuracy of its results, noted as Acc (G). All bold variables are vectors or matrices or a serial of coordinates.

2.2 Number Theoretic Transform(NTT)

For any $m \in \mathbb{Z}^+$, let $n = 2^m$, $q = 1 \bmod 2n$, and R_q be the polynomial ring. The NTT result of polynomial $\mathbf{p} = \sum_{i=0}^{n-1} x^i$, $\mathbf{p} \in R_q$ is defined as,

$$\mathrm{NTT}(\mathbf{p}) = \sum_{i=0}^{n-1} \hat{p}_i x^i,$$

in which $\hat{p}_i = \sum_{i=0}^{n-1} p^j w^{j(2i+1)}$ and w satisfies $w^{2n} = 1 \bmod 2n$.

Similarly, the inverse result NTT^{-1} is defined as,

$$\mathbf{p} = \mathrm{NTT}^{-1}(\hat{\mathbf{p}}) = \sum_{i=0}^{n-1} p_i x^i = \sum_{i=0}^{n-1} (n^{-1} \sum_{j=0}^{n-1} \hat{p}_j w^{j(2j+1)}) x^i,$$

NTT transform is sort of similar to the discrete Fourier transform to simplify and speed up polynomial multiplication over finite fields.

2.3 Ring-LWE based security models

Definition 1.

(R-LWE) A $(R_q, n, \mathbb{D})$ -LWE problem defined on a R_q: $\mathbb{Z}_q[x]/(x_n + 1)$ is the access to a challenge oracle $\mathcal{O}$ which is either a pseudo-random oracle $\mathcal{O}_p$ or a truly random oracle $\mathcal{O}_i$. $\mathcal{O}_p$ and $\mathcal{O}_i$ are defined as follows:

$\mathcal{O}_p$ generates a value $(u_i, v_i) = (u_i, u_i^T s + e_i) \in R_q \times R_q$, where s is a uniformly random vector in $\mathbb{R}_q$, u_i are randomly picked vectors in $\mathcal{D}_{\mathcal{L},s}$, and $e_i \in \mathbb{Z}_q$ are fresh noise samples in $\Psi_{q,n}$.

$\mathcal{O}_p$ generates a truly uniform random sample from $R_q \times R_q$.

Then the decisional R-LWE problem is to distinguish source oracle of the outputs, and the computational R-LWE problem is to recover s from (u_i, v_i).

Definition 2.

(Search problem on Ring-LWE with partial key leakage, R-SLWE). R-SLWE is defined on the parameter tuple (n, n', S, χ) in which $n' \in \{1, 2, 4, \cdots, n\}$, $S \subseteq \mathbb{Z}^+_{2n'}$ shows the set of index of possible leakage. $\mathbf{s} \leftarrow \chi$ is a randomly sampled secret key, and let $\hat{\mathbf{s}} = \mathrm{NTT}(\mathbf{s})$. Given the partial leakage $(\hat{\mathbf{a}}, \hat{\mathbf{u}} = \hat{\mathbf{a}} \cdot \hat{\mathbf{s}} + \hat{\mathbf{e}}, [\hat{s}_i]_{i \equiv \alpha \bmod 2n', \alpha \in S})$ in which $\mathbf{a}$ is a public matrix chosen from χ, and $\mathbf{e}$ is the noise vector sampled from χ, the R-SLWE problem is to recover $\mathbf{s}$.

Definition 3.

(Decisional problem on Ring-LWE with partial key leakage, R-DLWE). R-DLWE is defined on the parameter tuple (n, n', S, χ) in which $n' \in \{1, 2, 4, \cdots, n\}$, $S \subseteq \mathbb{Z}^+_{2n'}$ shows the set of index of possible leakage. $\mathbf{s} \leftarrow \chi$ is a randomly sampled secret key, and let $\hat{\mathbf{s}} = \mathrm{NTT}(\mathbf{s})$. Given the partial leakage $(\hat{\mathbf{a}}, \hat{\mathbf{u}} = \hat{\mathbf{a}} \cdot \hat{\mathbf{s}} + \hat{\mathbf{e}}, [\hat{s}_i]_{i \equiv \alpha \bmod 2n', \alpha \in S})$ in which $\mathbf{a}$ is a public matrix sampled from χ, and $\mathbf{e}$ is the noise vector sampled from χ, the R-DLWE problem is to distinguish $(\hat{\mathbf{a}}, \hat{\mathbf{u}} = \hat{\mathbf{a}} \cdot \hat{\mathbf{s}} + \hat{\mathbf{e}}, [\hat{s}_i]_{i \equiv \alpha \bmod 2n', \alpha \in S})$ from $(\hat{\mathbf{a}}, \hat{\mathbf{u}} \leftarrow \mathbb{Z}_q, [\hat{s}_i]_{i \equiv \alpha \bmod 2n', \alpha \in S})$.

We note INS_{real} and INS_{presdu} as the real samples and the samples constructed in primitives respectively. For all the R-LWE based secure instances, the following theorem holds.

Theorem 1.

[14] If a cryptographic primitive is polynomial time secure, then for an instance I,

(1) .P$\{I \in \mathrm{INS}^{R\text{-}LWE}_{real}\}$-P$\{I \in \mathrm{INS}^{R\text{-}LWE}_{presdu}\} = \epsilon_1$;

(2) .P$\{I \in \mathrm{INS}^{R\text{-}SLWE}_{real}\}$-P$\{I \in \mathrm{INS}^{R\text{-}SLWE}_{presdu}\} = \epsilon_2$;

(3) .P$\{I \in \mathrm{INS}^{R\text{-}DLWE}_{real}\}$-P$\{I \in \mathrm{INS}^{R\text{-}DLWE}_{presdu}\} = \epsilon_3$,

in which ϵ_1, ϵ_2 and ϵ_3 are negligible.

A key encapsulation mechanism consists three algorithms: $(pk, sk) \leftarrow KeyGen(1^k)$, $(v, c) \leftarrow Encap(pk)$, and $v \leftarrow Decap(sk, c)$, in which k is the security parameter. An IND-CCA secure KEM is defined by the following game between

a challenger $\mathcal{C}$and an adversary $\mathcal{A}$.

Definition 4.

（In Section 2 of ［28］）（IND-CCA secure KEM） $\mathcal{C}$establishes (pk, sk) by running$KeyGen(1^k)$， then uniformly samples a bit$m \in \{0, 1\}$， and generates (v_m^*, c^*) by $Encap(pk)$. $\mathcal{C}$sends (v_m^*, c^*, pk)as a challenge message. For any $c_i \neq c^*$， $\mathcal{A}$ adaptively choose to query the coordinatev_i. Finally， $\mathcal{A}$ makes a guess m'on m. If $m' = m$， $\mathcal{A}$wins the IND-CCA game. The advantage of$\mathcal{A}$is defined as，

$$Adv_{A,KEM}^{IND-CCA}(k) = |P[m' = m] - 1/2|,$$

The KEM is IND-CCA secure if the advantage is bounded by $n(k)$， which is a negligible polynomial of k.

2.4 Overview of NewHope

NewHope [3-4] is a RLWE based key exchange protocol of two parties. Two different routines are applied in designing a key exchange mechanism， namely the NewHope-KEM based on reconciliation [6] and the NewHope-KEM based on encryption [5] .

NewHope-KEM based on reconciliation is a quite complex but efficient key exchange protocol. With two rounds of reconciliation， the two parties negotiate a shared value. The protocol runs between two users Alice and Bob with the following steps：

1. The system initiates public parameters： $n = 1024$, $q = 12289$ and a Gaussian shaped distribution Ψ_{16}^n.

2. Alice samples$\mathbf{a} \leftarrow R_q$， and $\mathbf{s}, \mathbf{e} \leftarrow \Psi_{16}^n$. Then let $\hat{\mathbf{s}} = \mathrm{NTT}(\mathbf{s})$, $\hat{\mathbf{e}} = \mathrm{NTT}(\mathbf{e})$ and $\mathbf{b} = \mathbf{a} \cdot \mathbf{s} + \mathbf{e}$. The tuple $(\mathbf{a}, \mathbf{b})$is sent to Bob.

3 Bob samples $\mathbf{s}', \mathbf{e}', \mathbf{e}' \leftarrow \Psi_{16}^n$, computes $\hat{\mathbf{s}} = \mathrm{NTT}(\mathbf{s}), \hat{\mathbf{e}}' = \mathrm{NTT}(\mathbf{e}'), \hat{\mathbf{e}}' = \mathrm{NTT}(\mathbf{e}')$ and computes $\hat{\mathbf{u}} = \hat{\mathbf{a}} \cdot \hat{\mathbf{s}}' + \hat{\mathbf{e}}'$， $\hat{\mathbf{v}} = \hat{\mathbf{b}} \cdot \hat{\mathbf{s}}' + \hat{\mathbf{e}}'$， $\mathbf{v} = \mathrm{NTT}^{-1}(\hat{\mathbf{v}})$, $\mathbf{k} \leftarrow Encode(V)$.

4. Bob generates a reconsilation vector r by HelpRec$(\mathbf{v})$， and sends $(\hat{\mathbf{u}}, \mathbf{r})$to Alice.

5. Alice computes $\mathbf{w} = \mathrm{NTT}^{-1}(\hat{\mathbf{u}}, \hat{\mathbf{s}})$.

6. Alice generates v by Rec$(\mathbf{w}, \mathbf{r})$， and Bob generates v by Rec$(\mathbf{v}, \mathbf{r})$.

In the protocol， HelpRec（.） is defined as a r-bit reconciliation algorithm， see Algorithm 6.3 in ［14］ ，

$\mathrm{HelpRec}(\mathbf{x}, b) = \mathrm{CVP}_{\hat{D}_4}((\mathbf{x} + b\mathbf{g})2^r/q) \bmod 2^r$.

Rec（.） is an algorithm decoding $\mathbf{r}$from $\mathbf{x}$， see Lemma 6.4 in ［14］，

$\mathrm{Rec}(\mathbf{x}, \mathbf{r}) = \mathrm{Decode}(\mathbf{x}/q - \mathbf{Br}/2^r)$， in which $\mathbf{r}$is the output of HelpRec（$\mathbf{x}$）.

NewHope-KEM based on encryption is a much more brief one with a direct inheritance of Ring LWE public key encryption scheme. It can be described in the following steps：

1. The system initiates public parameters： $n = 1024$, $q = 12289$and a Gaussian shaped distribution Ψ_{16}^n.

2. Alice samples $\mathbf{a} \leftarrow R_q$， and $\mathbf{s}, \mathbf{e} \leftarrow \Psi_{16}^n$. Then let $\hat{\mathbf{s}} = \mathrm{NTT}(\mathbf{s})$, $\hat{\mathbf{e}} = \mathrm{NTT}(\mathbf{e})$ and $\mathbf{b} = \mathbf{a} \cdot \mathbf{s} + \mathbf{e}$. The tuple $(\mathbf{a}, \mathbf{b})$is sent to Bob.

3. Bob samples $\mathbf{s}', \mathbf{e}', \mathbf{e}' \leftarrow \Psi_{16}^n$， $V \leftarrow \{0, 1\}^n$， computes $\hat{\mathbf{s}} = \mathrm{NTT}(\mathbf{s}), \hat{\mathbf{e}}' = \mathrm{NTT}(\mathbf{e}'), \hat{\mathbf{e}}' = \mathrm{NTT}(\mathbf{e}')$and computes $\hat{\mathbf{u}} = \hat{\mathbf{a}} \cdot \hat{\mathbf{s}}' + \hat{\mathbf{e}}'$， $\hat{\mathbf{v}} = \hat{\mathbf{b}} \cdot \hat{\mathbf{s}}' + \hat{\mathbf{e}}'$， $\mathbf{v} = \mathrm{NTT}^{-1}(\hat{\mathbf{v}})$, $\mathbf{k} \leftarrow Encode(V)$.

4. Bob generates a vector $\mathbf{c} \leftarrow \mathbf{v} + \mathbf{k}$， and sends$(\hat{\mathbf{u}}, \mathbf{c})$to Alice.

5.Alice computes $\mathbf{v}' \leftarrow \mathbf{us}$ and $\mathbf{k}' \leftarrow \mathbf{c} - \mathbf{v}'$.

6. Finally， Alice and Bob obtain a shared $\mu$by running$Extract(\mathbf{k}')$ and $Extract(\mathbf{k})$.

The algorithm $Encode()$ is a map from$\{0, 1\}^n$to $\sum_{i=1}^{n} \mathbf{p}\left\lfloor \frac{q}{2} \right\rfloor$， and $Extract()$ is the reverse map.

The security of NewHope. The submission documentation ［4］ claims that KEM based on IND-CPA-PKE achieves IND-CCA secure， and a IND-CCA2-KEM can be constructed by the Fujisaki-Okamoto transform [18] in quantum random oracle model. KEM based on reconciliation is passively secure in the original paper [6] .

2.5 Attack on indistinguishability of output

To avoid identity forge and message re-send-

ing， indistinguishability is a basic property of the output of key exchange protocols. In this subsection， we define the indistinguishability security model of NewHope acceding the security model defined in Definition 4. Then the indistinguishability security characters are fully evaluated by instances of the model. We assume the two-pass key exchange protocol as a black box which is given a pair of identities， such as the tuple $\{id_a, id_b\}$， and yields two messages $(\hat{\mathbf{a}}, \hat{\mathbf{b}})$， $(\hat{\mathbf{u}}, \hat{\mathbf{r}})$. Then in the adversary's view on communication channels， he gets an instance of NewHope protocol， $\{(\hat{\mathbf{a}}, \hat{\mathbf{b}}), (\hat{\mathbf{u}}, \hat{\mathbf{r}})\} \leftarrow \mathrm{INS}(id_a, id_b)$.

Basic Attack. For $id^* = (id_0, id_1)_i \in \{(id_a, id_b), (id_c, id_d)\}$， in which$i \in \{0, 1\}$， the basic attack is a multiple rounds of interactions between a challenger and an adversary with the following steps.

Step 1.

The challenger initiates two instances of key exchange protocol in which one is a real protocol $\mathrm{INS}_{real}(\cdot)$ and the other one is an ideal random generator $\mathrm{INS}_{sim}(\cdot)$.

Step 2.

The adversary launches queries with $id = (id_0, id_1) \in S_{id}$， and the challenger takes id as an input of $\mathrm{INS}_{real}(\cdot)$and sands the outputs back to the adversary.

Step 3.

The adversary randomly selects $\{(id_a, id_b), (id_c, id_d)\} \in S_{id}$and sends it to the challenger. The challenger flips a coin to get a random bit $i \in \{0, 1\}$， and then he takes $id^* = (id_0, id_1)_i \in \{(id_a, id_b)_0, (id_c, id_d)_1\}$ as the challenged identities. Finally， the challenger sends the output $\{(\hat{\mathbf{a}}^*, \hat{\mathbf{b}}^*), (\hat{\mathbf{u}}^*, \hat{\mathbf{r}}^*)\}$ back to the adversary.

Step 4.

The adversary makes more additional queries with $id = (id_0, id_1) \in S_{id}$. If $id = id^*$， the challenger returns $\{(\hat{\mathbf{a}}, \hat{\mathbf{b}}), (\hat{\mathbf{u}}, \hat{\mathbf{r}})\} \leftarrow \mathrm{INS}_{sim}(id_a, id_b)$， else it returns $\{(\hat{\mathbf{a}}, \hat{\mathbf{b}}), (\hat{\mathbf{u}}, \hat{\mathbf{r}})\} \leftarrow \mathrm{INS}_{real}(id_a, id_b)$.

Step 5.

Finally， the adversary outputs a guess$b \in \{0, 1\}$. If $b=i$， the adversary wins in the attack.

Definition 5.

（Indistinguishability security on selective identities of NewHope key exchange） For two group of randomly selected users $\{(id_a, id_b), (id_c, id_d)\} \in S_{id}$， two instances are generated as follows，

$$\{(\hat{\mathbf{a}}, \hat{\mathbf{b}}), (\hat{\mathbf{u}}, \hat{\mathbf{r}})\} \leftarrow \mathrm{INS}(id_a, id_b),$$
$$\{(\hat{\mathbf{a}}', \hat{\mathbf{b}}'), (\hat{\mathbf{u}}', \hat{\mathbf{r}}')\} \leftarrow \mathrm{INS}(id_c, id_d),$$

Given the outputs， the probability that the adversary can win in the Basic Attack with the above two instances is a negligible variable $\epsilon(k)$， in which k is the security parameter.

2.6 Attack on the Linear Systems of NTT Coefficients

As the secret $\mathbf{s}$is an element inR_q， it can be flattened to a polynomial $\mathbf{s}(x) = \sum_{i=0}^{n-1} s_i x^i$， in which $n = 2^m$for $m \in \mathbb{Z}^+$. Let $w \in \mathbb{Z}_q$subjects to$w^{2n} \equiv 1 \bmod q$. Then according to the construction of NTT coefficients， $\mathbf{s}(x) = \sum_{i=0}^{n-1} (\mathrm{NTT}(\mathbf{s}(w^{2i+1}) \bmod q))$. Let $u = n/n'$， and $\mathbf{s}_u^{\alpha}(x) = \mathbf{s}(x) \bmod ((x^u - w^{\alpha})^u)$. By NTT（$\mathbf{s}(x)$）， we have the following equation about the i-th coefficient $s_{\{u,i\}}$ of $\mathbf{s}_u(x)$， $s_i + s_{i+u} w^u + s_{i+2u} w^{2u} + \cdots + s_{i+(n'-1)u} w^{(n'-1)u} = s_{u,i}$, $i \in \{0, 1, \cdots, u-1\}$. Then a linear system consisting of all the u coefficients is constructed with n' fragments of $\mathbf{s}$in each equation. For a leakage of possible partial key $S = S_1 \times S_2 \cdots \times S_u$， in which $S_i = \{s_i, s_{i+u}, \cdots, s_{i+(n'-1)u}\}$， we have the i-th linear equation，$[1, w^u, \cdots, w^{(n'-1)u}] \cdot S_i^T = s_{u,i}$.

Since we have the linear systems about secret key $\mathbf{s}$， we only need to find the most likely solution to the systems to recover $\mathbf{s}$. The up-to-date recent work in solving the linear systems is enumeration all the candidates in the $q^{n'}$space， and the target solution is the short vector with the shortest norm. Noting the component-wise equation $\hat{\mathbf{u}} = \hat{\mathbf{a}} \cdot \hat{\mathbf{s}} + \hat{\mathbf{e}}$， another similar enumeration can be applied on linear systems about coordinates of $\mathbf{e}$. Attacking on NTT coefficients

is applied and implemented in the subroutine in Section 5.2.

3 Related Works

Post-quantum key exchange mechanisms. We assume that key exchange mechanism is a special instantiation for key encapsulation mechanisms (KEM). The post-quantum KEM usually includes the authenticated protocols like [29], in which signatures or additional verifying structures are applied, and the direct KEM which is much brief and efficient, such as Ding [15], Peikert [26], and Alkim's NewHope [4, 6] that built on Ring-LWE assumption. Also, there are KEM based on standard LWE assumption, which makes the scheme more brief, such as Frodo protocol [9] and Kyber protocol [10]. Schemes in the first categray can easily satisfy strong security like IND-CCA and IND-CCA2 in quantum security model, despite there complex steps and heavy bandwidth costs. KEMs in the second and the third categray may only achieve passively secure, unless safe hash functions or FO transformation [18] are applied, such as Alkim's NewHope.

Attack on the lattice problem. Theoretically, security guarantees of lattice based cryptographic primitives including NewHope are the hardness of SVP, CVP and their variants. For efficiency reason, LWE and Ring-LWE problems are intriguing the attention of both designers and analyzers. Many works have been down in designing attack and analysis models against LWE problem and primitives based on it [7, 12, 24, 27]. A basic approach is to focus on attacking the SVP structure by basis enumeration or sieve under classic or quantum reduction, such as LLL [24] and BKZ [27] types of attacks. In basis reduction attacks, the purpose is to design an realizable and efficient SVP oracle by implementing a classic or a quantum algorithms. Then the oracle is applied in acquiring a ideal basis to fulfill the SVP problem in LWE instances.

BKZ and BKZ2.0 [12] in classic settings run in the exponential time complex of $2^{\mathcal{O}(n\log n)}$ with approximate factor complex of $O((6k^2)^{n/k})$. To the state-of-the-art, for k-dimension lattice, the best approximate factor is reduced on the level of $((1+\epsilon)\gamma_k)^{(n-k)/(k-1)}$, and is still accelerating. But according to the proof of Chen [12], the complex of enumerations and their variants is lower-bounded by blocksizes (or demension size $k \geqslant 250$).

Recent quantum sieve algorithms based on Locality Sensitive Hashing has reduced the complexity of reduction to $2^{0.292n}$ [8, 21], which brings a further advance scope in basis reduction. But the work in [22] shows that the best performance of quantum SVP algorithm may still have time complexity worse that $2^{0.2075n}$.

Semantic security. Semantic security analysis initiated by Cramer [13] is now a standard way of proving and evaluating the security strength of cryptographic primitives and protocols. The security strength or the costs of attacks are normally reduced to the costs of solving an open hard problems by mathematically implementing an attack between a challenger$\mathcal{C}$and an adversary$\mathcal{A}$. NewHope PKE scheme is semantically proved CPA secure under decisional Ring-LWE hard problem assumption in [11] with the following advantage for the adversary,

$$Adv^{IND-CPA}_{NewHope-PKE}(\mathcal{A}) \leqslant Adv^{dRLWE}_{n,q,\chi}(\mathcal{B}_1) + Adv^{dRLWE}_{n,q,\chi}(\mathcal{B}_2),$$

in which $\mathcal{B}_1$ and $\mathcal{B}_2$are quantum algorithms of solving decisional RLWE problems.

According to the work in [3, 4], the KEM schemes based on NewHope CPA PKE can be reduced to IND-CCA secure level both under classic random oracle model and quantum random oracle model. Then the key exchange protocol constructed both from NewHope SIMPLE [5] (based on NewHope CPA KEM) and NewHope USEIX [6] (based on NewHope CPA PKE) are at least CCA secure.

Leaky RLWE Assumptions. As partial key leaky attack is an important attack in security resilient evaluation, [14] introduced the computational and decisional assumptions of leaky RLWE assumptions, focusing on the NTT and NTT^{-1}coefficients

leakage during key exchange process. The partial key leakage attack indicates that under certain combinations of coefficient exposure， the secret keys can be recovered with noneligible probability， applying the techniques of constructing local CVP solvers and partial linear equation system of partial coefficients of secret keys or noise vectors.

4 Construction of Automatic Guess Oracle

The existing attack implementations mainly focus on partial key leakage or component exclusive analysis. Once an significant result achieved， a major obstacle in the previous attack is the accumulation of confidence in recovery of each partial key $s_i$to obtain a intact secret key. Assuming the noise vectors dependent with each other in each attack instances， we try to overcome this obstacle by categorizing the attacks with partial leakage and NTT components， constructing a wide guess oracle to output a tactic guess of e under each scenarios， and extracting the accumulative features potentially contained in each middle variables.

The main component of our guess oracle is a deep learning structure as a universal processor of inputs. For different attack instances， the deep learning structure is configured with different preprocessing components.

For the sake of overcoming the drawbacks of previous two attacks in confidence constrains and accumulative loss in RLWE instances， we developed two different subroutines in recovery of the noise vectors. As the NewHope based on PKE is much simpler than NewHope based on reconciliation， we use the variables of the later one in the following construction of our analysis framework， which can be easily simplified to be adopted in the former one.

4.1 An Automatic Analysis Framework

In this section， our main analysis framework to implementing the indistinguishability in details is introduced. The framework is constructed by a guess oracle including two alternative subroutines to adjust different attack models configured below. Each subroutine is a learning system controlled by instance configuration as well as time and memory boundaries to preprocess the query results. Subroutine 1 is partial leakage attack to boost the attack speed on NTT coefficients. Subroutine 2 is designed for the original key exchange protocol with noise decoder to enhance the significance of features in each queries. We will discuss the design of the two components in the next section. It takes queries under attack models as input， and outputs the guess result， see Fig.1.

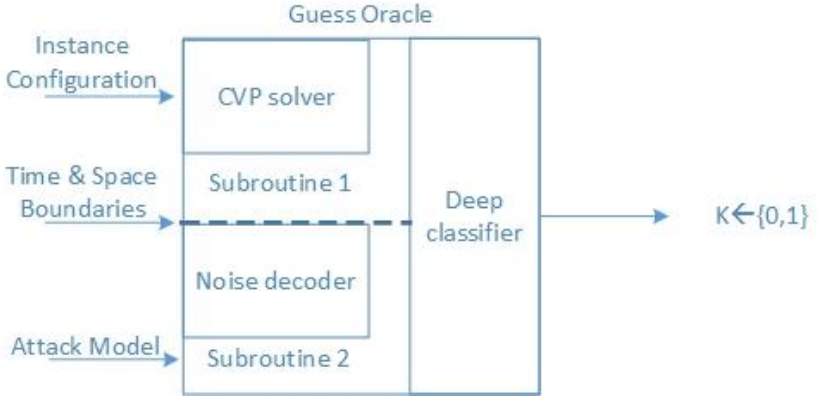


Fig 1 Automatic analysis framework

As in our attack framework， we mainly focus on the features exposed by NTT， NTT^{-1} and the two-pass messages， we have the following three heuristics to support the availability of the above framework.

Heuristic 1. In NTT， the co-efficients can be transformed into small CVP problems.

According to Lemma C. 1 [5] and the construc-

tion of linear system of coefficients in Section 3 in [14], the secret vector can be detected with significant probability with speicially chosed leakage points.

Heuristic 2. (Advantage over enumeration and guess) A solution to a linear system is learnable with accurate confidence of ϵ.

The method of solving the linear system in partial leakage attack is vector searching, such as the instances in Section 3 of [14]. But the success rate is quite low with approximate 0.57\% in one fourth leakage instances under a confidence of 98%. A learning system solves linear equations by constructing coefficient candidate from input samples of local CVP instances. So a solver based on learning system performs as good as searching secret keys in the worst cases. For a linear system that can be solved with an noneligible probability ϵ, there exists a learning system that can solve the linear system efficiently with success rate of at least ϵ.

Heuristic 3. (Existence of automatic analysis oracle to the linear system) A guess oracle based on learning system exists for identity distinguishability attacks in NewHope key exchange protocols.

By applying automatic learning approach to realize the heuristics, the attack framework acts like a detecting needle in revealing the weak points during the indistinguishability attack. Then following the framework, the details of each subroutine will be designed in the next section.

4.2 Subroutine 1: Neural Network based $\mathrm{CVP}_{D_{n'}}$ Solver

For the key leakage attack, a basic way to recover all the secret key s is by constructing linear systems from $u = a \cdot s + e$ as mentioned in Section 2. In this section, we construct a full connection neural network as the $\mathrm{CVP}_{D_{n'}}$ solver to improve the probability in searching solutions to the linear equations. The solver is basically a dynamic denoising model, by which n' coordinates of e' are generated under the constrains of CVP on $D_{n'}$. The model is expressed by the following processing method,

$$S_{\mathrm{CVP}}(u_i, a, [e_i]_{i \equiv \alpha \bmod 2n'}, v_i \in D_{n'}) = \mathrm{NN}(W, F(\cdot))$$

in which the loss function $F(\cdot)$ is,

$$F([e_i]_{i \equiv \alpha \bmod 2n'}) = \arg_w c_0(\min\|e_i^j - y_i\|^2) + c_1(\min\|\mathbf{e}_i^j - \omega\|^2), j \in \{1, \cdots, n'\}.$$

In the linear system L_i, the output e_i^* of $S_{\mathrm{CVP}}(\cdot)$ that supervised by the leakage e_i satisfies L_i, meanwhile, we gain $P\{e_i^* \text{ is a correct gues}\}$ is significantly higher than the direct enumeration in Section 3.1 and 3.2 of [14] according to Heuristic 3.

Then we can construct the guess oracle using the $\mathrm{CVP}_{D_{n'}}$ solver with the following three phases.

Initial Phase: For a RLWE instance $(a, u = a \cdot s + e)$ and leakage $[s]_{i \equiv \alpha \bmod 2n'}$, $S_{\mathrm{CVP}}(u_i, a, [e_i]_{i \equiv \alpha \bmod 2n'}, v_i \in D_{n'})$ instances are created based on a full connection neural network to hold the latent parameters in expressing the solution to L_i. We have n/n' solver instances in parallel.

Training Phase: Given a RLWE instance set S, for e_i^j of the linear equation L_i of the j-th RLWE instance, the solver S_{CVP}^i takes a random noise vector $v \in \Psi$ to optimize the latent connection in the neural network until the output of the solver satisfies L_i with the required confidence ϵ. For a chosen $\alpha \in 1, 2, \cdots, 2n'$, such as $\alpha = 1$, all the linear equation solver output the correct solution with the confidence of ϵ.

Generating Phase: Let $V = \{v_1, \cdots, v_{n/n'}\}$ be randomly generated noise vectors. The parallel solvers take V as input, and take $[u, e]_{i \equiv \alpha \bmod 2n}$ as the optimizing constrains of $F(\cdot)$. Finally, the solvers output all the e_i for the RLWE secret key. The final state of all solvers is capsulated as a guess oracle for the RLWE noise vector e.

As the input of the solver is random noise, it can applied as a CVP searcher for "the confident solutions" for each linear system. Loss function $F(\cdot)$ may be turbulent during training phase because of dispersion of target noise vectors. To partially avoid the turbulence, the training phase can be divided into two steps. In the first step, let $c_0 = 1$ and $c_1 = 1$. Then in the second step, candidates of solutions are feeded back to input end to train the solver again, while let

$c_0 = 0$ and $c_1 = 1$.

A adversary can launch a partial key leakage attack by applying the above guess oracle， and generate a more likely secret key compared with the random generated keys as the preprocessed results to win the indistinguishability attack. We will initiate the attack in Section 4.5.

4.3 Subroutine 2: A Wide Noise Decoder

Another category of attack configuration is the direct attack interaction without any knowledge of the secret key or the noise vectors. We use an automatic noise decoder to partially expose the coordinate information of secret key. A wide noise decoder is a deep structured classifier， such as learning network， which is designed for automatic guess on all the coordinates of secret key from attack queries both in scenarios with NTT and without NTT process. But it's also applicable for the original scheme of NewHope， in which the adversary takes the output of CVP solver as the input of the decoder.

The noise decoder is expressed by the following processing method，

$$S_{\mathrm{BD}}(\mathbf{u},\mathbf{r},\mathbf{a},\mathbf{b}) = \mathrm{NN}(W, F(\cdot))$$

in which the loss function $F(\cdot)$ is，

$$F(\mathrm{INS}(\cdot)) = \arg_w c_0(\min\|\mathbf{b} - \mathbf{a}\cdot\mathbf{s}\|^2) + c_1(\min\|\mathbf{u} - \mathbf{a}\cdot\mathbf{s}'\|^2) + c_2[\min(\mathrm{Rec}(\mathbf{b}\cdot\mathbf{s}',\mathbf{r}) - \mathrm{Rec}(\mathbf{a}\cdot\mathbf{s}\cdot\mathbf{s}',\mathbf{r}))],$$

If the input of the noise decoder is from instances without NTT， then $c_2 = 0$， and the decoder is a pure boundary linear decoder for $\mathbf{s} \leftarrow (\mathbf{a}, \mathbf{b})$ and $\mathbf{s}' \leftarrow \mathbf{a}, \mathbf{u}$， which is NP-hard to get correct solution， but is helpful to identify boundaries in distinguishing potentially correct secret from those selected from random instances. If the input is from instances with NTT， then $c_2 \neq 0$， and the potential s and s' are decoded by $Rec(\cdot)$ which is defined in [5]. The accuracy of the noise decoder can be allocated by a parameter γ defined as $\gamma = c_0 + c_1 + c_2$. We will discuss the value of γ in the implementations of different attacks.

Then we construct the noise decoder of guess oracle with an self-decoding network shaped learning system to capture the noise pattern by the following three phases.

Initial Phase： For a RLWE instance， let $(\mathbf{a}, \mathbf{b}, \mathbf{u}, \mathbf{r})$ and two selected identities $\{\mathrm{id}_a, \mathrm{id}_b\}$ as the input of the noise decoder based on two symmetric full connection neural networks to hold the latent parameters in expressing noise pattern of e and e'. We can process （a， b） and （u， r） in parallel simultaneously or independently， and the learning results may slightly different.

Training Phase： Taking a set $S = \{(\mathrm{INS}_i, s, s')\}$ of query results as input， the noise decoder extracts features of noise sampling from the two-pass message under the supervision of the loss function $F(\mathrm{INS}(\cdot))$ until the query limitation reached or the required confidence of decoding ϵ' is achieved.

Decoding Phase： Let $\mathrm{INS}^* = \{(\mathbf{a}^*, \mathbf{b}^*, \mathbf{u}^*, \mathbf{r}^*)\}$ be challenged instance. The trained decoder takes INS^* as input， and outputs an expected $\mathbf{a}^*\cdot\mathbf{s}^*$ and $\mathbf{a}^*\cdot\mathbf{s}'^*$.

As NTT transform is applied in the original NewHope key exchange protocol， the above construction of the noise decoder can be easily changed to comply with NTT variables by switching $(\mathbf{a}, \mathbf{b}, \mathbf{u}, \mathbf{r})$ to $(\hat{\mathbf{a}}, \hat{\mathbf{b}}, \hat{\mathbf{u}}, \hat{\mathbf{r}})$ in the initial phase and offering $S = \{(\mathrm{INS}_i, \hat{\mathbf{s}}, \hat{\mathbf{s}}')\}$ in training phase. Finally， the noise decoder yields the expected $\hat{\mathbf{a}}^*\cdot\hat{\mathbf{s}}^*$ and $\hat{\mathbf{a}}^*\cdot\hat{\mathbf{s}}'^*$ with a confidence ϵ'.

4.4 Deep Classifier

As the preprocess components of guess oracle， the CVP solver and the noise decoder offer an preliminary guess of $\mathbf{s}$ and $\mathbf{s}'$. In the query phase of the basic attack， the adversary constructs a simulation of challenge phase which is also a realization of basic attack taking the out layer attack result as its challenge message $\mathrm{INS}(\mathbf{s}^*, \mathbf{s}'^*)$ to train an applicably expressive structure in the guess oracle. Then it means that the adversary chooses a proper subroutine to train a guess oracle that can express the basic features of current system parameters. The deep classifier in the backend of the guess oracle takes the role as the above simulator accepting preprocessed candidates of $\mathbf{s}$ and $\mathbf{s}'$. In

the query phrase of basic attack， the simulator takes the following steps to get a guess.

Step 1.

For a set of query results $\{\mathrm{INS}_i^{id^*}\}$of selected id^*， the simulator pick a subroutine to preprocess $\{\mathrm{INS}_i^{id^*}\}$with a batch size of $|\{\mathrm{INS}_i^{id^*}\}|$. Then an $\mathrm{INS}(\mathbf{s}^*, \mathbf{s}'^*)$ is generated at the end of handling the batch of queries.

Step 2.

The simulator requests more batches of query results $\{\mathrm{INS}_i^{id^*}\}$ of reselected id^*， in which i is bounded by time or memory limitations.

Step 3.

The simulator learns on the set $\{\mathrm{INS}\mathbf{s}_j'\}$， in which j is the number of batches and is bounded by time or memory limitations.

Step 4.

The simulator initiates j simulating instances of id^{**}， $\mathrm{INS}(id^*)$， and generates $\mathbf{s}^{**}, \mathbf{s}'^{**}$. If the guess results are significant on distinguishing secret keys of id^* and id^{**}， the simulator terminates the query phase， or else go to Step 1.

In fact， the above steps describe a guess simulation for the instance randomly picked by the adversary. If the simulation returns a satisfactory result， the query phase ends. Note that $id^* = \{id_a^*, id_b^*\}$is independently selected in each query batch， the outputs of preprocess components are independently generated secret key candidates $\{\mathbf{s}^*, \mathbf{s}'^*\}$.

4.5 Attack instances

From all the above analysis on different conditions（NTT usage and partial key leakage）， we realized four possible ways of indistinguishability attack toward NewHope， that is attack （i） to （ix） acceding the Basic Attack， and instinctively， （i） is the hardest and （ix） is the easiest.

（i） *Attack on key exchange indistinguishability without NTT.*

For $id^* = (id_0, id_1)_i \in \{(id_a, id_b), (id_c, id_d)\} \subseteq S_{id}$， in which $i \in \{0, 1\}$， the attack is a realization of Basic Attack which contains multiple rounds of interactions between a challenger and an adversary. Details of the attack is given by the following steps：

Step 1.

The challenger initiates two instances of key exchange protocol in which one is a real protocol $\mathrm{INS}_{real}(n, \alpha, q, id \neq id^*)$and the other one is an ideal random generator $\mathrm{INS}_{sim}(n) \leftarrow \Psi_{16}^n$.

Step 2.

The adversary launches queries with $id = (id_0, id_1) \in S_{id}$. If $id = id^*$， the challenger returns $\{(\mathbf{a}, \mathbf{b}), (\mathbf{u}, \mathbf{r})\} \leftarrow \Psi_{16}^n$， else if $id \neq id^*$， the challenger returns $\{(\mathbf{a}, \mathbf{b}), (\mathbf{u}, \mathbf{r})\} \leftarrow \mathrm{INS}_{real}(n, \alpha, q, id \neq id^*)$， and sands the outputs back to the adversary. The adversary sets a accuracy limitation ϵ and takes $\{(\mathbf{a}, \mathbf{b}), (\mathbf{u}, \mathbf{r})\}$ as the input of the guess oracle with Subroutine 2， noted as G_2.

Step 3.

Repeat the previous step until achieving $\mathrm{Acc}(G_2) \geqslant \epsilon$.

Step 4.

The adversary randomly selects $id = (id_2, id_3) \in S_{id}$， and flips an unbiased coin to get a random bit $i = \{0, 1\}$. If i=1， he generates $(\mathbf{a}, \mathbf{b}), (\mathbf{u}, \mathbf{r}) \leftarrow \mathrm{IND}_{real}(n, \alpha, q, id)$， or else he generates $(\mathbf{a}, \mathbf{b}), (\mathbf{u}, \mathbf{r}) \leftarrow \mathrm{IND}_{sim}(n)$. Then if $P\{i = i^* \leftarrow (G_2((\mathbf{a}, \mathbf{b}), (\mathbf{u}, \mathbf{r})))\} \leqslant \frac{1+\epsilon}{2}$go to Step 2.

Step 5.

