基于预约束加密的可追溯加密恶意流量检测

张 迪,王 皓,张 颖,乐俊青,廖晓峰*
(重庆大学计算机学院,重庆 400044)

摘 要: 随着超文本传输安全协议(HyperText Transfer Protocol Secure,HTTPS)、传输层安全协议(Transport Layer Security,TLS)等安全通信协议的广泛部署,网络流量内容在获得有效保护的同时,也使依赖明文分析的恶意流量检测技术难以直接应用,给密态流量检测带来了新挑战。基于可搜索加密等密码技术的现有检测方案,通过半解密等操作能助力实现密态流量检测。然而,这类方案的中间盒可能借助检测过程推断检测规则,关联用户通信行为,挖掘用户隐私信息,容易引发严峻的隐私泄漏风险。此外,随着检测规则规模与流量分段Token数量不断增长,现有方案中大量使用的双线性配对、多轮交互或者复杂预处理等高开销操作导致计算与通信负担显著加重,难以满足实时检测与大规模部署需求。针对上述问题,本文提出一种基于预约束加密的可追溯加密恶意流量检测方案——CipherSight,旨在面向端到端通信场景,构建兼顾隐私保护与高效检测的主动检测机制。该方案首先结合集合预约束加密、轻量级布谷鸟哈希与对称加密技术,对检测规则与传输流量进行保护,在不泄露流量明文与规则语义的前提下实现恶意Token匹配。其次,基于签名机制对双向通信会话密钥进行承诺与绑定,使接收方能够快速验证并筛选其所需密文流量,从而避免对非目标流量进行冗余解密,提升解密阶段效率与可靠性。进一步地,基于Diffie-Hellman问题,在流量Token密文中隐式嵌入发送方身份信息,仅当流量匹配恶意规则时,中间盒才能恢复有效身份标识并实现精确溯源,而对合规流量仍无法获得任何有用信息。CipherSight在保证通信用户匿名性及恶意规则隐私性的前提下,实现了密态恶意流量的精确检测和恶意传播者的溯源。理论分析表明,所提方案基于不可区分性选择明文攻击(INDistinguishable Chosen Plaintext Attack,IND-CPA)安全的对称加密满足流量机密性,基于会话密钥签名的不可伪造性选择消息攻击(Existential UnForgeability under Chosen Message Attack,EUF-CMA)的安全性满足流量不可链接性,检测规则机密性则继承集合预约束加密的Element-hiding与Set-hiding安全性质。实验结果表明,CipherSight在密钥生成、规则生成、流量检测和流量解密等阶段均展现出较优性能:相较于BlindIDS,检测和解密效率分别提升40倍和15倍以上;相较于近期相关方案Pine与PrivBox,检测和解密效率均提升约5倍;在大规模规则与Token场景下,方案检测时间对规则规模变化不敏感,表现出更好的扩展性与实用性。综上,CipherSight为加密网络环境下实现高效、隐私保护的密态恶意流量主动检测和溯源提供了一种可行的新方案。

关键词: 加密流量;恶意流量检测;预约束加密;中间盒;可追溯

基金项目: 国家自然科学基金(No.62302072);重庆市自然科学基金创新发展联合基金(No.CSTB2023NSCQ-LZX0149)

中图分类号: TP309 **文献标识码:** A **文章编号:** 0372-2112(2026)04-1481-16

电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20251051

A Traceable and Encrypted Malicious Traffic Detection Scheme Based on Set Pre-Constrained Encryption

ZHANG Di, WANG Hao, ZHANG Ying, LE Junqing, LIAO Xiaofeng*
(*College of Computer Science*, *Chongqing University*, *Chongqing* 400044, *China*)

Abstract: With the widespread deployment of secure communication protocols such as hypertext transfer protocol secure (HTTPS) and transport layer security (TLS), while network traffic contents have been effectively protected, malicious traffic detection techniques that rely on plaintext analysis have become difficult to apply directly, posing new challenges to encrypted traffic detection. Existing detection schemes based on searchable encryption and related cryptographic techniques have facilitated the detection of malicious traffic over encrypted data through operations such as partial decryption. However, middleboxes in such schemes may infer detection rules during the detection process, correlate users' communication behaviors, and further mine users' private information, thereby causing severe privacy leakage risks. In addition, as the scale of detection rules and the number of traffic segmentation tokens continue to grow, the extensive use of high-overhead operations in existing schemes, such as bilinear pairings, multiple rounds of interaction, and complex preprocessing, significantly

收稿日期:2026-01-20;录用日期:2026-03-09;责任编委:程光
*通信作者:廖晓峰

increases computational and communication burdens, making it difficult to meet the demands of real-time detection and large-scale deployment. To address these issues, this paper proposes CipherSight, a traceable encrypted malicious traffic detection scheme based on set pre-constrained encryption, which aims to construct an active detection mechanism for end-to-end communication scenarios that achieves both privacy preservation and efficient detection. Specifically, the scheme combines set pre-constrained encryption, lightweight cuckoo hashing, and symmetric encryption to protect detection rules and transmitted traffic, enabling malicious token matching without revealing traffic plaintext or rule semantics. It further uses a signing mechanism to commit and bind bidirectional session keys, allowing the receiver to quickly verify and filter ciphertext traffic intended for itself, thereby avoiding redundant decryption of non-target traffic and improving the efficiency and reliability of the decryption phase. Moreover, based on the Diffie-Hellman problem, sender identity information is implicitly embedded into traffic token ciphertexts, so that only when traffic matches malicious rules can the middlebox recover a valid identity tag and achieve precise traceability, while no useful information can be obtained from benign traffic. CipherSight realizes accurate encrypted malicious traffic detection and malicious sender tracing while preserving communication user anonymity and malicious rule privacy. Theoretical analysis shows that the proposed scheme satisfies traffic confidentiality based on a symmetric encryption being indistinguishable chosen plaintext attack (IND-CPA) security, that the scheme satisfies traffic unlinkability based on the existential unforgeability under chosen message attack (EUF-CMA) security of the session key signature, and that the confidentiality of detection rules inherits the element-hiding and set-hiding security properties of set pre-constrained encryption. Experimental results demonstrate that CipherSight achieves strong performance in key generation, rule generation, traffic detection, and traffic decryption: compared with BlindIDS, it improves detection and decryption efficiency by about 40 times and more than 15 times, respectively; compared with recent related schemes such as Pine and PrivBox, it improves both detection and decryption efficiency by about 5 times; and in large-scale rule and token scenarios, its detection time is insensitive to changes in rule size, showing better scalability and practicality. Overall, CipherSight provides a feasible new solution for achieving efficient, privacy-preserving active detection and traceability of malicious traffic in encrypted network environments.

Keywords: encrypted traffic; encrypted malicious traffic detection; set pre-constrained encryption; middlebox; traceability

Foundation Item(s): National Natural Science Foundation of China (No.62302072); Natural Science Foundation of Chongqing (Innovation and Development Joint Fund) (No.CSTB2023NSCQ-LZX0149)

0 引言

随着5G网络的普及,互联网规模不断扩大,人们对移动流量中个人隐私数据的保护需求日益增强。截至2025年10月,Chrome等主流浏览器加速推进(HyperText Transfer Protocol Secure,HTTPS)的广泛使用,超过90%的用户使用加密技术来访问网络[1],以防止隐私泄漏,避免恶意中间人攻击等威胁。与此同时,高达80%的恶意软件和网络钓鱼等主要恶意流量也采用加密方式进行传输[2]。例如,利用安全套接层(Secure Sockets Layer,SSL)/(Transport Layer Security,TLS)协议等技术来实现加密传输,从而绕过传统的网络安全检测和拦截机制。然而,传统的基于内容识别、签名等技术的恶意流量分析技术难以直接应用于密态的恶意流量检测,导致网络安全问题面临新的威胁与挑战。

目前,根据密态恶意流量检测的主要思想,现有检测技术可以分为主动检测和被动检测。主动检测主要包括两类方法:一类是基于可明文检测技术的方案[3],通过对加密流量进行完全解密后再执行检测;另一类是基于可搜索加密技术的方案[4],在无需完全解密的情况下对加密流量进行关键词检索以实现检测。被动检测主要包括基于机器学习等人工智能技术,通过提取加密流量的特征,训练模型来实现恶意流量的识别[5-6]。在被动检测中,系统在不执行流量加密和解密操作的前提下实现检测,能较大程度保护用户隐私,但往往由于流量密态的随机性导致检测模型难以获得较高的准确率。在主动检测中,可明文检测技术最早利用中间人(Man in the Middle,MitM)技术开展解密检测[7]。随着硬件的发展,SGX-Box[8]、ShiledBox[9]与EndBox[10]主要依赖于可信执行环境,对加密流量进行解密并检测。然而,受限于可信执行环境的性能和规则,该类方法的检测效率普遍不高。可搜索加密技术是一种支持用户在密文上进行关键词查找的密码学原语,与可明文检测技术不同,该类检测方法不需要完全解密网络流量,而是在流量仍是加密的状态下,实现恶意信息关键词的检索。BlindBox[4]是第一个基于可搜索加密技术的加密流量检测方案,但其会话建立计算开销大,当检测规则数量达到3 000时,建立一次会话需要高达97 s。Embark[11]通过引入数据包头的前缀匹配,提升了加密恶意流量

的检测率,并维持网关和云之间的长期持久连接,降低了会话连接的计算开销,但其流量检测效率仍然较低。文献[12]、SplitBox[13]、ES-DPI[14]提出了将中间盒部署在云环境中,由可信中间盒转变为半可信中间盒,进一步提升流量的检测效率。BlindIDS[15]、PrivDPI[16]、SPABox[17]、Pine[18]、PrivBox[19]等方案通过设计可重用混淆规则来降低会话连接的计算开销,主要使用基于公钥双线性映射配对、离散对数,以及群运算等加密技术实现加密恶意流量检测。

然而,上述已有研究方案仍然存在效率低和隐私泄漏等局限性。具体而言,随着网络流量规模和检测规则集的增长,现有方案在流量的加解密和检测效率阶段仍面临高昂的计算开销,难以满足网络流量的实时性要求。同时,随着端到端加密技术的发展与应用,用户的隐私保护需求已从传统的传输数据保密性,延伸至对通信过程身份隐私的全面关注,而现有方案均未考虑对此隐私提供有效保护。即使部分中间盒方案只需要半解密即可实现密文流量检测,一定程度上保证了传输流量的机密性,但未对通信双方身份信息进行隐私保护,导致用户身份隐私泄露与行为轨迹可被关联分析。此外,这类方案也缺乏对恶意流量发起者的身份追溯与问责机制,无法在隐私保护与安全问责间实现平衡。因此,本文旨在解决以下重要挑战:如何在保护用户身份隐私、流量内容机密性及检测规则隐私性的前提下,实现高效快速的密态恶意流量检测,并提供恶意流量行为的可追溯性。

因此,本文提出了一个基于预约束加密的可追溯加密恶意流量检测方案 CipherSight,研究端到端场景下基于密码技术的恶意流量主动检测,在保障较高的密态流量检测准确率的前提下,实现用户匿名双向通信和恶意用户可追溯问责。为保护检测规则的隐私性并降低检测计算开销,CipherSight 结合轻量级 Cuckoo Hashing 技术和 Diffie-Hellman 元组思想,对检测规则进行轻量化加密存储。随后,为了实现匿名双向通信兼顾可追溯性,CipherSight 先结合基于集合预约束加密(Set Pre-Constrained Encryption,SPCE)和轻量级对称加密对流量和检测令牌(Token)进行加密,并在 Token 密文中隐式嵌入发送者身份信息。只有在传输的密态流量被检测出有恶意流量时,中间盒才能解密流量追溯出恶意传播者的身份信息。若流量合法,中间盒无法获取流量明文相关的任何有用信息和通信双方的身份信息,从而既保证了恶意行为的可追溯性,又保证了匿名通信的不可链接性。为进一步优化解密性能,CipherSight 采用 EdDSA 签名对通信会话密钥进行承诺,使得通信双方通过签名来确认流量密文来源正确可信,从而避免对非目标流量的冗余解密操作以降低开销。

