适用于分布式 DHR 系统的可追溯直接匿名认证方案

陈立全[1,2], 张子燕[1], 羊子煜[1], 刘苏慧[1]

[1] 东南大学网络空间安全学院 南京 中国 210096
[2] 紫金山实验室 南京 中国 211118

摘要 作为拟态防御技术的基本实现模型, 动态异构冗余(Dynamic Heterogeneous Redundancy, DHR)系统在分布式场景下存在通信安全问题: 由于系统内缺乏对异构执行体的匿名保护措施以及诚实性度量方法, 异构执行体可能在未经察觉的情况下被攻击者入侵, 进而使得系统整体失效。将可信计算模块(Trusted Platform Module, TPM)引入分布式 DHR 系统可以缓解上述问题。然而, 现有 TPM 标准中使用的直接匿名认证(Direct Anonymous Attestation, DAA)方案会破坏分布式 DHR 系统的动态反馈机制, 因此无法直接应用于分布式 DHR 系统。为此, 本文对 DAA 方案进行改进, 提出了一种适用于分布式 DHR 系统的可追溯直接匿名认证方案(Traceable Direct Anonymous Authentication Scheme, Tra-DAA)。本方案在维持系统内异构执行体对外匿名的同时, 为各异构执行体增加了内部追溯参数, 兼容了 DHR 系统的动态反馈性。此外, 我们引入了委托计算技术, 将 TPM 中的计算量降到了理论最低值。安全分析证明 Tra-DAA 在 DL、DH、DBDH 和 LRSW 假设下具备安全性, 即实现了匿名、证书不可伪造以及签名不可陷害。理论分析表明 Tra-DAA 相比多种代表性 DAA 方案在 TPM 运算量上具备显著优势。实验结果表明, Tra-DAA 中新增的可追溯功能在整体耗时中仅占 5%, 且 Tra-DAA 的整体效率相比 TPM v2.0 中的 DAA 方案有显著提升。具体来说, 在 Join 协议、伪名为空的 Sign/Verifier 协议, 以及伪名不为空的 Sign/Verifier 协议阶段, TPM 的计算耗时分别缩短了 33%、50%与 70%。

关键词 拟态防御; 动态异构冗余; 直接匿名认证; 可信计算
中图法分类号 TP309.7 DOI 号 10.19363/J.cnki.cn10-1380/tn.2023.11.01

Traceable Direct Anonymous Authentication Scheme for Distributed DHR System

CHEN Liquan[1,2], ZHANG Ziyan[1], YANG Ziyu[1], LIU Suhui[1]

[1] School of Cyber Science and Engineering, Southeast University, Nanjing 210096, China
[2] Purple Mountain Laboratories, Nanjing 211118, China

Abstract As the basic implementation model of mimic defense technology, dynamic heterogeneous redundancy (DHR) system has a communication security problem in distributed scenarios: due to the lack of honesty measures and anonymity protection measures for heterogeneous executives, heterogeneous execution party may be invaded by an attacker without being detected, resulting in the failure of the entire system. Introducing the trusted platform module (TPM) into the distributed DHR system can alleviate the above problems. However, the direct anonymous authentication (DAA) scheme used in the existing TPM standard will invalidate the dynamic feedback of the distributed DHR system, so it cannot be directly applied. In this paper, we improve the DAA scheme and propose a traceable direct anonymous authentication scheme (Tra-DAA) for distributed DHR system. The Tra-DAA scheme maintains the external anonymity of the heterogeneous executives in the system, and configures internal traceability parameters for each heterogeneous executive so as to realize the compatibility with the dynamic feedback of the DHR system. Meanwhile, by introducing the technology of delegation of computation, the computation amount of TPM is reduced to the theoretical minimum. Security analysis proves that the Tra-DAA scheme is secure under the assumption of DL, DDH, DBDH, and LRSW, and realizes anonymity, unforgeability of certificate and unforgeability of signatures. Theoretical analysis indicates that the Tra-DAA has a significant advantage in the computation overhead of TPM compared with other representative DAA schemes. Experiments results show that the new traceability function in the Tra-DAA scheme accounts for only 5% of the overall time-consuming, and the overall efficiency of the Tra-DAA scheme is significantly improved compared with the DAA scheme in TPM v2.0. Specifically, during the execution of the Join protocol, the Sign/Verifier protocol with an empty

通讯作者: 陈立全, 博士, 教授, Email: Lqchen@seu.edu.cn。
本课题得到国家重点研发计划项目(No. 2020YFE0200600)资助。
收稿日期: 2022-03-09; 修改日期: 2022-05-25; 定稿日期: 2023-09-01

pseudonym, and the Sign/Verifier protocol with a non-empty pseudonym, operation times of TPM are shortened by 33%, 50%, and 70%, respectively.