Challenge： The adversary randomly selects $\{(id_a, id_b), (id_c, id_d)\} \in S_{id}$and sends it to the challenger. The challenger flips a coin to get a random bit $i \in \{0, 1\}$， and then he takes $id^* = (id_0, id_1)_i \in \{(id_a, id_b)_0, (id_c, id_d)_1\}$ as the challenged identities. Finally， the challenger sends the output $\{(\mathbf{a}^*, \mathbf{b}^*), (\mathbf{u}^*, \mathbf{r}^*)\}$back to the adversary.

Step 6.

The adversary makes more additional queries with $id = (id_a, id_b)$and $id = (id\_c, id\_d)$. If$id = id^*$， the challenger returns $\{(\mathbf{a}, \mathbf{b}), (\mathbf{u}, \mathbf{r})\} \leftarrow \mathrm{INS}_{sim}(id)$， else it returns

$\{(\mathbf{a}, \mathbf{b}), (\mathbf{u}, \mathbf{r})\} \leftarrow \mathrm{INS}_{real}(id)$.

Step 7.

The adversary updates G_2 with Step 3 and Step 4.

Step 8.

Finally, the adversary outputs a guess$b \in \{0, 1\}$. If $b=i$, the adversary wins in the attack.

The essential part of the attack is the two phases of learning. The first phase (Step 2-Step 4) emphasizes on the accuracy of the two-pass messages and generalization of selected identities. The second phase (Step 7) focuses on the accuracy of guess oracle toward the two selected identities in Step 5 based on the knowledge acquired in the first phase. Without NTT component, there's no linear system to solve for the CVP solver in Subroutine 1, so that Subroutine 1 descends to a CVP solver for *s*and *s′*, which is a polynomially hard problem in $\mathbb{R}^n$. We only use Subroutine 2 in this attack instance.

(ii) *Attack on key exchange indistinguishability with NTT.*

For $id^* = (id_0, id_1)_i \in \{(id_a, id_b), (id_c, id_d)\} \subseteq S_{id}$, in which $i \in \{0, 1\}$, the attack is a multiple rounds of interactions between a challenger and an adversary acting the same as Attack (i) except that in the query phase, the guess oracle applies $(\hat{\mathbf{a}}, \hat{\mathbf{b}}), (\hat{\mathbf{u}}, \hat{\mathbf{r}})$instead of $(\mathbf{a}, \mathbf{b}), (\mathbf{u}, \mathbf{r})$in the query results.

(iii) *Attack on key exchange indistinguishability without NTT and partial leakage.*

For $id^* = (id_0, id_1)_i \in \{(id_a, id_b), (id_c, id_d)\} \subseteq S_{id}$, in which$i \in \{0, 1\}$, the attack is a multiple rounds of interactions between a challenger and an adversary acting the same as Attack (i) except that in the query phase, the guess oracle preprocesses the query results by Subroutine 1 and Subroutine 2 independently.

(iv) *Attack on key exchange indistinguishability with NTT and partial leakage.*

For $id^* = (id_0, id_1)_i \in \{(id_a, id_b), (id_c, id_d)\} \subseteq S_{id}$, in which $i \in \{0, 1\}$, the attack is a multiple rounds of interactions between a challenger and an adversary acting the same as Attack (i) except that in the query phase, the guess oracle applies $(\hat{\mathbf{a}}, \hat{\mathbf{b}}), (\hat{\mathbf{u}}, \hat{\mathbf{r}})$instead of $(\mathbf{a}, \mathbf{b}), (\mathbf{u}, \mathbf{r})$ in the query results, and preprocesses the query results by Subroutine 1 and Subroutine 2 independently.

(Using guess oracle) Given a RLWE instance set $S = \{(\hat{\mathbf{a}}, \hat{\mathbf{b}}, \hat{\mathbf{u}}, [\hat{s}]_{i \equiv \alpha \bmod 2n'})\}$, the adversary applies guess oracle to extract the features of the instance with the following steps.

1. Let $u = n/n'$, and divide S into u groups with $S = \{S_j = \{(\hat{a}, \hat{u}, [\hat{s}]_{i \equiv \alpha \bmod 2n'}\}\}$, in which $j \in \{1, \cdots, u\}$.

2. Taking S as the input, the adversary make guesses $s^*$with guess oracle.

3. For a confidence ϵ, the adversary evaluates the expected probability $P\{s^* = s\}$and go to step 2 until $P\{s^* = s\} \geqslant \epsilon$.

Table 1: The routine choice of attack

Attack type	α	Attack routine
Attack 1	0	Subroutine 2
Attack 2	$\alpha \in \mathbb{Z}^+$	Subroutine 2
Attack 3	$\alpha \in \mathbb{Z}^+$	Subroutine 1 and Subroutine 2
Attack 4	$\alpha \in \mathbb{Z}^+$	Subroutine 1 and Subroutine 2

5 Evaluation

5.1 Configuration

To fully evaluate the effectiveness and system costs, we designed a instantiation platform of our attack model. The platform contains a challenger terminal and a computationally powerful adversary backend. The challenger terminal accepts queries and generates the coordinate responses according to the four different attack instances in Section 4.5. The adver-

sary's backend collects all the responses for each key exchange process， and builds the guess oracles for four attack instances respectively.

The challenger terminal is a Lenovo laptop with Core i5-2430M CPU@2.4GHz and 4GB ram. The method of secret keys and the pass messages generation during key exchange are wrapped by the original CCA-KEM-512bits [3-4]. The adversary backend is a server running a 3.4GHz Core i5-7500 CPU and 8GB ram. The subroutines of the four attack instances are coded in python Jupyter notebook 3.7. Without losing of generality， we fixed all the basic parameters of the NewHope protocol， and only tune the NTT parameters and partial leakage. For the sake of fully training the learning system on each subroutine， we set loose constrains on timeout， query scale and accumulative guess rate. The parameters of challenger and adversary in the attack are configured as Table 2 and Table 3.

For objectiveness in each attack instance， we assume the adversary not only make queries from the challenger， but also can naturally acquire the passed messages on public channels to enhance the knowledge of the adversary. During the constructing of guess oracle， the adversary can arbitrarily make transformation on collected samples. One of the most simple transformations is global shifting or shrinking on features of samples. In each instance， we maintain a shrink rate to point the proportion of transformation. Obviously， these messages cannot change the alter the procedures of the original indistinguishability game defined in Section 2.

Table 2: Challenger configuration

Parameter type	Value	Attack model
Dimension	512	Attack (i-iv)
Modulus	12289	Attack (i-iv)
centered binomial width	8	Attack (i-iv)
Partial leakage position	$\alpha = \{1, 8, 16, \cdots x \mod 8\}$	Attack (iii) & (iv)
Hash function	SHAKE128	Attack (i-iv)
Root of unity in NTT	3	Attack (i) & (iii)

Table 3: Adversary backend configuration

Parameter type	Value	Attack model
Max query scale	10^9	Attack (ii&iv)
Phase I timeout	$8.64 * 10^4$s	Attack (ii&iv)
Phase II timeout	$8.64 * 10^4$s	Attack (ii&iv)
Success rate of effective guess	0.6	Attack (ii&iv)

Table 4: Subroutine 1 configuration(only the local CVP solver)

Parameter type	Value	Attack model
Structure	Full connection BP neural network	Attack (i-iv)
ϵ	0.99	Attack (i-iv)
α	1	Attack (i-iv)
batch size	512	Attack (i-iv)
(width,depth)	(64,10)	Attack (i-iv)

5.2 Results

Significance. Assuming the adversary controls the communication channels in each attack instance， we implemented the four types of attacks defined in Section 4. According to Definition 4， we listed all the conservative results yielded by the attack instances （i）-（ix） in Table 7， applying on both of the NewHope KEM series. From the results， it's obvious that NTT component has an important influence in both partial leakage instances （attack i and ii） and tactic instances （attack iii and iv）. Without NTT and partial leakage， the significance of guess

Table 5: Subroutine 2 configuration(noise decoder)

Parameter type	Value	Attack model
Structure	Auto-encoder	Attack (i-iv)
(c_0, c_1)	(1,1)	Attack (i-iv)
c_2	0	Attack (i) & (iii)
c_2	1	Attack (ii) & (iv)
input size	64	Attack (i-iv)
output size	1-16	Attack (i-iv)
feature width	32	Attack (i-iv)

Table 6: Deep classifier configuration

Parameter type	Value
Structure	Full connection BP neural network
input	$\{(0\text{-}255)\}^{64}$
output	{0,1}
batch size	512
(width,depth)	(64,100)

oracle is so negligible （attack iv） that we have to increase the optimal shrink rate to quantitatively evaluate it. We can safely draw the conclusion that the NTT component can increase the advantage of adversary， especially in PKE based KEM instances. In Rec. based KEM， NTT shows a weaker influence toward the advantage in the attack. Naturally， for the comparison of instances with and without partial leakage （attack ii and iv)， the adversary gains overwhelming advantage in the former attacks， which complies with the theoretical results. But the influence of NTT， NTT^{-1} and partial key leakage does not seem to accumulate the advantage for the guess oracle to compute advantage from the results of attack i，ii and iii.

Table 7: The overall significance of the guess oracle

Attack models	schemes	optimal shrink rate	significance
Attack (i)	PKE based KEM	0.19997	0.23333
Attack (ii)	PKE based KEM	0.19997	0.23416
Attack (iii)	PKE based KEM	0.24997	0.20667
Attack (iv)	PKE based KEM	0.54997	0.01258
Attack (i)	Rec. based KEM	0.19997	0.02416
Attack (ii)	Rec. based KEM	0.19997	0.21750
Attack (iii)	Rec. based KEM	0.24997	0.01500
Attack (iv)	Rec. based KEM	0.19997	0.00416

Cost of adversary. Table 8 shows the overall costs of adversary in constructing guess oracle， including space and time costs. Each sample in query phase consists of at least four major feature points which are encapsulations of polynomial coefficients. Restricted by computing power， the adversary is only allowed to make less than 10000 queries in each attack instance to avoid system crash. Compared with attack ii and iv， attack i and iii only cost about 1/3 time in period T1 （querying and training)， which shows NTT and NTT^{-1}play important role in enhancing system efficiency. In period T2 （ challenge and guess)， although the performance varies for different attack instance， time cost of all instances is less then 0.01 ms， which is undoubtedly acceptable. Instances based on reconciliation cost an average of extra 20\% time more than the PKE based KEM.

So far as we know， there no similar quantitative measurement mechanism for us to make a full comparison， such that we only demonstrated an average computing resource costs in our attack instances in Table 9. The peak memory is the maximum avail-

Table 8: The overall cost of adversary

Attack models	schemes	query	T1(seconds)	T2(seconds)
Attack (i)	PKE based KEM	10000	1317	0.00100
Attack (ii)	PKE based KEM	10000	4213	0.00083
Attack (iii)	PKE based KEM	10000	1615	0.00902
Attack (iv)	PKE based KEM	10000	4509	0.00041
Attack (i)	Rec. based KEM	10000	1401	0.00002
Attack (ii)	Rec. based KEM	10000	3889	0.00021
Attack (iii)	Rec. based KEM	10000	1570	0.00285
Attack (iv)	Rec. based KEM	10000	3716	0.00033

able ram space on the rim of system crash， and it shows that attack iv （original scheme without NTT） gains the most significant memory cost， while instances with partial leakage reaches the least memory cost （attack ii）. The storage in table 9 is volume of samples after they are pre-processed to fit coordinate learning models. As we only realized a prototype of our guess oracle on a platform with limited computing resources， the peak memory and storage costs are both in controllable levels.

Table 9: The maximum computing power cost of adversary

Attack models	schemes	peak memory(Gb)	storage(Gb)
Attack (i)	PKE based KEM	5.56022	0.27459
Attack (ii)	PKE based KEM	4.95308	0.27493
Attack (iii)	PKE based KEM	5.59003	0.28948
Attack (iv)	PKE based KEM	5.95913	0.28998
Attack (i)	Rec. based KEM	5.57011	0.27446
Attack (ii)	Rec. based KEM	5.65902	0.27470
Attack (iii)	Rec. based KEM	5.20087	0.28820
Attack (iv)	Rec. based KEM	5.89022	0.29001

6 Conclusions and Future Works

Quantized security analysis is an important aspect of cryptographic primitives. Post-quantum cryptographic designs of PKE， KEM and KTM schemes are usually only proved semantically secure under the assumptions of post-quantum hard problems. But it's hard to fully evaluate the influence of certain components in a complex cryptographic scheme because of the constrains of mathematical reduction in the semantic proof. So the construction of quantized security analysis system is quite necessary. In this paper， we explored the availability of quantized security analysis toward a post-quantum key exchange protocol （NewHope key exchange protocol with reconciliation） by applying an automatic learning methodology. Based on the routine of the challenge games in semantic proof， we carefully designed a serial of attack models， in which the adversary is granted the power of using a guess oracle to distinguish each instances based on automatic and statistical learning.

The evaluation results showed that different security strength under different circumstances （NTT and NTT^{-1}， partial key leakage， sampling methods） as we had expected. We can safely draw the conclusion that the NewHope key exchange protocol probabilistically do not satisfy selective identity CCA security in each circumstances.

There are still many unsolved problems in quantized security analysis， especially in cryptographic protocols， such as secure KEM and secure multi-party computing. These protocols are hard to compactly and universally describe in analysis systems. Artificial Intelligence is a powerful infrastructure in analysis systems， but how to mathematically design a rigid theory to achieve the optimal learning result in revealing the security level is also the leverage part in the exploration.

References:

[1] AgrawalShweta, BonehDan, and BoyenXavier. Efficient lattice (h)ibe in the standard model. In Annual International Conference on the Theory and Applications of Cryptographic Techniques，pages 553 – 572. Springer，2010.

[2] AlbrechtMartin R, PlayerRachel, and ScottSam. On the concrete hardness of learning with errors. Journal of Mathematical Cryptology，9(3)：169 – 203, 2015.

[3] AlkimErdem, AvanziRoberto, BosJoppe, DucasL´eo, Antonio de la Piedra, Po¨ppelmannThomas, SchwabePeter, and StebilaDouglas. Newhope-algorithm speci fi cations and supporting documentation. First Round NIST PQC Project Submission Document，2017.

[4] AlkimErdem, AvanziRoberto, BosJoppe, DucasL´eo, Antonio de la Piedra, Po¨ppelmannThomas, SchwabePeter, and StebilaDouglas. Newhope-algorithm speci fi cations and supporting documentation. Second Round NIST PQC Project Submission Document，2019.

[5] AlkimErdem, DucasL´eo, P¨oppelmannThomas, and SchwabePeter. Newhope without reconciliation. IACR Cryptology ePrint Archive，2016：1157，2016.

[6] AlkimErdem, DucasL´eo, Po¨ppelmannThomas, and SchwabePeter. Post-quantum key exchangea new hope. In 25th {USENIX} Security Symposium ({USENIX} Security 16)，pages 327 – 343，2016.

[7] AonoYoshinori, Yuntao Wang, HayashiTakuya, and TakagiTsuyoshi. Improved progressive bkz algorithms and their precise cost estimation by sharp simulator. In Annual International Conference on the Theory and Applications of Cryptographic Techniques，pages 789 – 819. Springer，2016.

[8] BeckerAnja, DucasL´eo, GamaNicolas, and LaarhovenThijs. New directions in nearest neighbor searching with applications to lattice sieving. In Proceedings of the twentyseventh annual ACM-SIAM symposium on Discrete algorithms，pages 10 – 24. Society for Industrial and Applied Mathematics，2016.

[9] BosJ, CostelloC, DucasL, and et al. Frodo: Take o ff the ring! practical, quantumsecure key exchange from lwe. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ，pages 1006 – 1018，2016.

[10] BosJoppe, DucasL´eo, KiltzEike, LepointTancr`ede, LyubashevskyVadim, SchanckJohn M, SchwabePeter, SeilerGregor, and Stehl´eDamien. Crystals-kyber: a ccasecure module-lattice-based kem. In 2018 IEEE European Symposium on Security and Privacy (EuroS&P)，pages 353 – 367. IEEE，2018.

[11] BosJoppe W, CostelloCraig, NaehrigMichael, and StebilaDouglas. Post-quantum key exchange for the tls protocol from the ring learning with errors problem. In 2015 IEEE Symposium on Security and Privacy，pages 553 – 570. IEEE，2015.

[12] Yuanmi Chen and Phong Q Nguyen. Bkz 2. 0：Better lattice security estimates. In International Conference on the Theory and Application of Cryptology and Information Security，pages 1 – 20. Springer，2011.

[13] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing，33(1)：167 – 226, 2003.

[14] Dachman-SoledDana, Huijing Gong, KulkarniMukul, and ShahverdiAria. Partial key exposure in ring-lwe-based cryptosystems: Attacks and resilience. IACR Cryptology ePrint Archive，2018：1068，2018.

[15] Jintai Ding. New cryptographic constructions using generalized learning with errors problem. IACR Cryptology ePrint Archive，2012：387，2012.

[16] Jintai Ding, Xiang Xie, and Xiaodong Lin. A simple provably secure key exchange scheme based on the learning with errors problem. IACR Cryptology ePrint Archive，2012：688，2012.

[17] FluhrerScott R. Cryptanalysis of ring-lwe based key exchange with key share reuse. IACR Cryptology ePrint Archive，2016：85，2016.

[18] FujisakiE. and OkamotoT. . Secure integration of asymmetric and symmetric encryption schemes. Advances in Cryptology，CRYPTO 1999，1666(6)：537 – 554, 1999.

[19] GentryCraig, PeikertChris, and VaikuntanathanVinod. Trapdoors for hard lattices and new cryptographic constructions. In Proceedings of the fortieth annual ACM symposium on Theory of computing，pages 197 – 206. ACM，2008.

[20] JiangH, ZhangZ, ChenL, and et al. Ind-cca-secure key encapsulation mechanism in the quantum random oracle model, revisited. Annual International Cryptology Conference，pages 96 – 125，2018.

[21] LaarhovenThijs. Sieving for shortest vectors in lattices using angular locality-sensitive hashing. In Annual Cryptology Conference，pages 3 – 22. Springer，2015.

[22] LaarhovenThijs, MoscaMichele, and Joop Van De Pol. Finding shortest lattice vectors faster using quantum search. Designs，Codes and Cryptography，77(2-3)：375 – 400, 2015.

[23] LangleyAdam. Cecpq1 results. Imperial Violet，2016.

[24] Arjen Klaas Lenstra, Hendrik Willem Lenstra, and Lova´szL´aszl´o. Factoring polynomials with rational coe ffi cients. Mathematische Annalen，261(4)：515 – 534, 1982.

[25] LyubashevskyVadim, PeikertChris, and RegevOded. On ideal lattices and learning with errors over rings. Journal of the ACM (JACM)，60 (6)：43，2013.

[26] PeikertChris. Lattice cryptography for the internet. In international workshop on post-quantum cryptography，pages 197 – 219. Springer，2014.

[27] Claus-Peter Schnorr and Martin Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. Mathematical programming，66(1-3)：181 – 199, 1994.

[28] XavierB. and QinyiL. . Direct cca-secure kem and deterministic pke from plain lwe. Post-Quantum Cryptography，PQCrypto 2019，LNCS 10505(6)：116 – 130, 2019.

[29] ZhangJ, ZhangZ, DingJ, and et al. Authenticated key exchange from ideal lattices. Annual International Conference on the Theory and Applications of Cryptographic Techniques，pages 719 – 751，2015.

SDI芯片夹具板频域S参数的仿真与验证

王锐[1]，张波[3]，张霞[2]，汪欣[3]

[1]天津芯海创科技有限公司，天津，100000；

[2]国家数字交换系统工程技术研究中心，河南郑州，450000；

[3]天津市滨海新区信息技术创新中心，天津，100000

摘　要：本文介绍了信号完整性设计的概况，信号完整性设计需贯串产品研发整个过程。介绍了S参数定义，基于信号完整性频域S参数分析，使用仿真软件提取SDI（Software Defined Interconnection）芯片夹具板的S参数，再进行高速互连链路的S参数级联。同时通过矢量网络分析仪（VNA）测试夹具板卡高速链路实际的S参数。两者数据对比后，进行仿真参数设置的优化，提炼出一套信号完整性频域S参数的仿真与测试闭环结合的设计方法，进一步提高仿真结果准确性，为后续高速板卡仿真及测试提供参考。

Simulation and verification of S-parameters in frequency domain of SDI chip fixture board

Wang Rui, Lv Ping, Wang Xin, Zhang Bo

Abstract: This article introduces the general situation of signal integrity design, which needs to run through the entire process of product development. The definition of S-parameters is introduced. Based on the frequency-domain S-parameter analysis of signal integrity, the simulation software is used to extract the S-parameters of the SDI(Software Defined Interconnection) chip fixture board, and then the S-parameters of the high-speed interconnect link are cascaded. At the same time, the actual S parameters of the high-speed link of the fixture board are tested by the vector network analyzer (VNA). After comparing the two data, the simulation parameter settings are optimized, and a set of design methods combining simulation and test closed-loop signal integrity frequency domain S parameters are refined, which further improves the accuracy of simulation results and provides reference for subsequent high-speed board simulation and testing.

一　信号完整性设计的概况

随着大数据的快速兴起，数据通信对总线带宽的要求越来越高。例如RapidIO 4.0的数据速率达到25.78Gbps，FC-AE-ASM 6.0的数据速率达到32Gbps，100GBASE-KR4的单lane数据速率达到25.78Gbps，PCIE 4.0的数据速率达到16Gbps，HDMI的数据速率达到6Gbps，USB3.1的数据速率达到10Gbps等。诸如谐振、反射、串扰、回波损耗、阻抗突变、EMI（电磁兼容）等射频微波邻域才会遇到的问题，现在成为高速数字系统设计领域要解决的关键问题。这就要求研发工程师要具备数字通信领域和射频微波领域的知识体系，还要掌握时域、频域、逻辑域的仿真测量分析技术。

信号完整性设计，包括仿真及测试。它应该贯串产品开发的整个阶段，改变传统的迭代投板设计理念，达到一次性设计成功和降低成本的目的。在产品设计的初期，从已有的模型数据库中调用模型进行预仿真分析，得到原理图设计规则以指导元件选型和Layout布局、布线设计。在Layout布线完成后，再进行后仿真以验证布局布线的设计是否满足约束规则。PCB板制作完成后，

开始互连的无源测试，验证板级互连的性能指标。焊接装配好元器件后，开始进行原型机的信号完整性测试验证，同时与前一阶段的仿真分析形成闭环，用于指导下一代产品的设计和开发。

图1是在产品开发的不同阶段，所需要的仪器和仿真工具图。信号完整性设计分析平台主要包括以下几个部分：信号完整性仿真分析、高速互连测试分析、高速信号测试分析、系统级激励响应测试分析、系统级总线测试分析等。

图1　信号完整性设计的不同阶段使用仪器和仿真工具组成

信号完整性频域仿真主要是提取高速互连系统的S参数，分析其插入损耗、回波损耗及串扰等指标是否符合协议规范。时域仿真主要是用IBIS、S参数、SPICE等模型搭建互连系统链路，分析接收端的眼图，加入协议中眼图模板，看是否符合规范。同时将频域S参数转换为TDR（Time Domain Reflectometry）阻抗，进一步分析链路中每一点的阻抗变化情况，是否符合协议阻抗要求。

电源完整性之前是从属于信号完整性知识体系，随着先进的FPGA、CPU、DSP技术发展，芯片的电压不断降低，但电流却越来越高，有的达到100A级别，这对PDN（Power Delivery Network）优化设计带来极大的挑战，因此目前将电源完整性独立成一个知识体系，通过仿真与测试分析相结合，保证芯片pad间的供电电压恒定，使地弹最小化，使电磁干扰问题最小化。

本文基于信号完整性频域S参数分析，使用仿真软件提取PCB段的S参数，再进行高速互连链路的S参数级联。同时通过矢量网络分析仪（VNA）测试高速链路实际的S参数。两者对比后，创造性提出修正仿真参数设置的方法，达到仿真与测试相结合的闭环设计。

二　S参数介绍

在信号完整性领域，S参数又称为行为模型，当信号作为激励作用于互连时，互连的行为会产生一个响应信号。在激励-响应波形中，隐含着的就是互连的行为模型。S参数可以描述除一些铁氧体以外的每一种互连电气行为，包括：电阻、电容、电路板走线、电路板平面、背板、连接器、封装、插座、电缆等。

当一个波形输入到互连系统时，它可以从互连系统散射回源端，这个波称为反射波。通过互连系统散射出去的波称为传输波。在频域中，用于测量正弦波反射响应和传输响应的仪器是矢量网络分析仪（VNA）。

每个S参数都是从被测互连系统某个特定端口散射出的正弦波与入射到被测互连系统某个端口

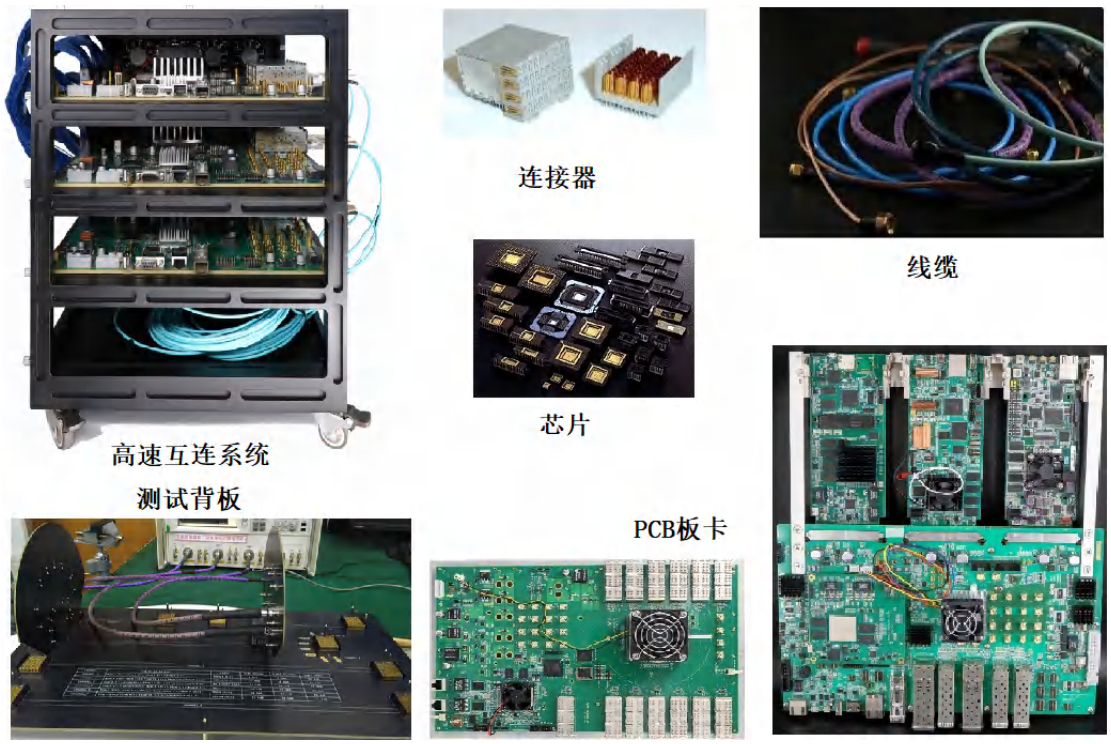


图2　S参数应用

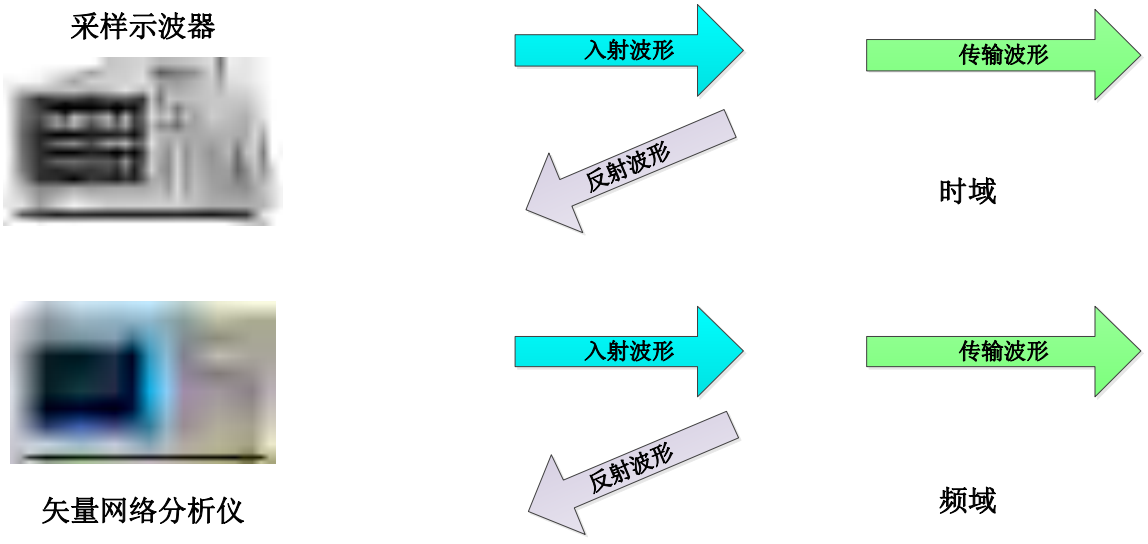


图3　S参数定义

的正弦波的比值。公式如下，S的单位为dB。

$$S = 20 X \log\left(\frac{\text{幅度 (输出正弦波)}}{\text{幅度 (输入正弦波)}}\right)$$

为了区别每个S参数所涉及的端口组合，使用两个下标值。第一个下标值是输出端口，第二个下标值是输入端口。

$$S_{kj} = \frac{\text{输出端口}k\text{的正弦波}}{\text{输入端口}j\text{的正弦波}}$$

例如S11称为回波损耗，代表从端口1进入并从端口1输出信号。S21称为插入损耗，代表从端口1进入并从端口2输出信号。S31称为近端串扰，S41称为远端串扰。

一般，如果互连系统不是物理对称的，S11和

S22是不相等的。对于所有线性无源元件而言，总有S12=S21。独立的S参数个数可由以下公式求出：

$$N_{unique} = \frac{n(n+1)}{2}$$

N_{unique}表示独立S参数元素的个数，n表示端口数。

高速串行链路经常处理的差分线，是用一个四端口来描述，其S参数矩阵如下。字母D和C分别代表差分信号和共模信号。S_{CD21}表示从端口2输出的共模信号与从端口1输入的差分信号比值。

由于所有的仪器都只能测量单端S参数，为了显示差分S参数，需要一些复杂的矩阵数学计算，参考公式如下。

$$S_{DD11} = 0.5\,X\,(S_{11} + S_{33} - 2\,X\,S_{31})$$

$$S_{DD21} = 0.5\,X\,(S_{21} + S_{43} - S_{41} - S_{23})$$

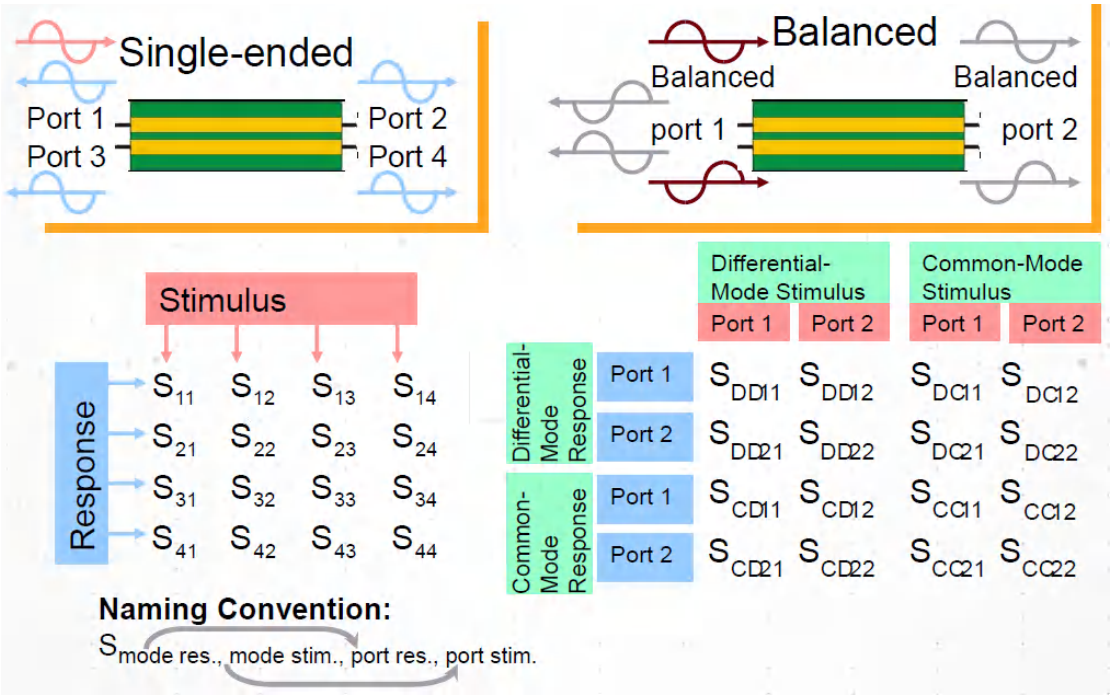


图4差分线的S参数矩阵

三　S参数测试与仿真

3.1　S参数测试

仿真和测试的板卡是天津市滨海新区创新中心的SDI交换芯片夹具板，SDI交换芯片具有32个高速lane，支持RapidIO 3.1、FC-AE-ASM 4.0、10GBase-KR、1000Base-X四种协议交换交换，并支持四种协议之间的混合协议转换和数据交换。单lane最高速率是10.312Gbps，交换能力为320Gbps。

芯片的32个高速lane全部由SMA同轴引起，用于对接高带宽示波器和误码仪，测试芯片Serdes的TX和RX的电参数是否满足协议规范。为了方便提取S参数，用于系统级联去嵌，在板卡右侧模拟实际高速走线，从第6层和第11层引出同轴。

S参数测试前先进行连接线缆的校准，测试使用Keysight的矢量网络分析仪E5071C和2端口电子校准件，校准全4端口时需要依次进行1-2、1-3、1-4、2-3、2-4、3-4端口校准才能完成。

校准和测试中都要保证线缆的自然延伸，否则会影响测试的精确性。测试的是第6层走线，S参数结果如下，并导出SMA同轴和PCB端全链路的S参数S4P文件。

3.2　S参数仿真

对板卡PCB走线段进行仿真时，将需要提取的差分线从整块PCB中剪切下来，可大大缩短仿真时间。根据Layout信息设置叠层参数、差分对、产生端口、仿真参数等。

仿真结果输出后，查看结果有无异常，导出PCB走线段仿真的S参数S4P文件

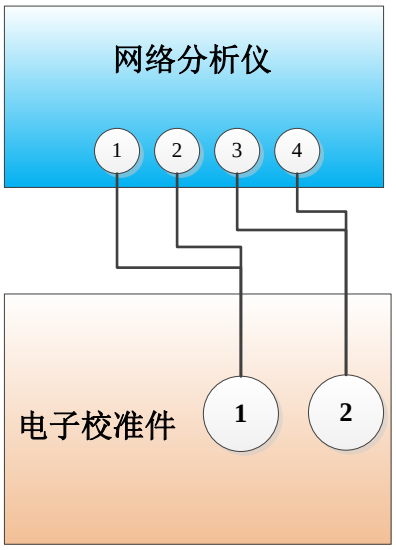


图5　测试线缆校准

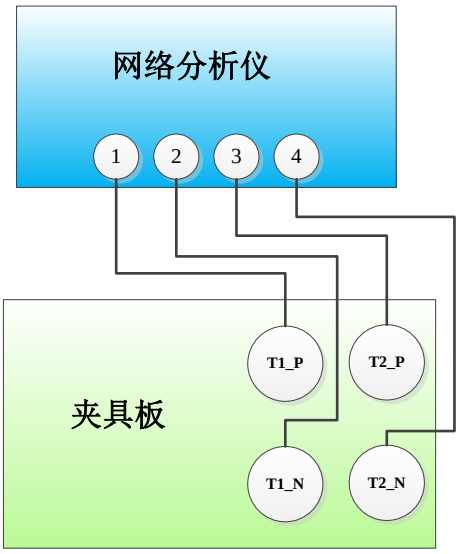


图6　夹具板S参数测试

四　S参数仿真与实际测试对比

对比仿真与实际测试的S参数，分别搭建仿真链路模型和实际测试链路模型。仿真链路模型中，仿真软件只能对PCB走线段进行仿真，全链路分析时要把信号链路所有部分的S参数进行级联分析，加入SMA同轴的S参数。实际测试时，矢量网络分析仪能测得全链路所有部分的S参数，其模型搭建较为简单。搭建模型如下图10。

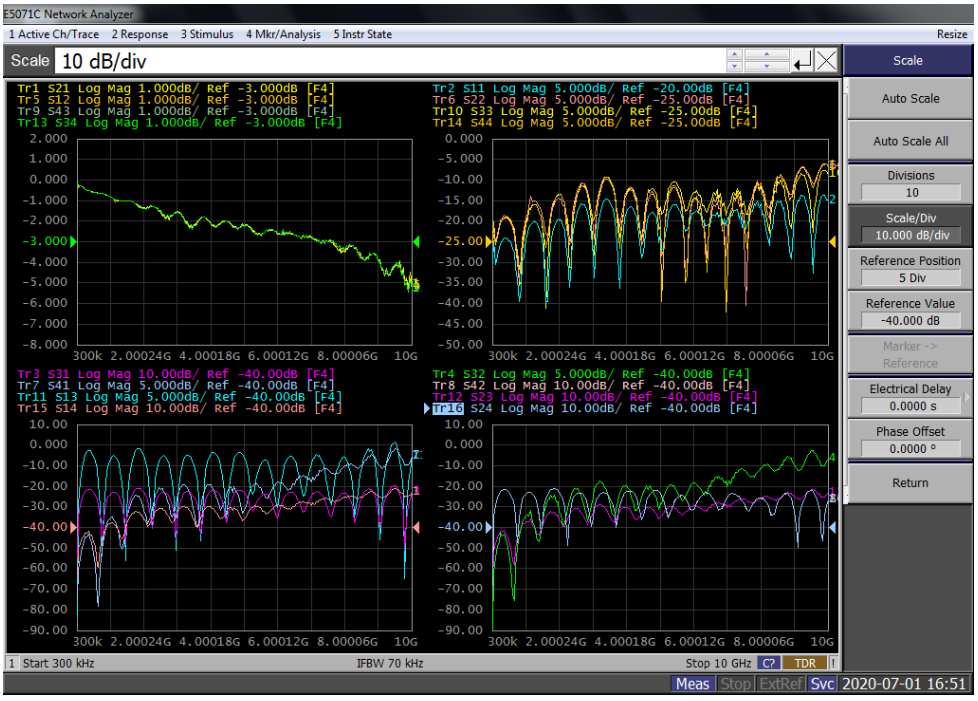


图7　板卡S参数测试结果

对比两者全链路的插入损耗和回波损耗，结果如下图11和12。

对比发现仿真结果与实际测试的曲线有较大差距，回波损耗存在明显的“频点偏移”现象。说明仿真设置不合理，需要修正仿真参数。对信号链路上每一节点器件模型分析，SMA同轴端口通常设置是以同轴GND通孔作为回流，导致回流路径增加，使S参数恶化。这里创造性提出缩短回