1　相关工作

第一个基于中间盒解密的加密恶意流量检测方案[7],提出了“客户端-代理”“代理-服务器”的加密通信新模型。代理对经过的流量解密以进行深度包检测,并重新加密后发给服务器,既实现了高准确率的恶意流量检测,又保证了原有流量的特性。文献[20]提出了 maTLS 协议,将中间盒与两个端点之间的 TLS 会话拆分为独立片段,解决了中间盒与 TLS 共同使用时容易遭受的证书验证失败、使用过时密码套件、未经授权内容修改等安全问题。然而,由中间盒进行完全解密或引入共享密钥等操作,给流量带来了极大的隐私泄漏风险。随后,文献[8]、文献[10]和文献[9]分别提出了基于软件保护扩展(Software Guard Extensions,SGX)的可信执行环境的恶意流量检测方案,主要思想是将解密和检测操作放在可信执行环境中,取代中间盒的角色,通过可信执行环境实现计算和内存的隔离,既能保护数据隐私,又能确保检测的可信性。然而,这类方案受可信执行环境的性能和规则限制,通常难以获得较高的准确率。

研究者使用可搜索加密技术来实现半解密的加密恶意流量检测。这类方案的中间盒往往以代理的角色存在于客户端和服务器之间,维护令牌化的数据加密传输链路,并且不具备解密的权限或者密钥,从而在一定程度上加强了对流量的隐私保护。文献[4]提出了 BlindBox 方案,在隐私保护的前提下把深度包检测(Deep-Packet Inspection,DPI)转化为可搜索加密上的匹配,终端对流量提取可检索的令牌用于检测,并且通过可搜索加密与混淆电路共同保护规则与密钥。但其代价是每个会话都需要执行一次规则加密与握手准备,计算开销较大。文献[11]提出了 Embark 方案,通过前缀/范围匹配支持对包头字段进行加密检测,从而扩大了可检测范围,并且在系统开销和安全性方面均较 BlindBox 有所提升。文献[12]基于可搜索对称加密(Searchable Symmetric Encryption,SSE)与 Cuckoo Hashing 构建了一个加密规则过滤器,把对加密恶意流量的主动检测转化为 SSE 的加密令牌匹配,利用分隔符或者滑动窗口切分流量为字符串,只有在令牌匹配成功时触发检测,并且中间盒始终看不到明文与规则,不泄露流量与语义规则。文献[17]提出了 SPABox 方案,分别用 3 类可搜索加密技术实现对加密恶意流量的主动检测。其一,基于离散对数数学的等值可搜索加密,使得中间盒在看不到明文与密钥的条件下实现关键词匹配;其二,采用混淆电路负责加密检测规则,保护规则和流量的隐私;其三,结合机器学习和同态加密技术,在密文上完成线性分类。文献[15]提出了 BlindIDS 方案,可信方使用可解密的

可搜索加密方案(Decryptable Searchable Encryption, DSE)为攻击关键词生成陷门并下发给服务提供方,中间盒据此在密文Token上做等值匹配,无需恢复明文流量或明文规则,即可实现加密恶意流量检测。

针对BlindBox在独立短链接场景中效率低的问题,文献[16]提出了PrivDPI方案,利用可重用的混淆规则技术,使得首个会话生成的规则可直接在后续会话中复用,大幅压缩初始化握手阶段的计算与准备时间,尤其适用于处理海量短连接场景下的流量检测。文献[18]提出了能在TLS加密下实现主动检测的Pine方案,相比于PrivDPI方案,对规则进行隐藏,并支持轻量级动态更新规则而无需再次运行协议,大幅降低了通信开销。文献[21]将"规则模式与动作"封装成可查询陷门,扩展对称隐藏向量加密(Symmetric Hidden Vector Encryption, SHVE),中间盒直接在密文上做等值位置匹配,而无需滑动窗口切分Token。为了降低开销,该方案基于SHVE的加密过滤器先做两级前缀筛选,大幅减少后续匹配次数。文献[22]面向密文环境的基于关键字的深度包检测(Keyword-based Deep Packet Inspection, KDPI),提出了MT-DPI、CE-DPI、BH-DPI这3套协议,在不暴露明文与规则的前提下实现密文上的关键词匹配,从而主动检测加密恶意流量。文献[19]提出了PrivBox,用一种双重双掩码的规则混淆生成在密文上做关键词匹配,从而实现对加密恶意流量的主动检测,在保证与BlindBox同等级隐私的同时,将效率做到接近PrivDPI。文献[23]在不解密TLS的前提下,设计了客户端-服务器一致性校验与正义裁决机制,使用轻量级密码原语如对称加密算法、哈希函数、伪随机函数(Pseudo-Random Function, PRF)等将Token随机化,从而实现中间盒对加密恶意流量的主动检测。文献[24]提出了S^4E和AS^3E两种加密流量模式匹配方案,用于保护加密数据的隐私,同时实现对恶意流量的主动检测。文献[25]基于公钥密码体系与双线性映射配对提出了带有可平移陷门的可搜索加密方案SEST,发送方仅用接收方公钥对整个明文序列加密,中间盒获得任意长度关键字生成的陷门后即可在密文上滑动匹配并返回命中位置,不需要预先分词或由于不同长度而反复重加密。

为了将中间盒部署到云端,文献[13]提出了云端分布式检测方案SplitBox,把流量检测任务切分后分配到多个云节点上,通过协同计算完成整体网络检测。由于单个节点仅能接触到局部信息,系统在实现功能的同时保护了全局流量与策略的隐私。文献[14]提出基于两台互不合谋的云服务器的方案ES-DPI,将加密恶意流量的深度包检测转换为密文令牌匹配与可搜索加密索引的组合,从而避免了流量与规则明文的暴露。文献[26]提出了SlimBox方案,将流量与规则统一切分为固定长度的k-gram子串,将位置信息巧妙嵌入OXT子线性可搜索加密的交叉检索流程,实现常数时间复杂度的快速筛包,并通过盐化计数等降低带宽并减少泄漏。文献[27]在可信执行环境Intel SGX内实现可搜索加密的内容相似检测中间盒,结合局部敏感哈希(Locality Sensitive Hashing, LSH)和Cuckoo Hashing技术,不需要向云泄露明文与密钥,实现了并行检测与高效线程管理。文献[28]提出了一种基于动态DSE的安全深度包检测方案,用于外包的中间盒主动检测,同时保证了规则更新的前向和后向隐私。文献[29]提出了一种隐私保护的加密数据包头检测方案,确保在外包中间盒的主动检测过程中不会泄露隐私信息,并且使用模糊可搜索加密技术来优化检查效率。文献[30]提出了P2DPI方案,利用密钥同态伪随机函数(Key-Homomorphic PRF, KH-PRF)在用户与中间盒之间交换混淆的检测规则,从而使云中间盒在不需要解密TLS流量的前提下,通过令牌匹配实现对加密流量执行深度包检测。

在应用架构方面,文献[31]基于KH-PRF,提出了MRv-DPI方案,支持多个规则提供者,并且基于可信硬件Trusted Hardware支持检测结果的验证。文献[32]基于隐私集合求交协议与过滤器,提出了一个面向云辅助工业互联网的隐私保护型加密恶意流量检测方案,把流量与规则都Token化后在密文层面求交集,用交集是否为空来判断流量是否命中规则,从而无需解密流量即可完成检测。文献[33]利用两层过滤架构、Bloom Filter与基于可搜索加密的动态规则匹配技术,提出了DCDPI方案,在两层中间盒架构的基础上,能够实现对不连续Token的检测并对连续的Token进行分组归类。文献[34]面向5G数据平面与多接入边缘计算场景,采用可编程交换机和过滤与验证机制(FIltering-plus-VErification, FIVE)两阶段多模式匹配算法,提出一种"内置式"的高效DPI检测方案NetDPI,利用可编程交换机兼具转发与计算能力,在网络中直接完成检测。文献[35]针对现有恶意加密流量检测方法对标注数据依赖较强且易受流量混淆干扰的问题,提出一种结合语义属性矩阵表示、数据增强与对比学习预训练机制的鲁棒检测方法,实现了少样本条件下对恶意加密流量的高精度识别。文献[36]基于安全多方计算技术,提出了Obliv-FW架构,将防火墙黑白名单使用Bloom Filter技术表示后进行秘密共享,将规则分散存储到多台不同安全域服务器,并在不暴露规则的情况下协同完成数据包过滤。文献[37]针对现有恶意软件检测方法在特征提取不充分、过度依

赖单一图像或单一模态特征,以及数据不平衡影响分类效果等问题,提出了一种基于多维度动态加权alpha图像融合与特征增强的恶意软件检测方法,实现了恶意软件家族的高精度检测与分类。文献[5]针对开放网络环境下恶意流量检测中未知攻击难识别、增量更新易误差累计的问题,提出了一种基于双层模型和指标分布的持续检测与分类方法DMDI,在NIDS等公开数据集上体现出较好的检测精度、更新效果与边缘部署可行性。文献[38]基于频域分析与机器学习检测技术,提出了一个面向高吞吐网络的实时恶意流量检测系统Whisper,对零日攻击、低速攻击和多种规避攻击实现鲁棒检测。文献[39]提出了一个面向未知加密恶意流量的实施无监督检测系统HyperVision,在不依赖标注数据和先验攻击知识的情况下识别异常交互模式,实现对加密与非加密恶意流量的通用检测。

2　预备知识

本章主要介绍本论文所提方案涉及的密码原语以及所用到的符号定义。具体的符号定义如表1所示。

表1　符号定义

Table 1　Notation definitions

符号	含义	符号	含义	符号	含义
$\mathcal{M}$	明文空间	λ	安全参数	DHT	Diffie-Hellman元组
$\mathbb{G}$	阶为p的乘法循环群	$\mathcal{U}$	元素全集	SEnc	对称加密方案SE的加密算法
g	$\mathbb{G}$的生成元	$\mathcal{K}$	对称加密算法的密钥空间	SDec	对称加密方案SE的解密算法
h_0,h_1	哈希函数$h_0,h_1:\mathcal{U}\to[n']$	$\mathcal{ID}$	端点身份标识符全集	σ_i	会话密钥的签名
ssk	会话密钥	CT_i	流量密文	S_i,R_j	发送方和接收方
$(\mathrm{pk},\mathrm{sk})$	签名方案的密钥对	PP	公共参数	(mpk,msk)	流量加密与检测的密钥对
H_1,H_2,H_3	哈希函数$H_1:\mathcal{U}\to\mathbb{G}\setminus\{0\},H_2:\mathbb{G}\to\mathcal{K},H_3:\{0,1\}^*\to\{0,1\}^*$	$\mathcal{D}$	规则集	MB	中间盒

2.1　Diffie-Hellman问题

如果存在一个$\alpha\in\mathbb{Z}_q$使得$g_1=g^\alpha$且$g_3=g_2^\alpha$,那么元组(g,g_1,g_2,g_3)称为Diffie-Hellman元组,简称DHT。根据群$\mathbb{G}$上的判定性Diffie-Hellman(DDH)假设断言:一个随机的DHT和一组均匀随机的群元素在计算上是难以区分的[40]。根据文献[41],均匀随机选择的$\beta,\gamma\in\mathbb{Z}_q$,令$g_2'=g^\beta\cdot g_2^\gamma$且$g_3'=g_1^\beta\cdot g_3^\gamma$,有以下两个性质成立:

(1)若(g,g_1,g_2,g_3)是DHT,那么(g_2',g_3')也是DHT。

(2)若(g,g_1,g_2,g_3)不是DHT,那么(g_2',g_3')是全新的、均匀随机的群元素。

2.2　布谷鸟哈希(Cuckoo Hashing,CH)

一个CH[42]由以下两个算法组成

(1)$\mathrm{Setup}(\lambda,n,\epsilon)\to(n',h_0,h_1)$:初始化算法,输入参数$\lambda$、整数上界$n$以及$\epsilon\geqslant 0$,输出整数$n'$和两个哈希函数$h_0,h_1:\mathcal{U}\to[n']$,其中,$n'$是一个关于$\lambda,n$和$\epsilon$的确定性函数,$\mathcal{U}$为元素全集。

(2)$\mathrm{Hash}(h_0,h_1,\mathcal{D})\to(\mathcal{T})$:确定性哈希算法,输入哈希函数$h_0,h_1:\mathcal{U}\to[n']$和集合$\mathcal{D}\in\mathcal{U}$,输出表$\mathcal{T}=[T_1,T_2,\cdots,T_{n'}]$,其中,每个$T_i$要么是$\mathcal{D}$中的一个元素,要么是$\perp$。

CH的正确性要求如下:

(1)对于任意$x\in\mathcal{U}$,都有$h_0(x)\neq h_1(x)$。假设对于每一对(甚至是)由敌手选择的哈希函数,该条件都成立。

(2)表$\mathcal{T}$中每个非$\perp$的元素都是唯一的。

(3)对于任意的n,ϵ和大小为n的集合$\mathcal{D}\subseteq\mathcal{U}$,对于$(n',h_0,h_1)\leftarrow\mathrm{Setup}(\lambda,n,\epsilon)$,以下事件成立的概率为$1-\mathrm{negl}(\lambda)$:存在一个集合$\mathcal{D}'\subseteq\mathcal{D}$,满足$|\mathcal{D}'|\geqslant(1-\epsilon)|\mathcal{D}|$,并且对于$\mathcal{D}'$中的任意元素$x$,都有$T_{h_0(x)}=x$或$T_{h_1(x)}=x$成立,其中,$\mathcal{T}=\mathrm{Hash}(h_0,h_1,\mathcal{D})$。

2.3　预约束加密

一个集合预约束加密(Set Pre-Constrained Encryption,SPCE)方案[40]由以下3个算法组成

(1)$\mathrm{Gen}(1^\lambda,\mathcal{D})\to(\mathrm{pk},\mathrm{sk})$:参数生成算法,输入一个安全参数$1^\lambda$和一个大小至多为$n$的集合$\mathcal{D}\subseteq\mathcal{U}$,输出公钥pk和私钥sk,其中,$\mathcal{U}$为元素全集。

(2)$\mathrm{Enc}(\mathrm{pk},x,m)\to(\mathrm{ct})$:加密算法,输入公钥pk、元素$x\in\mathcal{U}$和消息$m\in\mathcal{M}$,输出一个密文ct,其中,$\mathcal{M}$为消息空间。

(3)$\mathrm{Dec}(\mathrm{sk},\mathrm{ct})\to(m,\perp)$:解密算法,输入私钥sk和密文ct,输出一条消息$m\in\mathcal{M}$或符号$\perp$。

2.4　EdDSA签名

本文采用的EdDSA签名[43]方案由以下3个算法组成。

(1)$\mathrm{KeyGen}(1^\lambda)\to(\mathrm{pk},\mathrm{sk})$:密钥生成算法,随机选择私钥$\mathrm{sk}\leftarrow\{0,1\}^l$,计算哈希$H(\mathrm{sk})=a_0,a_1,\cdots,a_{2l-1}$。

令$a_0=a_1=a_2=a_{l-1}=0, a_{l-2}=1$,记$H_L(\mathrm{sk})\to(a_0,a_1,\cdots,a_{l-1})$,$H_R(\mathrm{sk})\to(a_l,a_{l+1},\cdots,a_{2l-1})$。令$s=\sum_{i=0}^{l-1}a_i 2^i \bmod p$,那么$\mathrm{pk}=g^s$。

(2)$\mathrm{Sign}(\mathrm{sk},\mathrm{pk},m)\to(\sigma)$:签名算法,计算伪随机数$r\leftarrow H(H_R(\mathrm{sk}),m)\bmod p$,$R\leftarrow g^r$,$c\leftarrow H(R,\mathrm{pk},m)$,$S\leftarrow r+c\cdot s$,输出签名$\sigma\leftarrow(R,S)$。

(3)$\mathrm{Verify}(\mathrm{pk},\sigma,m)\to(0,1)$:签名验证函数,计算$c'\leftarrow H(R,\mathrm{pk},m)$,若满足$\left[g^s\right]^8=R^8+\left[pk^{c'}\right]^8$,则验证通过,输出1;否则输出0。

3 所提方案 CipherSight

本论文提出了一个基于SPCE的可追溯加密恶意流量检测方案CipherSight,本节主要介绍CipherSight的系统模型和具体构造。

3.1 系统模型

CipherSight的系统模型由4类实体组成,包括发送方(Sender,S)、中间盒(Middlebox,MB)、规则生成器(Rule Generator,RG)、接收方(Receiver,R),如图1所示。

(1)发送方S:又称为端点(Endpoint),和接收方的角色可互换,负责消息的加密和发送。

(2)中间盒MB:负责对经过的加密流量进行检测,并追溯出恶意流量的发送方。

(3)规则生成器RG:负责制定恶意流量规则,规则可能与敏感信息高度相关,因此需要对规则加密,防止隐私泄漏。

(4)接收方R:又称为端点(Endpoint),和发送方的角色可互换,负责消息的接收和解密。

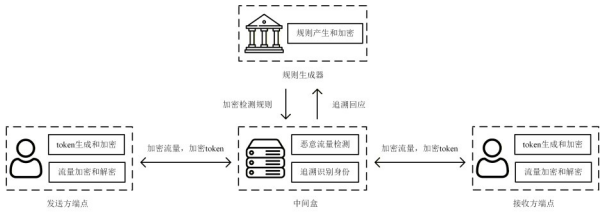


图1 系统模型

Figure 1 System model

通常情况下,发送方与接收方中最多有一方是恶意实体,如感染恶意软件的终端连接到恶意的目的地网络。本工作与现有方案一样,不考虑发送方与接收方均为恶意实体的情况。因为,如果两方均为恶意实体,他们可以协商新的密钥并使用其他隐蔽通信信道进行通信。RG是可信的诚实实体,它既不会向任何一方公开检测规则集合,也不会向端点授予规则访问权。MB是半诚实的实体,它忠实地提供检测服务,但可能会从经过的加密流量中挖掘敏感信息,或根据RG发送来的加密检测规则推断出规则相关的信息。另外,MB有可能被黑客入侵或窃听。

3.2 安全需求

流量机密性:系统针对流量明文m的加密方案$\Pi=(\mathrm{Setup},\mathrm{SEnc},\mathrm{SDec})$应满足不可区分性选择明文攻击(INDistinguishable Chosen Plaintext Attack,IND-CPA),即对任意概率多项式时间(Probabilistic Polynomial Time,PPT)敌手$\mathcal{A}$,即使$\mathcal{A}$能获得任意明文的加密结果,其仍无法有效地区分两条等长流量明文(m_0,m_1)的挑战密文对应的是哪一条明文。形式化地,敌手在IND-CPA挑战中的优势为$\mathrm{ADV}_{\mathcal{A},\Pi}^{\mathrm{IND-CPA}}(\lambda)=\left|\Pr[b'=b]-1/2\right|$对安全参数$\lambda$必须为可忽略函数。

流量不可链接性:系统应满足对于任意PPT中间盒敌手$\mathcal{A}$,即使可以观察到网络上所有传输的密文以及掌握所有公开信息,仅凭所见的密文信息而判断某段加密流量属于哪一对通信双方的能力应当是可忽略的。

检测规则机密性:规则产生器对检测规则$\mathcal{D}$进行加密后发送给中间盒,发送方在对流量m进行加密的过程中,隐式地将身份信息ID嵌入密文CT中,将CT传输给中间盒,中间盒对CT执行恶意流量检测功能。当CT对应的流量$m\notin\mathcal{D}$是合法时,发送方和中间盒不能推测出任何有关检测规则$\mathcal{D}$的相关信息。仅当流量$m\in\mathcal{D}$是恶意时,中间盒能够解密得到发送方的身份信息ID。

3.3 方案具体构造

CipherSight的整体工作流如图2所示,可以划分为5个阶段,分别是系统初始化阶段、密钥生成阶段、流量加密阶段、流量检测阶段与流量解密阶段。下面将按照阶段依次展开,对各环节的功能与执行流程进行详细说明。

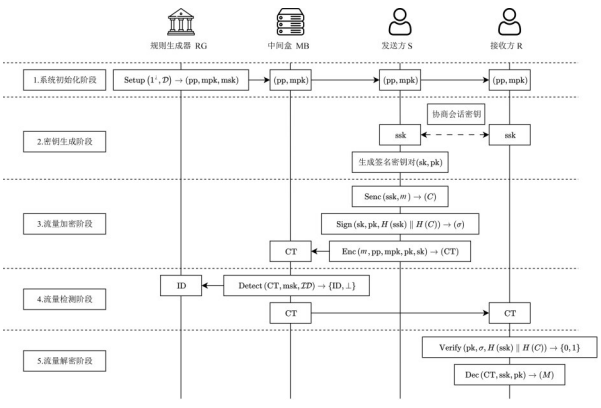


图2 CipherSight方案的工作流

Figure 2 Workflow of the CipherSight scheme

3.3.1 系统初始化阶段

$\mathrm{Setup}(1^{\lambda},\mathcal{D})\rightarrow(\mathrm{pp},\mathrm{mpk},\mathrm{msk})$:系统输入安全参数$\lambda$,初始化产生公钥mpk和私钥msk。首先,系统调用SPCE技术的Gen算法,初始化产生阶为p的乘法循环群$\mathbb{G}$,生成元为g。定义哈希函数$H_1:\mathcal{U}\rightarrow\mathbb{G}\backslash\{0\}$,$H_2:\mathbb{G}\rightarrow\mathcal{K}$,$H_3:\{0,1\}^{*}\rightarrow\{0,1\}^{*}$。规则生成器产生规则数据集$\mathcal{D}$,并调用布谷鸟哈希的Setup算法和Hash算法产生哈希函数h_0,h_1和表$\mathcal{T}\leftarrow\mathrm{Hash}(h_0,h_1,\mathcal{D})$。规则生成器随机选择$\alpha\leftarrow\mathbb{Z}_q$,计算$A=g^{\alpha}$,对于$i\in[n']$,如果$T_i=\perp$,那么选取$r_i\leftarrow\mathbb{Z}_q$,并且计算$\tilde{T}_i=g^{r_i}$;否则计算$\tilde{T}_i=H_1(T_i)^{\alpha}$。设$\mathrm{mpk}=\{h_0,h_1,A,\tilde{T}\}$,$\mathrm{msk}=\alpha$,$\mathrm{pp}=\{\mathbb{G},H_1,H_2,H_3\}$。

3.3.2 密钥生成阶段

$\mathrm{KeyGen}(1^{\lambda})\rightarrow(\mathrm{ssk},\mathrm{pk},\mathrm{sk},\mathrm{msk})$:通信双方协商确认会话密钥,并生成签名的密钥对,中间盒确认检测流量的密钥msk。令$\mathcal{ID}=\{\mathrm{ID}_i\}_{1,2,\cdots,N}$为网络中发送方和接收方的身份全集。首先,发送方$S_i$和接收方$R_j$协商会话密钥$\mathrm{ssk}_i=\mathrm{ssk}_j$。发送方$S_i$调用EdDSA签名的KeyGen算法,产生签名的密钥对$(\mathrm{pk}_i,\mathrm{sk}_i)$。规则生成器通过安全信道将密钥msk发送给中间盒以进行流量检测。