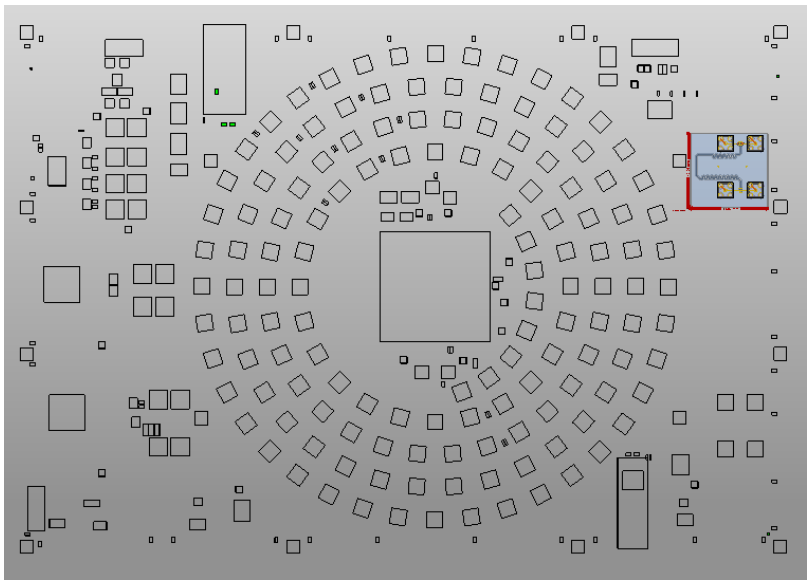


图8　仿真线路截取

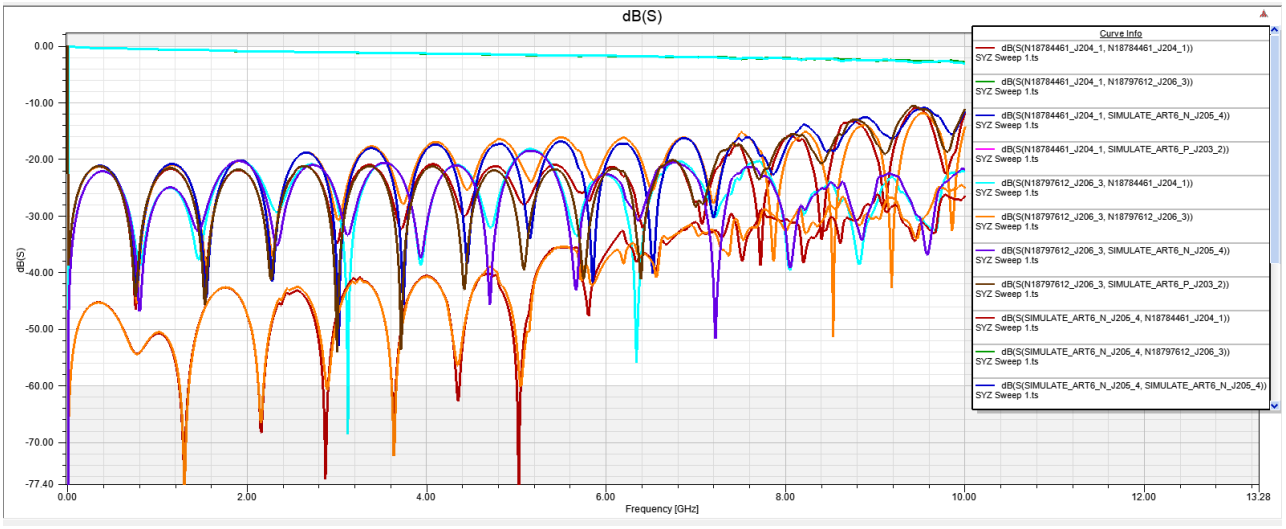


图9　仿真PCB段的S参数结果

流路径的方法，以同轴信号pad下方第5层铜皮作为回流参考。

同时分析信号过孔上是从Top层到第6层，而PCB层数是16层，过孔处有较长的短桩线，导致S参数部分频点产生谐振。利用仿真软件的优异性，修改过孔从Top层到第8层，模拟实际制作PCB时的背钻工艺。如下图13箭头处是仿真设置不合理的问题点。

进行PCB段的S参数仿真，导入仿真链路，对比实际测试、仿真修改前后的结果下。

在进行仿真设置优化后，插入损耗和回波损耗仿真曲线较好的拟合实际测试曲线，在实际测试时，测试环境周围的电磁场等也会对信号有干扰，所以实际测试的曲线与仿真结果有细微差异。通过实际测试数据反馈，创造性进行仿真优化，到达信号完整性闭环目的，验证了修正方法的有效性。

五　结束语

现代高速数字系统中面临的信号完整性问题越来越严峻，需要研发人员投入精力对SI/PI合理设计和测试验证。本文概述了信号完整性设计的概况，介绍贯串产品研发整个过程对信号完整性设计的要求。利用仿真软件提取PCB段走线S参数，用于系统级链路仿真。与矢量网络分析仪实际测试数据对比，分析两者误差原因。根据信号

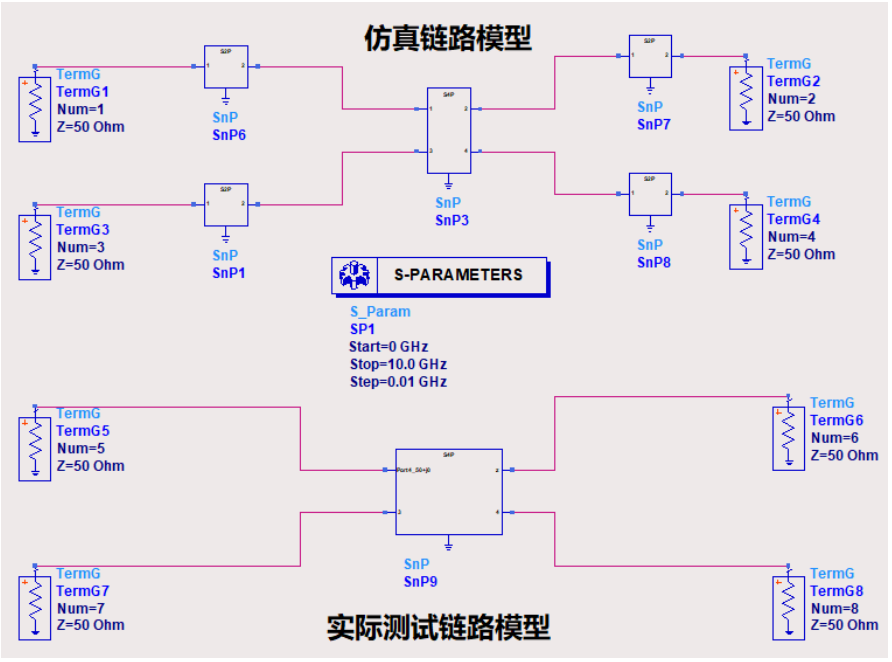


图10　搭建互连链路模型

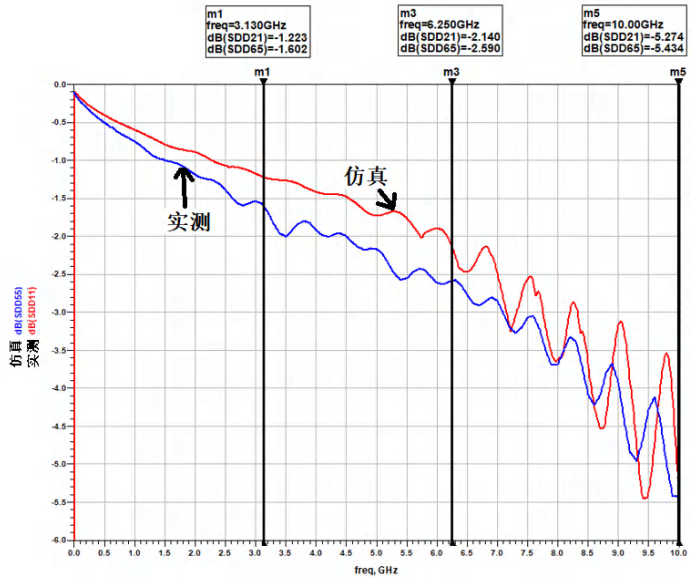


图11　仿真与实测插入损耗对比

完整性频域S参数的仿真和测试闭环结合的思想，创造性提出优化仿真设置，并进行有效验证。使仿真数据可以真实、准确等反映出芯片夹具板信号完整性是否满足设计，具有预判信号链路质量的优越性，为后续产品一次性设计成功和信号完整性仿真提供参考。

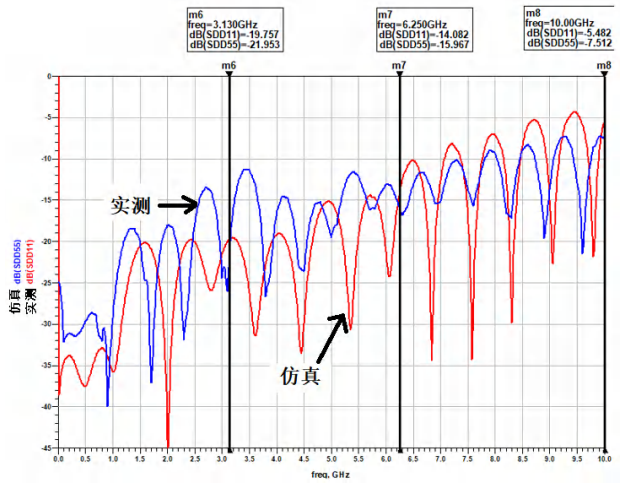


图12　仿真与实测回波损耗对比

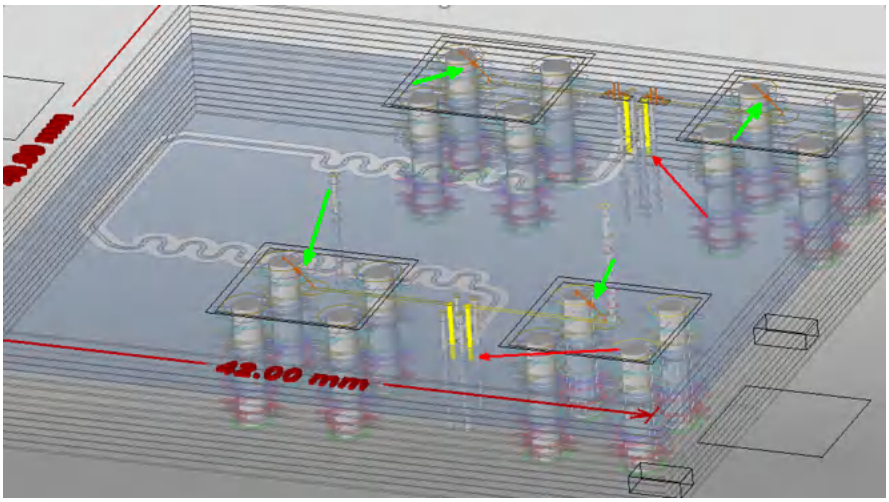


图13　仿真设置修改

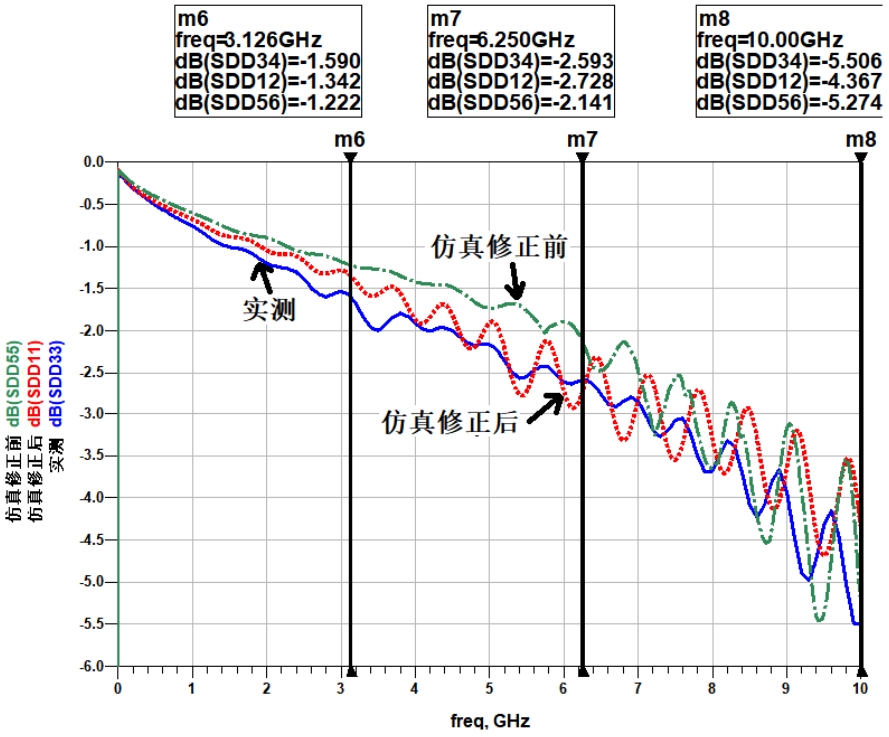


图14　仿真与实测插入损耗对比

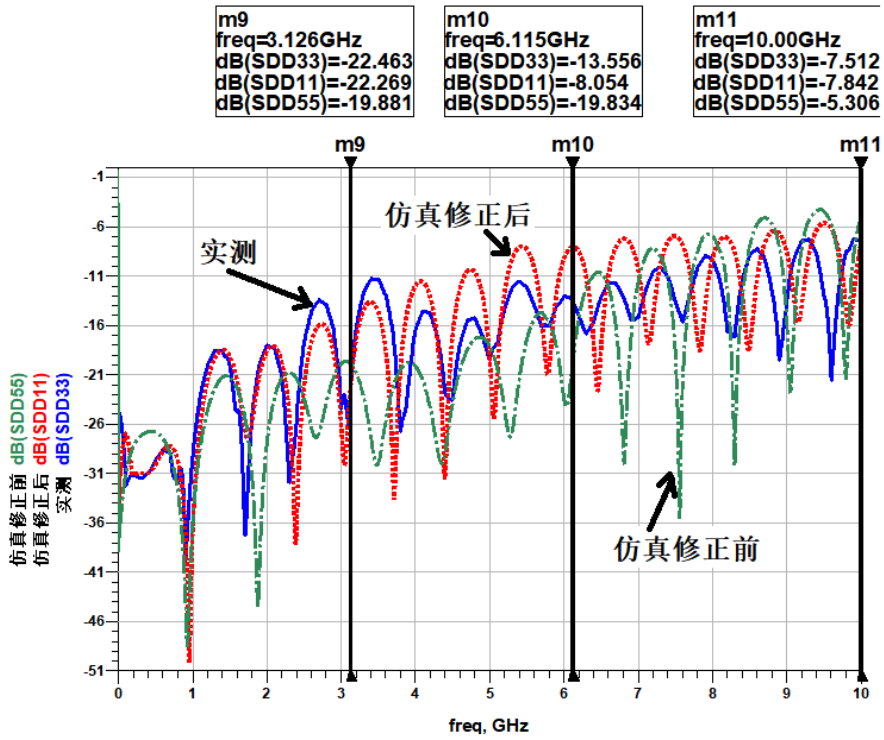


图 15　仿真与实测回波损耗对比

参考文献:

[1] 吕平,刘勤让,邬江兴,等,新一代软件定义体系结构[J]. 中国科学:信息科学,2018(3)

[2] Eric Bogatin,信号完整性与电源完整性分析(第3版)[M]. 电子工业出版社，2019

[3] Keysight,是德科技高速数字信号光电测试平台

[4] 张涛,ADS 高速电路信号完整性应用实例[M]. 电子工业出版社，2015

[5] Keysight，E5071C ENA Network Analyzers User’s Guide

[6] 房丽丽,章传芳,ANSYS 信号完整性和电源完整性分析与仿真实例[M]. 中国水利水电出版社，2018

[7] Stephen H. Hall,Howard L. Heck,高级信号完整性技术[M]. 电子工业出版社，2015

[8] Joel Dunsmore,微波器件测量手册 [M]. 电子工业出版社，2014

[9] 朱辉、冯云,实用射频测试和测量[M]. 电子工业出版社，2016

[作者简介]

王锐（1989-），男，硕士，高级工程师，主要研究方向：高速板卡设计、SI/PI 测试方法研究、时钟抖动测试方法研究。

张波（1986-），男，硕士，高级工程师，主要研究方向：大型复杂系统的硬件设计开发，高速板卡设计。

张霞（1980-），女，本科，助理研究员，主要研究方向：高速板卡设计，集成电路测试方法研究。

汪欣（1986-），男，硕士，高级工程师，主要研究方向：高速交换电路设计、大规模集成电路设计。

使用最佳拟态组件集解决假阳性问题

——邵昱文 1,张铮 1,王晓梅 1,潘传幸 1,邬江兴 2 1(信息工程大学,郑州 450001)

摘　要：针对目前拟态化改造技术中拟态界过大所带来的表决假阳导致的高误报问题，提出使用最佳拟态组 件集方法来选择被改造系统中需要异构冗余的组件，合理缩小拟态界，在不影响安全性的前提下有效降低 拟态防御中的“假阳率”，并提出两种不同的算法来有效选择最佳拟态组件。最后，通过结合实例的方式 来具体论述本文提出的选择算法，并使用对比实验证明该算法的有效性。

关键词：拟态界、最佳拟态组件集、假阳率、高误报

Using the Best Pseudo Positive Component Set to Solve the False Positive Problem

SHAO Yuwen[1], ZHANG Zheng[1], WANG Xiaomei[1], PAN Chuanxing[1], WU Jiangxing

1.Information Engineering University, Zhengzhou 450001, China

Key words: mimic bracket; best pseudo component set; false positive rate; high false positive rate

2（国家数字交换系统工程技术研究中心，郑州 450002）

2（China National Digital Switching System Engineering and Technological R&D Center， Zhengzhou 450002， China） **Absrtact：** In order to solve the problem of high false positives caused by false positives in the current mimicry transformation technology， this paper proposes the method of using the best pseudo component set to select the components that need heterogeneous redundancy， reasonably reduce the pseudo bound， and effectively reduce the "false positive rate" in the mimicry defense without affecting the security， and proposes two different algorithms to effectively select the most effective Good mimicry component. Finally， the selection algorithm proposed in this paper is discussed in detail by combining with examples， and the effectiveness of the algorithm is proved by comparative experiments.

1　引言

网络安全问题一直是盘踞在“互联网”大厦屋顶上的一朵乌云。拟态防御技术[1-2]作为新兴的防御技术， 能有效提高系统的安全性，随着拟态理论的不断完善与发展，拟态防御技术也更加成熟，但传统的拟态化改造[3]（将普通系统改造为具有动态异构冗余特性的系统）往往会造成过大的拟态界[4]。拟态界是指拟态括号（DHR 构造中的输入代理和多模表决器，左右括号在逻辑上或空间上一般是独立的， 且功能上联动）限定的保护范围，而过大的拟态界会使得改造者对一些非必要的组件进行异构冗余处理，这会造成表决假阳，

从而导致高误报，表决假阳问题会严重影响系统的可用性。为解决表决假阳问题，改造者往往采取“补” 的方式，先对整个系统进行异构冗余处理，然后再以功能测试中所发现的误报为依据（这些误报往往会导致系统无法向用户提供正

基金项目：国家重点研发计划资助项目（Grant No. 2018YF0804003，Grant No. 2017YFB0803204） 通信作者：张铮（ponyzhang@163.com）

常的服务），对系统进行“二次开发”，由于系统中包含多个异构冗余的执行体，因此改造成本是改造单路执行体成本的数倍，这样不仅耗费大量资源，也不能从根本上解决该问题。

本文从安全性量化的角度提出最佳拟态组件集理论，拟态化系统的改造人员可以将其作为选择需要拟 态化改造的组件标准，剔除不必要的组件，只选择必要的组件进行异构冗余，在不影响安全性的前提下有 效缩小拟态界，从根本上解决在对系统进行拟态化改造时产生的假阳性问题，也可降低拟态化改造的改造 成本。本文的第二部分对拟态化改造导致的假阳问题进行总体论述，第三部分提出最佳拟态组件集的概念 与最佳拟态组件集的构造算法，第四部分通过实例的方式来具体论述最佳拟态组件集的构造算法，并通过 实验来证明该构造算法的有效性。

2 拟态系统的假阳性问题

2.1 假阳性问题概述

假阳性原指医学中因为种种原因把不具备阳性症状的人检测出阳性的结果。在拟态防御系统中，表决假阳问题是指，表决器将执行体的正常行为裁决误报为攻击行为，对拟态系统而言，一次由表决假阳导致的误报可能会导致系统中断服务，因此，高误报会严重影响拟态防御系统的正常功能，降低系统的可用性。

2.2 产生原因

拟态界中包含具有随机性的组件：被拟态化改造的系统本身会提供一些存在随机性的服务，此类服务 的响应包含不可预测性的内容。如验证码，随机数，时间戳等。而提供这些服务的组件都被包含在拟态界 中。由于被拟态化改造的系统具有异构冗余性，因此当用户对该服务发起请求时，多个组件就会产生响应 不一致的情况，从而造成表决器误判。

拟态界中包含独立的组件：拟态界中涉及到多个执行体，这些执行体本身包含独立的组件，并且这些组件的某些特征在多个执行体之间是不一致的。

2.3 假阳性问题的危害

假阳性问题会将用户的正常请求误判为黑客的恶意请求，这会影响到用户的正常访问，也会有更大的 可能性暴露拟态架构，并占用大量的改造资源。

2.4 具体的假阳性问题

2.4.1 Web 服务器的 cookie

Cookie 是一小段文本信息，是网站为了识别用户的身份信息，进行会话跟踪而存储在用户本地终端上的数据。Cookie 由服务端生成并发送给客户端，之后的每次请求浏览器都会携带上 cookie 发送给客户端 [5]。

为了缓解 cookie 带来的安全问题，服务端往往会对 cookie 设置一个适当的有效期，另外对 cookie 的

value 进行处理，比如首先将 value 设置为用户名+ip 地址+有效时间+随机数，然后对该值进行加密。当服

务器进行了拟态化改造以后，一旦用户的请求涉及到了 cookie 服务，由于每台服务器生成的随机数不同， 加密算法不同，因此异构冗余的服务器会返回三个带有不同的 cookie 值的响应，这时表决器会检测到不一致而触发报警，从而造成了 web 服务器上的假阳性问题。

2.4.2 Web 服务器上的验证码功能

验证码功能是服务端提供的一串机器难以识别的字符串或物体，用户通过肉眼识别后进行正确的输入， 可以防止黑客的暴力破解与非法用户的恶意爬虫 [6]。但是其生成过程具有随机性和不可预测性，当服务器 进行拟态化改造之后，异构冗余的服务器会产生不同的验证码，从而造成用户无法正常访问。

2.4.3 程序的获取 pid 功能

基于软件多样化思想的多变体 [7] 可以为程序提供异构冗余执行环境。但是也会不可避免的遇到假阳性问题。比如很多程序都会涉及到 pid 的获取，如一些涉及到进程调度或是涉及网络通信的程序，这些程序会使用 getpid 功能来获得进程号，由于多变体提供了异构冗余环境，因此会将一个进程转化为多个异构冗余的进程同时执行，这些进程的进程号显然是不同的，那么当程序使用 getpid 功能时，异构冗余的进程会返回不同的进程号，这时表决器会检测到不一致而触发报警。

3 使用最佳拟态组件集解决假阳性问题

降低防御机制误报率的方式有基于对流量的分析，如采取机器学习或者深度学习的方式对访问请求进行分类，或者依据实际情况对规则进行适当的修改。但以上两种方式本质上都是基于规则，只适用于传统的防御机制，如防火墙，入侵检测系统等。而拟态系统无需依赖规则去进行攻击行为检查，因此，上述两种方式并不适用。对于拟态化改造的系统而言，产生假阳性的根本原因是拟态化改造以后的拟态界过大。具体来说，在对系统进行拟态化改造时，并未对需要拟态的组件进行合理的选择，对有概率产生假阳问题的组件进行了不必要的异构冗余。本文提出对传统拟态化改造技术的改进方案，提出最佳拟态组件集的概念，作为系统异构冗余的参考标准，有效缩小拟态界。在进行系统的拟态化改造时，基于最佳拟态组件集理论，选择有限的组件进行异构冗余，对于其他组件则不进行异构冗余，而是进行相应的兼容性改造，这样就可以在保证安全性的前提下缩小拟态界，从而解决拟态化改造时带来的假阳性问题。

为了便于描述，本文提出如下定义。

定义 1

完全拟态组件集：系统对用户提供的服务可以拆分为若干个小功能组件，这些小功能组件都存在于拟态界中，称为拟态组件，拟态组件的集合称为拟态组件集。一个系统的所有小功能组件的集合称 为完全拟态组件集。传统拟态化改造会对完全拟态组件集进行异构冗余化。本文假设在进行系统的拟态化 改造时，对完全拟态组件集进行异构冗余化会达到最大的安全增益。

定义 2

最佳拟态组件集：最佳拟态组件集是完全拟态组件集的子集。选择该子集中的组件进行系统的拟态化改造可以有效减少甚至消除假阳性问题。

3.1 使用攻击效费比来度量安全增益

参考文献[8]中作者 Manadhata 提出了利用攻击效费比来进行攻击面度量的理论，参考文献[9]中作者王立群提出了面向非相似余度信息系统的攻击面模型，本文基于两者理论，利用攻击效费比来度量系统的安 全性。将攻击面定义为系统输入与输出集合，系统通道集合，不可信数据项组成的三元组，形式化表示为

〈 〉 〈

simS M，C，I，此时，对于该攻击面 surf，其度量结果可以表示为 ΣNCM dem N

） （ ） （ ）〉

，ΣcCC dem c

，ΣdCI dem d，

（ ） （ ） （) 其中，dem N，dem c，dem d 分别表示系统 入口点和出口点组件、系统通道组件、不可信数据项组件的攻击效费比。上述公式从三个维度对攻击面进行了量化。为了便于描述，本文做出如下定义，并给出相关性质与假设。

定义 3

系统的安全性与系统的脆弱性成负相关，系统的脆弱性越大，则系统的安全性越小。

定义 4

系统的脆弱性可以表示为

vih =

Σ dem N

NCM

+ Σ dem c

cCC

+ Σ dem d # 1

） （ ） （ ）dCI

定义 5

对于完成拟态化改造的系统，从用户视角来说，拟态化后的系统与拟态前的系统提供的功能相一致，而系统入口点和出口点组件、系统通道组件、不可信数据项组件为原系统的三倍，文献[9]作者王立群在面向非相似余度信息系统的攻击面模型中提出了裁决攻击面的概念（RAS，ruling attack surface），即系统 S 的裁决攻击面由所有子系统间的共生资源组成，是系统 S 对外真实呈现的攻击面。$surf_A$表示拟态化改造前系统的攻击面，$surf_B$表示拟态化改造后系统的攻击面，即

$simS_A \leqslant simS_B$#（2）

定义 6

系统的安全增益等于拟态化改造前系统的脆弱性与拟态化改造后系统的脆弱性之差。sec 为系

统安全增益，vih_A表示拟态化改造前系统的脆弱性，vih_B表示拟态化改造后系统的脆弱性。

$sec = vih_A — vih_B$#（3）

性质 1 由于拟态架构的内生安全性[10]，针对

（ ） （ ） （ ）

（ ） （ ） （ ）

（ ） （ ） （ ）

$dem\ N_A \leqslant dem\ N_B$，*其中*$N_A\ C\ M_A\ C\ simS_A$，$N_B\ C\ M_B\ C\ simS_B$# 4 $dem\ c_A \leqslant dem\ c_B$，*其中*$c_A\ C\ C_A\ C\ simS_A$，$c_B\ C\ C_B\ C\ simS_B$ # 5 $dem\ d_A \leqslant dem\ d_B$，*其中*$d_A\ C\ I_A\ C\ simS_A$，$d_B\ C\ I_B\ C\ simS_B$# 6

假设 1

异构冗余可以有效增强系统的安全性和可靠性[11-13]，因此在进行系统的拟态化改造时，对完全拟态组件集进行异构冗余化会达到最大的安全增益。

3.2 求解最佳拟态组件集

本文提出两种不同的方法来求解最佳拟态组件集。

定义 8

$system_A$表示拟态化改造前的系统，$system_B$表示拟态化改造以后的系统。E 表示$system_A$中的所有组件的集合。E_o表示最佳拟态组件集，E_H表示完全拟态组件集。$surf_A$表示拟态化改造前系统的攻击面，$surf_B$表示拟态化改造后系统的攻击面。

3.2.1 增补法

步骤 1

求出基于完全拟态组件集的拟态化改造以后系统的安全增益。

$sec_{Nim} = vih_A — vih_B =$

Σ

NC MA

dem N

+ Σ dem c

cC CA

+ Σ dem d

dC IA

— Σ

NC MB

dem N

+ Σ dem c

（ ） (cC CB

+ Σ dem d # 7

） （ ） （ ）

dC IB

步骤 2 将E_o置为空集。$E_o = Ø$ 。

步骤 3

从E_H中任意取出一个组件 p 放入E_o中，$surf_o$表示只异构冗余E_o中的组件时系统的攻击面。此时的安全增益为

$sec_o = vih_A — vih_o =$

Σ

（ （

NC MA

dem N

+ Σ dem c

cC CA

+ Σ dem d

dC IA

— Σ

NC Mo

dem N

+ Σ dem c

cC Co

+ Σ dem d # 8

dC Io

若 sec_o为 0，则将组件 p 从E_o中取出，否则保

） （ ） （ ）

拟态后某功能的攻击效费比会小于拟态前某功能的攻击效费比。$surf_A$表示拟态化改造前系统的攻击面，$surf_B$表示拟态化改造后系统的攻击面，即

留 p 组件。

步骤 4

继续从E_H中任意取出一个组件 q（取过的组件不能再取），放入E_o中。此时求出对E_o中的所有组件进行异构冗余以后的安全增益。公式与步骤三中的公式相同，若此时的安全增益大于放入 q 组件之前的安全增益，则将 q 组件保留，否则将 q 组件舍去。

步骤 5

不断重复步骤 4，直至安全增益 sec_o等于 sec_{max}。此时，E_o中的组件即为最佳拟态组件集。

3.2.2 减元法

步骤 1

求出基于完全拟态组件集的拟态化改造以后系统的安全增益。

$sec_{Nim} = vih_A — vih_B =$

Σ

NC MA

dem N

+ Σ dem c

cC CA

+ Σ dem d

dC IA

— Σ

NC MB

dem N

+ Σ dem c

( ) (cC CB

(

+ Σ dem d # 9

) ( ) ( )

dC IB

步骤 2 将E_o赋值为E_H

Eo = EH

步骤 3

从E_o中任意取出一个组件 p（取过的组件不能再取），$surf_o$表示只异构冗余E_o组件时系统的攻击面（也就是取出组件以后系统的攻击面）。

此时的安全增益为

$sec_o = vih_A — vih_o =$

Σ

NC MA

dem N

+ Σ dem c

cC CA

+ Σ dem d

dC IA

— Σ

NC Mo

dem N

+ Σ dem c

cC Co

+ Σ dem d

dC Io

1t

( ) ( )

若 sec_o等于 sec_{max}，则将 p 组件舍去，若 sec_o小于 sec_{max}，则将 p 组件放回到E_o。

步骤 4

不断重复步骤 3，直至将E_o内的所有组件都取一遍，此时E_o中的组件即为最佳拟态组件集。

3.3 算法仿真实验

在对系统进行拟态化改造前，分别使用上文两种方法进行组件选择，计算每次迭代时的安全增益与假 阳率，仿真实验结果图如下。

可以看到，利用最佳拟态组件集进行拟态化改造时，系统所获得的安全增益并没有减少，但是却可以 对需要改造的组件进行精简，舍去一些不必要进行异构冗余的组件，有效降低拟态化改造后拟态系统的假 阳率。

4 应用实例

本文以某网上商城 web 系统为例来进行基于最佳拟态组件集的拟态化改造[14]。该系统主要功能有，卖家上架商品并展示，用户浏览商品并且可以下订单购买商品，另外商城提供用户抽奖功

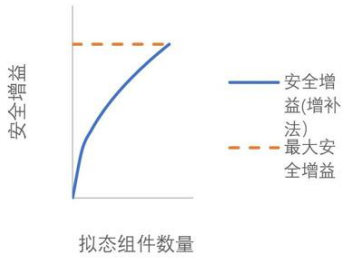


图 1 拟态系统安全增益图（增补法）

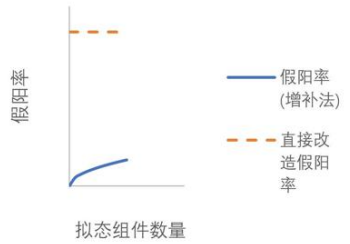


图 2 拟态系统假阳率统计图（增补法）

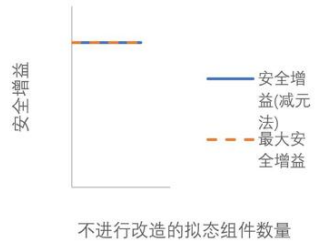


图 3 拟态系统安全增益图（减元法）

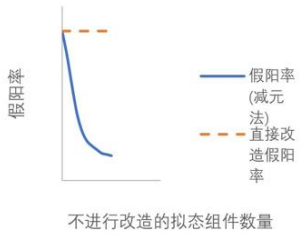


图 4 拟态系统假阳率统计图（减元法）

能。改造前的应用服务器操作系统为 linux，网页服务器为 apache，脚本语言为 php，后端数据库为 mysql。该网上商城的功能由四个模块组成，分别用户登录模块，用户购物/售货模块，用户注册模块，商品展示模块。这些模块

中包含的组件如下表 1。	表格 1 网上商城系统模块组成	
功能模块 用户登录模块	用户购物/售货模块 用户注册模块	商品展示模块
组件 验证码组件	php 脚本组件 用户 id 随机生成组件	php 脚本组件
会话维持组件	数据库服务组件 数据库服务组件	数据库服务组件
php 脚本组件数据库服务组件	数据库数据组件 数据库数据组件抽奖功能组件 php 脚本组件	数据库数据组件
数据库数据组件		

其中，验证码组件是网上商城在提供登录功能时所包含的图片验证码，该图片验证码在每次用户发出登录请求后都会随机刷新，功能是为了防止黑客的暴力破解攻击。会话维持组件指在进行身份认证以后，服务器返回给客户端的 cookie，session 等。用户 id 随机生成组件指用户在注册时，服务器给用户随机生成的一个标识，该数据会存放到数据库中。抽奖功能组件指网上商城在提供抽奖服务时所涉及到的组件，其

中包含了随机数生成器。php 脚本组件指提供某功能的 php 服务。数据库服务组件指与数据库发生交互的语句，如查询，增加，删除，修改等。数据库数据组件指存放在数据库内的组件，如订单数据，用户账号 与密码数据信息，商品数据信息等。

综上，提供网上商城服务的 web 系统的组成组件可以归纳为，验证码组件，会话维持组件，php 脚本组件，数据库服务组件，数据库数据组件，抽奖功能组件，用户 id 随机生成组件，操作系统组件，网页服务器组件。

下面分别使用增补法和减元法来对最佳拟态组件集进行求解。

4.1 增补法

首先，求出基于完全拟态组件集的拟态化改造以后的安全增益，这里假设未进行拟态化改造前，web 服务器的总攻击效费比为 25，改造后 web 服务器的总攻击效费比为 5，求出此时的最大安全增益 20。$surf_A$ 表示拟态化改造前系统的攻击面，$surf_B$ 表示拟态化改造后系统的攻击面。

$$sec_{Nim} = vih_A — vih_B$$

((

= Σ
NC MA
dem N
) (+ Σ dem c
cC CA
+ Σ dem d
dC IA
— Σ
NC MB
dem N
( ) (+ Σ dem c
cC CB
+ Σ dem d

) ( ))

dC IB
= 25 — 5
= 2t

然后将 E_o 置空，此时从 E_H 中任取一个组件放入 E_o 中，这里假设将 php 脚本组件放入 E_o 中，求出此时的安全增益：

$sec_o = vih_A — vih_o$

((

= Σ
NC MA
dem N
) (+ Σ dem c
cC CA
+ Σ dem d
dC IA
— Σ
NC Mo
dem N
( ) (+ Σ dem c
cC Co
+ Σ dem d

) ( ))

dC Io
= 8 — 5
= 3

安全增益>0，因此将 php 脚本组件保留。

继续从 E_H 中取一个组件，假设这里将用户会话组件放入 E_o 中，求出此时的安全增益：

$sec_o = vih_A — vih_o$

((

= Σ
NC MA
dem N
) (+ Σ dem c
cC CA
+ Σ dem d
dC IA
— Σ
NC Mo
dem N
( ) (+ Σ dem c
cC Co
+ Σ dem d

) ( ))

dC Io
= 8 — 8
= t

安全增益为 0，因此将用户会话组件从组件集 E_o 中删除。

不断重复增补法中的步骤四，当组件集 E_o 中的组件为 php 脚本组件，数据库服务组件，操作系统组件，

网页服务器组件时，此时的安全增益已经等于最大安全增益，因此此时集合内的组件集即为

最佳拟态组件 集。

4.2 减元法

首先，求出基于完全拟态组件集的拟态化改造以后的安全增益，这里假设未进行拟态化改造前，web 服务器的总攻击效费比为 25，改造后 web 服务器的总攻击效费比为 5，求出此时的最大安全增益 20。