3.3.3 流量加密阶段

$\mathrm{Enc}(m,\mathrm{pp},\mathrm{mpk},\mathrm{pk},\mathrm{sk})\rightarrow(\mathrm{CT})$:流量发送方为会话密钥生成签名,并对流量分段加密,生成密文流量与密文Token。考虑流量的实时连续性,发送方S_i的流量m_i按照滑动窗口(例如8字节)分段,记为$m_i=\{m_{i1},m_{i2},\cdots,m_{il}\}$,并计算$x_{ij}=H_1(m_{ij})$,其中,$j\in\{1,2,\cdots,l\}$。发送方$S_i$利用对称加密方案对其流量进行加密,获得密文$C_i=\{c_{ij}=\mathrm{SEnc}(\mathrm{ssk}_i,m_{ij})\}_{j\in\{1,2,\cdots,l\}}$。随后,$S_i$使用EdDSA签名算法对会话密钥进行承诺,即调用其算法Sign计算会话密钥ssk_i的EdDSA[43]签名$\sigma_i=\mathrm{Sign}(\mathrm{sk}_i,\mathrm{pk}_i,H_3(\mathrm{ssk}_i)\|H_3(C_i))$,后续接收方根据会话密钥的签名验证进行加密流量的筛选。同时,发送方S_i为该流量的Token并进行加密:针对每一段流量m_{ij},随机选取$\beta_{bj},\gamma_{bj}\leftarrow\mathbb{Z}_q$,然后计算$Q_{b,ij}=g^{\beta_{bj}}x_{ij}^{\gamma_{bj}}$,$S_{b,ij}=g^{\alpha\beta_{bj}}\tilde{T}_{h_b(m_{ij})}^{\gamma_{bj}}$,$\mathrm{ct}_{b,ij}=\mathrm{SEnc}(H_2(S_{b,ij}),\mathrm{ID}_i)$,其中,$b\in\{0,1\}$,$j\in\{1,2,\cdots,l\}$。因此,流量$m_{ij}$的密文可以表示为$\mathrm{CT}_{ij}=\{\sigma,\mathrm{pk}_i,c_{ij},Q_{b,ij},\mathrm{ct}_{b,ij}\}_{b\in\{0,1\},j\in\{1,2,\cdots,l\}}$。

3.3.4 流量检测阶段

$\mathrm{Detect}(\mathrm{CT},\mathrm{msk},\mathcal{ID})\rightarrow(\mathrm{ID},\perp)$:中间盒MB对通道中的加密流量$\mathrm{CT}_i$进行检测,如果检测到恶意流量,则追溯出该段恶意流量的用户身份,否则输出空值。具体来说,针对加密流量CT_{ij},中间盒MB可解析获得$(Q_{b,ij},\mathrm{ct}_{b,ij})$,计算$\mathrm{ID}_i'=\mathrm{SDec}(H_2(Q_{b,ij}^{\alpha}),\mathrm{ct}_{b,ij})$,其中,$b\in\{0,1\}$,$j\in\{1,2,\cdots,l\}$。如果中间盒恢复出的$\mathrm{ID}_i'$是有效的,即属于全集$\mathcal{ID}$,那么说明该加密流量属于规则数据集$\mathcal{D}$,即该加密流量是恶意的。随后,中间盒把检测结果和$\mathrm{ID}_i'$反馈给规则生成方,以便后续针对该恶意流量进行处置或者追责。

3.3.5 流量解密阶段

$\mathrm{Dec}(\mathrm{CT},\mathrm{ssk},\mathrm{pk},\mathcal{ID})\rightarrow(M)$:中间盒MB把检测后不满足规则的加密流量广播给网络中与其有通道链接的其他用户端点,用户端点对密文中的签名进行验证后选择解密正确的密文流量。具体而言,接收方R_j收到加密流量$\{\mathrm{CT}\}_{i=1,2,\cdots,N}$后,先利用公钥$\mathrm{pk}_i$和自己的会话密钥$\mathrm{ssk}_j$来验证签名$\sigma_i$,即$\mathrm{Verify}(\mathrm{pk}_i,\sigma_i,H_3(\mathrm{ssk}_j)\|H_3(C_i))\rightarrow(0,1)$。若验证输出1,则验证通过,说明该条加密流量未被篡改且是发送方S_i想发送给接收方R_j;否则,不是。随后,接收方R_j对该加密流量进行解密,即$M_i=\{m_{ij}=\mathrm{SDec}(\mathrm{ssk}_j,c_{ij})\}_{j\in\{1,2,\cdots,l\}}$。

4 方案分析

4.1 正确性分析

4.1.1 检测的正确性分析

中间盒收到加密流量后,可先解析密文$\mathrm{CT}_{ij}=\{\sigma_i,\mathrm{pk}_i,c_{ij},Q_{b,ij},\mathrm{ct}_{b,ij}\}_{b\in\{0,1\},j\in\{1,2,\cdots,l\}}$,利用检测密钥$\mathrm{msk}=\alpha$,计算$H_2(Q_{b,ij}^{\alpha})$,再计算出与该流量关联的身份$\mathrm{ID}_i'=\mathrm{SDec}(H_2(Q_{b,ij}^{\alpha}),\mathrm{ct}_{b,ij})$。当流量$m_{ij}\in\mathcal{D}$时,查询表$\mathcal{T}$能找到$m_{ij}$的位置$h_b(m_{ij})$,则能计算出$\tilde{T}_{h_b(m_{ij})}^{\alpha}=H_1(T_{h_b(m_{ij})})^{\alpha}=H_1(m_{ij})^{\alpha}$。因此,计算$Q_{b,ij}^{\alpha}=(g^{\beta_{bj}}x_{ij}^{\gamma_{bj}})^{\alpha}=(g^{\beta_{bj}}H_1(m_{ij})^{\gamma_{bj}})^{\alpha}=S_{b,ij}$,那么就有$H_2(Q_{b,ij}^{\alpha})=H_2(S_{b,ij})$,即利用对称加密方案SE的解密算法$\mathrm{SDec}(H_2(Q_{b,ij}^{\alpha}),\mathrm{ct}_{b,ij})\rightarrow(\mathrm{ID}_i')$能恢复出一个有效的身份标识符属于集合$\mathcal{ID}=\{\mathrm{ID}_i\}_{1,2,\cdots,N}$。当流量$m_{ij}\notin\mathcal{D}$时,查询表$\mathcal{T}$中位置$h_b(m_{ij})$获得的$T_{h_b(m_{ij})}$为随机值,记为$\mathrm{rr}_{ij}$,那么$S_{b,ij}=g^{\alpha\beta_{bj}}\tilde{T}_{h_b(m_{ij})}^{\gamma_{bj}}=g^{\alpha\beta_{bj}}H_1(\mathrm{rr}_{ij})^{\gamma_{bj}}$。计算$Q_{b,ij}^{\alpha}=(g^{\beta_{bj}}x_{ij}^{\gamma_{bj}})^{\alpha}=(g^{\beta_{bj}}(H_1(m_{ij})^{\gamma_{bj}}))^{\alpha}$,显然$H_2(Q_{b,ij}^{\alpha})\neq H_2(S_{b,ij})$。此时,解密算法所得$\mathrm{ID}_i'$不是一个有效的身份标识符。

4.1.2 解密的正确性分析

接收方 R_j 收到的一系列的加密流量 $\{\mathrm{CT}_i\}_{i=1,2,\cdots,N}$，可解析为 $\mathrm{CT}_i=\{\sigma_i\mathrm{pk}_i,C_i,Q_{b,i},\mathrm{ct}_{b,i}\}_{b=\{0,1\}}$。调用算法 $\mathrm{Verify}\left(\mathrm{pk}_i,\sigma_i,H_3(\mathrm{ssk}_j)\|H_3(C_i)\right)$ 验证签名，显然只有当会话密钥 $\mathrm{ssk}_i=\mathrm{ssk}_j$ 时，签名才能通过验证。又因为流量明文 m_i 使用密钥 ssk_i 进行对称加密，那么对应的流量密文 C_i，使用密钥 ssk_j 进行解密操作，即可恢复出正确流量明文。

4.2 安全性分析

4.2.1 流量不可链接性

CipherSight 将一次会话的密文流量与会话密钥签名进行可验证绑定，发送方传输的流量密文 CT 中包含了会话密钥 ssk 的签名 σ，接收方收到 CT 后解析出签名 σ，利用自己的会话密钥 ssk 进行签名验证。签名验证通过时（即 $\mathrm{Verify}\left(\mathrm{pk},\sigma,H_3(\mathrm{ssk})\|H_3(C)\right)=1$），对应的密文 C 即是该接收者应接收的密文流量 C。随着网络中参与者的增加，通信的匿名性越强，通信流量的不可链接性也越强。具体来说，对于任意 PPT 中间盒敌手 $\mathcal{A}$，即使能够观察到网络中的所有密文及其附带的签名信息，仍无法有效判断某段密文流量属于哪一对通信实体，其区分优势应为可忽略函数。该性质的成立依赖于签名方案的不可伪造性选择消息攻击（Existential Unforgeability under Chosen Message Attack，EUF-CMA）。如果敌手 $\mathcal{A}$ 在不知道签名密钥 sk 的前提下，在能自适应询问签名生成预言机后伪造出一个新的有效签名，即其可以构造未被合法签名绑定的会话密钥 ssk′和伪造签名 σ'，使得 $\mathrm{Verify}\left(\mathrm{pk},\sigma',H_3(\mathrm{ssk}')\|H_3(C)\right)=1$ 成立。此时，敌手 $\mathcal{A}$ 便可以通过伪造绑定或试探验证等方式，将观测到的密文流量与自己选定的候选通信对关联，进而获得关于通信双方身份的可区分信息，破坏流量的不可链接性。因此，要求签名方案满足 EUF-CMA 安全性，从而保证敌手无法生成任何新的可通过验证的 (ssk,σ) 对。下面将通过 EdDSA 签名方案满足 EUF-CMA 安全，进而证明本系统的会话密钥签名机制也满足 EUF-CMA 安全。

定理 1 假设 EdDSA 是一个 EUF-CMA 的签名方案，那么，CipherSight 方案中用于承诺会话密钥的签名机制也是 EUF-CMA 安全的。

证明 如果存在一个 PPT 的敌手 $\mathcal{A}$ 能够以不可忽略的概率 ϵ 攻破 CipherSight 方案中签名机制的 EUF-CMA 安全性，那么，我们就可以构造另一个概率多项式时间的算法 $\mathcal{B}$，利用 $\mathcal{A}$ 作为子程序，以不可忽略的概率 ϵ 攻破底层 EdDSA 签名方案的 EUF-CMA 安全性。

我们首先定义 CipherSight 签名的 EUF-CMA 安全游戏，该游戏由一个挑战者（Challenger）$\mathcal{C}$ 和一个敌手 $\mathcal{A}$ 参与。

1. 初始化（Setup）：（1）挑战者 $\mathcal{C}$ 运行 EdDSA 的密钥生成算法 $\mathrm{KeyGen}_{\mathrm{EdDSA}}$，产生一个签名密钥对 $(\mathrm{sk},\mathrm{pk})$；（2）$\mathcal{C}$ 将 pk 发送给 $\mathcal{A}$。

2. 查询（Query Phase）：（1）$\mathcal{A}$ 被允许访问签名预言机（Signing Oracle）$\mathcal{O}_{\mathrm{sign}}(\mathrm{sk},\cdot)$；（2）$\mathcal{A}$ 可以自适应地选择 q 个消息 $m_1,m_2,\cdots,m_q$ 并提交给 $\mathcal{O}_{\mathrm{sign}}$；（3）对于每次查询 m_j，$\mathcal{C}$ 计算 $\sigma_i\leftarrow\mathrm{sign}(\mathrm{sk},\mathrm{m}_j)$，并将 σ_i 返回给 $\mathcal{A}$。

3. 伪造（Forge Phase）：在查询阶段结束后，$\mathcal{A}$ 输出一个它声称是伪造的消息-签名对 (m^*,σ^*)。

4. 获胜条件：敌手 $\mathcal{A}$ 在此游戏中获胜，当且仅当以下两个条件同时成立：（1）$\mathrm{Verify}(\mathrm{pk},\sigma^*,m^*)=1$；（2）$m^*\notin\{m_1,m_2,\cdots,m_q\}$。

$\mathcal{A}$ 获胜的概率记为 ϵ。如果对于所有 PPT 敌手 $\mathcal{A}$，ϵ 都是可以忽略的，则称该方案是 EUF-CMA 安全的。现在，我们构造算法 $\mathcal{B}$，它将尝试攻破 EdDSA 方案，$\mathcal{B}$ 会与一个外部的 EdDSA 挑战者 $\mathcal{C}_{\mathrm{EdDSA}}$ 交互，同时，$\mathcal{B}$ 会为 $\mathcal{A}$ 模拟上述的 EUF-CMA 游戏。

构造 $\mathcal{B}$ （1）Setup（B）：$\mathcal{B}$ 从其挑战者 $\mathcal{C}_{\mathrm{EdDSA}}$ 处接收到 EdDSA 的公钥 $\mathrm{pk}_{\mathrm{EdDSA}}$，$\mathcal{B}$ 的目标是伪造一个签名。（2）Setup（A）：$\mathcal{B}$ 将这个 $\mathrm{pk}_{\mathrm{EdDSA}}$ 作为公钥 pk 发送给 $\mathcal{A}$，启动与 $\mathcal{A}$ 的游戏。（3）Query（A）：当 $\mathcal{A}$ 向 $\mathcal{B}$ 发起对消息 m_j 的签名查询时：①$\mathcal{B}$ 无法自己计算签名；②$\mathcal{B}$ 将 $\mathcal{A}$ 提交的 m_j 原封不动地转发给他自己的挑战者 $\mathcal{C}_{\mathrm{EdDSA}}$；③$\mathcal{C}_{\mathrm{EdDSA}}$ 返回 m_j 的签名 σ_j；④$\mathcal{B}$ 将 σ_j 返回给 $\mathcal{A}$。（4）Forgery（A）：最终，$\mathcal{A}$ 输出一个伪造的 (m^*,σ^*)。（5）Forgery（B）：$\mathcal{B}$ 收到 (m^*,σ^*)，并将其作为自己对 $\mathrm{pk}_{\mathrm{EdDSA}}$ 的伪造结果，输出给挑战者 $\mathcal{C}_{\mathrm{EdDSA}}$。

如果 $\mathcal{A}$ 成功赢得了游戏，这意味着 $\mathrm{Verify}(\mathrm{pk},\sigma^*,m^*)=1$ 并且 $m^*\notin\{m_1,m_2,\cdots,m_q\}$。由于 $\mathcal{B}$ 只是将 $\mathcal{A}$ 的查询转发给了 $\mathcal{C}_{\mathrm{EdDSA}}$，并将 $\mathcal{A}$ 伪造的 (m^*,σ^*) 作为自己的输出，且 $\mathcal{A}$ 的公钥 pk 正是 $\mathcal{B}$ 需要攻击的公钥 $\mathrm{pk}_{\mathrm{EdDSA}}$。因此，如果 $\mathcal{A}$ 成功，那么 $\mathcal{B}$ 也必然成功。所以，$\mathcal{B}$ 攻破 EdDSA 方案的概率=$\mathcal{A}$ 攻破 CipherSight 签名机制的概率=ϵ。

CipherSight 所使用的 EdDSA 签名算法，基于椭圆曲线离散对数困难性假设，是 EUF-CMA 安全的。这意味着，任何 PPT 算法攻破 EdDSA 的 EUF-CMA 安全的概率 ϵ 是可忽略的。

综上所述，CipherSight 方案中使用的 EdDSA 签名满足 EUF-CMA。证毕

4.2.2 流量机密性

在 CipherSight 方案中，对称加密 SE = {KeyGen,

SEnc, SDec}用于加密实际的流量内容 m_i,得到密文 $C_i=\{c_{ij}\}$。此部分的安全目标是保护流量内容的机密性,使其免受半诚实中间盒 MB 的窃听。我们将证明 CipherSight 是 IND-CPA。

定理 2　假设 SE 是一个 IND-CPA 安全的对称加密方案,那么,CipherSight 中的流量加密阶段对于半诚实的中间盒是 IND-CPA 安全的,即中间盒无法从密文 C_i 中获取有关明文 m_i 的任何信息。

证明　如果存在一个 PPT 敌手 $\mathcal{A}_{CS}$ 能够以不可忽略的优势 ϵ 攻破 CipherSight 方案中流量的 IND-CPA 安全性,那么,我们就可以构造另一个 PPT 算法 $\mathcal{B}$。该算法 $\mathcal{B}$ 可以利用 $\mathcal{A}_{CS}$ 作为子程序,以相同的不可忽略优势 ϵ 破坏 SE 加密方案的 IND-CPA 安全性。

我们首先定义敌手 $\mathcal{A}$ 攻击一个对称加密方案的 IND-CPA 安全游戏,该游戏由挑战者 $\mathcal{C}$ 和敌手 $\mathcal{A}$ 参与。

1. 初始化(Setup):$\mathcal{C}$ 运行密钥生成算法 KeyGen_{Sym} 产生一个密钥 K。

2. 查询阶段(Query Phase 1):$\mathcal{A}$ 可以访问一个加密预言机(Encryption Oracle)$\mathcal{O}_{enc}(K,\cdot)$。$\mathcal{A}$ 可以提交任意明文 m_j,预言机返回密文 $c_j \leftarrow \mathrm{SEnc}(K, m_j)$。

3. 挑战阶段(Challenge Phase):(1)$\mathcal{A}$ 选择两个等长的挑战明文 m_0, m_1 提交给 $\mathcal{C}$;(2)$\mathcal{C}$ 随机选择一个比特 $b \in \{0,1\}$;(3)$\mathcal{C}$ 计算挑战密文 $c_b \leftarrow \mathrm{SEnc}(K, m_b)$,并将其发送给 $\mathcal{A}$。

4. 查询阶段(Query Phase 2):$\mathcal{A}$ 可以继续访问加密预言机(但不能查询 m_0 或者 m_1)。

5. 猜测(Guess):$\mathcal{A}$ 输出一个猜测 b'。

6. 获胜条件:敌手 $\mathcal{A}$ 获胜,如果 $b'=b$。$\mathcal{A}$ 的优势定义为 $\mathrm{ADV}_{\mathcal{A}}^{\mathrm{IND-CPA}}=\left|\Pr[b'=b]-1/2\right|$。如果对于所有 PPT 敌手 $\mathcal{A}$,该优势 ϵ 都是可忽略的,则称该方案是 IND-CPA 安全的。

现在,我们构造算法 $\mathcal{B}$,它将尝试攻破 SE 方案。$\mathcal{B}$ 会与一个外部的 SE 挑战者 $\mathcal{C}_{SE}$ 交互,同时 $\mathcal{B}$ 会为敌手 $\mathcal{A}_{CS}$ 模拟 CipherSight 的完整环境。

构造 $\mathcal{B}$　(1)Setup(B):$\mathcal{B}$ 与其挑战者 $\mathcal{C}_{SE}$ 开始游戏,$\mathcal{C}_{SE}$ 生成一个 $\mathcal{B}$ 不知道的 SE 密钥 K_{SE},访问其加密预言机 $\mathcal{O}_{enc}$。(2)Setup(A):$\mathcal{B}$ 为 $\mathcal{A}_{SE}$ 模拟 CipherSight 的系统环境,生成方案的所有公共参数 PP,将使用来自 $\mathcal{C}_{SE}$ 的 K_{SE} 作为其模拟的会话密钥 ssk。(3)模拟(Simulation):当 $\mathcal{A}_{CS}$ 期望观察流量 CT_i 时,它期望看到完整的密文结构 $\mathrm{CT}_i=\{\sigma_i, \mathrm{pk}_i, C_i, Q_{b,i}, \mathrm{ct}_{b,i}\}$:①$\mathcal{A}_{CS}$ 发起挑战,提交两个等长的挑战流量明文 m_0, m_1;②挑战(B):$\mathcal{B}$ 将 $\mathcal{A}_{CS}$ 提交的 m_0, m_1 转发给自己的挑战者 $\mathcal{C}_{SE}$;③ $\mathcal{C}_{SE}$ 随机选择 $b \in \{0,1\}$,计算挑战密文 $c_b \leftarrow \mathrm{SEnc}(K_{SE}, m_b)$,并返回 c_b 给 $\mathcal{B}$;④模拟响应:$\mathcal{B}$ 将收到的 c_b 作为对称加密部分 C_i;⑤同时,$\mathcal{B}$ 自行生成 CT_i 的其余部分;⑥$\mathcal{B}$ 将组装好的完整密文发送给 $\mathcal{A}_{CS}$。(4)猜测(A):$\mathcal{A}_{CS}$ 检查 CT_i,并输出其对 b 的猜测 b'。(5)猜测(B):$\mathcal{B}$ 将 $\mathcal{A}_{CS}$ 的猜测 b' 作为自己的猜测,输出给挑战者 $\mathcal{C}_{SE}$。

$\mathcal{B}$ 攻破 SE 方案的优势 $\mathrm{ADV}_{\mathcal{B}}^{\mathrm{IND-CPA}}$ 等于 $\mathcal{A}_{CS}$ 攻破 CipherSight 的优势 $\mathrm{ADV}_{\mathcal{A}_{CS}}^{\mathrm{IND-CPA}}$。CipherSight 采用 IND-CPA 安全的对称加密。因此,任何 PPT 敌手 $\mathcal{A}_{CS}$ 攻破 CipherSight 方案的优势 ϵ 也必定是可忽略的,方案的流量机密性得到了保障。证毕

4.2.3　检测规则的机密性

CipherSight 基于 SPCE 对检测规则进行加密。敌手可能通过自适应尝试 m_i 是否属于检测规则集合 $\mathcal{D}$ 来获取集合 $\mathcal{D}$ 的相关信息。所采用的 SPCE 技术具备 Element-hiding 安全性质,该性质指的是在仅获得公开参数、公钥与密文的情况下,对任意两对信息,敌手在计算上无法区分挑战密文究竟由哪个消息生成,即密文对规则元素保持隐藏,且同时隐藏与绑定的规则内容。同时,SPCE[40] 具备 Set-hiding 性质,即公钥不会泄露集合 $\mathcal{D}$ 的隐私信息,保证其机密性。因此,在 CipherSight 中,检测规则集合 $\mathcal{D}$ 的机密性由 SPCE[40] 的 Element-hiding 安全性质保障,且其形式化定义与安全性证明已经在文献[40]中给出并得到证明,因此本方案继承了 SPCE 结构,满足检测规则机密性。