$sec_{Nim} = vih_A — vih_B$

$$\left(\quad (\right.$$

$= \Sigma$
NC MA
dem N
) (+ Σ dem c
cC CA
+ Σ dem d
dC IA
— Σ
NC MB
dem N
( ) (+ Σ dem c
cC CB
+ Σ dem d

$$) \quad ( ) \left. \right)$$

dC IB
= 25 — 5
= 2t

将最佳拟态组件集的内容置为完全拟态组件集。

然后从E_o中任意取出一个组件 p，这里将 php 服务组件取出，$surf_o$表示只异构冗余E_o组件时系统的攻击面。此时的安全增益为：

$sec_o = vih_A — vih_o$
$= \Sigma$
NC MA

$$\left(\quad (\right.$$

dem N
) (+ Σ dem c
cC CA
+ Σ dem d
dC IA
— Σ
NC Mo
dem N
( ) (+ Σ dem c
cC Co
+ Σ dem d

$$) \quad ( ) \left. \right)$$

dC Io
= 2t — 5
= 15

此时的安全增益小于最大安全增益，因此将 php 服务组件放回。继续从E_o中任意取出一个组件 p，这里将用户会话组件取出。此时的安全增益为：

$sec_o = vih_A — vih_o$

$$\left(\quad (\right.$$

$= \Sigma$
NC MA
dem N
) (+ Σ dem c
cC CA
+ Σ dem d
dC IA
— Σ
NC Mo
dem N
( ) (+ Σ dem c
cC Co
+ Σ dem d

$$) \quad ( ) \left. \right)$$

dC Io

= 25 — 5

= 2t

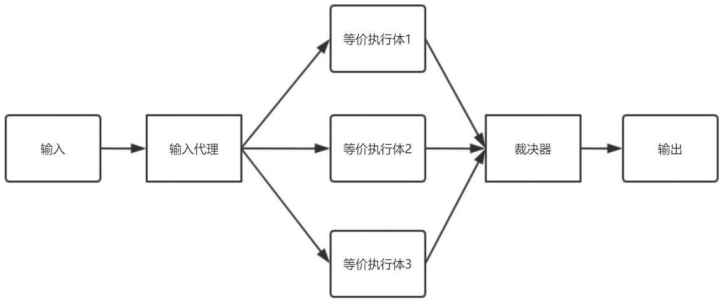
此时的安全增益等于最大安全增益，因此不将会话组件放回。不断重复步骤三，直至将E_o内的所有组件都取一遍，此时E_o的组件为php 脚本组件，数据库服务组件，操作系统组件，网页服务器组件，该组件的集合即为最佳拟态组件集，这个结果也与增补法求出的结果相一致。

基于以上两种方法求出的最佳拟态组件集对web 应用服务器进行拟态化改造，web 应用服务器的改造内容分别为 php 语言加入不同标签并异构 php 版本，数据库交互语句加入不同标签，异构操作系统，将原来的单一 linux 系统异构冗余为三个操作系统，ubuntu，centos，windows。异构网页服务器，将原来的 apache 异构冗余为 nginx，apache，iss。可以看到基于最佳拟态组件集去进行拟态化改造时，那些会造成假阳性问

题的组件并未加入到拟态化改造的组件中，因此可以有效降低拟态系统的假阳率。

4.3 有效性验证

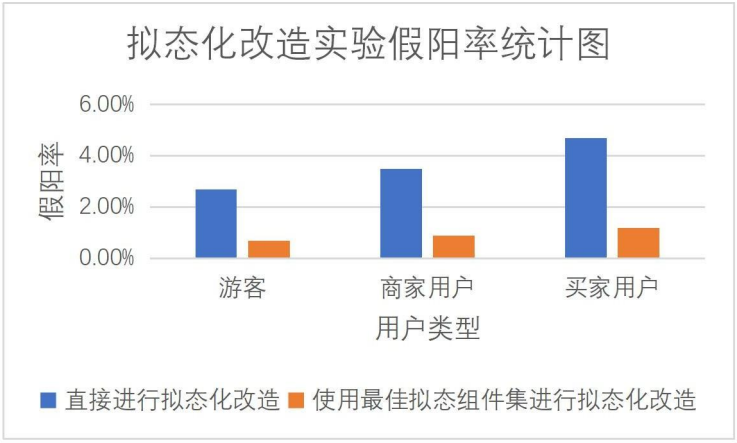
为了对本文提出的理论进行有效性验证，本实验采取直接改造与基于最佳拟态组件集进行组件选择这 两种改造方式，分别对上文提及的网上商城系统进行拟态化改造，本实验对网上商城系统拟态化改造之后 的系统架构图如图 5，输入代理的功能与裁决器的功能都由代理服务器提供。方法一对网上商城系统进行直接的拟态化改造，方法二首先使用最佳拟态组件集理论选择需要改造的组件，然后再对网上商城系统进 行拟态化改造。本实验结合访问日志分析，将用户正常请求（50000 次任意请求）中的假阳率作为参考依据。具体情况如图 6。



代理服务器与执行体的具体配置如下表：

因此，在系统的拟态化改造过程中，使用最佳拟态组件集来选择必要的组件进行改造可以有效降低拟 态系统的假阳率。但是假阳性问题并不能完全消除，考虑到可能是由于其他因素（如时

	代理服务器	执行体 1	执行体 2	执行体 3
系统类型	Windows10	Ubuntu16	Centos7.0	Windows7
服务器类型	Nginx	Apache	Nginx	Iss
PHP 版本		PHP5.6	PHP5.5	PHP7.0
PHP 标签		PHP 标签一	PHP 标签二	PHP 标签三
数据库标签		数据库标签一	数据库标签二	数据库标签三



间延迟等）造成了 该问题。

5 结束语

针对目前在对系统进行拟态化改造时，难以解决的假阳性问题，本文提出最佳拟态组件集理论，并提 出使用增补法和减元法两种不同的算法选择最佳拟态组件，两种算法得到的最佳拟态组件集可以作为系统 拟态化改造时的参考标准，在不影响安全性的前提下合理缩小拟态界，有效降低甚至消除假阳率。

但是如何更加准确的量化拟态化组件所带来的安全增益，如何更高效的选择组件构成最佳拟态组件集， 将是本文未来的研究方向。

参考文献:

[1] 邬江兴．网络空间拟态防御原理：广义鲁棒控制与内生安全(上册)[M]．北京：科学出版社，2018. WU J X. Principles of mimic defense in cyberspace: generalized ro bust control and endogenous security (Volume 1)[M]. Beijing：Science Press，2018.

[2] 邬江兴．网络空间拟态防御原理：广义鲁棒控制与内生安全(下册)[M]．北京：科学出版社，2018. WU J X. Principles of mimic defense in cyberspace: generalized ro bust control and endogenous security (Volume 2)[M]. Beijing：Science Press，2018.

[3] 郭威．分布式存储系统拟态化架构与关键技术研究[D]．战略支援部队信息工程大学，2019.
Guo Wei. Research on mimicry architecture and key technologies of distributed storage system[D].
Information Engineering University of strategic support forces，2019

[4] 邬江兴．网络空间拟态防御研究[J]．信息安全学报，2016（4）：1-10.
Wu Jiangxing. Research on Cyber Mimic Defense [J]. Journal of information security，2016（4）：1-10

[5] 林琳．详细了解 Cookie Session Token[J]．计算机与网络，2019（22）：37.
Lin Lin. learn more about cookie session token [J]. Computer and network，2019（22）：37

[6] 刘金源．常见的 Web 安全漏洞及防御技术[J]．网络安全和信息化，2020（08）：123-124.
Liu Jinyuan. Common web security vulnerabilities and defense technologies [J]. Network security and informatization，2020（08）：123-124

[7] 姚远, 潘传幸, 张铮,等．多样化软件系统量化评估方法[J]．通信学报，2020，041（003）：120-125. Yao yuan，pan Chuanxing, Zhang Zheng, et al. Quantitative evaluation method of diversified software systems [J]. Acta Telecom Sinica，2020，041（003）：120-125

[8] Manadhata P K, Wing J M. A formal model for a system's attack surface [M]//Moving Target Defense. Springer, New York, NY，2011：1-28.

[9] 张铮, 王立群, 李卫超．面向非相似余度信息系统的攻击面模型[J]．通信学报，2018，39（S2）：227-234. Zhang Z, Wang L Q, Li W C. Research on formal model for an information system's attack surface with dissimilar redundant architecture. Journal of Communications[J]，2018，39（S2）：227-234.

[10] 邬江兴．内生安全：重新定义新基建的安全属性[J]．中国科技产业，2020（5）：7-9.
Wu Jiangxing. Endogenous security: redefining the security attribute of new infrastructure [J]. China Science and technology industry，2020（5）：7-9

[11] CAPIZZI R, LONGO A, VENKATAKRISHNAN V N, et al. Preventing information leaks through shadow executions[J]. 2008：322-331.

[12] COX B, EVANS D, FILIPI A, et al. N-variant systems: a secretless framework for security through diversity[C]//Conference on Usenix Security Symposium. 2006：9.

[13] DEVRIESE D, PIESSENS F. Noninterference through secure multi-execution[C]// Security and Privacy，2010：109-124.

[14] 仝青, 张铮, 张为华, 等．拟态防御 Web 服务器设计与实现[J]．Journal of Software，2017，28（4）.
Tong Qing, Zhang Zheng, Zhang Weihua, et al. Design and implementation of pseudo defense web server [J]. Journal of Software，2017，28（4）.

[作者简介]

邵昱文（1997—），男，学士，博士生，主要研究方向为网络空间安全，漏洞挖掘。邮箱 735011726@qq. com。

张铮（1976—），性别，博士，副教授，主要研究方向为网络空间安全，主动防御技术。邮箱ponyzhang@163. com。

王晓梅（1976—），女，博士，副教授，主要研究方向为通信网络，大数据。

潘传幸，（1996—），男，学士，博士生，研究方向为主动防御、内生安全。

邬江兴（1953– ），男，中国工程院院士，教授、博士生导师，主要研究方向为网络通信与安全。

新一代集成电路发展趋势思考

邬江兴，刘勤让，沈剑良，吕平，汤先拓
国家数字交换系统工程技术研究中心，郑州450002

摘　要：随着摩尔定律和登纳德缩放定律逐渐失效，新材料、新工艺、新器件和新方法等成为集成电路创新发展的热点。本文从未来信息基础设施基本特征出发，详细分析了集成电路领域所面临的物理极限、良率极限和封装极限等三大发展困局，并从多个维度系统阐述了未来基础设施对集成电路提出的新需求以及新一代集成电路的发展趋势。

关键词：集成电路 信息基础设施 片上系统 软件定义 晶上系统

Thinking on the development trend of the new generation of integrated circuits

WU JiangXing, LIU QinRang, SHEN JianLiang, Tang Ping LV Xiantuo
National Digital Switching System Engineering and Technical RD Center, Zhengzhou 450002

Abstract: With the gradual failure of the Moore's law and Dennard's scaling laws, new materials, new processes, new devices and new methods have become the hot spots of IC innovation and development. Starts from the basic characteristics of future information infrastructure, This paper analyzes in detail the three dilemmas in the development of IC technology, including physical limit, yield limit and package limit, and systematically describes the new requirements of future infrastructure for integrated circuits from multiple dimensions,and the development trend of the new generation integrated circuit。

Key words: Integrated circuit; Information infrastructure; system-on-chip

1　引言

随着摩尔定律和登纳德缩放定律逐渐失效，无论高性能计算机、大型云服务基础设施，还是超大规模芯片先进制程工艺等，都陷入了按传统行动路线在技术和经济上都难以持续发展的困局。5nm，3nm等先进工艺的寡头效应加剧，新材料、新工艺、新器件和新方法等成为集成电路创新发展的热点，集成电路的技术与产业格局进入重大变革期。

2　集成电路技术发展困局

摩尔定律是由英特尔的创始人之一戈登·摩尔提出来的。摩尔定律的内容为：当价格不变时，集成电路上可容纳的晶体管的数目，约每隔18-24个月便会增加一倍，性能也将提升一倍。在定律被提出后的一段时间里，集成电路的发展动力较为强劲，约每18个月工艺就进行一次迭代。随着技术节点不断下探，工艺的迭代速度已经有所放缓。2015年国际半导体行业机构联合发布的国际

半导体技术线路图（ITRS）显示，随着集成电路尺寸不断缩小，技术瓶颈制约工艺的发展越来越明显，从2015年以来产品换代速度已由之前的18个月下降到24个月，这种现象预计将会持续到2030年。

技术是衡量摩尔定律发展的主要因素之一，但是经济因素也是设计公司考量的重点。3nm制程的开发费用预计耗资40亿至50亿美元。台积电计划投入3nm的资金高达6000亿新台币，约合190亿美元。此外，设计成本也是一个问题。

集成电路一直按照“晶圆-划片-封装”的工程技术路线。伴随工艺制程技术的进步，晶体管和线宽越做越小（已达纳米级）、功能越加越多、规模越来越大、成本越来越低。尤其是SoC系统的出现，提高了产品性能、增加了产品功能和可靠性、缩短了开发周期、降低了开发成本，带动了电子信息系统的跃迁式发展。但随着芯片工艺制程越来越接近量子效应区域，单位面积中容纳的晶体管数量越来越多，制造和设计成本呈指数级攀升、良率控制越来越难、研发周期越来越长、功耗问题越来越突出，集成电路按照SoC基于单一工艺节点进行IP复用的技术路线正遭遇物理节点失效、经济学定律失效、PPA指标难以为继等困局。当前的集成电路发展面临着物理极限、良率极限和封装极限三个维度发展的困局，如图所示：

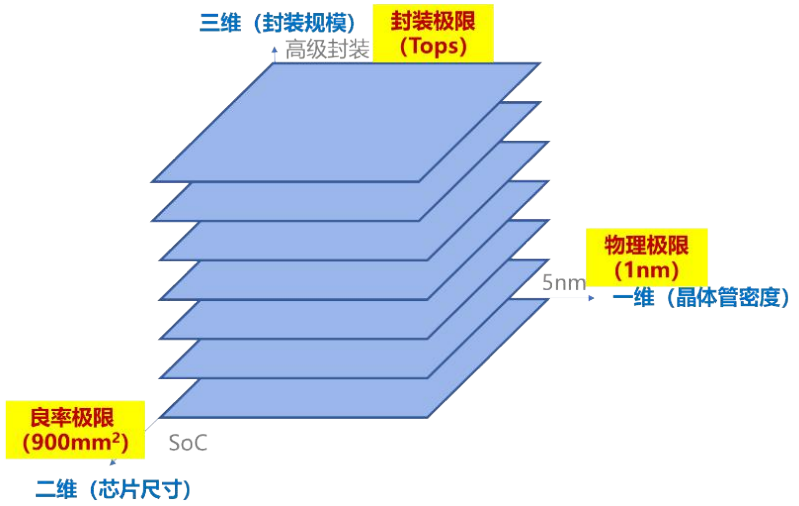


图1　集成电路发展遇到的三个困局

2.1　困局一：工艺节点持续进步逼近物理极限

在芯片设计中，工艺尺寸是指晶体管栅极的宽度。栅极越小，可以封装到给定空间中的处理能力就越大。3nm以下工艺一直被公认为是摩尔定律最终失效的节点，随着晶体管的缩小将会遇到物理上的极限考验。2019三星代工论坛”（Samsung Foundry Forum 2019）上，发布新一代3nm闸极全环（GAA，Gate-All-Around）工艺，预计2021年量产3nm GAA工艺。台积电也在积极推进3nm工艺，预计在2021年进入风险试产阶段，2022年下半年量产。虽然台积电与三星电子都在积极推进3nm的技术开发与生产，但是3nm之后的硅基半导体工艺路线图暂时没有指定，主要是由于集成电路加工线宽达到3nm之后，将进入介观（Mesoscopic）物理学的范畴，各种物理障碍——如杂质涨落、量子隧穿等，都会成为集成电路发展的阻力。介观物理和基于量子化的处理方法是解决这些物理障碍的有效手段。但目前而言，这些技术在商业化上还尚未成熟，这将制约集成电路发展的一大因素。

2.2　困局二：单芯片尺寸增大遭遇良率极限

众所周知，工艺越先进，芯片面积越大，所能承载的功能越多，性能也优于多个小芯片拼装而成的产品，为了延缓摩尔定率的失效，很多厂商都在研发超大规模的芯片，尤其是广泛应用于人工智能、自动驾驶和虚拟现实领域的GPU，一直在追求大面积。2019年9月25日，阿里巴巴在杭州正式发布含光800AI芯片，采用台积电12nm工艺，核心面积高达709mm2，性能接近Nvidia 2017年发布的GPU Tesla V100（台积电12nm工

艺、芯片面积815 mm2，210亿个晶体管）。2020年5月14日Nvidia公司正式发布了世界最大的单芯片A100 GPU，该芯片历时4年研发，Die的面积为826mm2，采用了台积电7nm工艺，集成了540亿晶体管，达到了7nm工艺的极限水平了，与上一代GPU Tesla V100相比较，性能提升了约20倍。但是大芯片的带来的一个主要问题是芯片良率降低，成本升高。

芯片成本的一个重要依据是每个Wafer能制造多少个Die，因为Die一般是长方形或者正方形，所以圆形的Wafer边缘部分被浪费了，芯片尺寸越大，被浪费的边缘部分越多。在Die的尺寸达到足够小的尺寸时，理论上整张Wafer都可以被利用，即面积利用率达100%。但是在晶圆在制造过程中存在不可避免的缺陷，这些缺陷分布在整个Wafer上，Die的大小会将会影响到芯片的良率，如图所示：

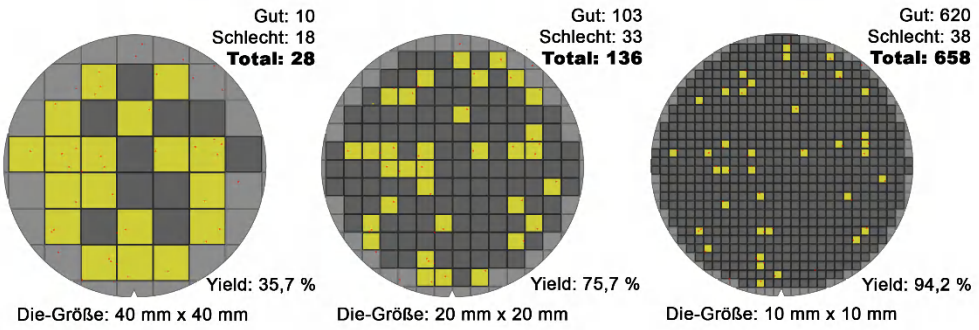


图2　芯片尺寸与良率

黄色代表有缺陷的Die，黄色Die上的红色小圆点代表芯片的缺陷，三种Die的尺寸分别为40nm x 40nm、20nm x 20nm和10nm x 10nm，对应的Die良率分别为35.7%、75.7%和94.2%，因此，良率和面积并不是线性关系，尺寸越小对应的Die良率越高，芯片尺寸和面积比是线性，分别是4：1和16：1，而是呈指数关系但是好的Die与有缺陷的Die的比例则是62：1。

从理论上分析，同等功能的芯片会在先进制程下面积变小，随之良率也会提高，同步制程提高还能使得相同功能的芯片功耗更低，性能更好，但是在实际操作过程中，先进的制程也会使得芯片的缺陷更严重，有可能使得良率更低，同时漏电流增大，待机功耗增加。因此通过提高制程来减小面积提升良率也不是很好的解决办法。

2.3　困局三：高级封装技术的散热难题

作为芯片制造过程的最后一步，封装在电子供应链中看似不起眼，却一直发挥着极为关键的作用。在后摩尔时代，各Foundry厂在继续推进先进制程工艺的同时，也在积极发展先进封装技术，希望能够通过提升多芯片的集成封装密度、降低整体的面积、提升带宽及连接速度，来实现对于摩尔定律经济效益的继续推动。2018年Intel将先进封装技术作为其六大战略布局方向之一，陆续推出了EMIB2.5D、Foveros 3D封装技术；AMD的EPYC（Naples）[5]、每个EPYC处理包括4个Zeppelin die，使用了2D MCM（Multi-chip module）封装；Altera Stratix 10 FPGA采用了EMIB封装技术[6]，中心是FPGA die，周围是6个chiplet；Altera Lakefield SoC则采用了3D Foveros封装技术[7]。为了应对人工智能对高效能运算芯片的需求，台积电将于2020年推出第五代CoWoS（Chip-on-Wafer-on-Substrate）封装工艺。三星也推出了x-Cube封装制程，并在2018年三星晶圆代工论坛日本会议上，公布了其封测领域的路线图，目前三星的x-Cube封装制程已在其自家7nm和5nm制程上得到了验证。

高级封装技术在一定程度上解决了摩尔定律失效的问题，但是也带来了散热问题[8]：1）芯片堆叠后发热量将增加，但散热面积并未相对增加，因此发热密度大幅提高；2）多芯片封装虽然仍保有原散热面积，但由于热源的相互连接，热耦合增强，从而造成更为严重的热问题；3）内埋置基板中的无源器件也有一定的发热问题，由于有机基板或陶瓷基板散热不良，也会产生严重的热问题；4）由于封装体积缩小，组装密度增加，

使得散热不易解决，因此需要更高效率的散热设计。另外高级封装技术受功耗、面积等的影响，也仅能支持较少数量的Die堆叠，无法真正解决摩尔定律失效带来的问题。

3 未来智能信息基础设施对集成电路的需求

随着云化集约、人工智能、大数据等的飞速发展，体系架构、集成电路的发展也出现了新的趋势。当前，软硬件协同计算正在成为新的计算模式，以面向应用的软件定义为中心，通过软件去定义硬件系统、网络平台乃至基础设施成为了新的服务模式。在2018年世界计算机体系结构大会（ISCA，International Symposium on Computer Architecture）上，图灵奖得主John L. Hennessy、David A. Patterson也指出，领域专用软硬件协同计算成为计算机体系结构发展的新方向。另外，“牧村浪潮”（Makimoto’s Wave）和许氏循环均指出，半导体产品每十年波动一次，沿着“通用”与“定制”交替发展，2018年～2028年之间半导体将会重新走向通用。通过对原始芯片的配置编程就可以得到用户自定义的功能电路，从而引导半导体产业结构演变，使得“硬”、“软”均可编程，即算法可编程、可重构器件也可编程的U-SoC，集成电路进入到软件定义芯片的时代，为变结构处理、变结构计算等打下了坚实的技术基础。

集成电路作为未来智能技术载体与智能产业基石，不论是超算中心、大数据中心、工业互联网，还是边缘计算、人工智能终端、物联网，都正在对集成电路功能、性能、成本等提出了新的需求。随着软硬件协同计算、软件定义芯片技术的不断发展，本文从工程系统、芯片能力、新型应用和服务模式几个维度阐述了未来基础设施对集成电路提出的新需求：

3.1 工程系统角度：超高密度与超强能力芯片需求；

信息系统一直遵循“芯片、模组、机匣、机架、系统”的工程实现路线，如图3所示。对信息基础设施，如超级计算、云计算、边缘计算、数据中心及核心网络骨干节点等，为实现最大化的集约服务效能，服务足够多的用户，提供尽可能多的计算、存储和网络资源，对CPU、DSP、GPU、RAM、SWITCH、ADC、RF等不同用途的芯片，往往要层次化的堆叠数万乃至数十万规模之巨，典型系统功耗达数兆瓦乃至数十兆瓦量级，机架数量达到上百个之多，占地总面积达到数千平方米。更为严重的是，这种“拼规模”的发展模式会导致系统性价比和效能急剧下降，功耗和延迟则会急剧增加，基于不同功能芯片逐层堆叠系统的工程技术路线遭遇可持续发展瓶颈。以连续两年（2016年、2017年）斩获世界超级计算机排名榜单TOP500第一名的太湖之光超级计算机为例，太湖之光由40个运算机柜和8个网络机柜组成，集成了40960个中国自主研发的“申威26010”众核处理器，占地面积605平方米，功耗达15.37MW，运行的实际功耗＞18MW。人类大脑有100多亿个神经细胞组成，如果把大脑的活动转换成电能，相当于一只20瓦灯泡的功率。虽然在处理速度、存储容量上，人脑无法与超级计算相比较，但人类大脑的推理判断等智能特征是未来信息基础设施所需具备的基本要求，在形态上也将能够接近人类智能的大脑，但是当前的基础设施都不满足智能社会的需求。

图3 “芯片、模组、机匣、机架、系统”的工程实现路线

根据公式（1）可知，系统性能与芯片的处理密度强相关，芯片处理密度越大，芯片处理能力越强，系统性能越优。

P=f（C）…… 公式（1）

因此，从系统工程的角度看，需要更高处理密度，更强处理能力的芯片来构建未来智能信息基础设施。

3.2 芯片能力角度：“摩尔定律”持续有效需求；

当前摩尔定律逼近物理学与经济学双重极限，随着万物互联时代的到来，大数据却呈现爆炸式增长，数据量与处理能力出现巨大的剪刀差，大量的数据存不下、算不急，导致大量有用数据丢失。另外，工艺进步对计算性能提升明显降低，红利逐渐消耗殆尽，如图4所示。

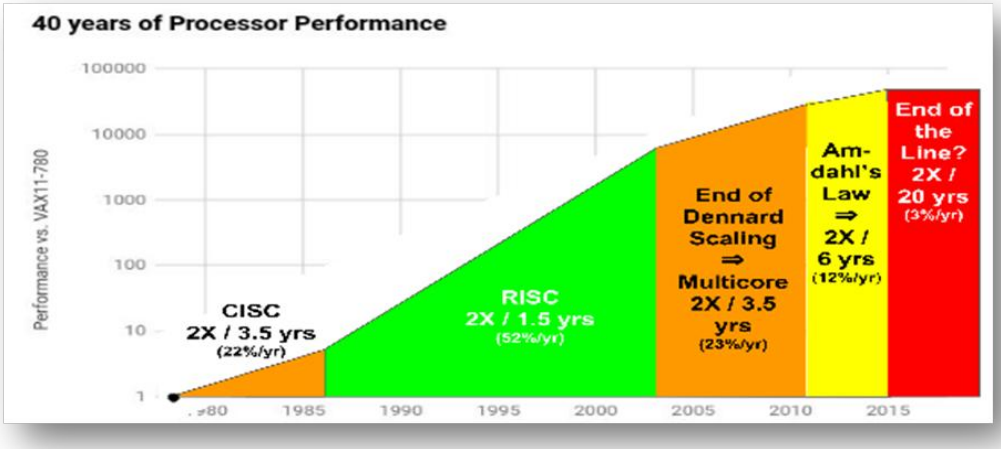


图4 计算性能随工艺与体系结构创新的发展状态

图4源自John Hennessy和David Patterson的《计算机体系结构：定量方法》[9]，从这张图中可以看出，从1986年到2004年，RISC微处理器在摩尔定律（每个新的半导体工艺节点处的晶体管多2倍）和登纳德缩放定律（MOSFET的功率密度是常数，即工艺尺寸缩减，功耗跟随降低）的推动下，获得了近20年的快速性能提升。

随着登纳德缩放定律在晶体管与连线（wire）两方面都遇到困难，登纳德缩放定律逐渐失效，各个处理器都停止了提高速度。每个晶体管的功耗也停止在每个节点上降低一半。通过完全依靠摩尔定律（每个节点增加2倍的晶体管）并迅速增加一个芯片上的处理器数量来迎接多核时代。在这个时代，处理器性能加倍的间隔从1.5年延长到3.5年，持续五年左右的时间。虽然登纳德缩放定律，在阿姆达尔定律的推动下，处理能加倍的间隔从3.5年延长到6年，但是随着摩尔定律的放缓，自2015年以来处理器性能的提高已急剧下降到每年仅3%。处理器性能加倍的时间也由1.5年、3.5年延长至当前的20年了。

因此，如何让摩尔定律速度持续有效？甚至用更快能力增长速度来匹配数据增长速度是智能时代对芯片能力的基本需求。

3.3 新型应用角度：领域专用软硬件协同需求

当前的互联网已经从浏览式互联网演进到支持商业应用的消费互联网，随着智能时代的到来，将进入到万物互联的产业互联网，低时延、高带宽是未来网络必须满足的基本要求，虽然5G技术在普及过程中，但5G技术更多的关注在边缘网络，并没有支持核心网络，而当前的其它网络技术也无法在时延和带宽等方面满足未来产业互联网应用的需求。众所周知，传统的计算机将数据储存在内存中，然后传送到处理器运算，这种来回“搬运”数据的活动耗费能源和时间，是冯·诺依曼计算架构的核心瓶颈。未来的网络将会像人脑一样直接在记忆体里面计算，领域专用软硬件协同计算、存储、网络融为一体的网络将会满足这一需求，传统计算、存储、网络分离的信息基础设施的形态将会被彻底打破。

3.4 服务模式角度：重新定义经济性指标需求

芯片行业是典型的人才密集和资金密集型高风险产业，按照现有的经济模式，如果没有大量用户摊薄费用，芯片成本将直线上升，制程工艺的研发和生产成本逐代上涨，由图5可知5nm制程下的设计费用已高达5.4亿美元。根据市场研究机构International Business Strategies（IBS）的数据，3nm芯片的设计费用约5~15亿美元，工艺开发费用约40~50亿美元，兴建一条3nm产线的成本约为150-200亿美元。3nm芯片仅比5nm芯片提升15%性能、降低25%功耗。

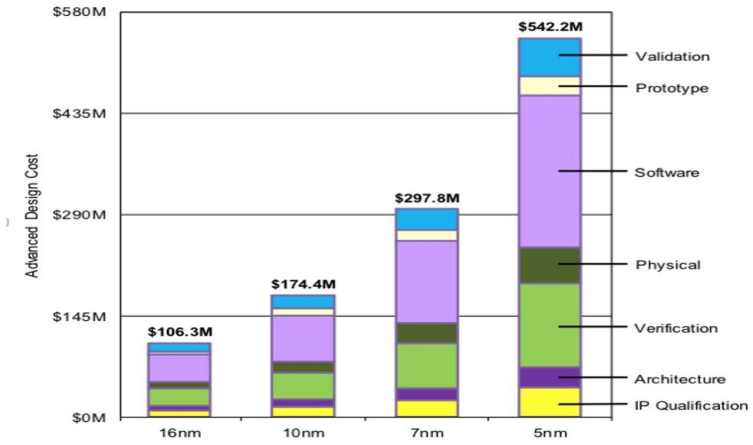


图5 不同工艺制程下设计开销对比

资金、技术壁垒提升，先进制程的供给端向寡头垄断发展，先进制程供不应求。目前先进制程的供给端只有台积电、三星、英特尔。英特尔为IDM（Integrated Design and Manufacture）公司，不对外提供流片服务。受益于5G、智能手机、高性能计算、人工智能、物联网等需求，7nm及以下先进制程需求旺盛。台积电为先进制程的核心晶圆代工厂，目前10nm工艺客户已经超过10家，7nm EUV客户至少5家（苹果（APPLE、海思、高通、三星、AMD），6nm客户除了7nm EUV的5家还多了博通、联发科。使用先进节点的好处很多，晶体管密度更大、占用空间更少、性能更高、功率更低，但是设计制造成本缺越来越高，在当前的集成电路商业模式下，采用先进制程的芯片只有达到足够的出货量才能确保当前的商业模式成立，以5nm制程下的芯片为例，出货量需达到1000万片。随着云服务［10］的普及，靠芯片销售数量的经济模式，在云化集约服务和多样化碎片应用场景都无法持续，亟需重新定义芯片的经济性指标，未来将以服务频次作为衡量集成电路的经济指标。

5 展望

不论是大型信息基础设施面临的基于细粒度芯片的层层堆叠式工程技术路线发展瓶颈，还是集成电路发展面临的单片功能、性能和经济学天花板困局，其根本原因在于SoC（SoC，System On Chip）基于单一工艺节点进行复杂度越来越高、周期越来越长、投入越来越大的IP复用设计、验证和流片所导致，只有颠覆工程技术路线、打破片上系统边界条件束缚，才能有效破解超大规模芯片和大型信息基础设施面临的发展困局，找到可共同持续发展的新基线。

超高密度晶圆级互连和软硬件协同计算结构应该代表着工艺节点逼近物理极限、摩尔定律逐渐失效的集成电路发展的可行出路，其基本思路是将软件定义架构贯穿到集成电路设计、加工和封装的全流程，融合预制件组装和晶圆集成等先进理念，借助晶圆级互连的高带宽、低延迟、低功耗等显著优势，可以实现单一晶圆上集成成千上万的传感、射频、计算、存储、通信等“预制件”颗粒，打破现有集成电路的设计方法、计算范式、实现材料、集成方式等边界条件，相比基于PCB的芯片焊装成组件堆叠出更大系统的工程

实现模式，晶圆级集成系统通过TCB（Thermal Compression Bonding），可以实现带宽可提升一个数量级，延迟可降低一个数量级，功耗可降低一个数量级，再连乘体系结构创新带来的两个数量级构造增益，整体系统性能可提升四到六个数量级。同时通过硬件资源的在线编译重构，可灵活满足多样化应用领域与场景，贡献一条与工艺进步弱相关的全新发展路线。同时也颠覆芯片按销售规模获利的传统商业模式，催生按服务次数计费的崭新商业模式。需要重点发展的主要技术方向包括：

（1）领域专用软硬件协同计算结构探索设计；
（2）晶圆级层次化互连接口与标准设计；
（3）领域专用异构“预制件”选取方法；
（4）“预制件”软件定义生成方法；
（5）晶圆级先进硅互连工艺技术研究；
（6）预制件和晶圆互连系统的测试方法；
（7）软硬件协同资源调度算法；
（8）软硬件协同编译工具；
（9）硬件级内生安全技术。

参考文献:

[1] 思科发布云产业调研报告(2016-2021. http://www. ecas. cas. cn/xxkw/kbcd/201115_124731/ml/xxhjsyjcss/ 201803/t20180305_4538378. html

[2] 腾讯人工智能白皮书:泛在智能. 腾讯研究院,2020

[3] 全球产业展望GIV 2025 :打开智能世界产业版图. http://www. 199it. com/archives/739297. html

[4] IDC发布最新版«数据时代2025»白皮书. http://www. chinastor. com/market/12214001R018. html

[5] AMD EPYC（Naples） https://www. aspsys. com/solutions/hpc-processors/amd-epyc/

[6] 英特尔展示EMIB封装技术 跟AMD2. 5D封装类似但技术水平更高. http://www. elecfans. com/emb/dsp/20180820734341. html

[7] 英特尔3D封装技术深度解读. https://www. sohu. com/a/291109996_132567

[8] 曾理, 陈文媛, 谢诗文,等. 集成电路封装高密度化与散热问题[J]. 电子与封装, 2006(9):15-21.

[9] 计算机系统结构——一种定量的方法[M] 郑纬民,译. 2版. 北京: 清华大学出版社, 2002

[10] Hossain M , Khan R , Noor S A , et al. Jugo: A Generic Architecture for Composite Cloud as a Service [C]// IEEE International Conference on Cloud Computing. San Francisco，CA，IEEE，2017. 806-809

[作者简介]

邬江兴（1953—），男，工程院院士，教授，博士生导师，主要研究方向

刘勤让（1975—），男，博士，研究员，博士生导师，网络信息安全、新型体系结构

沈剑良（1982—），男，博士，副教授，硕士生导师，新一代网络信息系统架构、集成电路设计

吕平（1977—），女，博士，副研究员，新型体系结构，集成电路设计

汤先拓（1985—），男，博士，助理研究员，片上网络设计，集成电路设计

物联网下的智能物理层认证机制

李兴璐[1]，黄开枝[1]，王少禹[1]，许晓明[1]，张波[2]
[1]战略支援部队信息工程大学，河南，郑州，450002；
[2]国防科技大学，江苏，南京，210012

摘　要：由于无线信道的广播特性，物联网节点在接入网络时极易受到来自攻击者的欺骗攻击和 Sybil 攻击。为增强区域内物联网节点接入安全认证，本文提出了一种基于卷积神经网络（Convolutional Neural Network，CNN）的智能物理层认证机制。此机制基于位置信息，将物联网区域进行划分，并提取信道状态信息（Channel State Information， CSI）作为无线信道指纹区分位于不同位置的节点。在本文中，将获取的CSI 进行离线训练以提取深层空时环境特征，并将节点身份信息与空时环境特征绑定，通过 CSI 在线识别外来未知节点身份，达到非法节点入侵检测及区域安全接入效果。最后本文通过实验分析了所提出的智能 物理层认证机制的认证精度，对比了在不同信噪比情况下和使用不同梯度下降算法的认证性能。实验结果 表明，此机制具有良好的认证精度，可以实现对未知节点身份的精准识别，并进一步根据识别结果定位未 知节点所在的区域范围，实现实时、高效的非法节点入侵检测及区域管控。

关键词：物联网、CSI、CNN、认证、入侵检测

Intelligent physical layer authentication mechanism under the IoT

Li Xinglu[1], Huang Kaizhi[1], Wang shaoyu[1], Xu Xiaoming[1], Zhang Bo[2]
1.PLA Strategic Support Force Information Engineering University, Zhengzhou, Henan, 450002;
2.National University of Defense Technology, Nanjing, Jiangsu, 210012

Abstract: Due to the broadcast characteristics of the wireless channel, the Internet of Things (IoT) nodes areextremely vulnerable to spoofing attacks and Sybil attacks from attackers when they access the network. In order to enhance the security authentication of the IoT node access in the area, this paper proposes an intelligent physical layer authentication mechanism based on Convolutional Neural Network (CNN). This mechanism divides the IoT area based on location information and extracts Channel State Information (CSI) as a wireless channel fingerprint to distinguish nodes located at different locations. In this paper, the obtained CSI is trained offline to extract the deep space-time environment characteristics, while the node identity information is bound with the space-time environment characteristics, and the CSI is used to identify the identity of unknown foreign nodes online, so as to achieve the effect of illegal node intrusion detection and regional security access. Finally, this paper analyzes the authentication accuracy of the proposed intelligent physical layer authentication mechanism through experiments, then compares the authentication performance under different signal-to-noise ratios and using different gradient descent algorithms. The experimental results show that this mechanism has good authentication accuracy， can accurately identify the identity of unknown nodes and further locate the area where the unknown node is located according to the identification results. This mechanism can ultimately realize real-time and efficient illegal node intrusion detection and area control.

Key words: IoT; CSI; CNN; authenticate; intrusion detection

基金项目：国家自然科学基金项目（No. 61601516）；非理想条件下异构蜂窝网络物理层安全技术（No. 61871404）。

1 引言

物联网的飞速发展大大便利了人们的生活，但由于无线信道的广播特性，在物联网节点通过接入点

（Access Point， AP）接入网络的过程中，AP易受到例如欺骗攻击和 Sybil 攻击［1］等伪造合法用户身份的攻击。具体而言，攻击者可以俘获合法节点身份并在传输过程中对数据进行监听、篡改或丢弃，以破坏数据完整性；或声称大量虚假身份从而向 AP 发送高速率的接入请求消息，消耗 AP 资源，造成网络拥塞。

由于欺骗攻击和 Sybil 攻击对网络造成的严重损伤，针对其已经提出了多种检测方案以确保物联网安全。传统基于密钥协商、分发的解决方案，如组密钥协商［2］、自适应私钥建立［3］、一次性密钥颁发［4］等， 大多面临着公钥基础设施部署的困难以及密码算法的高复杂度问题。并且随着海量设备入网，密钥的管理与分发将会产生大量开销并造成网络延迟。与传统基于密钥的认证方案相比，物理层认证不需要密钥分发和管理，且攻击者很难模拟无线信道的物理层属性。物理层认证利用 RSS［5］、接收信号强度指示符（RSSI） ［6］、信道脉冲响应［7］、信道频率响应［8］、信道状态信息（CSI）［9］等无线信道物理层属性作为无线信道指纹来区分非法节点与合法节点。以上基于无线信道指纹的认证方案本质上是对相邻数据包的比对，由于信道估计存在误差，这些方案缺乏鲁棒性和稳健性，容易受信道波动影响。近年来，机器学习技术由于其强大的数据分析能力，已被逐渐应用于与物理层认证方案的结合，并展现出良好的鲁棒性、可靠性和认证精确性。L.Xiao 等人［10］提出了一种基于 Q 学习的物理层欺骗攻击检测系统，能够预先获得欺骗攻击检测中的最佳测试门限。并在［11］中对其进行了扩展，采用 Dyna-Q 学习来提高认证速度。这种采用机器学习技术对门限处理的方式虽进一步提高了认证精度，但由于多用户需要采用多个门限，且越多的用户区分越小， 这种技术不适用于多用户的认证，无法确保物联网节点接入安全。

为确保区域内物联网节点接入安全，本文基于环境和位置特征，提出了一种基于 CNN 的智能物理层认证机制以实现对区域范围内外来非法节点的入侵检测和身份识别。CNN 能够提取原始数据更深层次的信息，在处理分类问题时具有一定优势。本文所提出的机制分为离线训练阶段和在线测试阶段，在离线训练阶段将收集的区域范围内的 CSI 信息用作 CNN 的输入，以提取该区域的无线环境特征作为该区域内节点的身份标识。一旦有外来节点入侵，CNN 智能认证器可根据当前的 CSI 信息判断未知节点的身份类型， 并进一步判断出其所处区域和位置，从而实现在物联网区域内对未知节点实时、高效的身份鉴别，保障区域物联网节点接入安全。

2 系统模型

系统模型如图 2 所示，考虑一个区域内的物联网场景，其中假定单个 AP 和多个物联网合法节点。AP 和合法节点经过人为部署且其位置固定已知。在实际通信场景中，多个物联网合法节点如传感器节点等， 进行信息采集并通过 AP 接入网络层，在网络层进行数据编码、高层认证和传输。但在物联网节点接入 AP 的过程中，AP 可能会遭受欺骗攻击和 Sybil 攻击。将物联网场景中的整片区域划分为大小相等的网格，并将合法节点所在网格标记为绿色，其余所有的网格区域均被标记为红色，定义为如欺骗攻击节点或 Sybil 节点等非法节点可能入侵的“风险区”。其中欺骗攻击节点可以俘获合法节点身份，并在向 AP 传输数据的过程中对数据进行监听、篡改或丢弃；Sybil 节点可以声称大量虚假合法节点身份，从而向 AP 发起高速率的接入请求，占用和消耗 AP 资源，造成网络拥塞。

由于无线信道具有唯一性、时变性、空变性和互易性，物理层属性很难被伪造，以 CSI 作为无线信道指纹来标识节点身份是物理层认证的一个重要手段。

令 X 和 Y 表示发送方和接收方的信号矢量。MIMO 信道可以被建模为：

$$Y = \mathrm{H}X + N$$

其中 N 为加性高斯白噪声，CSI 则表现为信道的频率响应 H，可以由 X 和 Y 估计：

（1）

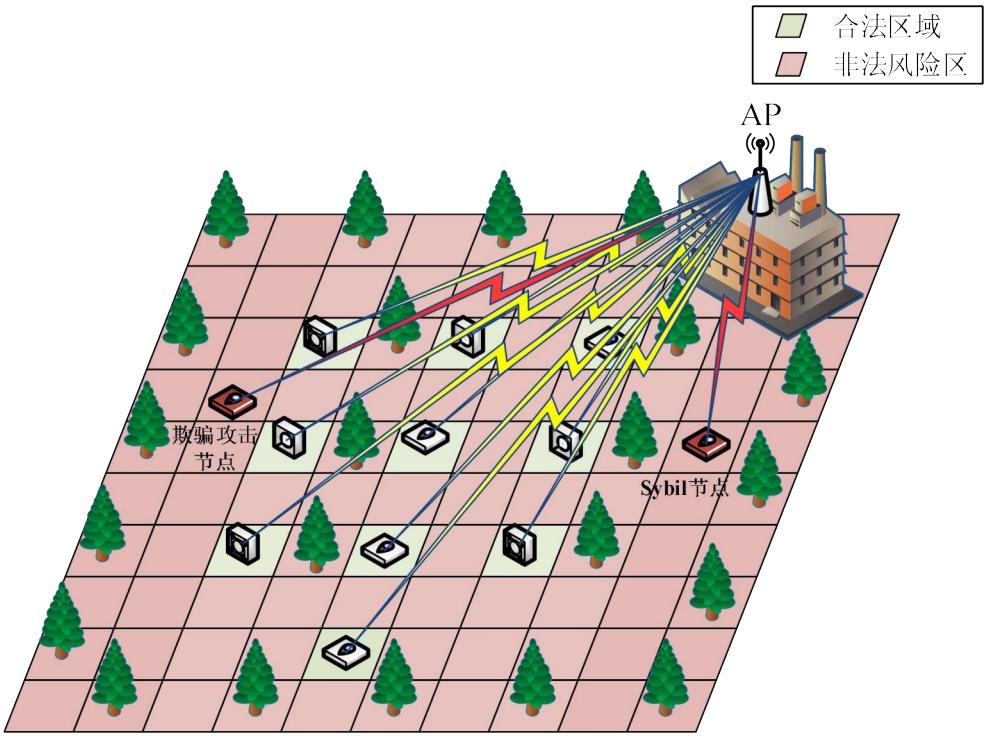


图 2　系统模型

=ΞCSI = Hˆ *Y X*

（2）

来自发送方的无线信号可能会遭受由例如多径效应、衰落、阴影和延迟失真引起的损伤，CSI 记录信号传播期间经历的信道变化，具有以下特性：

（1） **由于多径效应和环境中的信道衰落，在不同位置获取的 CSI 不同，基于 CSI 可以达到厘米级的**

高空间分辨率［12］；

（2） 与 RSS 相比，CSI 包含了更多的位置信息，反映更了深层次的信道差异。

基于以上特性，CSI 蕴含的时空特征与网格代表的身份类型进行绑定，可以被很好地用于对不同“网格”中的用户身份类型进行区分，通过识别 CSI 特征，达到识别身份的目的从而实现认证。

3　CNN 智能认证机制

3.1　智能认证机制框架

基于 CNN 的智能物理层认证机制框架如图 3 所示，物联网区域被划分为 N 个网格，在 N 个网格内分别进行 n 次采样，得到 N 个 CSI 数据组。将 N 个 CSI 数据组进行预处理，添加不同的身份标签以区分位于不同网格的节点。将预处理后的 CSI 数据输入 CNN 智能认证器中进行训练以提取深层空时环境特征，并得到 N 个合法或非法的身份类型输出结果，从而建立一个身份指纹库。最终通过身份指纹库对外来未知节点身份进行识别，达到非法节点入侵检测及区域管控的安全接入效果。整个过程分为两个阶段：离线训 练阶段和在线测试阶段。

1） *离线训练阶段*

在离线训练阶段，分别在区域内不同网格进行多次采样以获取大量合法或非法节点的 CSI 信息。在输入 CNN 进行训练前，需要对数据进行归一化，将输入数值映射到［0，1］范围内，以提高 CNN 的学习效率。

将归一化后的不同网格内获取的 CSI 的数据进行标记，输入到 CNN 智能认证器中进行训练，以得到不同身份类型的输出，从而建立一个较为完备的身份指纹库和稳健的智能物理层认证机制，用于在线测试阶段 对未知节点的身份认证。

2） *在线测试阶段*

在在线测试阶段，未知节点闯入区域内，向 AP 发送接入请求。收集其 CSI 信息输入已经训练完备的智能物理层认证机制中，从而实时、高效

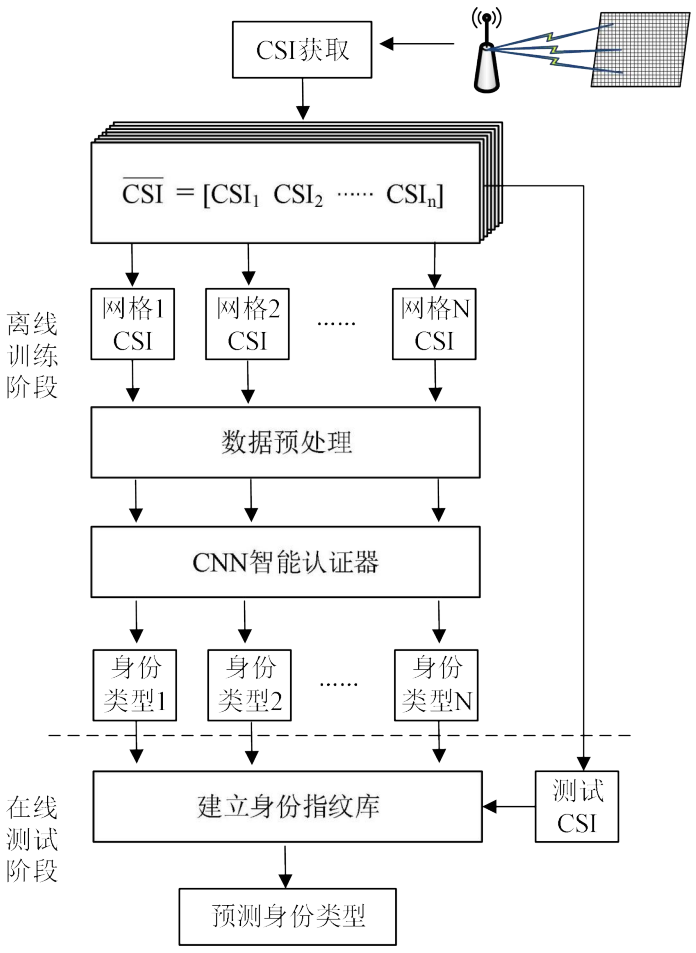


图 3　基于 CNN 的智能物理层认证机制

地判别出未知节点身份类型，检测其是否为非法入侵者。在此阶

段的数学表达如式（3）所示：

C't =

f（H（t））

（3）

其中，C't 表示输出的认证身份类型，在本文中，定义了 0-99 共 100 个身份类型，包括 10 个合法身份类别和 90 个非法身份类别。f（·）为本文所提出的可以对用户身份类型进行有效识别的智能物理层认证机制；H（t）表示在 t 时刻的信道观测。

3.2　CNN 智能认证器设计

CNN 智能认证器是智能物理层认证机制的重要组成部分，其结构设计如图 5 所示。

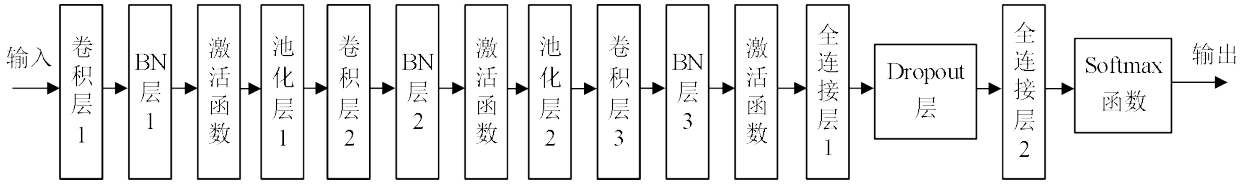


图 5　CNN 智能认证器结构

本文提出的 CNN 智能认证器包含了 3 个卷积层、2 个池化层和 2 个全连接层。其中，在每个卷积层后都跟有一个批归一化（Batch Normalization，BN）层和一个 ReLU 激活函数。虽然在输入网络前已经对数据进行了归一化，但数据进入网络后进行了矩阵运算和非线性变换，这就使得其分布可能会发生很大的变化。BN 层可以在网络训练的过程中对一批样本的同一维度特征做归一

化处理，从而加快训练速度，提高训练精度。在训练过程中，由于数据存在抽样误差或训练数据不足，可能会导致过拟合。具体表现为模型在训练时表现良好，有着较高的预测准确率，但在测试时预测准确率很低。为了防止发生过拟合现象，在两个全连接层间引入了一个 dropout 层。在最后一个全连接层后连接了一个 softmax 函数激活的输出层，softmax 函数对输出结果进行归一化，将所有输出数值映射到［0，1］范围之间。

由图 5 可以看出，卷积层、池化层、激活函数以及全连接层是 CNN 智能认证器不可或缺的组成，下面分别对 CNN 卷积层、池化层、激活函数以及全连接层进行具体讲述。

（1） *卷积层*

卷积层是 CNN 的核心，其引入使 CNN 具有以下特征：

• 稀疏交互：CNN 绝大部分待训练参数在全连接层，稀疏交互是指 CNN 输入层与最后的全连接层之间的“间接连接”是非全连接的。通过多次卷积可以使输入分成各种“小区域”用作全连接层的输入，从而

找出一种合理的间接连接方式，并减少全连接层的输入参数。

• 参数共享：是指在一个模型的多个函数中使用相同的参数。传统神经网络在计算一层的输出时，权重矩阵的每一个元素只使用一次，当它乘以输入的一个元素后就再也不会用到了。而在 CNN 中，卷积核的每一个元素都作用在输入的每一位置上。卷积运算中的参数共享保证了只需要学习一个参数集合，而不是 对于每一位置都需要学习一个单独的参数集合，大大提升了网络效率，降低了网络复杂度。

• 等变表示：CNN 参数共享的特殊形式使得 CNN 有等变表示的性质。等变表示是指一个函数能够满足输出以同样的方式随着输入的改变而改变。

在卷积层中，数据的卷积计算将通过卷积核实现。卷积层中每个神经元的卷积计算由公式（4）给出。

$a（l+1）（i，j）=$

$f（［a（l）\otimes wl（i，j）+\Xi］）$

（4）

$\iota Kl\ gl\ gl$

$=f（\sum\sum\sum［a（l）（s\cdot i+x，s\cdot j+y）wl（x，y）］+\Xi）$

$l\ k\ l\ l\ k\ k=1\ x=1\ y=1$

其中，$（i，j）\in\{0，1，\cdots，nl+1\}$，$nl+1$

$=nl+2\ pl-ml+1$ 。

sl

$a（l）$ 表示第 l 层神经元的输出；$f\ l$ （·）表示第 l 层神经元的激活函数；$nl\times nl$ 表示第 l 层神经元的输出维

度；$m\times m$ 表示第 l 层中卷积核的尺寸；wl 表示第 l 层的第 k 个卷积核；K 表示第 l 层中卷积核的数量；

l1k l

gl 表示第 l 层中神经元的个数；sl 和 pl 分别表示第 l 个卷积层中的卷积步长和填充大小；Ξ 表示阈值矩阵。其中填充是指卷积核在进行卷积运算时，对输入矩阵四周填充 0 元素、1 元素或相同元素（即 0 填充、1 填充和 same 填充），以使卷积后的输出矩阵维度与输入矩阵维度保持一致；步长则规定了卷积核每次移动的固定距离。

（2） *线性整流函数*（Rectified Linear Unit, ReLU）

ReLU 函数是在 CNN 中常用的激活函数，定义了在经过线性变换 $W（l）x+b（l）$ 后的神经元的非线性输出结果，其中 $W（l）$ 和 $b（l）$ 为第 l 层的权重矩阵和偏置项。ReLU 函数表达式如公式（5）所示。

$f（x）=\max（0，x）=\max（0，W（l）x+b（l））$

（5）

（3） *池化层*

池化层使用某个区域的总体特征来代替此区域的网络输出。池化层能够实现下采样，即通过对数据进 行分区采样以降低矩阵维度，减少计算量，降低网络复杂度。常见的池化方式有最大池化和平均池化，即 计算区域内最大值或平均值作为该区域池化后的值。最大池化和平均池化计算方式可以分别由公式（6）

和公式（7）表示：

$a（l+1）（i，j）=\max（al（s\cdot i+x，s\cdot j+y））$

（6）

k， x， y k l l

（l +1）

1 *Kl* g*l* g*l*

（l ）

a （i， j） =

Kl

fl fl

∑ ∑ ∑ ［ak

k =1 x=1 y =1

（sl · i + x， sl · j + y）］

（7）

（4） 全连接层

全连接层连接所有要素，将局部特征合并为全局特征，并将输出值发送到分类器，实现对输入的分类。

4 实验设计及结果分析

4.1 CSI 的获取

在本研究中，使用了准确定性无线信道产生器（Quasi-Deterministic Radio Channel Generator，QuaDRiGa）[13] 仿真模型来模拟产生信道环境，获取 CSI。QuaDRiGa 仿真模型是由 WINNER 信道模型发展而来，相比 WINNER II [14] 信道模型，QuaDRiGa 有着更详细的仿真场景几何拓扑参数输人、连续时间演进特性以及 3D MIMO 特性。另外，QuaDRiGa 仿真模型还引入了 Drifting 模型，能够实现移动终端在一个片段（segment）内移动时，多径功率、时延、发射角与到达角等小尺度参数在微小时间间隔内的平滑演进 [15]。文献 [15] 基于实测数据验证了 QuaDRiGa 仿真模型大尺度参数统计特性和时间演进特性的准确性。

本次实验模拟了一个 20m×20m 的区域，其中合法节点位置固定已知，AP 位于此区域中心正上方 5m 处。将此区域划分为 100 个 2m×2m 的网格，在每个网格内进行 500 次采样，AP 获取每个网格内的 CSI 数据，并入数据集。具体仿真参数设置如表 1 所示：

通过 QuaDRiGa 信道模型，可以得到一个 64×52×50000 的 CSI 复数矩阵。将复数矩阵的实部和虚部进行拆分，并合并为一个 62×52×2×50000 的四维矩阵，以方便用作 CNN 的输入。

表 1 仿真参数设置

名称	参数设置
信道模型中心频率带宽	3GPP_38.901_UMa_NLOS
子载波个数天线配置	
	2.4 GHz 3 MHz
	52
极化方向	AP:64 根天线节点:单天线垂直极化

4.2 数据预处理

在数据输入 CNN 前，首先需要对数据进行归一化。采用最大-最小归一化方式对原始数据进行线性变换，将数据限制在［0，1］区间内，以提高学习效率，加快收敛速度。最大-最小归一化数学表达如式（8）所示：

y =（x - max）（max- min）

（8）

在归一化后，需要对数据添加标签。将每个网格中的 CSI 数据归为一类，共分为 100 个类别，即 10 个合法身份类别（分别为 Alice1-Alice10）和 90 个非法身份类别（Eve1-Eve90）。如此，在识别未知节点身份后，可直接根据其身份类型确定其所处的网格位置。

4.3 CNN 网络及训练参数设置

CNN 在训练时采用了反向传播算法 [16]。首先通过前向传播获得各层激活输出，在更新参数时，采用梯度下降算法（例如 adam 算法、sgdm 算法、rmsprop 算法等），从最后一层依次向前更新，即实现反向传播的过程。CNN 网络参数设置如图 6 所示：

在每个卷积层都使用 same 填充的方式，步长均设置为 1。池化层均采用最大池化的方式，步长为 2。图 6 显示了数据进入每层后输出的矩阵维度。输入维度为 64×52×2，在卷积层 1，由 8 个尺寸为 3×3×2 的

卷积核对输入数据进行卷积运算，再经过归一化和非线性激活，得到维度大小为 64×52×8 的输出。到达池化层 1 后，池化层对数据进行降维处理并提取主要特征，输出维度为 32×26×8。之后数据经过卷积层 2、池化层 2、卷积层 3 到达全连接层，最终得到维度大小为 1×1×100 的输出。

将数据集按照 8：1：1 的比例划分为训练集、

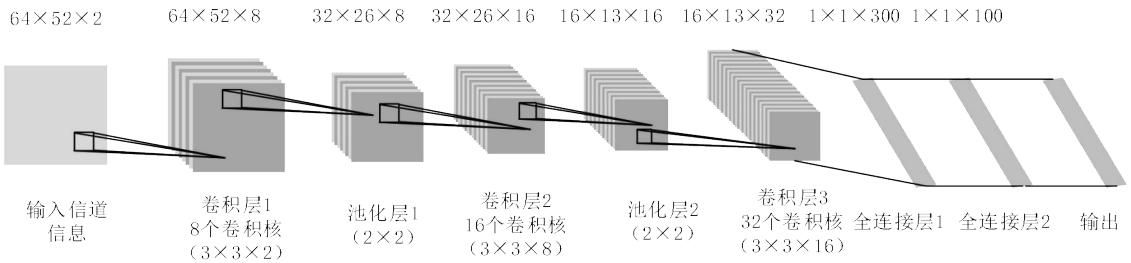


图 6　CNN 网络参数设置

验证集和测试集，采用 adam 算法进行训练。在网络训练时，几个重要训练参数的设置如表 2 所示。

表 2　训练参数设置

参数	数值
Mini Batch Size	200
iteration	200
epoch	10
Initial Learn Rate	0.001
Learn Rate Drop Factor	0.5
Learn Rate Drop Period	5
Validation Frequency	200

其中批次（batch）是指在训练集中选择的一部分用来权值更新的样本的大小；迭代（iteration）是指

用一个 batch 完成一次权值更新的过程；一个 epoch 是指对训练集中所有数据进行一次完整训练；初始学习率是训练时的重要参数，学习率决定了训练的效率，若学习率太低，训练将花费很长时间，但如果学习率太高，训练可能会陷入次优结果。学习率可以在训练过程中自适应调整，随着训练过程中 epoch 的增加，训练结果离最优值越来越近，可以将学习率适当减小从而继续训练过程。Learn Rate Drop Factor 是一个用于对学习率进行调整的乘数因子，Learn Rate Drop Period 则决定在几个 epoch 后对学习率进行调整。在本次实验中，将其设定为在每 5 个 epoch 后将学习率降为原来的一半。Validation Frequency 用于决定在多少次 iteration 后使用验证集对模型进行验证。

4.4　仿真结果分析

本文提出的基于 CNN 的智能物理层认证机制的认证性能如图 7 所示。初始学习率为 0.001，在第 5 个epoch 后学习率下降为 0.0005。训练精度和验证精度随着 epoch 和 iteration 的增加不断升高，最终趋于平稳。在训练过程结束后，使用测试集对训练模型进行测试，测试精度由公式（9）计算得到。

accuracy

$= sum（Y$

pred

$= Y）$

（9）

test num

test

其中 *Ypred* 为预测结果，*Y* 为真实结果，num_{test} 为测试集总数据数量。即测试精度为预测成功的结果占测试集的比例，最后得到的测试精度为 99.12%。

损失函数反映了训练模型对数据的拟合程度，即模型预测值与真实值之间的差距。在使用深度学习网络处理分类问题时，训练损失和验证损失数值多由交叉熵（cross-entropy）损失函数来衡量，损失值越小，代表模型越精确。交叉熵损失函数可以由公式（10）表示：

$_{n}C = 1 \sum [Y \ln（Y）+（1 - Y）\ln（1 - Y）]$

pred pred

（10）

n i=1

图 8 显示了 CNN 损失函数变化情况，可以看到在训练过程中的训练损失和验证损失随 epoch 和 iteration

的增加而快速下降，最终逐渐收敛于 0，训练模型与数据拟合程度较高。

由上述分析可以看出，本文提出的基于 CNN 的智能物理层认证机制可以达到较高的认证精度，且在模型训练结束后，能快速识别未知节点身份，并根据分类预测结果定位其所在的网格区域位置，

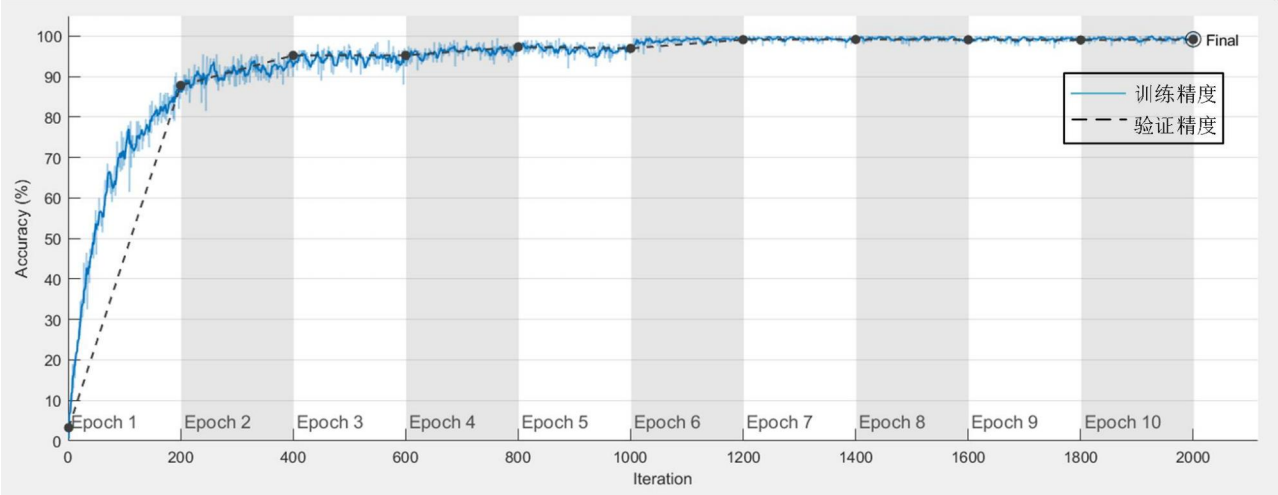


图 7　认证精度

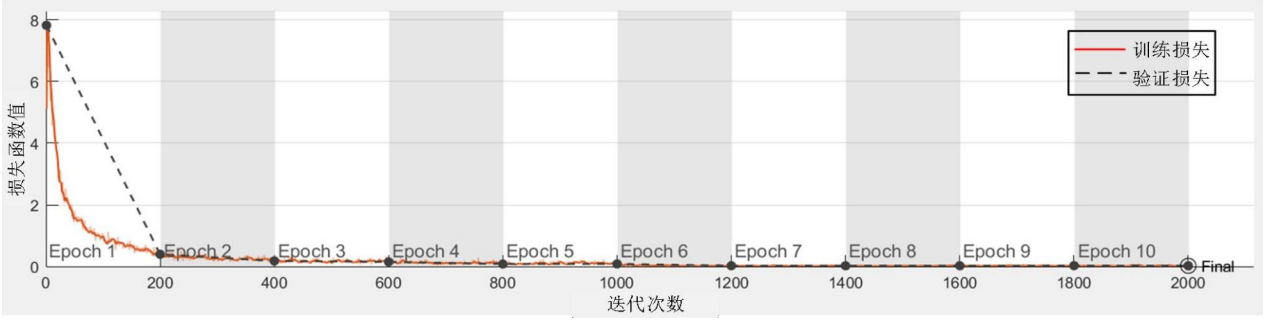


图 8　CNN 损失函数变化

从而实现 对非法入侵节点的实时检测。

由于信道中存在噪声，为了进一步模拟噪声对网络训练模型认证精度的影响，分别利用在不同信噪比 情况下取得的数据对模型进行训练得出其测试精度。图 9 中显示了 SNR 为 0dB、5dB、10dB、15dB、20dB 时的测试精度，随着 SNR 的增大，信号特征更易于提取，测试精度不断上升。

不同的梯度下降算法会产生不同的训练结果，图 10 对比了分别采用 adam 算法、sgdm 算法、rmsprop 算法时随 epoch 变化的验证精度和验证损失。其中 rmsprop 算法基于权重梯度最近量级的均值为每一个参数适应性地保留学习率；sgdm 算法在梯度下降的过程中加入了惯性，使得梯度方向不变的维度上速度变快，梯度方向有所改变的维度上的更新速度变慢，从而可以加快收敛并减小震荡；Adam 是一种可以替代传统随机梯度下降过程的一阶优化算法，它能基于训练数据迭代地更新神经网络权重，它集合了适应性梯度算法和 rmsprop 算法的优点。可以看出，在本实验中，adam 算法性能略优于 sgdm 算法和 rmsprop 算法。

5　结束语

图 10 使用不同梯度下降算法的 CNN 验证精度和验证损失

为了确保区域物联网节点接入安全，本文提出了一种基于 CNN 的智能物理层认证机制。该机制本质上基于位置信息，提取 CSI 作为无线信道指纹以区分位于不同位置的发送节点，分为离线训练和在线测试两个阶段。将收集的不同网格内的 CSI 信息输入 CNN 智能认证器中进行离线训练，形成完备的身份指纹库，以完成在线测试阶段对未知节点身份的识别，并进一步判断出其所处区域和位置，实现物联网区域入 侵检测和安全接入。实验结果表明，训练后的智能物理层认证机制能够达到 99.12% 的测试精度，可以实现对未知节点身份的精准识别，并进一步根据识别结果定位未知节点所在的区域范围，实现即时、高效

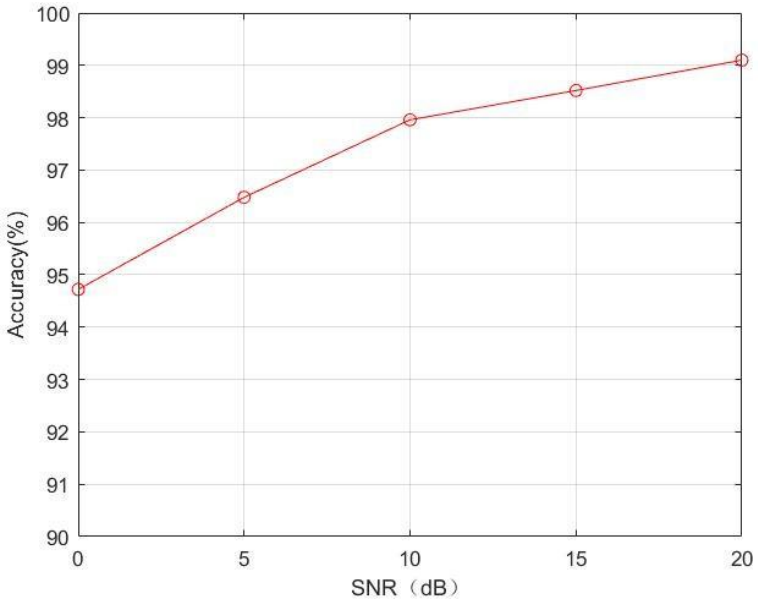
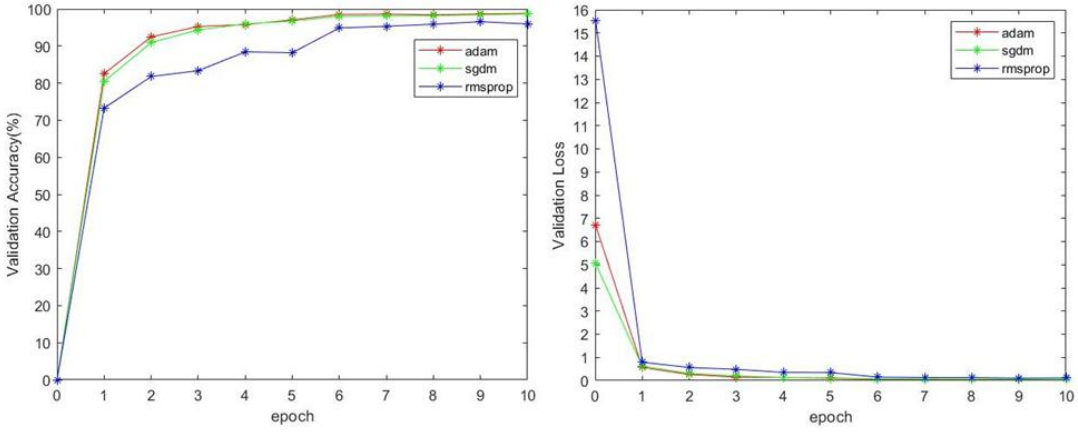


图 9　不同 SNR 下的 CNN 训练精度



的非 法节点入侵检测及区域管控。

参考文献:

[1] Douceur J R. The Sybil Attack［C］//Peer-to-Peer Systems，First International Workshop (IPTPS 2002)，Mar，2002：251-260.

[2] Halford T R, Courtade T A, Chugg K M, et al. Energy-Efficient Group Key Agreement for Wireless Networks［J］. IEEE Transactions on Wireless Communications，2015，14(10)：5552-5564.

[3] Zhao H, Zhang Y, Huang X, et al. An Adaptive Secret Key Establishment Scheme in Smart Home Environments［C］//ICC 2019 - 2019 IEEE International Conference on Communications (ICC)，Shanghai，China，2019：1-6.

[4] Sukma N, Chokngamwong R. One time key Issuing for Verification and Detecting Caller ID Spoofing Attacks［C］//International Joint Conference on Computer Science & Software Engineering，Nakhon Si Thammarat，2017：1-4.

[5] Yang J, Chen Y, Trappe W, et al. Detection and Localization of Multiple Spoofing Attackers in Wireless Networks［J］. IEEE Transactions，Parallel and Distributed Systems，2013，24(1)：44-58.

[6] Li W, Zhang D. RSSI Sequence and Vehicle Driving Matrix Based Sybil Nodes Detection in VANET［C］//2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN), Chongqing, China，2019：763-767.

[7] Liu F, Wang X, Primak S，A two dimensional quantization algo rithm for CIR-based physical layer authentication［C］//Proceedings of the Global Communications Conference，IEEE ICC，2013：4724－4728.

[8] Xiao L, Reznik A, Trappe W, et al. PHY-Authentication Protocol for Spoofing Detection in Wireless Networks［C］//2010 IEEE Global Telecommunications Conference GLOBECOM 2010，Miami，Florida，USA，2010：1-6.

[9] Jiang Z，Zhao J，Li X Y, et al. Rejecting the attack: Source authentication for Wi-Fi management frames using CSI Information［C］//2013 Proceedings IEEE INFOCOM，Turin，2013：2544-2552.

[10] Xiao L, Li Y, Liu G, et al. Spoofing detection with reinforcement learning in wireless networks［C］//IEEE GLOBECOM，San Diego，

CA，USA，2015：1-6.

[11] Xiao L, Li Y, Han G, et al. PHY-Layer Spoofing Detection With Reinforcement Learning in Wireless Networks［J］. IEEE Transactions on Vehicular Technology，2016，65（12）：10037-10047.

[12] 廖润发．基于无线信道特征和智能算法的物理层安全技术研究［D］. 电子科技大学，2019.
Liao R F. Research on physical layer security technology based on wireless channel characteristics and intelligent algorithms［D］. University of Electronic Science and Technology of China，2019.

[13] Stephan J, Leszek R, Lars T, Quasi Deterministic Radio Channel Generator User Manual and Documentation［OL］. http://www.quadriga-channel-modle. de，2019.

[14] KYOSTI P, MEINILA J, HENTILAL, et al. Dl. 1. 2: WINNER II Channel Models［R/OL］. http://projects
，
itiativeceltic-in. org/winner+/ phase 2- model，html，2007.

[15] 李树, 赵雄文, 王琦．5G 毫米波 QuaDRiGa 平台信道仿真与验证研究[J]. 电波科学学报，2017，
32（2）：176-183. LI S, ZHAO X W, WANG Q. Simulation and validation for 5G millimeter wave channels model through QuaDRiGa platform［J］. Chinese journal of radio science，2017，32（2）：176-183.

[16] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. nature，2015，521（7553）：436-444.

[作者简介]

李兴璐（1996-），女，战略支援部队信息工程大学在读硕士，主要研究方向为物理层认证。

黄开枝（1973-），女，国家数字交换系统工程技术研究中心教授、博士生导师，主要研究方向为通信　信号处理及无线通信安全。

王少禹（1993-），男，战略支援部队信息工程大学在读博士，主要研究方向为无线通信安全、物理层　认证。

许晓明（1988-），男，博士学位，国家数字交换系统工程技术研究中心副研究员，主要研究方向为网络空间安全、无线移动通信、无线内生安全。

张波（1993-），男，博士学位，国防科技大学讲师，主要研究方向为通信安全。

物联网准静态场景下基于智能超表面的密钥生成方法

郝一诺，金梁，黄开枝，肖帅芳
信息工程大学，河南 郑州 450002

摘　要：针对物联网场景中准静态信道变化缓慢、密钥生成速率低的问题，本文提出了一种基于智能超表面的密钥生成方法。首先，利用智能超表面构造快速变化的信道并从中提取密钥；然后，在相干时间内优化密钥生成和数据传输的功率分配，实现了“一次一密”安全传输；最后，仿真验证了该方法的有效性。仿真结果表明，在信噪比为20dB、反射单元为60个的条件下，该方法的密钥生成速率与无智能超表面方法相比提高了0.47倍以上，并且随着反射单元个数和变化次数的增加，密钥生成速率会有进一步的提高。

关键词：物理层安全、密钥生成、智能超表面、准静态信道

Key Generation Method based on Intelligent Reflecting Surface in Quasi-static Scene of Internet of Things

Hao Yi-nuo, Jin Liang, Huang Kai-zhi, Xiao Shuai-fang
Information Engineering University, Zhengzhou Henan 450002, China

Abstract: Aiming at the problems of slow change of quasi-static channel and low key generation rate in IoT scenarios, this paper proposes a key generation method based on intelligent reflecting surface. First, use the intelligent reflecting surface to construct a rapidly changing channel and extract the key from it. Then, optimize the power distribution of key generation and data transmission within the coherent time, and realize the "one time one secret" secure transmission. Finally, the simulation verified this effectiveness of the method. The simulation results show that under the conditions of a signal-to-noise ratio of 20dB and 60 reflection units, the key generation rate of this method is increased by more than 0. 47 times compared with the non-intelligent reflecting surface method. As the number of reflection units and its changing-times increases, the key generation rate will be further improved.