在 CipherSight 中,规则生成器 RG 会将密钥 msk 通过安全信道发送给中间盒 MB,使得 MB 能实现恶意加密流量的检测和追溯。然而,MB 在获得 msk 后也将具备构造或修改检测集合 $\tilde{\mathcal{T}}$ 的能力。此时,MB 可能根据自身意图向检测集合中插入其他任意规则,从而破坏检测结果的正确性与可信性。为了解决这个问题,进一步引入认证签名机制,对检测集合的完整性与来源合法性进行约束,具体解决方案如下:引入一个满足 EUF-CMA 安全性的数字签名方案,记为 $\Pi_{sig}=\{\mathrm{Sig.Gen}, \mathrm{Sig.Sign}, \mathrm{Sig.Verify}\}$。规则生成器 RG 在 Setup 阶段通过算法 $(\mathrm{vk}_{RG}, \mathrm{sk}_{RG}) \leftarrow \mathrm{Sig,.Gen}(1^{\lambda})$ 生成认证签名密钥对,其中,sk_{RG} 由 RG 私有保存,vk_{RG} 作为公开验证密钥发布。随后,RG 生成检测规则集合 $\mathcal{D}$ 的加密检测集合 $\tilde{\mathcal{T}}$,并对其进行签名,即 $\sigma_{\tilde{\mathcal{T}}} \leftarrow \mathrm{Sig.Sign}(\mathrm{sk}_{RG}, \tilde{\mathcal{T}})$。RG 发布 $\{\mathrm{mpk}, \mathrm{vk}_{RG}, \sigma_{\tilde{\mathcal{T}}}\}$。在发送方执行流量加密与 Token 生成、加密之前,发送方首先对检测集合 $\tilde{\mathcal{T}}$ 及其签名 $\sigma_{\tilde{\mathcal{T}}}$ 进行验证 $\mathrm{Sig.Verify}(\mathrm{vk}_{RG}, \tilde{\mathcal{T}}, \sigma_{\tilde{\mathcal{T}}})$。如果通过验证,则说明当前检测集合 $\tilde{\mathcal{T}}$ 确实由合法的规则生成器 RG 生成,且在发布后未被中间盒 MB 篡改,发送方继续执

行后续的流量加密与Token加密过程；若没有通过验证，则说明检测集合可能被修改或者签名无效，发送方应立即终止协议并报告异常。通过引入签名机制，中间盒MB即使拥有主密钥msk，也无法修改检测集合$\tilde{\mathcal{T}}$。因此，该机制能够有效防止MB恶意篡改检测集合，保证检测规则集合的完整性、来源真实性以及后续检测过程的正确性。

4.3 性能分析

表2对比了本方案与现有方案在系统模型和功能上的差异。表3和表4分别基于理论分析对文献[4, 12, 15, 18, 19, 23]和所提方案CipherSight的计算开销和通信开销进行了具体对比。

表2 系统模型和功能对比

Table 2 Comparison of system models and functionalities

方案	Middlebox	双向通道	匿名性	隐私性	可追溯性
文献[4]	半诚实	√	×	√	×
文献[23]	半诚实	√	×	√	√
文献[12]	半诚实	√	×	√	×
文献[15]	半诚实	√	×	√	×
文献[18]	半诚实	√	×	√	×
文献[19]	半诚实	√	×	√	×
本方案	半诚实	√	√	√	√

表3和表4中，M表示规则数据集大小，N表示通信过程中将消息分段后的片段Token数目，g表示一次群幂次计算，p表示一次双线性映射配对计算，h表示一次哈希运算，s表示一次对称加密或解密计算，e表示一次签名计算，f表示一次PRF计算，x表示一次异或运算，c表示一次电路生成计算，–表示该方案不适用于该项对比。如表2所示，在计算开销方面，与文献[4]提出的BlindBox方案相比，在流量加密、解密计算复杂度相差不大的前提下，CipherSight在流量检测阶段大幅度降低了计算开销。与文献[12]提出的方案对比，在同样使用轻量级PRF计算的前提下，CipherSight在流量解密阶段通过减少对称解密的次数降低了计算开销。与文献[15]提出的BlindIDS方案对比，CipherSight不但在流量加密、流量检测、流量解密阶段移除了高计算开销的双线性映射配对计算，并且在流量检测阶段也大幅降低了计算开销。与文献[18]提出的Pine方案和文献[19]提出的PrivBox方案对比，CipherSight降低了流量检测阶段和解密阶段的群幂次计算次数，从而降低计算开销。在通信开销方面，如表3所示，所提方案CiphreSight与文献[4]、文献[23]、文献[12]、文献[15]、文献[18]、文献[19]在流量加密、流量检测、流量解密的通信复杂度上保持一致。表3和表4的对比结果表明，所提方案CipherSight无需高计算开销的pairing操作，在检测和解密阶段仅需低开销的群运算，且在检测过程中计算开销仅与Token数目有关，与检测规则数量无关。因此，相较于基于pairing操作的对比方案，CipherSight的计算开销具有显著优势；相较于不需要pairing操作的轻量化对比方案，CipherSight仍然具备较优的整体性能。

表3 计算开销对比

Table 3 Comparison of computational overhead

方案	初始化	密钥生成	流量加密	流量检测	流量解密
文献[4]	$\mathcal{O}(Mc)$	–	$\mathcal{O}(2Ns)$	$\mathcal{O}(2N\log Ms)$	$\mathcal{O}(2Ns)$
文献[23]	$\mathcal{O}(1)$	–	$\mathcal{O}\big(N(h+2f+x+s)\big)$	$\mathcal{O}\big(MN(h+x)\big)$	$\mathcal{O}\big(N(h+2f+x+s)\big)$
文献[12]	$\mathcal{O}\big(M(4f+x)\big)$	–	$\mathcal{O}\big(N(3f+2s)\big)$	$\mathcal{O}\big(N(3f+x)\big)$	$\mathcal{O}\big(N(3f+2s+h)\big)$
文献[15]	$\mathcal{O}(1)$	–	$\mathcal{O}\big(N(3g+x+3h+p)\big)$	$\mathcal{O}\big(MN(p+h)\big)$	$\mathcal{O}\big(N(2g+x+h+p)\big)$
文献[18]	$\mathcal{O}\big(M(f+g+e)\big)$	–	$\mathcal{O}\big(N(2f+g)\big)$	$\mathcal{O}(MNf)$	$\mathcal{O}\big(N(2f+3g)\big)$
文献[19]	$\mathcal{O}\big(M(6h+10g+4e)\big)$	–	$\mathcal{O}\big(N(f+3g+2h)\big)$	$\mathcal{O}(MNf)$	$\mathcal{O}\big(N(f+3g+2h)\big)$
本方案	$\mathcal{O}\big(M(2h+g)\big)$	$\mathcal{O}(1)$	$\mathcal{O}\big(N(e+3s+8g+3h)\big)$	$\mathcal{O}\big(N(2h+2g+2s)\big)$	$\mathcal{O}\big(N(s+e)\big)$

5 实验评估

5.1 实验设置

本文实验在Linux服务器的环境中完成，CPU为i9-12900KF，基准频率为3.2 GHz，内存大小为128 GB，操作系统为Ubuntu 22.04。CipherSight由Rust语言实现，采用了开源Cuckoo Filter库[44]，使用Montgomery曲线Curve25519，集成了EdDSA中的Ed25519签名算法（基于Edwards曲线edwards25519），对称加密技术采用AES。对比方案BlindIDS由C++语言实现，基于MCL开源库[45]并使用BN254椭圆曲线完成密码学计算。对比方案Pine、PrivBox使用Python语言实现，基于Charm-Crypto开源库并使用Prime256v1曲线。实验数据集来自文献[22]所公开的代码仓库中的数据集[46]。本文的性能核心目标在于提升中间盒单次检测的算法效率，单次检测算法效率的优化是多线程系统吞吐的基础和支撑。因此，我们采用上述配置的服

表4　通信开销对比

Table 4　Comparison of communication overhead

方案	初始化	密钥生成	流量加密	流量检测	流量解密
文献[4]	$\mathcal{O}(M)$	–	$\mathcal{O}(N)$	$\mathcal{O}(M+N)$	$\mathcal{O}(N)$
文献[23]	$\mathcal{O}(1)$	–	$\mathcal{O}(N)$	$\mathcal{O}(M+N)$	$\mathcal{O}(N)$
文献[12]	$\mathcal{O}(M)$	–	$\mathcal{O}(N)$	$\mathcal{O}(M+N)$	$\mathcal{O}(N)$
文献[15]	$\mathcal{O}(1)$	$\mathcal{O}(M)$	$\mathcal{O}(N)$	$\mathcal{O}(M+N)$	$\mathcal{O}(N)$
文献[18]	$\mathcal{O}(M)$	–	$\mathcal{O}(N)$	$\mathcal{O}(M+N)$	$\mathcal{O}(N)$
文献[19]	$\mathcal{O}(M)$	–	$\mathcal{O}(N)$	$\mathcal{O}(M+N)$	$\mathcal{O}(N)$
本方案	$\mathcal{O}(M)$	$\mathcal{O}(1)$	$\mathcal{O}(N)$	$\mathcal{O}(M+N)$	$\mathcal{O}(N)$

务器来模拟中间盒的计算环境,并在此环境下进行了CipherSight和对比方案的各算法运行时间测试。测试过程不考虑网络I/O、数据包接收/分发等网络因素,专注于测量核心算法的运行时间。

5.2　实验分析

本节对CipherSight与已有方案BlindIDS[15]、Pine[18]、PrivBox[19]进行实验仿真和性能测试,所有实验测试结果采取10次测试结果的平均值。实验测试主要针对CipherSight与3个对比方案在检测规则的生成、流量加密、加密流量检测、流量解密4个阶段的运行时间进行,以展示CipherSight在效率方面的优越性。

首先,我们将规则数目设为10个,Token数目设为1个,综合测试和对比了所提方案CipherSight与3个对比方案的密钥生成、流量加密、流量检测,以及流量解密4个算法的时间开销。由于BlindIDS、Pine与PrivBox对比方案并未在流程中定义独立的密钥生成阶段,为公平评估所提方案CipherSight的密钥生成性能,实验通过测量并累加各对比方案中签名的公私钥对生成、流量加密与解密密钥生成、检测规则加密密钥生成等核心准备步骤的运行时间,作为其等效密钥生成开销,以此进行密钥生成性能对比。对比结果如图3所示,CipherSight在计算效率上的表现显著优于对比方案。具体来说,在密钥生成阶段,CipherSight耗时约0.025 ms,低于PrivBox的0.64 ms,Pine的0.36 ms和BlindIDS的0.057 ms。在关键的流量加密与解密阶段,CipherSight加密耗时仅0.201 ms,较BlindIDS(0.446 ms)、Pine(0.457 ms)、PrivBox(0.407 ms),CipherSight的加密速度是3个对比方案的2倍有余。解密耗时仅0.027 ms,解密速度约为BlindIDS(0.405 ms)、Pine(0.332 ms)的2倍。在流量检测阶段,CipherSight耗时仅0.067 ms,相比BlindIDS(2.655 ms),检测效率提升近40倍,相较Pine(0.336 ms)和PrivBox(0.319 ms)提升5倍,表明CipherSight在检测阶段显著降低了计算开销,检测效率具有显著优势。

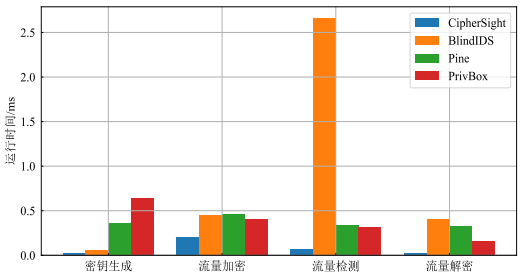


图3　方案各阶段运行时间对比

Figure 3　Comparison of running time across different phases

我们将进行所提方案CipherSight和3个对比方案的检测规则生成时间的测试和对比。在所提方案中,检测规则生成主要体现在系统初始化阶段;在BlindIDS方案中,主要体现在Setup和RuleGen阶段;在Pine方案中,主要体现在Setup、Preprocessing、Preparation of Session Detection Rule阶段;在PrivBox方案中,主要体现在Setup、Preprocessing和Session Rule Preparation阶段。测试过程在Token数量设为10 000时进行,累计各方案的各相关阶段的运行时间以进行对比。对比结果如图4所示,随着检测规则数目从1 000依次递增到10 000,所有方案的检测规则生成时间随着规则数目增长呈线性增长。当规则数目达到10 000时,CipherSight所需时间约为390 ms,相比BlindIDS的952 ms,效率提升约1.44倍;较Pine的5 096 ms,效率提升约13倍;而相较于PrivBox的21 540 ms,规则生成速度提升约55倍。总体而言,CipherSight显著提升了检测规则生成效率。