Key words: physical layer security; key generation; intelligent reflecting surface; quasi-static channel

1　引言

物理层密钥生成技术以信道状态信息（Channel State Information， CSI）作为密钥源，密钥更新速度依赖于信道信息的快速变化。然而，在智能家居、环境监测等典型的物联网场景中，信道变化往往十分缓慢，严重制约了密钥生成速率。这类通信节点和物理环境均固定不变的信道被定义为“准静态信道”[1]。在准静态环境中，多径相位不变、节点不移动导致多普勒效应不明显，密钥源随机性差，密钥生成速度低，难以满足高速通信中的加密需求。要解决准静态环境下密钥生成速率低的问题，其本质在于提高密钥源的随机性。

目前，针对准静态信道中提高密钥速率的研究已经取得了一些进展，主要分为三类：（1）借助频域、空域等多维资源，扩大密钥源信息，例如：文献［2］提出了多进多出（Multiple-In Multiple-Out， MIMO）场景下利用随机波束成形对信道进行随机加权处理的密钥生成方法，利用权值的随机性提高密钥源的随机性；文献［3］利用协作干扰的方法构建等效信道，提高准静态环境下

基金项目：国家自然科学基金资助项目（No. 61871404）。
Foundation item: The National Natural Science Foundation of China（No. 61871404）.

密钥源的随机性；（2）利用人工随机源提高密钥源随机性，例如：文献［4］提出了一种基于人工随机源生成密钥的方法；文献［5］利用MIMO接收信号进行人工随机源设计，提高密钥生成速率。（3）利用中继协作辅助密钥生成，例如：文献［6］研究了双向中继系统中密钥生成的问题，提出了4种基于放大转发的密钥生成方案。文献［7］在文献［6］的基础上，研究了多径环境下基于相对幅度和相对相位的密钥生成方案。文献［8］基于中继节点辅助和安全网络编码，将中继节点与合法通信双方之间的信道引入密钥源，提高了密钥源的随机性。

然而，目前针对于准静态环境密钥生成方案存在一些局限性，例如：借助多维资源扩大密钥源信息的方法依赖大量的频域、空域资源，在资源受限的环境中无法实现[8]；基于人工随机源和中继辅助的方法随机性有限，密钥协商复杂度高等。

近年来，随着射频微机电系统（Radio Frequency Micro-Electro-Mechanical Systems，RFMEMS）的快速发展和可编程可重构超表面的广泛应用，智能超表面（Intelligent Reflecting Surface，IRS）作为一种低功耗、高效率的数据通信技术，被越来越多的应用于无线数据传输中，利用超材料技术定制无线传输环境以获得传输或安全增益的方法越来越多的被提出[9]。IRS基于电磁材料，由无源阵列结构组成，该结构能够以低功耗连续或离散的调整每个无源元件的相位。IRS中的无源元件又被称为反射单元，每个反射单元的反射系数都自主可控、代表一个独立的反射链路[13]。大量文章证明[9-12]，在物联网场景中将基站（Base Station，BS）与IRS进行有线连接的联合设计方法，可以显著增强光谱和能量效率。

基于以上分析，本文提出了一种面向物联网准静态场景的IRS辅助密钥生成方法。首先，在基于IRS的物联网准静态场景中进行信道探测，得到IRS信道的信道状态信息；然后，多次改变IRS的反射系数构造快速变化的信道，每次变化后都进行信道探测，最后从IRS信道状态信息和反射系数中提取密钥。本文仿真分析了不同信噪比、反射单元变化次数和反射单元个数对IRS辅助密钥生成方法的密钥生成速率的影响。在此基础上提出了“一次一密”功率分配算法。在相干时间内对密钥生成和数据传输进行功率分配，保证密钥长度不小于数据长度，实现“一次一密”安全传输。最后，仿真分析了不同信噪比下，IRS辅助密钥生成方法可实现“一次一密”安全传输的最大速率。

2 系统模型

2.1 信道模型

本小节针对面向物联网准静态场景的IRS辅助密钥生成系统进行建模，如图2.1所示。在该模型中，存在基站BS、用户Alice和Bob、被动窃听者Eve以及一个智能超表面IRS。BS与IRS控制器通过有线链路进行连接，BS通过IRS控制器控制IRS上每个反射单元的反射系数的动态变化。BS、Alice和Bob在TDD模式下传输窄带信号，保证了信道的互异性。其中，BS设有M根天线，用户与窃听者均设为单天线，IRS设有N个反射单元，每个反射单元可控且独立变化。

在IRS辅助密钥生成系统中，本文考虑如下三类信道：直接信道、间接信道和反射信道[9]。直接信道代表BS与合法用户（Alice或Bob）之间的信道h_A或h_B，$h_A, h_B \in \mathbf{C}^{M\times 1}$；间接信道代表BS与IRS之间的信道$h_I \in \mathbf{C}^{N\times 1}$，反射信道代表IRS与合法用户（Alice或Bob）之间的信道h_1或h_2，$h_1 = \left[h_{11}, h_{12}, \cdots . h_{1N}\right]^T \in \mathbf{C}^{N\times 1}$，$h_2 = \left[h_{21}, h_{22}, \cdots . h_{2N}\right]^T \in \mathbf{C}^{N\times 1}$。对于窃听者而言，$h_{EI} \in \mathbf{C}^{N\times 1}$表示Eve和IRS之间的信道，$h_{EB} \in \mathbf{C}^{M\times 1}$表示Eve和BS之间的信道。

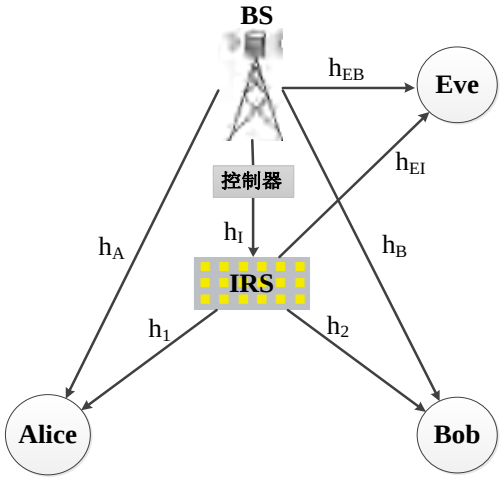


图2.1 IRS辅助密钥生成系统模型

考虑准静态环境的信道特性，本文将信道建模为准静态块衰落模型，即在相干时间内信道状态信息保持不变，在相干时间之间为衰落信道，信道状态信息发生变化。不同信道之间具有独立性，上下行信道之间具有互易性。假设每种信道均服从均值为0、方差为σ^2的高斯分布，Alice、Bob、Eve接收到的噪声是独立同分布的高斯白噪声。

对于窃听者Eve来说，为了不暴露自身位置，Eve只进行被动窃听，即只对信息进行窃听，不对信息传输过程进行干扰，不参与数据传输和密钥生成过程。由于窃听者与用户之间的距离满足几倍波长以上，信道之间互不相关，所以窃听者不能根据自身的信道状态信息推测出合法用户与BS和IRS之间信道的信道状态信息。

2.2 IRS反射系数模型

IRS的反射系数可以用一个对角矩阵Θ表示[14]，即$\Theta = diag(\theta) \in \mathbf{C}^{N\times N}$。其中，$\theta = \left[\theta_1, \theta_2, \cdots, \theta_N\right]^T \in \mathbf{C}^{N\times 1}$，$\theta_m \in \Phi$，$\Phi$表示第n个反射单元的反射系数$\theta_n$的取值范围。

本文考虑反射系数连续变化的情况，在这种情况下：

$$\Phi = \left\{\theta_n \middle| \theta_n = e^{j\varphi_n}, \varphi_n \in [0, 2\pi)\right\} \tag{2.1}$$

其中，θ服从均值为0的高斯分布，即$\theta \sim N(0, \sigma_\theta^2)$。

3 IRS辅助密钥生成方法

基于上述模型，本文提出一种物联网通信中基于IRS的准静态环境下的密钥生成方法。首先，由BS、Alice和Bob互发导频，经过信道探测之后，Alice和Bob分别获得各自与BS和IRS之间信道$\{h_A, h_B, h_I, h_1, h_2\}$的估计值；然后，BS控制IRS进行L次反射系数的随机变化，每次变化后，BS都向用户发送新的导频$x_i, i \in [1, \mathrm{L}]$。每次接收到导频后，用户都可根据已知信息获得各自对于当前状态下IRS反射系数Θ_i的估计值；最后，Alice和Bob利用已知的信道状态信息和多次信道探测后得到的多个Θ_i的估计值进行密钥生成。由于Eve所获得的信道状态信息与合法用户不同，所以该密钥是安全的。

同时，本文方法对该通信模型的密钥生成和数据传输进行联合设计。由于在相干时间内，密钥生成和数据传输时间分别影响着密钥和数据的长度，因此本文根据数据传输速率和密钥生成速率进行功率分配，保证密钥长度不小于数据长度，实现了理论上的“一次一密”。本文研究了不同信噪比下，IRS辅助密钥生成方法可实现“一次一密”的最大速率。

3.1 密钥生成方法

在本文所提的密钥生成方法中，首先由BS与合法用户之间进行多次信道探测。根据文献［15］的结论，在多次信道探测后，Alice可以获得信道$\{h_A, h_B, h_I, h_1, h_2\}$的估计值$\{\tilde{h}_{AA}, \tilde{h}_{BA}, \tilde{h}_{IA}, \tilde{h}_{1A}, \tilde{h}_{2A}\}$，Bob可以获得信道$\{h_A, h_B, h_I, h_1, h_2\}$的估计值$\{\tilde{h}_{AB}, \tilde{h}_{BB}, \tilde{h}_{IB}, \tilde{h}_{1B}, \tilde{h}_{2B}\}$。

在获得全部信道信息后，BS控制IRS的反射单元系数多次随机变化，每次变化后，BS向合法用户发送一次导频，合法用户接收到导频后可由已知信息得到当前的IRS反射系数。

以反射系数第i次变化为例，假设Θ_i代表第i次变化后IRS的反射系数，$x_i \in \mathbf{C}^{M\times 1}$代表第i轮中BS发送的导频信息，则此时Alice接收到的信号为[16]：

$$y_{Ai} = \left(h_A + h_1 \Theta_i h_I\right) x_i + n_{Ai} \tag{3.1}$$

采用迫零算法，则Alice对于信道的估计值为：

$$\tilde{h}_{Ai} = \frac{x_i^T}{\|x_i\|^2} y_{Ai} = \left(h_A + h_1 \Theta_i h_I\right) + \frac{x_i^T}{\|x_i\|^2} n_{Ai} \tag{3.2}$$

由于在最初的多次信道探测后，Alice已经得到了$\{h_A, h_I, h_1\}$的估计值$\{\tilde{h}_{AA}, \tilde{h}_{IA}, \tilde{h}_{1A}\}$，因此Alice可以得到此时IRS反射系数$\Theta_i$的估计值$\tilde{\Theta}_{iA}$。同理，Bob也可得到$\Theta_i$的估计值$\tilde{\Theta}_{iB}$。因为$\Theta_i$是随机变化的，所以$\Theta_i$与信道无关，$\Theta_i$与$\Theta_k (\mathrm{k} \neq \mathrm{i})$也无关。因此，每次Alice和Bob在反射系数随机变化后进行信道估计，都可得到一组新的共享随机源$(\tilde{\Theta}_{iA}, \tilde{\Theta}_{iB})$。

在本文所提的信道模型中，间接信道h_I和反射信道$\{h_1, h_2, h_{EI}\}$为小尺度衰落信道且动态变化，随机性很高。与以上两种信道相比，直接信道$\{h_A, h_B\}$随机性很差，对密钥生成的贡献不高，因此本文所提的密钥生成方法仅考虑前两种信道。假设Θ共变化了L次，$\tilde{\Theta}_A = \left(\tilde{\Theta}_{1A}, \tilde{\Theta}_{2A}, \cdots, \tilde{\Theta}_{LA}\right)$，

$\tilde{\Theta}_B=\left(\tilde{\Theta}_{1B},\tilde{\Theta}_{2B},\cdots,\tilde{\Theta}_{LB}\right)$。Alice 对随机源 $\tilde{h}_{AA}\tilde{h}_{IA}\tilde{h}_{1A}\tilde{\Theta}_A$、Bob 对随机源 $\tilde{h}_{AB}\tilde{h}_{IB}\tilde{h}_{1B}\tilde{\Theta}_B$ 进行量化、密钥协商、隐私放大等一系列密钥生成步骤后，合法用户双方可以生成一致密钥。IRS 辅助密钥生成具体实现过程如表 3.1 所示。

表 3.1　IRS 辅助密钥生成方法实现过程

步骤 1：Alice、Bob 和 BS 分别发送导频，相互进行多次信道探测，Alice 估计出 $\left\{\tilde{h}_{AA},\tilde{h}_{BA},\tilde{h}_{IA},\tilde{h}_{1A},\tilde{h}_{2A}\right\}$，Bob 估计出 $\left\{\tilde{h}_{AB},\tilde{h}_{BB},\tilde{h}_{IB},\tilde{h}_{1B},\tilde{h}_{2B}\right\}$。
步骤 2：BS 控制 IRS 的反射单元系数多次随机变化，每次变化后，BS 向合法用户发送一次导频，合法用户接收到导频后得到当前的 IRS 反射系数。
步骤 3：Alice 利用随机源 $\tilde{h}_{AA}\tilde{h}_{IA}\tilde{h}_{1A}\tilde{\Theta}_A$、Bob 利用随机源 $\tilde{h}_{AB}\tilde{h}_{IB}\tilde{h}_{1B}\tilde{\Theta}_B$ 生成一致密钥。

3.2　密钥生成与数据传输联合功率分配

Shannon 指出[17]，实现完美保密的条件是“一次一密”，此时需满足密钥 K 的长度不小于明文的长度，即

$$H(K)\geqslant H(M) \tag{3.3}$$

其中，H 代表随机变量的熵。

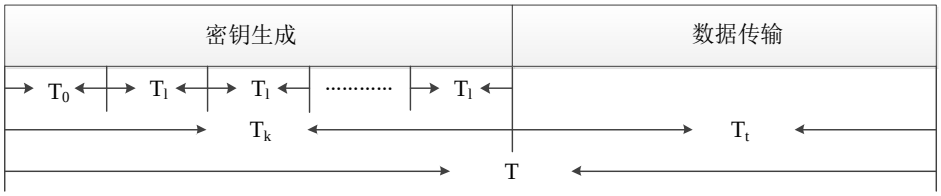


图 3.1　密钥生成与数据传输时隙分布图

对应于本文方法，如图 3.1 所示，在相干时间 T 内，设用于密钥生成的时间为 T_k，用于数据传输的时间为 T_t。在密钥生成过程中，设初始的多轮信道探测所占时间为 T_0，后续每次 Θ 变化后进行的信道探测所占时间为 T_l，Θ 共变化 L 次，密钥生成速率为 R_s，数据传输速率为 R_b，每次 Θ 变化后可提取的新的密钥长度为 n_k，则有：

$$T_k=T_0+L\cdot T_l \tag{3.4}$$

$$R_s=L\cdot n_k/\mathrm{T} \tag{3.5}$$

在“一次一密”的条件下，最优“一次一密”速率 $R_{k,t}$ 的联合最优化问题可表示为：

$$\begin{aligned}&\max R_{k,t}\\&s.t.0\leqslant T_k\leqslant T,0\leqslant T_t\leqslant T\\&L\cdot n_k\geqslant T_t\cdot R_b\\&T_t+T_0+L\cdot T_l\leqslant T\end{aligned} \tag{3.6}$$

由式（3.6）可知，用于密钥生成的时间越多，生成的密钥长度越大，但是相干时间 T 是固定的，约束了密钥的长度；用于数据传输的时间越长，相干时间内可传输的数据越多，但是为了满足“一次一密”的条件，密钥长度需不小于明文长度，约束了数据传输时间，因此该问题存在最优解。

4　IRS 辅助密钥生成方法的理论分析

4.1　安全性分析

在上述的密钥生成过程中，假设 Eve 采用与合法用户相同的模式进行密钥生成：首先，Eve 和 BS 进行多轮信道探测后，获得了自身与 BS 和 IRS 之间信道 $\left\{h_{EI},h_{EB},h_I\right\}$ 的全部信息 $\left\{\tilde{h}_{EI},\tilde{h}_{EB},\tilde{h}_{IE}\right\}$；然后，在每次 Θ 变化、BS 发送导频后，Eve 同样进行信道探测并由已知信息得到当前反射系数的估计值 $\tilde{\Theta}_{iE}$。因此，最终 Eve 可用于密钥生成的随机源为：

$$Z=\tilde{h}_{EI}\tilde{h}_{IEB}\tilde{\Theta}_E=\sum_{i=1}^{L}\tilde{h}_{EI}\tilde{h}_{IEB}\tilde{\Theta}_E\in\mathbf{C}^{1\times N} \tag{4.1}$$

对于合法用户 Alice 和 Bob 而言，Alice 的密钥源为：

$$X=\tilde{h}_{AA}\tilde{h}_{IA}\tilde{h}_{1A}\tilde{\Theta}_A=\sum_{i=1}^{L}\tilde{h}_{AA}\tilde{h}_{IA}\tilde{h}_{1A}\tilde{\Theta}_{iA}\in\mathbf{C}^{1\times N} \tag{4.2}$$

Bob 的密钥源为：

$$Y=\tilde{h}_{AB}\tilde{h}_{IB}\tilde{h}_{1B}\tilde{\Theta}_B=\sum_{i=1}^{L}\tilde{h}_{AB}\tilde{h}_{IB}\tilde{h}_{1B}\tilde{\Theta}_{iB}\in\mathbf{C}^{1\times N} \tag{4.3}$$

由于 Eve 与 Alice 和 Bob 的距离满足几倍波长以上，信道之间没有相关性，所以有

$$I\left(\tilde{h}_{AA}\tilde{h}_{IA}\tilde{h}_{1A};\tilde{h}_{EI}\tilde{h}_{IEB}\right)\approx I\left(\tilde{h}_{AB}\tilde{h}_{IB}\tilde{h}_{1B};\tilde{h}_{EI}\tilde{h}_{IEB}\right)<$$
$$<I\left(\tilde{h}_{AA}\tilde{h}_{IA}\tilde{h}_{1A};\tilde{h}_{AB}\tilde{h}_{IB}\tilde{h}_{1B}\right) \tag{4.4}$$

因此

$$I(X;Z)\approx I(Y;Z)<<I(X;Y) \tag{4.5}$$

由式（4.4）和（4.5）可知：

$$I(X;Y)=I(X;Y|Z)+I(Y;Z) \tag{4.6}$$

在从共享随机源$(X;Y)$中提取密钥时，存在部分信息是窃听方能够窃取到的，因此就导致了密钥的不安全性[18]。随着IRS反射单元的个数N逐渐增大，窃听者窃取到的共享随机源信息$I(Y;Z)$不断减小，密钥的安全性不断增加。

定理1

假设IRS反射单元个数为N，当发送总功率保持不变，IRS反射单元个数趋近于无穷大的时候，合法方的密钥源与窃听方的密钥源的互信息量为0，窃听方无法观察到任何共享随机源互信息。

证明：

假设

$$\tilde{Z}=h_{EB}+h_I\Theta_i h_{EI} \tag{4.7}$$

则窃听方根据接收到的信号所得到的信道估计值为

$$Z=\tilde{Z}+n_e \tag{4.8}$$

$Y\rightarrow\tilde{Z}\rightarrow Z$构成马尔可夫链，根据文献［19］定理1可得

$$R_{Y\tilde{Z}}=0 \tag{4.9}$$

$$I\left(Y;\tilde{Z}\right)=0 \tag{4.10}$$

其中，$R_{Y\tilde{Z}}$为协方差矩阵。根据马尔可夫链的性质可得

$$I(Y;Z)=0 \tag{4.11}$$

因此，在定理1的情况下，窃听者无法获取任何共享随机源互信息。证毕。

所以

$$\lim_{N\rightarrow\infty}I(X;Z)=\lim_{N\rightarrow\infty}I(Y;Z)=0 \tag{4.12}$$

本文将在第五节对定理1进行仿真验证。

4.2 密钥生成速率

假设Alice和Bob生成密钥K的可达密钥速率为R_s，则对于$\forall\varepsilon>0$和充分大的n，R_s和K需要满足以下关系：

$$\left.\begin{aligned}&\Pr I\{K\neq K'\}\leqslant\varepsilon\\&\frac{1}{n}I(K;\gamma)\leqslant\varepsilon\\&\frac{1}{n}H(K)\geqslant R_s-\varepsilon\\&\frac{1}{n}\log|\kappa|\leqslant\frac{1}{n}H(K)+\varepsilon\end{aligned}\right\} \tag{4.13}$$

其中，κ表示密钥空间，$K,K'\in\kappa$，$|\kappa|$是κ的基数即密钥空间的大小，γ代表密钥协商过程中发送的协商信息，n为密钥源符号长度。

据文章[20]可得，满足公式（4.13）的可达密钥速率为：

$$R_s=\lim_{\Delta\rightarrow 0}\frac{1}{T}I\left(\tilde{h}_{1A}\tilde{h}_{IA}\tilde{h}_{2A}\tilde{\Theta}_A;\tilde{h}_{2B}\tilde{h}_{IB}\tilde{h}_{1\mathrm{B}}\tilde{\Theta}_B\middle|\tilde{h}_{EI}\tilde{h}_{IE}\tilde{\Theta}_E\right) \tag{4.14}$$

5 仿真结果

为了验证所提方法的可行性和有效性，本文在MATLAB R2016a环境下进行了一系列的仿真实验。其中，式（4.14）中的互信息由ITE工具箱计算得出，并假设收发端均进行数字信号处理，采用16 bit量化。本文使用蒙特卡洛方法进行10000次实验，每次试验都随机产生信道和噪声数据，确保了实验的准确性。

5.1 密钥生成速率与信噪比的变化关系

本小节在相同环境下，对IRS辅助密钥生成方法、放大转发（Amplify and Forward， AF）方法和无IRS加入的一般密钥生成方法进行了对比，仿真结果如图5.1所示。由图5.1可知，本文所提方法与其他方法相比，在密钥生成速率上有着明显的优势，这是因为本文所提方法将不断变化的反射信道作为密钥源，并进行了多次信道探测，大大增加了密钥源的随机性。

5.2 密钥生成速率与轮数的变化关系

为了研究密钥生成速率与反射系数变化次数L的关系，本文在信噪比SNR=10 dB的条件下，对于不同的L进行了仿真实验，仿真结果如图5.2所示。由仿真可得，在SNR=10 dB的条件下，当反射单元个数为40时，每次变化可多产生约0.35比特密钥；当反射单元个数为60时，每次变化可多产生约0.49比特密钥；当反射单元个数为80时，每次变化可多产生约0.7比特密钥。

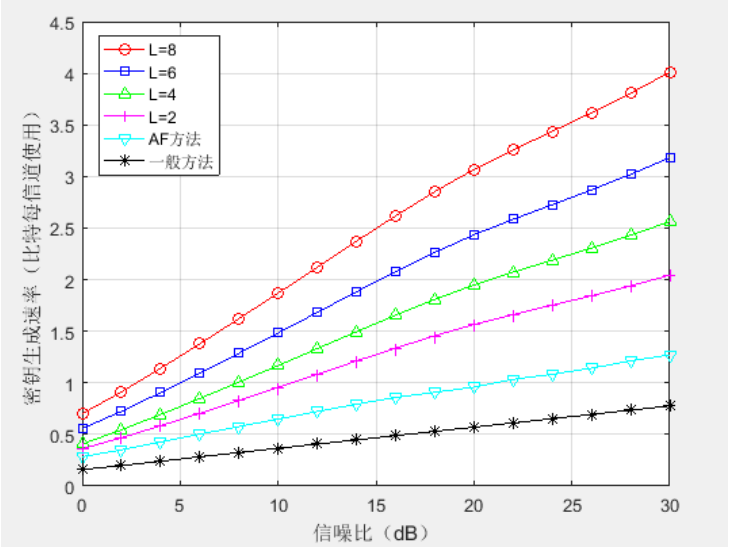


图5.1　信噪比对密钥生成速率影响

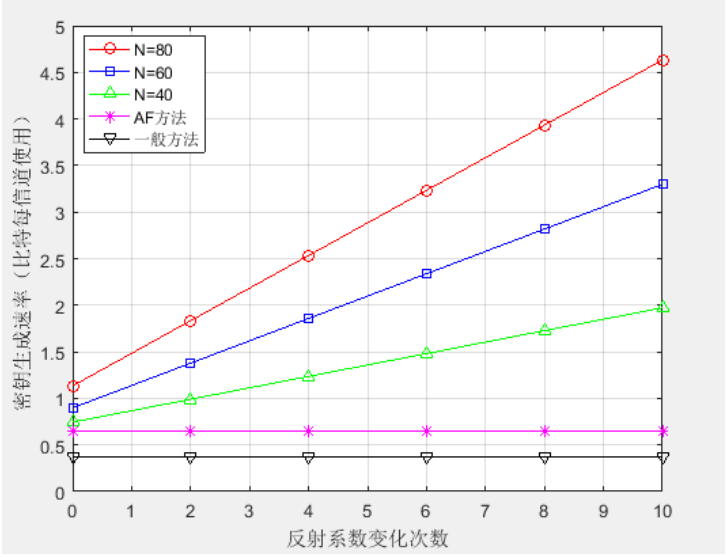


图5.2　反射系数变化次数对于密钥生成速率的影响

5.3　密钥安全性分析

本文研究了反射单元个数对于密钥生成安全性的影响，仿真结果如图5.4所示。IRS反射单元的个数N越大，合法通信双方的共享随机源互信息越大，窃听者窃取共享随机源互信息的难度越大，$I(Y;Z)$越小，密钥越安全。

5.4　“一次一密”最优速率

通过对于相干时间内密钥生成与数据传输时间的联合分配，本文在N=60、$R_b = 10$的条件下，对不同信噪比下的“一次一密”最优速率进行了仿真实验，实验结果如图5.5所示。随着信噪比的不断增加，“一次一密”最优速率因为密钥生成速率的增加而增加。

6　结束语

本文主要研究了物联网通信场景中准静态环境下的密钥生成方法，针对准静态环境中信道随机性差导致密钥生成速率低的问题，提出了基于IRS的密钥生成方法。在该方法中，利用BS控制IRS反射系数的不断变化，构造一组额外的快速变化的信道，并将该信道信息与反射系数作为共享随机源提取密钥。同时，本文在所提方法的基础上进行了密钥生成与数据传输的联合最优功率分配设计，在密钥生成功率与数据传输功率相匹配时，实现了理论上的“一次一密”。蒙特卡洛仿真

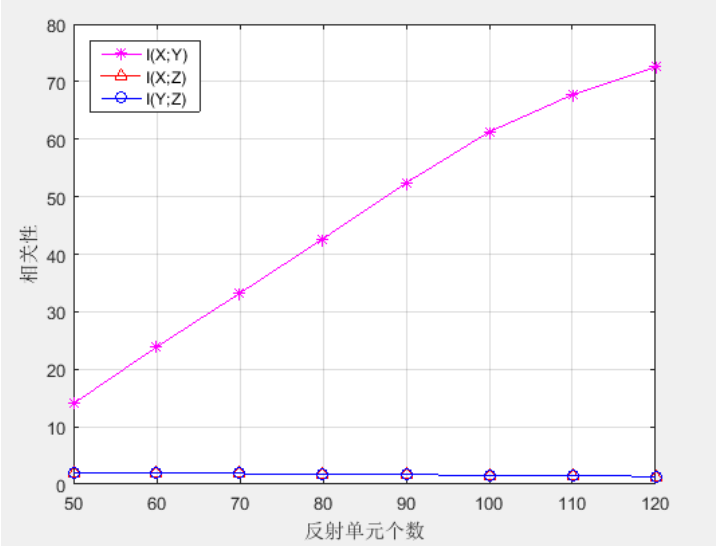


图5.3　反射单元个数对密钥安全性的影响

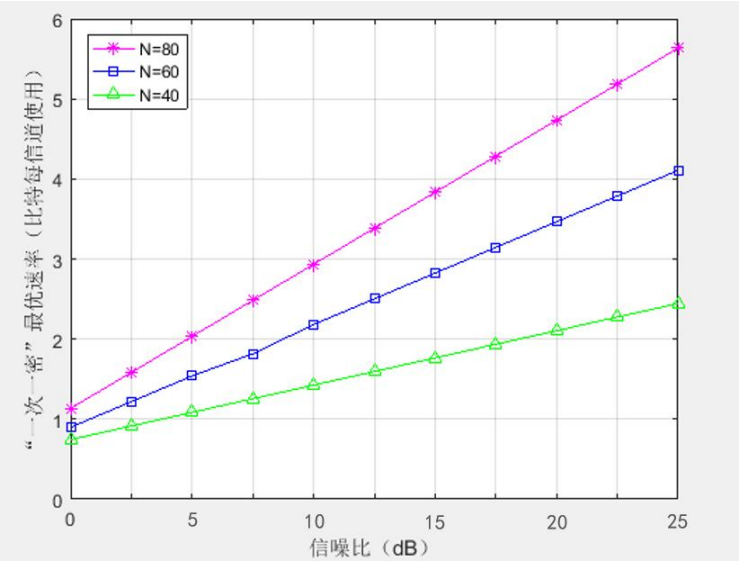


图5.4　信噪比对于“一次一密”最优速率的影响

表明，本文所提方法在信噪比为20dB、反射单元为60个的条件下，密钥生成速率与无智能超表面方法相比提高了0.47倍以上，并且随着反射单元个数和变化次数的增加，密钥生成速率会有进一步的提高。本文所提的IRS辅助密钥生成方法，为准静态环境下的密钥生成问题提供了一个良好的解决方案，并且为超材料技术在物理层安全中的应用提供了一条新的思路。

参考文献:

[1] 李古月,俞佳宝,胡爱群. 基于设备与信道特征的物理层安全方法[J]. 密码学报,2020,7(2):224-248.

[2] Madiseh M, Neville S, and Mc Guire M. Applying beamforming to address temporal correlation in wireless channel characterization based secret key generation[J]. IEEE Transactions on Information Forensics Security，2012，7(4)：1278-1287.

[3] Chen D, Qin Z, Mao X, et al. Smoke Grenade: An efficient key generation protocol with artificial interference[J]. IEEE Transactions on Information Forensics Security，2013，8(11)：1731-1745.

[4] Yang B, Wang W, and Yin Q. Secret key generation from multiple cooperative helpers by rate unlimited public communication[C]. IEEE International Conference on Acoustics，Speech and Signal Processing，Florence，Italy，2014：8183-8187.

[5] Lou Y, Jin L, Zhong Z, et al. Secret key generation scheme based on MIMO received signals spaces [J]. SCIENTIA SINICA Information，2016，47(3)：362-373.

[6] Takayuki S, Hisato I, and Hideichi S. Physical-layer secret key

agreement in two-way wireless relaying systems［J］. IEEE transactions on Information Forensics and Security，2011，6（3）：650-660.

[7] 魏浩,郑宝玉,候晓赟,等. 基于放大转发的双向中继信道密钥生成［J］. 电子与信息学报,2013,（6）:1344-1350.

[8] 肖帅芳,郭云飞,白慧卿,等. 面向物联网准静态信道的中继协作密钥生成方法[J]. 电子与信息学报,2013,（6）:1344-1350.

[9] ZhouG. , PanC. , RenH. , WangK. , Di RenzoM. and NallanathanA. , "Robust Beamforming Design for Intelligent Reflecting Surface Aided MISO Communication Systems，" in IEEE Wireless Communications Letters，doi: 10. 1109/LWC. 2020. 3000490.

[10] GuoH. , LiangY. , ChenJ. and LarssonE. G. , "Weighted Sum-Rate Maximization for Intelligent Reflecting Surface Enhanced Wireless Networks，" 2019 IEEE Global Communications Conference (GLOBECOM)，Waikoloa，HI，USA，2019，pp. 1-6，doi: 10. 1109/GLOBECOM38437. 2019. 9013288.

[11] Björnson and LE. . Sanguinetti, "Demystifying the Power Scaling Law of Intelligent Reflecting Surfaces and Metasurfaces, " 2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)，Le gosier，Guadeloupe，2019，pp. 549-553，doi: 10. 1109/CAMSAP45676. 2019. 9022637.

[12] DongL. and WangH. , "Enhancing Secure MIMO Transmission via Intelligent Reflecting Surface，" in IEEE Transactions on Wireless Communications，doi: 10. 1109/TWC. 2020. 3012721.

[13] WuQ. and ZhangR. , "Towards smart and reconfigurable environment: Intelligent reflecting surface aided wireless networks," 2019. Available：https：//arxiv. org/abs/1905. 00152.

[14] ChenJ, LiangYC, PeiY, et al. Intelligent Reflecting Surface: A Programable Wireless Environment for Physical Layer Security［J］. IEEE Access，2019，7：82599-82612.

[15] Wu Q, Zhang S, Zheng B, et al. Intelligent Reflecting Surface Aided Wireless Communications: A Tutorial［J］. 2020. Available: https：//arxiv. org/abs/2007. 02759

[16] Yu X , Xu D , Schober R . Enabling Secure Wireless Communications via Intelligent Reflecting Surfaces［C］// 2019 IEEE Global Communications Conference (GLOBECOM). IEEE，2020.

[17] Shannon C E. Communication theory of secrecy systems［J］. Bell System Technical Journal，1949，28（4）：656-715.

[18] PASOLINI G and DARDARI D. Secret key generation in correlated multi-dimensional Gaussian channels［C］. 2014IEEE International Conference on Communications，Sydney，Australia，2014：2171 - 2177. Doi：10. 1109/ICC. 2014. 6883645.

[19] 金梁, 蔡奥林, 黄开枝,等. 基于多随机信号流的密钥生成方案[J]. 电子与信息学报，2019，041（006）：1405-1412.

[20] CSISZAR I and NARAYAN P. Common randomness and secret key generation with a helper［J］. IEEE Transactions on Information Theory，2000，46（2）：344-366. doi: 10. 1109/18. 825796.

[21] 楼洋明, 金梁, 钟州, 等. 基于MIMO接收信号空间的密钥生成方案［J］. 中国科学F辑，2017，047（003）：362-373.

[作者简介]

郝一诺（1997—），女，信息工程大学硕士研究生，研究方向为无线物理层安全。

金 梁（1969—），男，博士，信息工程大学教授，博士生导师，研究方向为移动通信技术、阵列信号处理、无线物理层安全。

黄开枝（1973—），女，博士，信息工程大学教授，博士生导师，研究方向为宽带移动通信与异构无线网络安全、无线物理层安全。

肖帅芳（1989—），男，博士，信息工程大学助理研究员，研究方向为无线物理层安全。

无标度网络鲁棒性智能优化策略与应用

彭亚斌，刘彩霞，刘树新，李海涛
中国人民解放军战略支援部队信息工程大学，河南 郑州 450002

摘　要：现实世界中的大部分网络都具有无标度特性，无标度网络作为研究真实网络的一种重要技术手段，其鲁棒性增强问题是近年来的研究热点。随着智能优化算法和机器学习的快速发展，将其用于网络鲁棒性的优化成为重要研究趋势。通过对国内外相关文献进行系统的整理和分析，介绍了无标度网络鲁棒性的研究基础，然后重点对鲁棒性智能优化策略进行分类总结，并以物联网为例阐述了无标度网络的应用，最后对无标度网络鲁棒性的研究趋势进行了展望。

关键词：无标度网络、鲁棒性、智能优化算法、物联网

Robustness Intelligent Optimization Strategy and Application ofScale-Free Network

PENG Ya-bin, LIU Cai-xia, LIU Shu-xin, LI Hai-tao
PLA Strategic Support Force Information Engineering University, Zhengzhou 450002, China

Abstract: Most networks in the real world have the characteristics of scale-free. As an important technical means of studying real networks, the problem of increasing robustness of scale-free networks has been a research hot spot in recent years. With the rapid development of intelligent optimization algorithms and machine learning, using them for network robustness optimization has become an important research trend. By systematically sorting out and analyzing relevant documents at home and abroad, this paper introduces the research basis of the robustness of scale-free networks, and then focuses on classification and summary of robust intelligent optimization strategies, and uses the Internet of Things as an example to illustrate scale-free The application of the network, finally the research trend of the robustness of the scale-free network is prospected.

Key words: scale-free; robustness; intelligent optimization algorithm; internet of things

1　引言

随着科学技术的发展，现实生活中的各种系统变得越来越庞大、复杂、难以掌控，而复杂网络作为一种数据表现形式和科学研究的技术手段，可以很好的表示和分析现实世界中的复杂系统。例如，将人群中感染流行病的信息构建成网络，分析病毒传播特点[1]，有助于病毒隔离和预防；将真实通信系统建模为网络，调整其拓扑连接，可以有效降低网络攻击造成的伤害等。因此，运用复杂网络理论解决现实系统中的问题成为研究热点。

无标度网络作为复杂网络中最接近于真实世界的基本网络模型，被广泛应用于各类复杂系统的研究[2]，其中无标度网络的鲁棒性是应用研究的重点。网络鲁棒性是指网络遭受随机故障或恶意攻击后，网络维持

原有功能的能力，它是网络高效稳定工作的保证。现实生活中鲁棒性较差的网络可能造成巨大的经济和时间损失，比如2013年10月，美国阿肯色州的某高压传输装置被一名男子破坏，直接导致超过10000人以上规模的停电[3]。而网络鲁棒性增强策略可以有效的降低网络遭受随机故障或恶意攻击后造成的损失，现有鲁棒性增强策略包

基金项目：国家自然科学基金资助项目（No. 61803384）

括：防护加强策略（即事前防护，如关键节点保护、边权调整、容量冗余等）、智能优化策略（即事后优化，如遗传算法、禁忌搜索算法、机器学习算法等）、恢复策略（即事后治理，如节点恢复等）。其中智能优化策略以网络鲁棒性最大化为目标函数，通过各类优化算法搜索目标问题的最优解，调整网络节点的连接方式以增强网络鲁棒性。

目前关于无标度网络鲁棒性的智能优化策略的研究多见于外文文献，中文文献较少。因此，本文对近几年部分相关文献进行整理和总结，并以物联网为例，介绍了无标度网络的应用和鲁棒性优化流程，希望能为相关研究者提供一定的思考。

2 无标度网络鲁棒性的研究基础

2.1 无标度网络模型

万维网、交通网络以及引文网络等诸多现实网络具有如下共同特性：绝大多数节点仅拥有几个连边，而少数节点却拥有大量连边，表现在节点度分布为幂律（power-law）分布，即$P(k)\sim k^{-\gamma}$，此类网络称为无标度网络。

无标度网络拓扑是从小到大逐渐增加规模而形成的，更符合现实网络的演化过程。以论文引用关系网络为例，不断有新的论文加入到网络中，且新加入的论文偏向于引用已经被多次引用的论文，即论文被引用的概率与其已经被引用的次数成正相关，从而造成个别经典论文被多次引用，而大多数论文被引用次数较少。Barabási 和 Albert 为解释这一现象的产生机理，提出了BA无标度网络模型（简称BA模型）。

为考虑现实网络的实际情况，研究者们在BA模型基础之上进行扩展，使其能够更准确的模拟现实网络。文献［4］考虑了节点的内在性质，提出了适应度模型（fitness model），其优先连接概率与节点的度数和适应度的乘积成正比。文献［5］考虑了节点的连接范围有限，提出了局域世界演化网络模型（local-world evolving network），该模型将优先连接机制从全局限定到局域范围。文献［6］考虑了节点的老化和死亡，网络中节点并不是只增不减的，而应该是动态变化的，提出了动态演化模型（dynamical evolution model），该模型通过节点增长概率p和节点删除概率q控制网络中节点数量的动态变化。

2.2 网络攻击类型

网络攻击根据其攻击目标的不同，可分为节点攻击和连边攻击。节点攻击指被攻击节点以及其连边从网络中全部移除，而连边攻击指被攻击边从网络中移除。相比之下，节点攻击的破坏力更强、攻击速度更快，移除一定比例节点后，网络将处于瘫痪状态。因此，节点攻击被攻击者广泛采用。

网络攻击根据其攻击方式的不同，可分为随机攻击和恶意攻击[7]。随机攻击指随机选择网络中的任一节点进行攻击，所有节点被攻击概率相同，而恶意攻击指有目标性的选择网络中最“重要”的节点进行攻击，其目的是以最少的攻击代价对网络造成最大的破坏。根据节点重要性的评估方法不同，恶意攻击又分为：

1） 高度节点自适应攻击（HDA， high degree adaptive attack）[8]，其攻击策略为：每次攻击当前网络中度数最大的节点，并删除其连边，然后对剩余网络中的节点根据度的大小重新排序，重复以上步骤，直到网络中所有节点都为孤立节点。HDA是当前研究中普遍采用的恶意攻击方式；

2） 高介数节点自适应攻击（HBCA， high Betweenness centrality adaptive attack）[9]，其攻击策略与HDA相似，仅需把度数改为节点介数中心性。由于节点介数的计算成本较高，故HBCA策略在研究中并不常用。

网络攻击根据其攻击范围的不同，可分为全局攻击和局域攻击[10]。全局攻击指攻击范围为整个网络，可以攻击网络中的任意节点，但实际操作时，网络的全局信息往往难以获取，因此，攻击者常根据已掌握的局域信息进行攻击[11]。

2.3 鲁棒性评价指标

网络鲁棒性是指网络遭受攻击或故障后，保持其网络结构和功能完整性的能力[12]。它对于现实网络的搭建、评价及优化具有指导作用，成为近几年学术界研究的热点问题。

基于渗流学理论和统计学方法，文献［8］提出一种网络鲁棒性度量指标R，该指标考虑了所有可能的恶意攻击下的最大连通子图的大小，其定义如下：

$$R = \frac{1}{N}\sum_{Q=0}^{N-1} s(Q) \tag{1}$$

其中N表示整个网络的节点数目，$s(Q)$表示移除Q个节点后，剩余网络的最大连通子图所包含的节点数目与N的比值，系数$1/N$是归一化处理，便于不同规模的网络进行鲁棒性对比。全连接网络拓扑的鲁棒性最强，对应的R值为0.5；星型网络拓扑的鲁棒性最弱，对应的R值为$1/N$，因此指标R的取值范围为$[1/N, 0.5]$[13]。该指标及其变形已被研究者们广泛应用于网络鲁棒性的研究。

除此之外，网络鲁棒性的度量指标还有网络连通效率、可用性度量、图谱以及渗流阈值[14-15]等，这些指标从不同角度衡量网络鲁棒性，适用于不同的应用场景。因此，在选择鲁棒性度量指标时，应根据所研究应用的实际需求确定。

3 无标度网络鲁棒性的智能优化研究

无标度网络的“长尾”特征，使其对于随机攻击具有很强的抵抗能力，而对于恶意攻击的抵抗能力很差。因此，本文主要针对恶意攻击下的无标度网络鲁棒性优化方法进行总结。此外，为了便于各优化方法的对比，均采用指标R作为鲁棒性评价指标。

鲁棒性增强最直观的方法就是增加连边，但这会导致网络本身的无标度特性遭到破坏，另一方面，对于大多数现实网络，节点所能连接的边的数量是有限的，过多的连边可能导致节点的功能下降或故障。因此，无标度网络的鲁棒性优化需要保证其节点度分布不变，本文提及的优化方法均满足此条件。

随着优化理论和计算机科学的高速发展，智能优化算法在各方面的应用得到突破，本文主要从爬山算法、模拟退火算法、遗传算法、文化基因算法以及机器学习算法出发，研究其在无标度网络鲁棒性优化方面的应用。

3.1 传统的鲁棒性优化方法

1) 爬山算法

爬山算法（HCA， hill climbing algorithm）是经典的网络鲁棒性智能优化算法。它是一种局部择优的贪心搜索算法，通过从当前解的相邻解空间搜索一个最优解作为当前解，直到当前解没有更优相邻解则停止搜索。该方法实现简单，但很容易陷入局部最优。

文献［16］基于爬山算法，提出一种随机交叉换边策略增强网络鲁棒性的方法。通过随机交叉交换拓扑中的两条边，计算交换前后鲁棒性指标R是否增加，如果增加，则接受此次交换，否则，将拓扑还原。该方法能够保证R值持续增加，但爬山算法存在“多峰”问题，不能保证搜索到全局最优解，且换边效率较低导致计算成本增加。

另一方面，该方法在鲁棒性优化的过程中发现，网络拓扑结构逐渐演化为“类洋葱（onion like）”的拓扑结构，其特征为度数相近的节点更倾向于相互连接，度数较大的节点在拓扑中心，被度数逐渐减小的节点环绕。此结构能够保证中心节点遭受攻击后，网络仍能处于较高的连通状态，该结构也被后续的研究中广泛采用。

2) 模拟退火算法

针对爬山算法易陷入局部最优问题，文献［17］基于模拟退火算法（SA， simulated annealing），提出了一种概率性换边策略。该策略创新点在于以一定的概率接受搜索过程中的非最优解，该概率以模拟退火的方式逐渐降低到0，这使得算法在搜索过程中有更大的解空间去寻找全局最优解。但更大的解空间意味着更多的冗余换边操作和R值计算，导致计算成本较高。

为此，文献［18］对其搜索算子进行改进。利用“类洋葱”结构特征，构造了一种搜索能力较强的启发式搜索算子，并结合随机搜索算子提出了启发式混合搜索算子，结果表明其优化性能和计算开销都有所改善。此外，文献［19］以自然连接性为优化目标，利用此算法对无标度网络鲁棒性进行优化，结果证明了其优化能力优于爬山算法和智能重连策略。

3) 智能重连策略

针对爬山算法换边效率较低问题，文献［20］基于恶意攻击下网络最大连通子图的状态演化，提出一种智能重连策略（SR， smart rewiring），通过对具有相似度数的节点分优先级的进行连接，形成高度模块化的拓扑结构，这种“模块化类洋葱”结构属于“类洋葱”结构的特例。通过在真实网络中验证，结果表明只需智能交换9%的连边，可以将鲁棒性提高30%。该方法相比于模拟退火算法的计算成本降低了数个数量级，但鲁棒

性优化效果一般，适用于对时效性要求高、优化性能要求较低的真实网络系统。此外，文献［21］在此基础之上，提出交叉换边时在高度节点间建立连边的方法，该方法解释了正度-度相关性可以增强网络鲁棒性的原因。

3.2 遗传算法

遗传算法（GA， genetic algorithm）是基于“物竞天择，适者生存”的生物进化论思想而提出，通过模拟生物遗传、种群进化过程实现全局搜索，在求解复杂最优化问题中表现突出。因此，遗传算法也被广泛应用于网络鲁棒性优化问题。

1) *传统的遗传算法*

遗传算法的基本流程包括编码设计、种群初始化、交叉、变异、选择等操作。文献［22］基于传统的遗传算法，实现了网络鲁棒性增强，但传统的遗传算法存在早熟收敛问题[23]，这是由于进化过程中，个体间差异减小，基因多样性急剧降低，造成交叉和变异无法产生更优个体，种群逐渐失去进化动力，而收敛到某个适应度非最优的个体。另一方面，收敛速度慢、计算成本高也是传统遗传算法应用中的一大难题，特别是当搜索空间较大时。此外，遗传算法还被应用于具有无标度特性的无线传感器网络的部署，比如能量利用效率和能量平衡的取舍均衡问题[24]，以及求解最优网络聚类问题等[25，26]。

2) *多智能体遗传算法*

为了解决传统遗传算法收敛速度慢和计算成本高问题，文献［27］利用多智能体对环境的感知和反作用能力，结合遗传算法的搜索方式，提出了多智能体遗传算法（MAGA， muti-agent genetic algorithm），它包括邻域竞争、邻域正交交叉、变异、智能体自学习算子，可以有效地解决全局数值优化问题。

文献［28］在多智能体遗传算法框架基础上，添加局部搜索算子，各智能体通过与相邻区域内的智能体进行竞争、交叉、变异和自学习操作，从而提升自身能量，实现无标度网络鲁棒性的增强，在BA网络和真实网络中的鲁棒性优化效果都优于传统方法，且收敛速度快、计算成本较低。此外，多智能体遗传算法还被应用于电网规划[29]、交通管控[30]以及资源调度[31]等现实问题。

3) *多种群遗传算法*

为了解决传统遗传算法的早熟收敛问题，文献［32］提出了多种群遗传算法（MPGA， multi-population genetic algorithm），通过不同种群分别设置不同的交叉和变异概率，使不同种群在全局和局部搜索上有不同的平衡[33]，有效规避了由于概率取值造成的早熟收敛问题，同时通过对比观察不同种群的进化过程，可以发现哪些种群陷入局部最优而停止进化。

文献［34］在多种群遗传算法基础之上，引入移民算法，从而提出了多种群协同进化算法（MPCA， multi-population co-evolution algorithm），如图1所示。其创新点在于移民算法实现了多个种群之间的基因交流，通过不同种群的最优个体替换其他种群的最差个体，避免了个体间的基因差异的趋同性，丰富基因的多样性有利于种群的持续优化，并保证了算法最优解是各种群协同进化所得。该算法的鲁棒性优化效果明显优于前文所提算法，但多种群同时进化导致计算开销很大，当网络规模达到一定程度后，如何降低优化时间成为有待解决的难题。

3.3 文化基因算法

文化基因算法（MA， memetic algorithm），又称密母算法，它将种群的全局搜索与个体的局部启发式搜索相结合[35]，其本质为：

$$\text{Memetic} = \text{GA} + \text{Local Search} \tag{2}$$

即算法实质相当于在遗传算法的基础之上，添加局部搜索算子，局部搜索算子可以根据不同的策略进行设计。

文献［36］基于文化基因算法，考虑到“类洋葱”结构的度相似节点倾向于相互连接的特性，通过计算节点度差设计了局部搜索算子，进而以拓扑竞争的方式得到全局鲁棒性最优的拓扑。文献［37］同时考虑了恶意攻击和随机攻击下的无标度网络鲁棒性优化，相比于文献［36］，其创新点在于设计了新的鲁棒性评价指标和局域搜索算子，该算子能保证在优化恶意攻击下的鲁棒性时，不破坏其抵御随机攻击的能力。

文化基因算法在网络鲁棒性优化问题中应用广泛。文献［38］基于此算法，采用遗传算法作为全局搜索，两级学习策略作为局部搜索，在保持网络的社区结构的前提下，增强了网络的社区

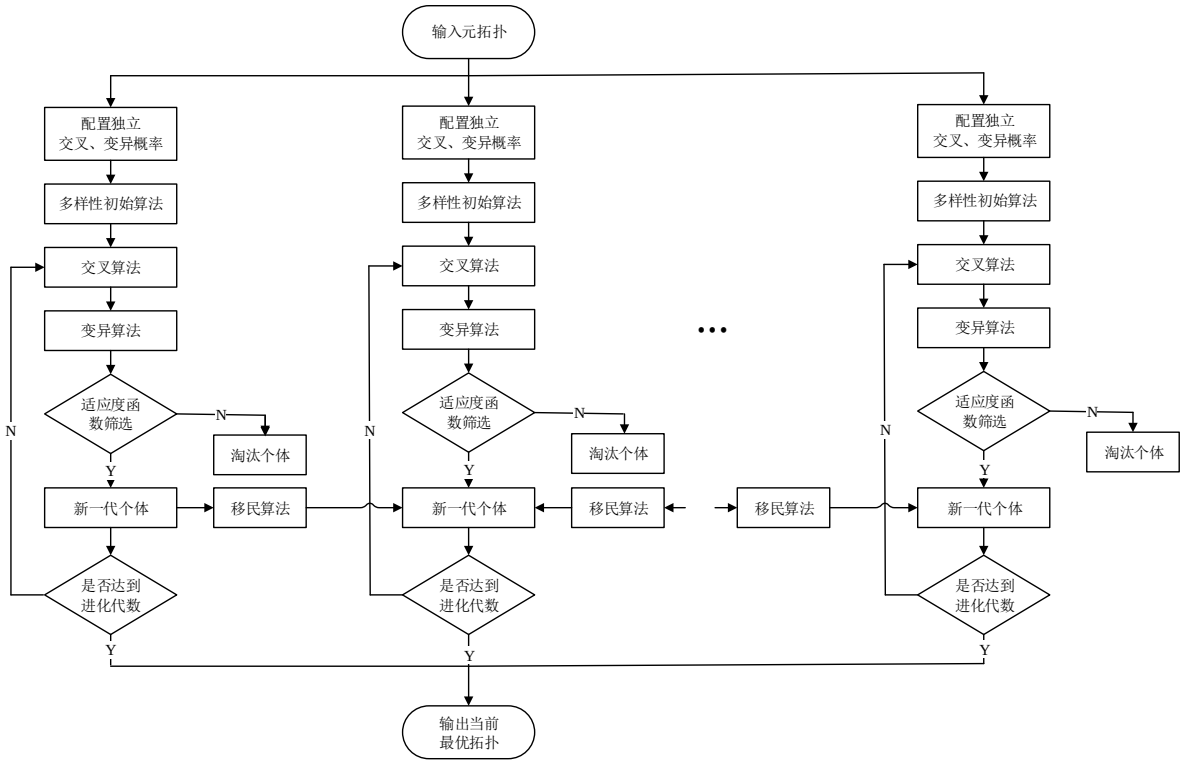


图1　MPCA算法流程图

鲁棒性。文献［39］基于此算法，解决了主机-路由器组成的通信网络中主机位置的最优配置问题，有效地增强了模型对级联失效的鲁棒性，对于异构网络系统的设计和优化有指导作用。文献［40］考虑了sink节点对网络负载分配的影响，基于此算法增强了无线传感器网络抵抗级联故障的鲁棒性，并得出网络通信效率、模块化程度和聚类系数与鲁棒性呈正相关，平均最短路径长度与鲁棒性负相关。此外，该算法还应用于航空网络的重要连边识别和保护[41]、无标度网络的级联失效鲁棒性[42]、相依网络的社区鲁棒性[43]等问题。

遗传算法与文化基因算法都属于进化算法，且原理相似，将二者进行对比，如表1所示。

表1　各遗传算法与文化基因算法对比

算法	算法框架组成	种群数目	优缺点
传统的遗传算法	编码设计、种群初始化、交叉、变异、选择操作	单种群	算法简单，计算成本仅高于多智能体遗传算法；易陷入早熟收敛，优化效果不佳
多智能体遗传算法	编码设计、种群初始化、邻域竞争、邻域正交交叉、变异、智能体自学习算子	单种群	收敛速度快，计算成本低，适用于对时效性要求较高网络；优化效果一般
多种群遗传算法	编码设计、种群初始化、交叉、变异、选择操作	多种群	解决了“早熟收敛”问题，优化效果较好；算法复杂，计算成本较高
多种群协同进化算法	编码设计、种群初始化、交叉、变异、选择操作、移民算子	多种群	优化效果最佳，移民算子增加了基因交流和多样性；计算成本高，仅适用于对鲁棒性要求较高的网络
文化基因算法	编码设计、种群初始化、交叉、变异、选择操作、局部搜索算子	单种群	可针对不同网络的需求设计局部搜索算子，适用于多种类型的网络；优化效果和计算成本都一般

3.4　机器学习

分析前文提及的无标度网络鲁棒性的优化算法，可以发现主要存在两个问题：鲁棒性优化效果不佳（易陷入局部最优）和算法计算成本较高，二者在同一算法中难以兼顾，算法设计时都有所取舍。而机器学习（ML， machine learning）能够很好的解决这一难题，它可以通过训练大量的网络拓扑数据，提取高鲁棒性网络的拓扑特征，将

学习到的模型直接应用于初始拓扑的鲁棒性优化，这样既可以保证算法的优化效果，又可以显著降低计算成本。

1) 神经网络

神经网络（NN， neural network）算法是当前研究的热门方向，其在诸多应用领域内取得了显著成果，其原理是模拟人类大脑的神经元传导机制，具有强大的学习能力，根据其层数的不同可分为浅层神经网络和深度神经网络。

反向传播（BP， back propagation）神经网络是浅层神经网络中应用最成功的代表，具有很强的非线性映射、自学习和自适应、泛化、容错能力，可以很好地解决复杂优化问题。文献［44-45］基于误差和动量扰动因子的BP神经网络，提出一种智能演化算法（ROML， robustness optimization scheme by machine learning），如图2所示。该算法以多种群协同进化算法优化前后拓扑为训练数据集，通过设计模型的决策函数和阈值来学习网络节点的连接特征，用学习参数表征网络拓扑优化过程中的特征变化。该方法在训练过程中需要一定的计算成本，但生成后的学习模型可以迅速完成初始拓扑的优化。另一方面，BP神经网络作为一种监督学习算法，限制了优化上限，未来可以考虑采用神经网络中的无监督学习算法，进行网络鲁棒性的进一步优化。

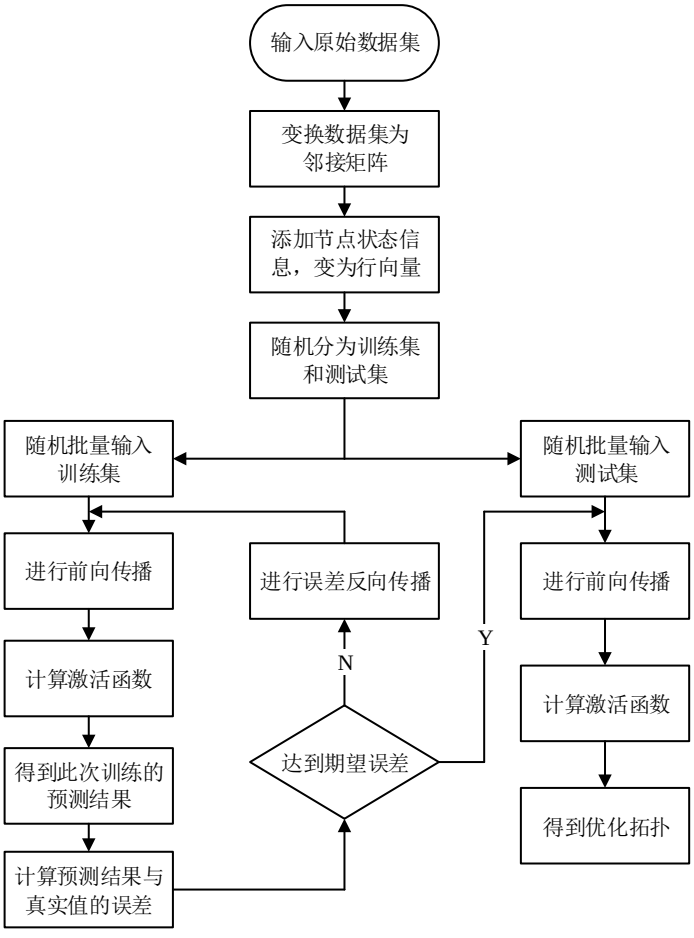


图2 ROML算法流程图

2) 强化学习

强化学习（RL， reinforcement learning）不同于监督学习和无监督学习，它在没有标注数据集的前提下，通过动作执行后所得回报函数，判断动作是否正确来进行学习，其中回报函数相当于一种延迟和稀疏的标签数据形式。即给定数据，通过不断“试错”学习如何选择一系列动作，以取得最大化长期收益。

基于强化学习的算法特性，文献［46］结合深度神经网络和强化学习的Actor-Critic算法，提

出了一种深度确定性策略学习（DDLP， deep deterministic policy learning）算法，如图3所示。该算法以网络鲁棒性评价指标为回报函数（目标函数），将网络拓扑转换为环境空间，利用深度学习神经网络来模拟规则函数（actor网络）和回报函数（critic网络），减小了问题模型的样本复杂度，并提高了系统模型的收敛性。而对于动作空间较大问题，提出一种映射操作，降低了搜索成本和存储开销。实验结果表明，该算法在计算成本和优化性能方面都大大优于现有算法。未来可以考虑通过简化神经网络层次以实现更轻量级的DDLP算法，用于时效性要求较高的现实网络拓扑的优化。

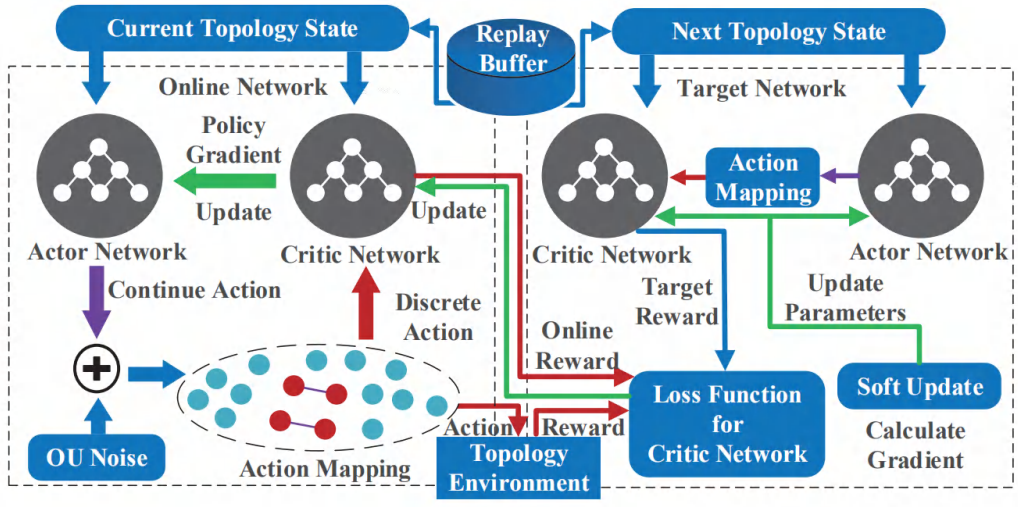


图3　DDLP概述

3.5　其他优化方法

1)　基于“类洋葱”结构的鲁棒性优化算法

前面研究已经证明“类洋葱”结构具有高鲁棒性，文献［47］利用度相似节点趋于连接的结构特性，设计了度差交换策略；利用拓扑中存在大量的垂直于网络中心的边，使得度数相近的节点组成类环形的结构特性，设计了角度和交换策略，两种策略相结合有效的实现了初始拓扑向“类洋葱”结构拓扑的转变。另一方面，该算法限制了换边的条件，提出独立边的概念，减少了换边操作和R值的比较次数，使该方法更具时效性。该方法的鲁棒性优化效果一般，但该方法所提策略与其他优化算法相结合，会有更好的优化效果。

2)　基于边分类的启发式算法

现有无标度网络鲁棒性优化算法存在一个“通病”，即当网络规模较大时优化性能较差，这是由于现有算法没有发现影响网络鲁棒性的根本因素。为此，文献［48］分析了影响最大连通子图大小的边的类型，根据连边的节点是否失效，将边分类为有效边、弹性边和无效边，并提出三种调整算子和两种复原算子，用于对不同类型的边进行变换，在不改变度分布的约束下，调整不同类型边的数量和改变同一类型的边的连接关系，从而提高网络的鲁棒性。此外，由于该算法分析并改进了影响网络鲁棒性的重要因素，所以在优化大规模网络时，性能并没有显著下降。

3.6　多目标优化算法

前文所提算法都仅以网络鲁棒性为优化目标，称之为单目标优化问题。但真实网络优化时，需要根据网络的现实需求，确定多个不同的优化目标，当优化目标达到两个或两个以上，则称之为多目标优化问题。

进化算法由于其全局搜索、并行性等算法特性，成为解决多目标优化问题的主流算法，称之为多目标进化算法（MOEA， muti-objective evolutionary algorithm）。文献［49］以非支配排序的多目标进化算法为框架，根据网络结构特点设计了交叉和局部搜索算子，实现了节点和连边鲁棒性的同时优化。文献［50］针对多种恶意节点和连边攻击组成的混合攻击，提出两阶段多目标进化算法MOEA - RSF_{MMA}，两阶段优化模式有利于平衡两个目标的计算成本，提高搜索效率，充分

发挥了多目标进化算法在组合优化领域的优势。

网络分析是社会计算和大数据的理论基础之一，它包括网络鲁棒性分析、社团结构分析、结构平衡分析等[51]，其中网络结构平衡问题的研究对于现实网络具有实际价值。因此，文献［52］将网络结构平衡和网络鲁棒性同时优化建模为多目标优化问题，提出一种多目标进化算法MOEA/D - RSB，并在无标度网络和实际符号网络中验证了算法具有良好的优化性能。此外，分析了不同鲁棒性能分区的特点，以说明不同的平衡策略对网络鲁棒性的影响，对于处理社会和自然动力学中的某些问题具有参考价值。

4　无标度网络的应用--以物联网为例

现有研究表明，现实世界中的大多数复杂网络都具有无标度特性[53]，属于无标度网络，如互联网、通信网、交通运输网络、电网以及物联网。无标度网络特性及鲁棒性优化相关研究也被应用于各个领域，文献［54］根据无标度网络的最短路径特征，提出一种更高效的交通运输系统。文献［39］通过优化主机和路由器的配置，增强了通信网络抵抗恶意攻击和级联失效的能力。文献［55］利用航路点的位置来重构航空网络中航线，使其趋向于更为稳定的结构，从而大幅提高航空网络的整体鲁棒性。

在无标度网络的众多应用中，物联网的高速发展受到研究者们关注，特别是5G和窄带物联网的逐渐商用，使物联网成为继通信网后又一个万亿级市场，是人类生活走向智能化的必经之路[56]。物联网涉及到智慧城市、智能工业、现代农业、医疗护理以及国防安全等多个领域[57]，预计到2020年底，物联网设备的连接数量将超过250亿[56]。因此，对物联网网络拓扑结构和演化行为的研究具有重要意义。

4.1　物联网拓扑构建

物联网拓扑结构的构建是研究的基础，将物联网设备看做节点，设备间的通信看做连边，就可以将物联网抽象为一个复杂网络拓扑。由于现有物联网感知层设备多采用无线通信方式，因此物联网感知层拓扑也相当于无线传感器网络拓扑。另一方面，根据设备节点的功能是否相同，可将物联网感知层拓扑分为同构拓扑和异构拓扑，小世界网络模型多用于物联网异构拓扑的建模[58-59]，而无标度网络模型多用于物联网同构拓扑的建模[60]，两类拓扑分别适用于不同的应用场景。

物联网设备节点的通信距离、存储空间以及节点能量有限，因此，传统的无标度网络的构造方法需要进行改进才能适用于物联网拓扑。文献［61］基于BA网络的构建方案，提出将择优连接限制在通信范围内，以及限制节点的最大度数，构建了一种同构物联网拓扑模型。考虑到山区等特殊地理环境下的物联网节点部署，文献［62-63］首先运用高斯分布理论，构建了一个随机多峰山地地形，随后考虑地形遮挡、噪声、信号衰减等环境因素，得出每个坐标的侦测概率，最后按照择优连接和轮盘赌法原则，实现了三维环境下的物联网拓扑构建。考虑到节点的能量消耗，文献［64］通过可调系数来平衡无线传感器网络的连通性和能量消耗，提出一种能量感知的无标度网络拓扑。文献［65］通过限定节点的连接数量，使得无线传感器网络的能力消耗更加均衡。

4.2　物联网拓扑鲁棒性优化

物联网拓扑结构的稳定健壮抗毁是研究的重点。当物联网设备遭受随机故障或恶意攻击时，如何尽可能降低整体损失，保证未被攻击的设备能够正常运行，网络拓扑鲁棒性的优化是解决此问题的根本。

物联网作为一个庞大且复杂的网络系统，如何构造高鲁棒性的物联网拓扑成为难题。在物联网的众多应用中，智慧城市与人民的生活紧密相关，它包括智能家居、城市公共安全、智慧能源、智慧交通、智慧医疗等多个领域，因此，本文以智慧城市为例介绍物联网鲁棒性优化流程[34]。如图4所示，其具体流程如下：

1）各应用领域通过设备节点将拓扑信息上传至智慧城市云端服务器；

2）云端服务器将收集到的信息传送到大数据中心服务器；

3）大数据中心存储着每个设备节点的地理信息和其他拓扑信息；

4）使用鲁棒性优化算法对大数据中心提供的数据进行建模和优化；

5）得到一个高鲁棒性的物联网网络拓扑，再根据此拓扑结构调整设备节点间的连接关系，从

而实现物联网网络拓扑的优化。

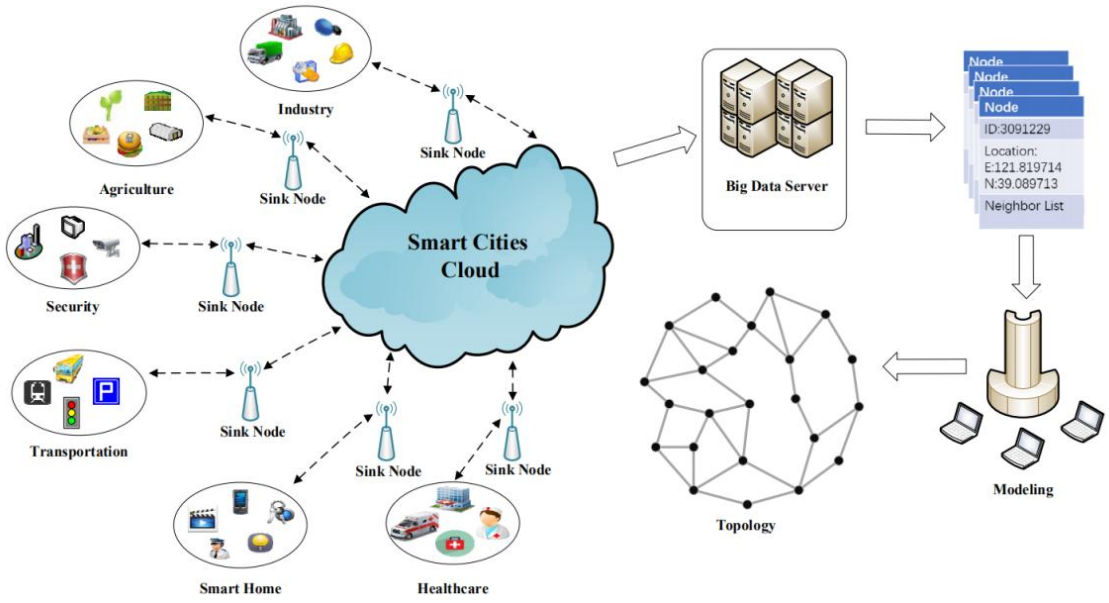


图4　智慧城市中物联网鲁棒性优化流程

物联网拓扑鲁棒性优化过程中，鲁棒性优化算法至关重要。由于物联网节点的特殊性，前文所提到的部分算法需要改进后才能应用于物联网拓扑优化，此处不再赘述。选择物联网拓扑鲁棒性优化算法时，不仅要考虑算法的优化性能，更要考虑算法的时效性，这样才能实时地进行拓扑结构优化。

4.3　物联网拓扑研究展望

现有研究中，针对物联网拓扑结构的相关研究还比较少，随着物联网应用的大量部署，拓扑结构存在的相关问题也将逐渐显现，因此，结合物联网和复杂网络相关知识，总结可能存在的问题和研究展望如下：

1） 如何设计高可用性的物联网拓扑。考虑拓扑鲁棒性、设备节点能量、负载、通信距离以及成本等因素，设计对应场景下的网络拓扑，用于指导真实物联网设备的部署。

2） 现有的网络鲁棒性的优化方法，需要集中式进行拓扑信息收集、优化、再分发，所需时间较长，面对快速的恶意攻击，优化速度赶不上攻击速度，并不能在实际应用中起到效果。因此需要考虑设计轻量级、实时在线的网络拓扑优化算法。

3） 随着物联网的快速发展，其网络规模巨大，很难将整个网络拓扑一次性放入求解问题中，因此如何解决超大规模的物联网拓扑优化问题值得深思。可以考虑设计分布式计算的拓扑优化算法解决此问题。

4） 物联网规模的增大还会导致现有优化算法的效果逐渐降低，计算成本难以接受。而机器学习在鲁棒性优化问题上的研究较少，可以考虑采用机器学习算法提取拓扑优化特征，提高优化效果，降低计算开销。特别是强化学习和无监督对比学习是研究的重点。

5） 现实应用中，物联网的拓扑优化是一个多目标优化问题，不仅要考虑拓扑鲁棒性，还要考虑成本、效率以及安全性等因素，针对不同场景的需求不同，有所取舍，这样优化方案才具备实用性。

5　结束语

现实世界中的众多复杂系统，时刻面临着自身随机故障或各类恶意攻击的危险，将它们抽象为网络拓扑结构进行鲁棒性优化，可以一定程度的解决此问题。无标度网络作为一种最贴近于现实网络的基本网络模型，其鲁棒性优化问题成为研究的重点，特别是近年来各类智能优化算法、深度学习、强化学习的快速发展，为网络鲁棒性

优化提供了新的解决思路，但当前学界对相关研究比较缺乏，有待研究者们探索。

无标度网络是一种研究工具，研究者应多结合自身领域经验，以实际系统出发，建立面向真实系统的网络模型，并对其进行分析和研究，进而解决实际问题，这也是未来的研究趋势。物联网拓扑研究存在的不足和展望同样适用于无标度网络，可供参考。

参考文献:

[1] DAW M A, EL-BOUZEDI A H. Modelling the epidemic spread of COVID-19 virus infection in Northern African countries[J]. Travel Medicine and Infectious Disease，2020，35：101671.

[2] ZHANG D, DU F, HUANG H, et al. Resiliency assessment of urban rail transit networks: Shanghai metro as an example [J]. Safety Science，2018，106：230－243.

[3] HU S, LI G. TMSE: A topology modification strategy to enhance the robustness of scale-free wireless sensor networks [J]. Computer Communications，2020，157：53－63.

[4] BIANCONI G, A-LBARABÁSI. Bose-Einstein Condensation in Complex Networks[J]. Physical Review Letters，American Physical Society，2001，86(24)：5632－5635.

[5] LI X, CHEN G. A local-world evolving network model[J]. Physica A: Statistical Mechanics and its Applications，2003，328(1)：274－286.

[6] DEO N, CAMI A. A birth-death dynamic model of scale-free networks [C]//Proceedings of the 43rd annual Southeast regional conference. New York，NY，2005：26－27.

[7] LI R H, YU J X, HUANG X, et al. Measuring robustness of complex networks under MVC attack [C]//Proceedings of the 21st ACM international conference on Information and knowledge management. New York，NY，2012：1512－1516.

[8] SCHNEIDER C, MOREIRA A, ANDRADE J, et al. Mitigation of malicious attacks on networks [J]. Proceedings of the National Academy of Sciences，2011，108：3838－41.

[9] HOLME P, KIM B J, YOON C N, et al. Attack vulnerability of complex networks[J]. Physical Review E，2002，65(5)：1-15

[10] 董政呈,方彦军,田猛. 相互依存网络抗毁性研究综述[J]. 复杂系统与复杂性科学,2017,14 (03):30-44.

[11] DONG G, DU R, HAO H, et al. Modified localized attack on complex network[J]. Europhysics Letters，2016，113(2)：28002.

[12] ALBERT R, A-LBARABÁSI. Statistical mechanics of complex networks[J]. Reviews of Modern Physics，2002，74(1)：47－97.

[13] 柴文壹. 复杂网络鲁棒性优化及其在推荐系统的应用研究[D]. 西安电子科技大学，2017.

[14] LIU J, ZHOU M, WANG S, et al. A comparative study of network robustness measures[J]. Frontiers of Computer Science，2017，11(4)：568－584.

[15] TANIZAWA T, HAVLIN S, STANLEY H E. Robustness of onionlike correlated networks against targeted attacks[J]. Physical Review E，2012，85(4)：046109.

[16] HERRMANN H J, SCHNEIDER C M, MOREIRA A A, et al. Onion-like network topology enhances robustness against malicious attacks [J]. Journal of Statistical Mechanics-Theory and Experiment，2011：P01027.

[17] BUESSER P, DAOLIO F, TOMASSINI M. Optimizing the Robustness of Scale-Free Networks with Simulated Annealing[C]// International Conference on Adaptive and Natural Computing Algorithms. Berlin, Heidelberg：Springer，2011：167－176.

[18] 慕彩红,柴文壹,刘逸,刘敬. 一种改进的网络鲁棒性与有效性增强方法[J]. 西安电子科技大学学报,2018,45(04):6-11.

[19] DUAN B, LIU J, TANG X. Optimizing the natural connectivity of scale-free networks using simulated annealing [J]. Physica A: Statistical Mechanics and its Applications，2016，457：192－201.

[20] LOUZADA V H P, DAOLIO F, HERRMANN H J, et al. Smart rewiring for network robustness[J]. Journal of Complex Networks，2013，1(2)：150－159.

[21] LIANG B, YAN-DONG X, LV-LIN H, et al. Smart Rewiring: Improving Network Robustness Faster[J]. Chinese Physics Letters，2015，32(7)：078901.

[22] REN H-P, HUANG X-N, HAO J-X. Finding Robust Adaptation Gene Regulatory Networks Using Multi-Objective Genetic Algorithm[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics，2016，13(3)：571－577.

[23] PANDEY H M, CHAUDHARY A, MEHROTRA D. A comparative review of approaches to prevent premature convergence in GA[J]. Applied Soft Computing，2014，24：1047－1077.

[24] SHUKLA R N, CHANDEL A S, GUPTA S K, et al. GAE3BR: Genetic algorithm based energy efficient and energy balanced routing algorithm for Wireless Sensor Networks [C]// 2015 International Conference on Advances in Computing，Communications and Informatics (ICACCI). 2015：942－947.