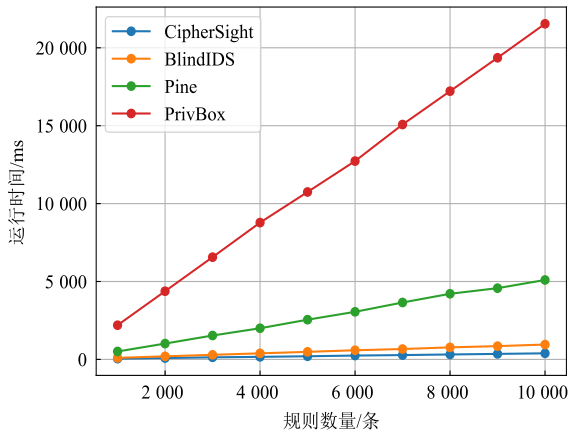


图4　检测规则生成时间对比

Figure 4　Comparison of detection rule generation time

令检测规则数目为10 000,分别对CipherSight和3个对比方案的流量加密阶段的运行时间进行测试与对比。对比结果如图5所示,所有方案的流量加密

时间随着Token数量由1 000递增到10 000呈线性增长。在Token数目达到10 000时，CipherSight的运行时间约为1 731 ms，低于BlindIDS的4 020 ms，效率提升了约1.3倍，稍高于Pine的873 ms和PrivBox的1 267 ms。这是因为在CipherSight的流量加密过程中采用了Cuckoo Hashing技术，将传输流量的用户身份标识隐式嵌入Token中，并对每一个Token进行两次加密，以实现对传输加密恶意流量身份的追溯。所以，CipherSight对比Pine、PrivBox方案在流量加密阶段时间开销稍高，但实现了恶意流量传播者身份追溯功能。

在3个对比方案中，只有BlindIDS是基于双线性配对的。根据图3的综合对比可知，方案BlindIDS的流量检测运行时间开销远高于采用轻量级原语的方案CipherSight、Pine、PrivBox。因此，我们将CipherSight分别与BlindIDS和Pine、PrivBox进行流量检测运行时间对比，对比结果如图6和图7所示。首先，根据图6，CipherSight展现了数量级的性能优势。具体来说，图6(a)展示了当检测规则数量为1 000时，Token数量从10依次递增到100，BlindIDS的检测时间高达23 813 ms，而CipherSight只需大约6.163 ms，效率提升约3 860倍，提升超过3个数量级。图6(b)展示了当Token数量为100时，检测规则数量从100递增到1 000，BlindIDS的检测时间快速增长到23 662 ms，而CipherSight只需大约7.685 ms，检测效率优势仍然保持3个数量级。图7(a)展示了规则数量为10 000时，Pine、PrivBox的检测时间随Token数量的增长曲线斜率呈递增趋势，而CipherSight基本保持线性增长趋势。Token数量递增到10 000时，Pine、PrivBox的检测时间上升至秒级(2.606 s和2.984 s)，而CipherSight检测时间仍然在毫秒级(614.878 ms)，其检测速度分别为Pine和PrivBox的近5倍。图7(b)展示了Token数量为10 000时，Pine、PrivBox的检测时间随规则数量的增长而增长，而CipherSight的检测时间几乎不受规则数量的影响。综上，CipherSight的检测效率受策略规模(即规则数与Token数)扩增的影响程度轻微，相较于对比方案具有显著优越性。

令检测规则数量为10 000，对CipherSight和3个对比方案的流量解密进行测试和对比。结果如图8所示，随着Token数量由1 000依次递增到10 000，4个方案的解密时间均随着Token的数量递增呈线性增长。当Token数目达到1 000时，CipherSight的解密时间约为243 ms，仅为BlindIDS(3 788 ms)的6.4%，解密速度提升约15.6倍；相较于Pine(1 527 ms)和PrivBox(1 269 ms)，解密速度提升约6.3倍和5.2倍。综上，CipherSight方案在流量解密方面显著优于对比方案。

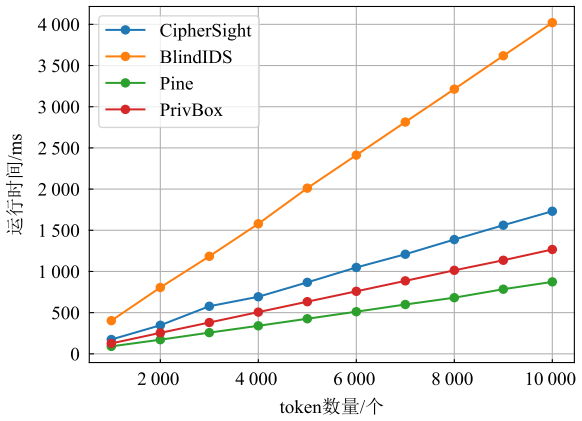
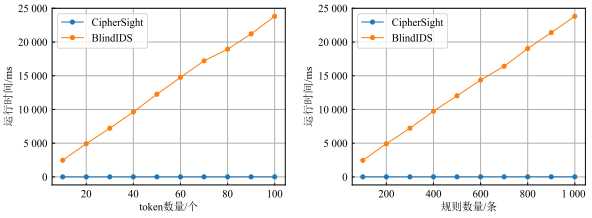


图5 流量加密时间对比

Figure 5 Comparison of traffic encryption time

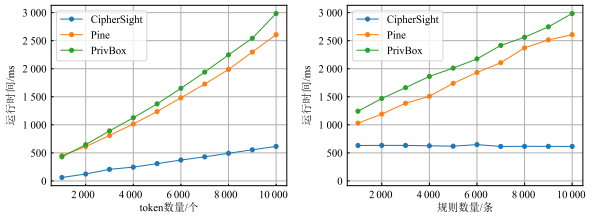


(a) 规则数量为1 000 (b) Token数量为100

(a) Number of rules = 1 000 (b) Number of tokens = 100

图6 CipherSight与BlindIDS的流量检测时间对比

Figure 6 Comparison of traffic detection time between CipherSight and BlindIDS



(a) 规则数量为10 000 (b) Token数量为10 000

(a) Number of rules = 10 000 (b) Number of tokens = 10 000

图7 CipherSight与Pine、PrivBox的流量检测时间对比

Figure 7 Comparison of traffic detection time among CipherSight, Pine and PrivBox

本节对CipherSight与已有方案BlindIDS[15]，分别针对8个流量文本数据集进行了实验仿真和测试，8个流量文本数据集格式相同，但在Token数目方面存在差异。实验测试结果如图9所示，显然，CipherSight在8个数据集上的加密和解密操作效率都显著优于对比方案BlindIDS。

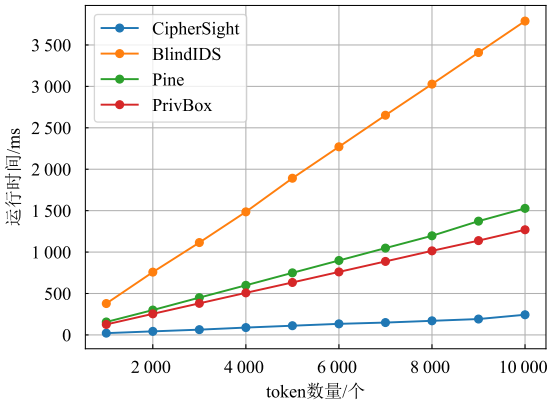
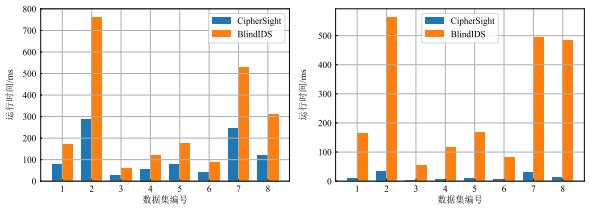


图 8　流量解密时间对比

Figure 8　Comparison of traffic decryption time



(a) 流量加密　　(b) 流量解密

(a) Traffic encryption　　(b) Traffic decryption

图 9　不同数据集下 CipherSight 与 BlindIDS 的加解密时间对比

Figure 9　Comparison of traffic encryption and decryption time between CipherSight and BlindIDS on different datasets

6　总结与展望

6.1　总结

本文面向端到端加密通信环境下的恶意流量主动检测需求,在 TLS 等加密机制广泛部署后,传统依赖明文载荷的 DPI 等难以直接开展检测;而现有“可检测密文”类方案往往在规则与流量、隐私保护与检测效率之间存在矛盾,难以同时满足合规审查、性能开销与可追责要求。针对上述矛盾,本文提出一种基于预约束加密的可追溯加密恶意流量检测方案 CipherSight。方案融合轻量级布谷鸟哈希与对称加密实现检测规则与流量的双重机密性保护;引入 EdDSA 签名对会话密钥承诺与密文解密过程进行约束,增强通信双方的不可否认性与解密可靠性;并基于 Diffie-Hellman 问题的困难性在 Token 加密的过程中隐式嵌入发送者身份信息,使系统在检测到恶意流量的同时能够实现精确溯源,为后续审计取证与责任认定提供依据。理论分析与实验评估表明,相比依赖双线性映射配对等现有方案,CipherSight 在核心流程中避免高开销计算,在显著提升检测效率的同时保持了较强的隐私保护能力,为加密网络环境下的安全监测提供了一种可行路径。

6.2　未来展望

本文所提方案 CipherSight 实现了可追溯的、不可链接性的匿名密态恶意流量检测,适配于端到端架构,但端到端架构对发送方生成 Token 的强依赖,导致其在公共互联网等非受控环境中的部署面临新的挑战。因此,CipherSight 在受控域内的端到端架构中部署可能更具优势,更适用于边界清晰、主体可管理、需要合规审计与责任追踪的应用场景,如企业园区网、零信任架构等此类通常存在可部署的中间盒与边缘网关以及明确的安全策略与合规边界的环境,以便于 CipherSight 发挥快速检测与可追溯问责的体系化价值。因此,面向完全开放的公共网络全局优化和高效部署端到端加密的恶意流量检测,是未来极具挑战且至关重要的研究方向。

尽管 CipherSight 方案在隐私性与效率方面展现出良好特性,但当前方案的双向通信匿名性依赖于系统中的端点数量,数量越多匿名隐私性越强,随之而来的是更高的带宽开销。未来可引入可信执行环境(Trusted Execution Environment,TEE)等技术,降低中间盒广播加密流量的带宽开销,是值得深入探讨的研究方向。此外,CipherSight 的核心工作在于提升中间盒单次检测的算法效率,而非优化其作为网络设备的网络吞吐量。单次检测算法效率的优化是多线程系统吞吐的基础和支撑,因此,如何将 CipherSight 的快速检测算法集成至高性能数据平台,并在真实流量下优化端到端的吞吐量与延迟,也是非常具有实际意义的研究方向。

参考文献

[1] Google. HTTPS encryption on the web[EB/OL]. [2025-11-14]. https://transparencyreport.google.com/https/overview?hl=en.

[2] Cisco. Encrypted Traffic Analytics with the New Cisco Network and Secure Network Analytics At-a-Glance[EB/OL]. [2025-11-14]. https://www.cisco.com/c/en/us/solutions/collateral/enterprise-networks/enterprise-network-security/at-a-glance-c45-740079.html.