[25] ELHOSENY M, YUAN X, YU Z, et al. Balancing Energy Consumption in Heterogeneous Wireless Sensor Networks Using Genetic Algorithm [J]. IEEE Communications Letters，2015，19(12)：2194－2197.

[26] PEIRAVI A, MASHHADI H R, JAVADI S H. An optimal energy-efficient clustering method in wireless sensor networks using multi-objective genetic algorithm [J]. International Journal of Communication Systems，2013，26(1)：114－126.

[27] 钟伟才,薛明志,刘静,焦李成. 多智能体遗传算法用于超高维函数优化[J]. 自然科学进展,2003(10):72-77.

[28] 安柏慧. 恶意攻击下网络鲁棒的多智能体遗传算法优化[D]. 西安电子科技大学,2017.

[29] VICTOR A, VIORICA S, SILVIA M, et al. Multi-agent cognitive system for optimal solution search[C]//2018 International Conference on Development and Application Systems (DAS). 2018：53－56.

[30] DIEZ C, SANCHEZ-ANGUIX V, PALANCA J, et al. A Multi-agent Proposal for Efficient Bike-Sharing Usage [C]//PRIMA 2017: Principles and Practice of Multi-Agent Systems. 2017：468－476.

[31] 李小涛,彭翀. 基于混合多智能体遗传算法的作业车间调度问题研

究[J]. 北京航空航天大学学报，2017，43(02)：410-416.

[32] QIU T, LIU J, SI W, et al. A Data-Driven Robustness Algorithm for the Internet of Things in Smart Cities [J]. IEEE Communications Magazine，2017，55(12)：18－23.

[33] 刘杰．基于多种群协同进化的物联网鲁棒性策略研究[D]. 大连理工大学，2018.

[34] QIU T, LIU J, SI W, et al. Robustness Optimization Scheme With Multi-Population Co-Evolution for Scale-Free Wireless Sensor Networks [J]. IEEE/ACM Transactions on Networking，2019，27(3)：1028－1042.

[35] RUIZ L G B, CAPEL M I, PEGALAJAR M C. Parallel memetic algorithm for training recurrent neural networks for the energy efficiency problem [J]. Applied Soft Computing，2019，76：356－368.

[36] ZHOU M, LIU J. A memetic algorithm for enhancing the robustness of scale-free networks against malicious attacks [J]. Physica a-Statistical Mechanics and Its Applications，2014，410：131－143.

[37] TANG X, LIU J, ZHOU M. Enhancing network robustness against targeted and random attacks using a memetic algorithm [J]. Europhysics Letters，2015，111(3)：38005.

[38] LIU W, GONG M, WANG S, et al. A two-level learning strategy based memetic algorithm for enhancing community robustness of networks[J]. Information Sciences，2018，422：290－304.

[39] Optimizing robustness of complex networks with heterogeneous node functions based on the Memetic Algorithm[J]. Physica A: Statistical Mechanics and its Applications，2018，511：143－153.

[40] FU X, PACE P, ALOI G, et al. Topology optimization against cascading failures on wireless sensor networks using a memetic algorithm[J]. Computer Networks，2020，177：107327.

[41] DU W, LIANG B, YAN G, et al. Identifying vital edges in Chinese air route network via memetic algorithm [J]. Chinese Journal of Aeronautics，2017，30(1)：330－336.

[42] TANG X, LIU J, HAO X. Mitigate Cascading Failures on Networks using a Memetic Algorithm [J]. Scientific Reports，2016，6(1)：38713.

[43] WANG S, LIU J. Community robustness and its enhancement in interdependent networks [J]. Applied Soft Computing，2019，77：665－677.

[44] CHEN N, QIU T, ZHOU X, et al. An Intelligent Robust Networking Mechanism for the Internet of Things [J]. IEEE Communications Magazine，2019，57(11)：91－95.

[45] 陈宁．面向物联网鲁棒性优化的智能演化策略研究[D]. 大连理工大学，2019.

[46] CHEN N, QIU T, MU C, et al. Deep Actor－Critic Learning-Based Robustness Enhancement of Internet of Things[J]. IEEE Internet of Things Journal，2020，7(7)：6191－6200.

[47] QIU T, ZHAO A, XIA F, et al. ROSE: Robustness Strategy for Scale-Free Wireless Sensor Networks [J]. IEEE/ACM Transactions on Networking，2017，25(5)：2944－2959.

[48] RONG L, LIU J. A heuristic algorithm for enhancing the robustness of scale-free networks based on edge classification [J]. Physica A: Statistical Mechanics and its Applications，2018，503：503－515.

[49] LI Z, WANG S, MA W. Multi-objective Evolutionary Algorithm for Enhancing the Robustness of Networks[C]//Bio-inspired Computing－Theories and Applications. Singapore：Springer，2016：322－327.

[50] ZHOU M, LIU J. A Two-Phase Multiobjective Evolutionary Algorithm for Enhancing the Robustness of Scale-Free Networks Against Multiple Malicious Attacks [J]. IEEE Transactions on Cybernetics，2017，47：1－14.

[51] CAI Q, GONG M, RUAN S, et al. Network Structural Balance Based on Evolutionary Multiobjective Optimization: A Two-Step Approach [J]. IEEE Transactions on Evolutionary Computation，2015，19(6)：903－916.

[52] WANG S, LIU J, JIN Y. Robust Structural Balance in Signed Networks Using a Multiobjective Evolutionary Algorithm [J]. IEEE Computational Intelligence Magazine，2020，15(2)：24－35.

[53] 唐向龙．基于进化算法的复杂网络鲁棒性优化与分析[D]. 西安电子科技大学，2017.

[54] DU W B, WU Z X, CAI K Q. Effective usage of shortest paths promotes transportation efficiency on scale-free networks [J]. Physica a-Statistical Mechanics and Its Applications，2013，392(17)：3505－3512.

[55] LORDAN O, SALLAN J M, ESCORIHUELA N, et al. Robustness of airline route networks[J]. Physica A: Statistical Mechanics and its Applications，2016，445：18－26.

[56] 赵傲阳．无标度物联网拓扑鲁棒性优化策略研究[D]. 大连理工大学，2017.

[57] MILLS J, HU J, MIN G. Communication-Efficient Federated Learning for Wireless Edge Intelligence in IoT[J]. IEEE Internet of Things Journal，2020，7(7)：5986－5994.

[58] YU R, XUE G, ZHANG X. Application Provisioning in FOG Computing-enabled Internet-of-Things: A Network Perspective[C]// IEEE INFOCOM 2018 - IEEE Conference on Computer Communications. 2018：783－791.

[59] WEI K, GUO S, ZENG D, et al. Exploiting Small World Properties for Message Forwarding in Delay Tolerant Networks [J]. IEEE Transactions on Computers，2015，64(10)：2809－2818.

[60] QIU T, LUO D, XIA F, et al. A greedy model with small world for improving the robustness of heterogeneous Internet of Things [J]. Computer Networks，2016，101：127－143.

[61] BIANCONI G, RAHMEDE C. Complex Quantum Network Manifolds in Dimension d > 2 are Scale-Free[J]. Scientific Reports，2015，5(1)：13979.

[62] ZHAO A, QIU T, XIA F, et al. A Scale-Free Network Model for Wireless Sensor Networks in 3D Terrain [C]// Industrial IoT Technologies and Applications. 2016：201－210.

[63] LIU J, QIU T, ZHANG S, et al. A Three Dimensions Deployment Model for Internet of Things [C]//2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design (CSCWD). 2018：859－863.

[64] YUHUI JIAN, ERWU, YUE WANG, et al. Scale-free model for wireless sensor networks[C]//2013 IEEE Wireless Communications and Networking Conference (WCNC). 2013：2329－2332.

[65] ZHU H, LUO H, PENG H, et al. Complex networks-based energy-efficient evolution model for wireless sensor networks[J]. Chaos, Solitons & Fractals, 2009, 41(4): 1828 - 1835.

[作者简介]

彭亚斌（1996— ），男，硕士研究生，主要研究方向：复杂网络、移动通信网络安全。

刘彩霞（1974— ），女，博士，研究员，主要研究方向：移动通信网络新技术，网络与信息安全。

刘树新（1987— ），男，博士，助理研究员，主要研究方向：链路预测、移动通信网络安全。

李海涛（1982— ）男，硕士，副研究员，主要研究方向：通信网络安全、数据处理和嵌入式设计。

基于集成约减的网络入侵检测方法

李召召[1]，李彧[1]，孙远航[1]，成诚[1]，宋克[2]

[1]网络通信与安全紫金山实验室，江苏南京 211111；

[2]解放军战略支援部队信息工程大学，河南郑州 450002

摘　要：网络入侵检测数据集NSL-KDD剔除了KDD99中的冗余重复样本，一定程度上减轻了数据集带来的样本偏斜问题，被越来越多地应用于网络入侵检测的研究中。本文基于NSL-KDD数据集，提出了一种利用集成约减进行网络异常检测的方法。首先，通过随机森林中每个基学习器在验证集上的决策输出，计算出各个分类树之间的多样性指标。其次，结合遗传算法找到随机森林中最佳的基学习器组合，剔除重复冗余的分类树，以降低基学习器数量，实现集成约减。最后，通过实验测试验证所提方法的有效性。实验结果显示，基于集成约减的方法能够有效减小随机森林的集成规模，提升网络入侵检测的精度。

关键词：网络入侵检测、集成约减、多样性分析

Network Intrusion Detection Method Based on Ensemble Pruning

Li Zhaozhao[1], Li Yu[1], Sun Yuanhang[1], Cheng Cheng[1], Song Ke[2]

1.Purple Mountain Laboratories, Nanjing, Jiangsu, 211111;

2.Information Engineering University, Zhengzhou, Henan, 450002

Abstract: Network intrusion detection(NID) data set NSL-KDD has removed the redundant samples from KDD99. It alleviates the sample skew problem of KDD99 and is being used in NID algorithms more and more. In this study, we have proposed an ensemble pruning NID method based on NSL-KDD data set. Firstly, the diversity indexes between the base classifiers are calculated on the decision outputs of each base tree of random forest. Then, the redundant base trees are eliminated based on genetic algorithm(GA) to find the optimal base classifier combination and reduce the ensemble size. Lastly, a series of experiments are conducted to validate the the effectiveness of the proposed method. The results show that the proposed ensemble pruning method can reduce the size of the base classifiers of the random forest and improve the network intrusion detection accuracy.

Key words: network intrusion detection; ensemble pruning; diversity analysis

1　引言

随着互联网技术在工控领域的不断发展和延伸，针对工业网络的攻击行为日趋增多。从伊朗核设施爆发的震网病毒Stuxnet，到去年席卷全球的勒索软件Wannacry，都显示出了网络攻击行为对国家基础设施的严重威胁。在工业互联网中部署入侵检测系统（Intrusion Detection System，IDS）是保障网络安全的主要手段之一。IDS能够将网络特征进行分类，通过分类器识别网络入侵行为，进而采取措施保障网络安全。基于机器学习的人工智能技术以其强大的模式识别能力，在网络入侵检测方面被广泛应用[1-2]。但是，如何在复杂多变的网络攻击行为下提升入侵检测的分类精度，如何在低算力、分布式的工业控制网络中部署入侵检测系统，成为了工业互联网下网络入侵检测亟待解决的问题。因此，本文提出了一种基于集成学习和集成约减的网络入侵检测方法。

基金项目：国家科技重大专项核高基项目（No.2017ZX01030301）

集成学习由多个较为简单的弱分类器组合形成强分类器，通过弱分类器之间的互补性和多样性提升总体的分类性能。同时，相比于深度神经网络等单一分类器，集成学习以若干个弱分类器投票决策的方式组合在一起，能够更加方便的部署于分布式、低算力的工业控制网络节点中去。

本文组织结构如下：第二部分简要介绍了当前网络入侵检测和集成学习的相关研究；第三部分详述了本文所用的基于集成约减的入侵检测方法；第四部分基于所提方法进行了实验测试和分析；第五部分对本文进行了总结和展望。

2 相关研究

当前，网络入侵检测方法的研究多基于KDD99数据集展开，主要研究方法包括：基于人工神经网络（Artificial Neural Network，ANN）的方法[3]；基于贝叶斯网络的方法[4]；基于聚类的方法[5]；基于决策树的方法[6]；基于支持向量机（Support Vector Machine，SVM）的方法[7]；基于随机森林的方法[8-9]。传统的机器学习方法通常基于单一分类器对网络特征数据进行分类，分类精度相对较低。Vinayakumar等人基于卷积神经网络等深度学习方法，通过深度神经网络对网络行为数据进行特征学习，提取蕴含在网络状态数据中的内在特征，达到提升分类精度的目的[10-11]。但是，基于KDD99数据集包含大量冗余重复样例，严重影响了分类器的分类精度，使得入侵检测系统的异常检测能力难以得到客观的评价。因此，Tavallaee等人在KDD99数据集的基础上，删减了其中的冗余重复样本，形成了精简的NSL-KDD数据集，并基于典型的机器学习方法对其进行了分类测试[12]。在NSL-KDD数据集中，KDDtrain数据集为剔除冗余样本的训练集，共包含24中攻击行为；KDDtest数据集为剔除冗余样本的测试集，测试集中包含38种攻击行为；KDDtest21数据集为剔除KDDtest中易分类样本后的子集。

与单一的分类器相比，集成学习已经被证明具有更好的分类性能。其主要原因在于，多个分类器的结合可以消除基分类器之间不相关的分类误差，弥补彼此在识别特定模式时的缺陷，使集成后的分类器整体分类性能得到提升。随机森林（Random Forest，RF）是目前应用最为广泛的集成算法之一[13]。Tavallaee也在NSL-KDD99上应用了随机森林方法，结果显示基于随机森林的集成学习方法在入侵检测精度上明显优于其他分类器[12]。然而在随机森林中，许多分类树并不能提高集成分类器的性能，通常可以在不影响分类精度的前提下将其剔除，以适应有限计算和通信资源的工控网络场景。集成约减（Ensemble Pruning）通过优化集成分类器的多样性、分类精度指标，移除冗余重复的基分类器，以便在降低集成规模的同时保持甚至提升其整体的分类精度[14]。Martınez等人基于基学习器之间的互补性，采用顺序添加的方法寻找随机森林中的最优基学习器组合[15]，但该方法并不能保证约减子集的分类精度。Li等人则基于Kmean的聚类方法先将相似性较高的基学习器聚类成簇，并在每个类簇中挑选满足多样性指标的子集，以达到集成约减的目的[16]。Adam等人则基于遗传算法，通过最优化基学习器组合在训练集上的分类精度达到集成约减的目的[17]。但其以子集成的分类精度为适应度函数进行最优化求解，容易造成过拟合，对子集成的泛化性能造成了不利影响。因此本文用KDDtrain数据集进行模型训练，基于KDDtest进行集成约减，基于KDDtest21进行测试，结合多样性指标和遗传算法，开展了基于集成约减的网络入侵检测方法研究。

3 基于集成约减的网络入侵检测方法

3.1 多样性度量

多样性指标是集成学习中的一个重要特征，集成约减主要通过最优化不同的多样性度量达到缩小集成规模的目的。典型的多样性度量分为成对性度量和非成对性度量。成对性度量基于基学习器两两之间的混淆矩阵定义，如表1所示。主要包括Q统计量、不一致度量$Dis_{i,k}$、双次失败度量DF等，其定义如式1、式2、式3所示。非成对性度量包括κ度量、Kohavi-Wolpert方差KW等，其定义如式4、式7所示。

给定数据集D=｛x_1，x_2，x_3，…，x_n,｝，其中

表1 基学习器C_i和C_k的混淆矩阵

	C_i分类正确(1)	C_i分类错误(0)
C_k分类正确(1)	N_{11}	N_{10}
C_k分类错误(0)	N_{01}	N_{00}

n为数据集样本个数，基学习器对样本分类正确标记为1，分类错误标记为0。L为随机森林中基学习器的个数，$y_{j,i}$为第i个基分类器在样本x_j上的决策输出。$Dis_{i,k}$为第i个和第k个基学习器之间的不一致性度量。N_{00}，N_{10}，N_{01}，N_{11}为基学习器C_i和C_k在所有样本上的混淆矩阵。N_{00}代表两分类器均分类失败的样本个数，N_{11}代表两分类器均分类正确的样本个数，N_{01}代表C_i分类失败C_k分类正确的样本个数，N_{10}代表C_i分类正确C_k分类错误的样本个数，其满足$N_{00}+N_{10}+N_{01}+N_{11}=N$。式7中$l(x_j)$为随机森林在样本$x_j$上分类正确的基分类器数量。

$$Q_{i,k}=\frac{N_{00}\cdot N_{11}-N_{01}\cdot N_{10}}{N_{00}\cdot N_{11}+N_{01}\cdot N_{10}} \tag{1}$$

$$Dis_{i,k}=\frac{N_{01}+N_{10}}{N_{00}+N_{10}+N_{01}+N_{11}} \tag{2}$$

$$DF_{i,k}=\frac{N_{10}}{N_{00}+N_{10}+N_{01}+N_{11}} \tag{3}$$

$$\kappa=1-\frac{1}{2p_{av}(1-p_{av})}Dis_{av} \tag{4}$$

$$p_{av}=\frac{1}{NL}\sum_{j=1}^{N}\sum_{i=1}^{L}y_{j,i} \tag{5}$$

$$Dis_{av}=\frac{2}{L(L-1)}\sum_{i=1}^{L-1}\sum_{k=i+1}^{L}Dis_{i,k} \tag{6}$$

$$KW_{i,k}=\frac{1}{NL^2}\sum_{j=1}^{N}l(x_j)(L-l(x_j)) \tag{7}$$

3.1 现有的集成约减方法

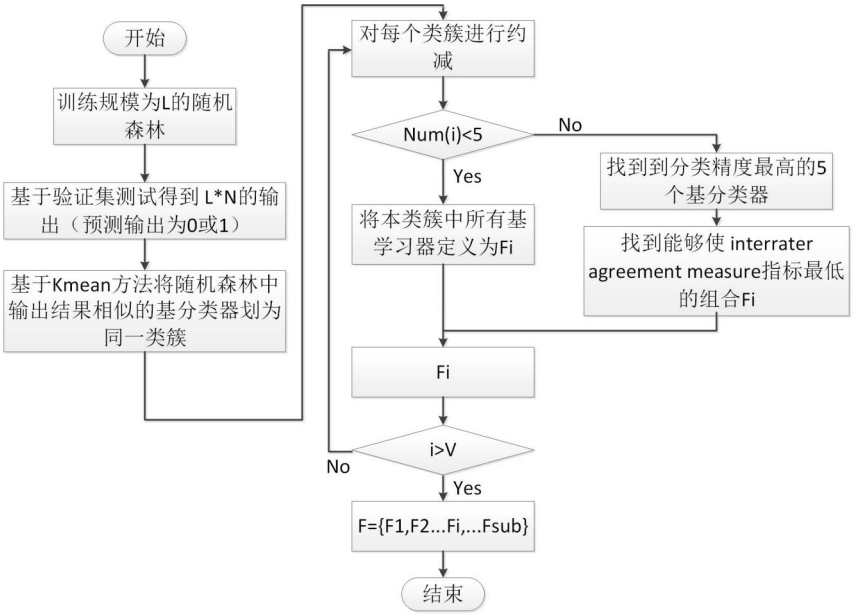


图1 基于K-mean聚类的集成约减方法

基于K-mean聚类的集成约减（Random Forest Kmean ensemble pruning， RF Kmean）过程如图1所示。其原理为：基于训练集训练规模为L的集成分类器，进而在验证集上得到二元化的预测输出矩阵，并基于每个基学习器输出结果进行Kmean聚类。通过对基学习器输出结果进行聚类，性能相似的基学习器将被划分至同一类簇，进而在每个类簇中基于多样性指标剔除性能相似的基学习器，最后得到约减子集Fi。其中V为定义的约减子集最小规模。

基于互补性指标的集成约减方法（Random Forest Diversity ensemble pruning， RF Div）如图2

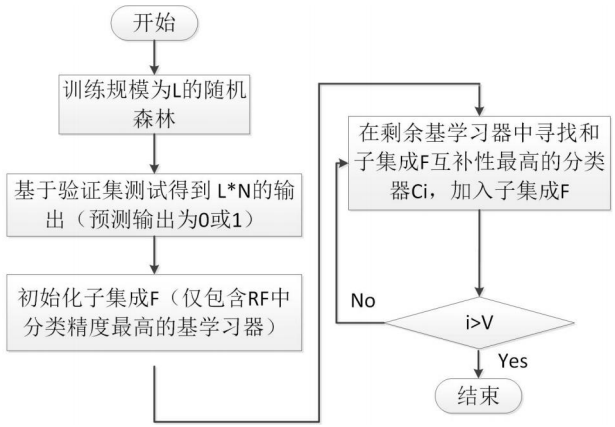


图2 基于互补性指标的多样性集成约减方法

所示。其原理为：基于训练集训练规模为L的集成分类器，进而在验证集上得到二元化的预测输出矩阵，依据验证集的输出矩阵寻找互补性最高的子集F，实现集成约减。互补性指标C的定义如式8所示，其旨在计算待测基学习器C_k与当前子集成F_i在验证集上输出的差异程度。

$$C=\sum_{i=1}^{N}(C_k(x_i)\neq F_i(x_i)) \tag{8}$$

3.2 基于遗传算法的集成约减原理

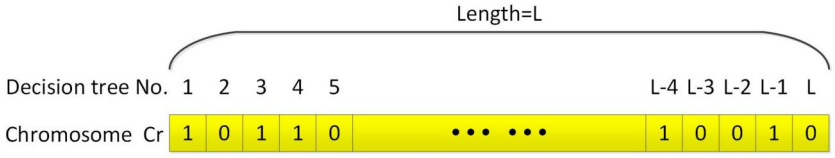


图3 RF GA集成约减方法中个体的编码方式

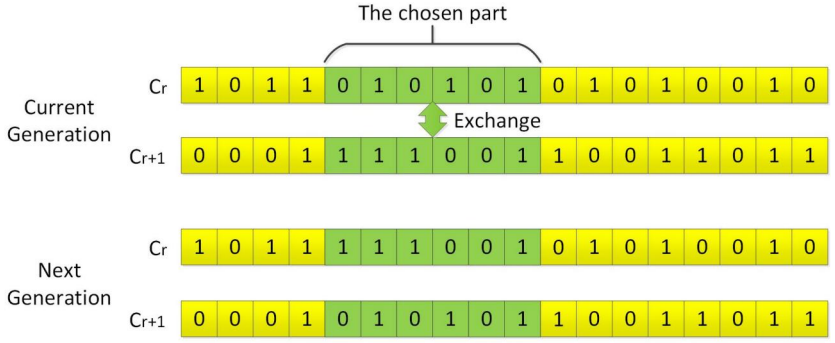


图4 RF GA方法中种群中基因的交叉操作

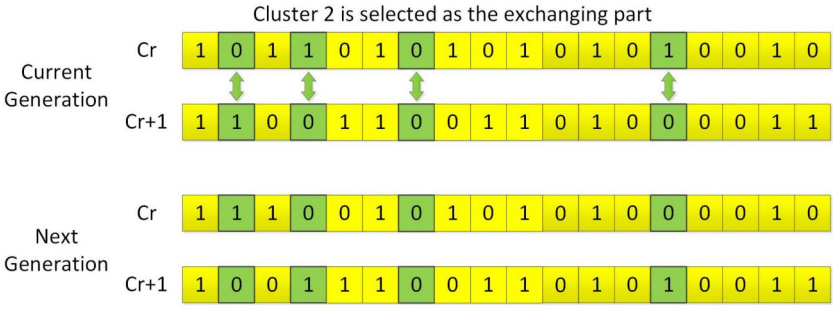


图5 RF GA方法中种群进化的变异操作

基于遗传算法的集成约减方法（Random Forest Genetic Algorithm ensemble pruning， RF GA）将集成中的每一个基分类器抽象成种群中个体上的一个二进制编码，编码为1代表在子集成中加入该基学习器，编码为0表示在子集成中剔除该基分类器，如图3~5所示。集成规模为L，则种群中每个个体有L个二进制编码。假设原始集成中包含10个基分类器，则原始集成的编码方式为Cr=“1111111111”。在种群进化过程中不断进行集成约减，剔除若干个基分类器后，Cr变为“1100010001”，表示第1、2、6、10个基学习器被选择保留在子集成中，其余六个基学习器被约减剔除。种群中每个个体的基因编码方式Cr即为一种约减子集成的组合方式。通过不断的交叉、编译等操作，最优化适应度函数，即可得到最优的个体（约减子集组合），达到集成约减的目的。本文以子集成的平均双次失败度量DF_{av}指标为适应度函数，使得种群不断进化，最终迭代得到最优的子集成组合，其过程如图6所示。

$$DF_{av}=\frac{2\sum_{i=1}^{L}\sum_{j=i+1}^{L}DF_{ij}}{L(L-1)} \tag{9}$$

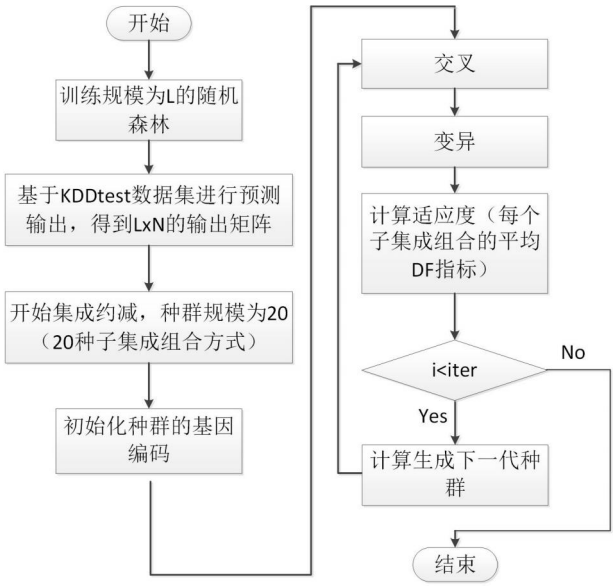


图6　RF GA 集成约减方法流程

4　实验及结果分析

4.1　实验环境和实验设定

本文采用NSL-KDD数据集进行实验，基于其中的KDDtrain数据集进行模型训练，基于KDDtest进行集成约减，基于KDDtest21对约减后的子集成进行测试。仿真环境matlab R2017b，PC为Win10操作系统，CPU为i7-8700，内存为32G。实验中采用分类准确率（Accuracy）、分类精度（Precision）、召回率（Recall）和F1-score指标评价集成约减算法的分类性能，通过子集成的基学习器数量衡量约减效果。实验中测试了随机森林规模从10到100递增过程中，几种集成算法在数据集上的分类性能和约减规模。

4.2　实验结果分析

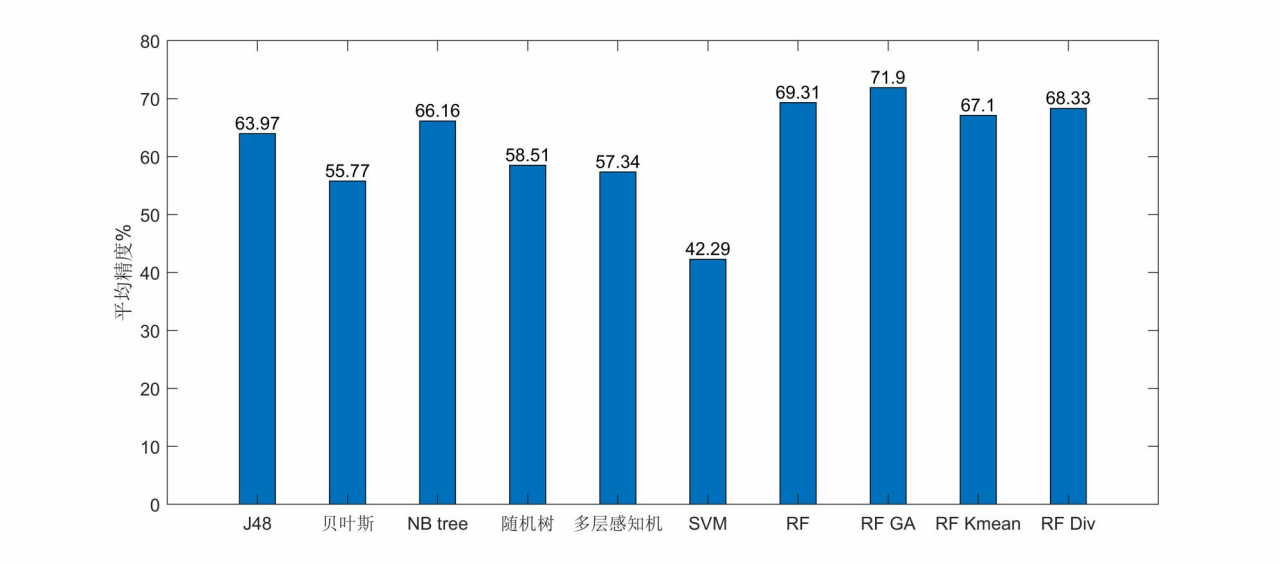


图7　不同分类方法的准确率

以随机森林基学习器数量L等于50时为例，与决策树、贝叶斯网络、多层感知机、支持向量机等分类方法相比，随机森林方法能显著提升网络入侵检测的分类精度，如图7所示。与现有的随机森林方法、聚类约减方法、互补性约减方法相比，本文所提出的RF-GA方法能够以原始集成一半的基学习器数量得到最高的分类精度，约减效果明显，如图7-8所示。

实验结果显示，随着基学习器数量的增加，集成分类器的分类性能逐步提升，但一味的增加基学习器数量并不总是能提升分类器的分类精度，如图9所示。在随机森林达到一定规模后，对其进行集成约减，四种分类性能指标均得到显著提升，基于遗传算法的集成约减方法亦明显优于随机森林方法、RF kmean、RF Div方法，提升算法在网络入侵检测数据集上的分类能力。

5　结论

本文提出了一种基于遗传算法和多样性指标的集成约减方法，以适应工业控制网络低算力、分布式应用环境下的网络入侵检测需求。实验结果表明，相比于现有集成约减算法，所提方法能

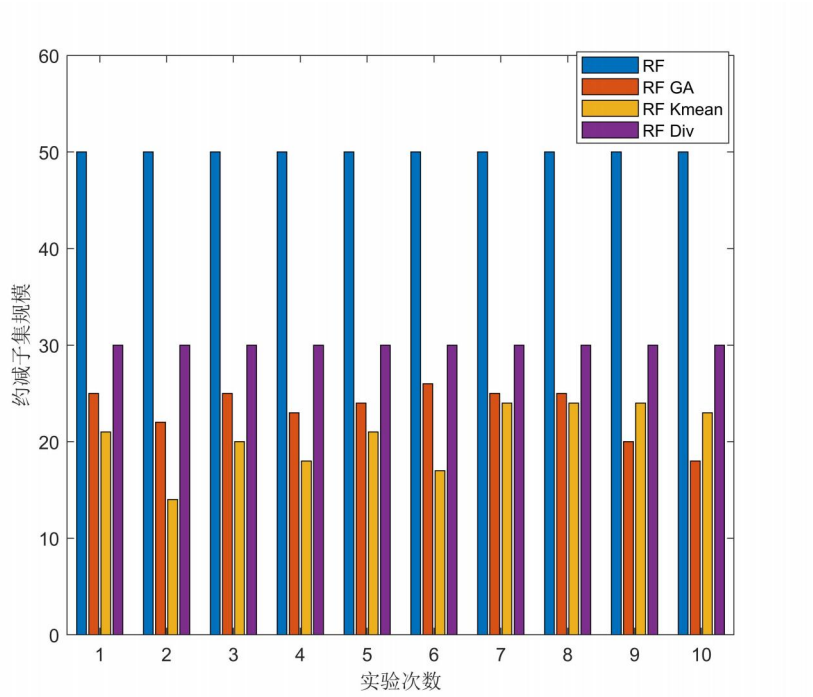


图8　不同集成约减方法的约减子集规模

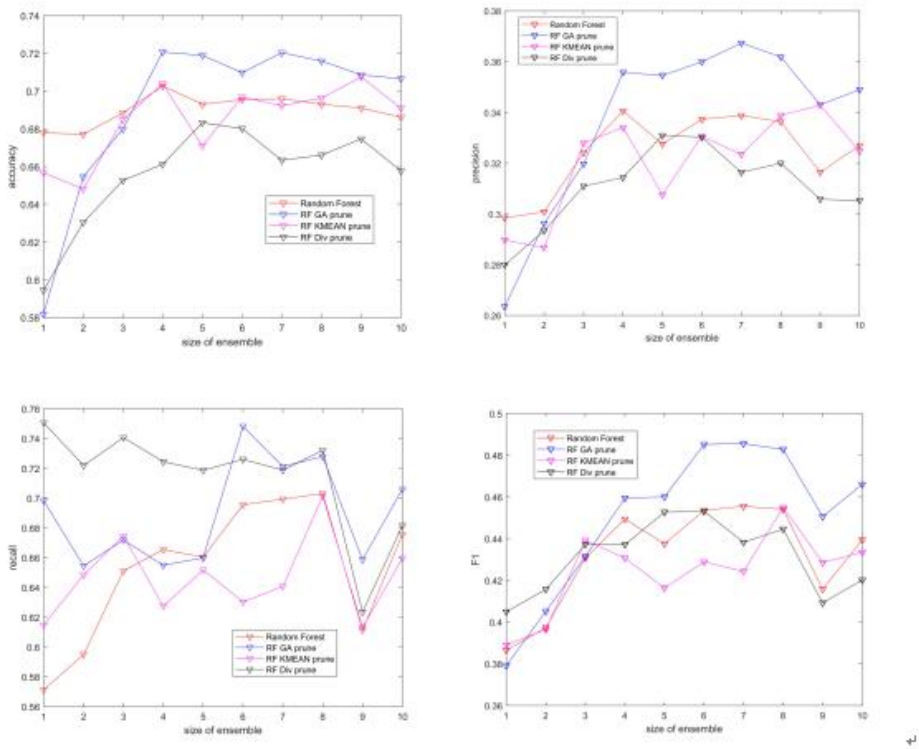


图9　不同森林规模下的约减子集分类性能

够在降低随机森林集成规模的同时提升分类精度，加强子集成对网络入侵行为的检测能力。但该集成约减方法与多样性指标的选取息息相关，后期应进一步研究不同多样性指标对集成分类器的影响。集成分类器的分布式部署也是本文未来的研究方向之一。

参考文献：（参考文献格式参照2015年新标准GBT 7714-2015）

［1］ Buczak， Anna， L， et al. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection［J］. Communications Surveys & Tutorials， 2016.

［2］ Bhuyan M H， Bhattacharyya D K， Kalita J K . Network Anomaly Detection： Methods， Systems and Tools［J］. IEEE Communications Surveys & Tutorials， 2014， 16（1）：303-336.

［3］ B. Morel. Artificial intelligence and the future of cybersecurity［C］//Proceedings of the 4th ACM workshop on Security and artificial intelligence， ACM， 2011.

［4］ Livadas C， Walsh B， Lapsley D， et al. Using Machine Learning Techniques to Identify Botnet Traffic［C］// IEEE Conference on Local Computer Networks. IEEE Xplore， 2006.

［5］ M. Blowers and J. Williams. Machine Learning Applied to Cyber Operations［J］. Network Science and Cybersecurity， 2014， 55： 155－175

［6］ Bilge L， Kirda E， Kruegel C， et al. EXPOSURE： Finding Malicious Domains Using Passive DNS Analysis［C］// Proceedings of the Network and Distributed System Security Symposium， NDSS 2011， San Diego， California， USA， 6th February - 9th February 2011. 2011.

［7］ Chitrakar R， Huang C . Selection of Candidate Support Vectors in incremental SVM for network intrusion detection［J］. Telecom Power Technology， 2014， 45（sep.）：231-241.

［8］ J Zhang， M Zulkernine and A Haque. Random-forests-based network intrusion detection systems［J］. IEEE Transactions on Systems， Man， and Cybernetics， 2008， 38（5）：649-659.

［9］ Gharibian F， Ghorbani A A . Comparative Study of Supervised Machine Learning Techniques for Intrusion Detection［C］//Conference on Communication Networks & Services Research. IEEE， 2007.

［10］ R Vinayakumar， K P Soman and P Poornachandran. Applying convolutional neural network for network intrusion detection［C］//2017 International Conference on Advances in Computing， Communications and Informatics （ICACCI）， Udupi， 2017， 1222-1228.

［11］ R Vinayakumar， K P Soman and P Poornachandran. Evaluating effectiveness of shallow and deep networks to intrusion detection system［C］// 2017 International Conference on Advances in Computing， Communications and Informatics （ICACCI）， Udupi， 2017， 1282-1289.

［12］ Tavallaee M， Bagheri E， Lu W， et al. A detailed analysis of the KDD CUP 99 data set［C］//IEEE International Conference on Computational Intelligence for Security & Defense Applications. IEEE， 2009.

［13］ Cutler A， Cutler D R， Stevens J R. Random Forests［J］. Machine Learning， 2004， 45（1）：157-176.

［14］ 孙博，王建东，陈海燕，等. 集成学习中的多样性度量［J］. 控制与决策， 2014， 000（003）：385-395.

SUN B， WANG J D， CHEN H Y， et al. Diversity measures in ensemble learning［J］. Control and Decision， 2014， 000（003）：385-395.

［15］ Martınez-Munoz G， Suárez A. Aggregation ordering in bagging［C］//Proc. of the IASTED International Conference on Artificial Intelligence and Applications. 2004， Citeseer， 258-263.

［16］ Li Z， Wang L， Shen P， et al. Fault Diagnosis of MVB Based on Random Forest and Ensemble Pruning［M］// Proceedings of the 4th International Conference on Electrical and Information Technologies for Rail Transportation （EITRT） 2019.

［17］ Adnan M N， Islam M Z. Optimizing the number of trees in a decision forest to discover a subforest with high ensemble accuracy using a genetic algorithm［J］. Knowledge-Based Systems， 2016， 110：86-97.

[作者简介]

李召召（1989-），男，博士，中级工程师，主要研究方向

为网络通信，异常检测，芯片设计。

李彧（1979-），男，博士，副高职称，主要研究方向为网络通信，异常检测，芯片设计。

孙远航（1982-），男，硕士，中级工程师，主要研究方向为网络通信，异常检测，芯片设计。

成诚（1984-），男，硕士，工程师，主要研究方向为网络通信，异常检测，芯片设计。

宋克（1976-），男，副研究员，博士生，主要研究方向为网络空间安全，芯片设计。

拟态公众号