[3] 侯剑, 鲁辉, 刘方爱, 等. 加密恶意流量检测及对抗综述[J]. 软件学报, 2024, 35(1): 333-355.
Hou Jian, Lu Hui, Liu Fangai, et al. Detection and countermeasure of encrypted malicious traffic: A survey[J]. Journal of Software, 2024, 35(1): 333-355. (in Chinese)

[4] Sherry J, Lan Chang, Popa R A, et al. BlindBox: Deep packet inspection over encrypted traffic[C]//Proceedings of 2015 ACM Conference on Special Interest Group on Data Communication. New York: ACM, 2015: 213-226.

[5] 陆浩天, 董育宁, 全宇轩. 一种基于双层模型和指标分布的恶意网络流持续检测和分类方法[J]. 电子学报, 2025, 53(5): 1637-1649.
Lu Haotian, Dong Yuning, Quan Yuxuan. A method for continuous detection and classification of malicious network traffic based on double-layer model and distribution of indexes[J]. Acta Electronica Sinica, 2025, 53(5): 1637-1649. (in Chinese)

[6] 周奕涛, 张斌, 刘自豪. 基于多模态深度神经网络的应用层 DDoS 攻击检测模型[J]. 电子学报, 2022, 50(2): 508-512.
Zhou Yitao, Zhang Bin, Liu Zihao. Application layer DDoS detection model based on multimodal deep learning neural network[J]. Acta Electronica Sinica, 2022, 50(2): 508-512. (in Chinese)

[7] Jarmoc J. SSL/TLS interception proxies and transitive trust[EB/OL]. (2012-03-14)[2025-11-14]. https://media.blackhat. com/bh-eu-12/Jarmoc/bh-eu-12-Jarmoc-SSL_TLS_Interception-WP.pdf.

[8] Han J, Kim S, Ha J, et al. SGX-Box: Enabling visibility on encrypted traffic using a secure middlebox module[C]//Proceedings of the 1st Asia-Pacific Workshop on Networking. New York: ACM, 2017: 99-105.

[9] Trach B, Krohmer A, Gregor F, et al. ShieldBox: Secure middleboxes using shielded execution[C]//Proceedings of Symposium on SDN Research. New York: ACM, 2018: 2.

[10] Goltzsche D, Rüsch S, Nieke M, et al. EndBox: Scalable middlebox functions using client-side trusted execution[C]//Proceedings of the 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. Piscataway: IEEE, 2018: 386-397.

[11] Lan Chang, Sherry J, Popa R A, et al. Embark: Securely outsourcing middleboxes to the cloud[C]//Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation. Berkeley: USENIX Association, 2016: 255-273.

[12] Yuan Xingliang, Wang Xinyu, Lin Jianxiong, et al. Privacy-preserving deep packet inspection in outsourced middleboxes[C]//Proceedings of the 35th Annual IEEE International Conference on Computer Communications. Piscataway: IEEE, 2016: 7524526.

[13] Asghar H J, Melis L, Soldani C, et al. SplitBox: Toward efficient private network function virtualization[C]//Proceedings of 2016 Workshop on Hot Topics in Middleboxes and Network Function Virtualization. New York: ACM, 2016: 7-13.

[14] Ren Hao, Litt H, Liu Dongxiao, et al. Toward efficient and secure deep packet inspection for outsourced middlebox[C]//Proceedings of 2019 IEEE International Conference on Communications. Piscataway: IEEE, 2019: 8761954.

[15] Canard S, Diop A, Kheir N, et al. BlindIDS: Market-compliant and privacy-friendly intrusion detection system over encrypted traffic[C]//Proceedings of 2017 ACM on Asia Conference on Computer and Communications Security. New York: ACM, 2017: 561-574.

[16] Ning Jianting, Poh G S, Loh J C, et al. PrivDPI: Privacy-preserving encrypted traffic inspection with reusable obfuscated rules[C]//Proceedings of 2019 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2019: 1657-1670.

[17] Fan Jingyuan, Guan Chaowen, Ren Kui, et al. SPABox: Safeguarding privacy during deep packet inspection at a middlebox[J]. IEEE/ACM Transactions on Networking, 2017, 25(6): 3753-3766.

[18] Ning Jianting, Huang Xinyi, Poh G S, et al. Pine: Enabling privacy-preserving deep packet inspection on TLS with rule-hiding and fast connection establishment[C]//Proceedings of the 25th European Symposium on Research in Computer Security. Heidelberg: Springer, 2020: 3-22.

[19] Wu Pengfei, Ning Jianting, Huang Xinyi, et al. PrivBox: Privacy-preserving deep packet inspection with dual double-masking obfuscated rule generation[J]. IEEE Transactions on Dependable and Secure Computing, 2025, 22(5): 4954-4970.

[20] Lee H, Smith Z, Lim J, et al. maTLS: How to make TLS middlebox-aware? [C]//Proceedings of the 26th Annual Network and Distributed System Security Symposium. San Diego, CA, USA: The Internet Society, 2019: 23547.

[21] Lai Shangqi, Yuan Xingliang, Sun Shifeng, et al. Practical encrypted network traffic pattern matching for secure middleboxes[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 19(4): 2609-2621.

[22] Wang Weicheng, Lee H, Huang Yan, et al. Towards efficient privacy-preserving deep packet inspection[C]//Proceedings of the 28th European Symposium on Research in Computer Security. Heidelberg: Springer, 2023: 166-192.

[23] Chen Dajiang, Wang Hao, Zhang Ning, et al. Privacy-preserving encrypted traffic inspection with symmetric cryptographic techniques in IoT[J]. IEEE Internet of Things Journal, 2022, 9(18): 17265-17279.

[24] Bkakria A, Cuppens N, Cuppens F. Privacy-preserving pattern matching on encrypted data[C]//Proceedings of

the 26th International Conference on the Theory and Application of Cryptology and Information Security. Heidelberg: Springer, 2020: 191-220.

[25] Desmoulins N, Fouque P A, Onete C, et al. Pattern matching on encrypted streams[C]//Proceedings of the 24th International Conference on the Theory and Application of Cryptology and Information Security. Heidelberg: Springer, 2018: 121-148.

[26] Liu Qin, Peng Yu, Jiang Hongbo, et al. SlimBox: Lightweight packet inspection over encrypted traffic[J]. IEEE Transactions on Dependable and Secure Computing, 2023, 20(5): 4359-4371.

[27] Yao Jing, Meng Xiangyi, Zheng Yifeng, et al. Privacy-preserving content-based similarity detection over in-the-cloud middleboxes[J]. IEEE Transactions on Cloud Computing, 2023, 11(2): 1854-1870.

[28] Guo Yu, Wang Cong, Jia Xiaohua. Enabling secure and dynamic deep packet inspection in outsourced middleboxes [C]//Proceedings of the 6th International Workshop on Security in Cloud Computing. New York: ACM, 2018: 49-55.

[29] Guo Yu, Wang Mingyue, Wang Cong, et al. Privacy-preserving packet header checking over in-the-cloud middleboxes[J]. IEEE Internet of Things Journal, 2020, 7(6): 5359-5370.

[30] Kim J, Camtepe S, Baek J, et al. P2DPI: Practical and privacy-preserving deep packet inspection[C]//Proceedings of 2021 ACM Asia Conference on Computer and Communications Security. New York: ACM, 2021: 135-146.

[31] Long Zhentao, Wu Pengfei, Zhang Kai, et al. Verifiable and privacy-preserving deep packet inspection for multiple rule service providers[C]//Proceedings of the 21st International Conference on Information Security and Cryptology. Heidelberg: Springer, 2025: 275-295.

[32] Feng Jingyu, Zhang Jing, Zhang Wenbo, et al. Detecting malicious encrypted traffic with privacy set intersection in cloud-assisted industrial internet[J]. Journal of Information Security and Applications, 2024, 85: 103831.

[33] Deng Minjun, Zhang Kai, Wu Pengfei, et al. DCDPI: Dynamic and continuous deep packet inspection in secure outsourced middleboxes[J]. IEEE Transactions on Cloud Computing, 2023, 11(4): 3510-3524.

[34] Zhou Chengjin, Xiang Qiao, Pu Lingjun, et al. NetDPI: Efficient deep packet inspection via filtering-plus-verification in programmable 5G data plane for multi-access edge computing[J]. IEEE Transactions on Mobile Computing, 2024, 23(12): 15031-15047.

[35] Shen Meng, Wu Jinhe, Ye Ke, et al. Robust detection of malicious encrypted traffic via contrastive learning[J]. IEEE Transactions on Information Forensics and Security, 2025, 20: 4228-4242.

[36] Allami A, Nicewarner T, Goss K, et al. Oblivious and distributed firewall policies for securing firewalls from malicious attacks[J]. Computers & Security, 2025, 150: 104201.

[37] 谢丽霞, 魏晨阳, 杨宏宇, 等. 基于多维度动态加权alpha图像融合与特征增强的恶意软件检测方法[J]. 电子学报, 2025, 53(3): 849-863.
Xie Lixia, Wei Chenyang, Yang Hongyu, et al. Malware detection method based on multi-dimensional dynamic weighted alpha image fusion and feature enhancement[J]. Acta Electronica Sinica, 2025, 53(3): 849-863. (in Chinese)

[38] Fu Chuanpu, Li Qi, Shen Meng, et al. Frequency domain feature based robust malicious traffic detection[J]. ACM Transactions on Networking, 2023, 31(1): 452-467.

[39] Fu Chuanpu, Li Qi, Xu Ke. Detecting unknown encrypted malicious traffic in real time via flow interaction graph analysis[C]//Proceedings of the 30th Annual Network and Distributed System Security Symposium. San Diego: The Internet Society, 2023: 1-18.

[40] Bartusek J, Garg S, Jain A, et al. End-to-end secure messaging with traceability only for illegal content[C]//Proceedings of the 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques. Heidelberg: Springer, 2023: 35-66.

[41] Naor M, Reingold O. Number-theoretic constructions of efficient pseudo-random functions[J]. Journal of the ACM, 2004, 51(2): 231-262.

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

[43] Xie Yumeng, Fan Qing, Zhang Chuan, et al. Accountable and secure threshold EdDSA signature and its applications[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 7033-7046.

[44] Anon. Cuckoo filter[EB/OL]. [2025-11-14]. https://github.com/axiomhq/rust-cuckoofilter.git.

[45] Anon. mcl[EB/OL]. [2025-11-14]. https://github.com/herumi/mcl.git.

[46] Anon. OpenSSL implementation integrating MT-DPI with TLS[EB/OL]. [2025-11-14]. https://github. com/mt-dpi/openssl-integration.git.

作者简介

张 迪 女,1993年11月出生于贵州省毕节市。现为重庆大学计算机学院副教授、硕士生导师。主要研究方向为隐私保护、区块链、应用密码学。

E-mail: dizhang@cqu.edu.cn

王 皓 男,2001年8月出生于辽宁省海城市。现为重庆大学计算机学院硕士研究生。主要研究方向为隐私保护、应用密码学。

E-mail: wang_hao@cqu.edu.cn

张 颖 女,2002年4月出生于重庆市。现为重庆大学计算机学院硕士研究生。主要研究方向为信息安全。

E-mail: zhang_ying@stu.cqu.edu.cn

乐俊青 男,1991年4月出生于江西省抚州市。现为重庆大学计算机学院副教授、硕士生导师。主要研究方向为隐私保护、联邦学习、信息安全。

E-mail: junqingle@cqu.edu.cn

廖晓峰 男,1964年10月出生于重庆市。现为重庆大学计算机学院教授、博士生导师。主要研究方向为神经网络、隐私保护、密码学。

E-mail: xfliao@cqu.edu.cn