Key words mimic defense; dynamic heterogeneous redundancy; direct anonymous authentication; trusted computing

1 引言

网络空间存在泛在化的不确定性威胁, 公众隐私保护、信息基础设施安全乃至网络空间秩序的稳定都急需有效的网络空间安全防御技术支撑。拟态安全防御技术是在充分考虑目标内部位置漏洞、后门攻击等多种安全威胁后, 通过模拟生物学的“拟态现象”, 构建出防御者控制下攻击者难以确定的动态变化环境[1-2]。其基本实现模型为动态异构冗余(Dynamic Heterogeneous Redundancy, DHR)架构, 可以抵御黑客对特定漏洞的攻击, 并规避由未知系统或硬件漏洞导致的系统异常[3]。

DHR 主要由异构执行体集合、裁决器和策略调度器组成。各异构执行体将结果输出至裁决器, 裁决器向策略调度器反馈裁决结果, 策略调度器据此进行异构执行体的清洗、筛选、替换。随着应用范围的扩大, 单机 DHR 系统逐渐无法满足用户的需求, 异构执行体开始分布于不同的应用场景, 形成了分布式 DHR 系统。

然而, 分布式 DHR 系统的安全面临挑战。异构执行体是攻击者经常针对的薄弱环节: 攻击者为了破坏拟态防御的主动防御特性, 会对异构执行体的数据进行监听, 以期实现渗透。若多个正在运行的异构执行体均为被入侵的不诚实设备, 则敌手可以在特定时间点改变这些异构执行体的输出信息, 导致多模裁决错误标记功能正常的异构执行体, 实现对分布式 DHR 系统的操控。在分布式应用场景下, 异构执行体与裁决器、策略调度器不再位于同一个安全的内网环境中, 既无法保证异构执行体与其他系统组件之间的交互安全可信, 又难以确保异构执行体诚实可信。

因此, 从安全角度考虑, 我们希望分布式 DHR 系统下组件间的交互方案满足: i) 保证异构执行体在数据交互的过程中保持匿名; ii) 实现对异构执行体诚实性的实时度量。具体而言:

1) 由于异构执行体与裁决器、策略调度器之间的通信安全难以保障, 需要通过身份认证来完成可信的数据交互。传统的基于身份信息的认证协议容易暴露通信实体的身份信息, 大大增加异构执行体被敌手追踪分析的风险。因此, 异构执行体在认证过程中需要匿名保护, 以实现重要身份信息的隐藏。

2) 现有构建方案大都建立在异构执行体可信的基础上, 如 Wang 等人[4]在概率分析和仿真实验中均以异构执行体安全可信为基础; Wu 等人[5]的安全分析给出异构执行体均诚实可信的假设。而分布式场景下, 难以度量异构执行体是否诚实可信。因此, 需要实现对异构体诚实性的实时度量。

针对上述问题, 目前现有的解决方案无法满足分布式 DHR 系统的安全需求。Chen 等人[6]基于椭圆曲线密码(Elliptic Curve Cryptosystem, ECC)和对称密码, 提出了面向分布式系统的接入协议。该协议可以抵御中间人、重放等常见攻击手段, 但未针对其他攻击进行说明, 且忽视了终端设备身份信息的隐私保护问题。Chen 等人[7]设计了一种安全网关来保证分布式系统中的传输安全, 并通过中间服务器来保证终端之间密钥交换和端到端通信的安全。然而, 该方案未考虑部分协议方被敌手攻破的场景, 且未对终端身份信息进行保护。Zhang 等人[8]给出了一种使用置信度来衡量异构执行体诚实性的方案, 然而置信度对异构执行体的评估在协议执行之后, 这种“先运行, 后修正”的思想与分布式 DHR 系统的动态变换特性相矛盾, 无法满足分布式 DHR 系统的安全需求。

为了解决分布式 DHR 系统中的安全问题, 我们试图将可信计算引入分布式 DHR 系统中。可信计算被认为是解决分布式系统安全问题最可靠方案[9]之一, 其通过在硬件设备中植入可信平台模块(Trusted Platform Module, TPM)芯片, 能够确保终端设备的完整性。将可信计算引入分布式 DHR 系统, 可以带来如下安全收益:

1) 直接匿名认证(Direct Anonymous Attestation, DAA)方案是可信计算中的常用认证技术, 通过零知识证明, 实现了不展示身份信息的身份认证。采用 DAA 方案, 可以使得异构执行体在与其它系统组件交互时不泄露身份信息, 从而满足异构执行体的匿名性需求。

2) TPM 芯片可以通过度量与验证来判断异构执行体是否按照设计者的预期运行, 并且可信平台提供了异构执行体诚实性的可靠度量方案。

然而, DAA 的完全匿名特性会打破 DHR 动态反馈的有效性: 由于无法实现异构执行体的定位, 策略调度器无法根据裁决器的反馈信息及时对异构执

行体进行调整。因此, 将 DAA 方案直接应用于分布式 DHR 系统是不可行的。为此, 如何在不破坏 DAA 安全性、保持异构执行体匿名性的前提下, 设计并实现异构执行体的身份信息追溯, 从而实现对异常异构执行体的定位, 仍是一个具有挑战的研究问题。此外, 由于分布式 DHR 系统的动态、异构、冗余等特性, 其在运行时需要占用大量的系统资源, 现有的 DAA 方案与分布式 DHR 系统存在冲突。因此, 如何改进 DAA 方案, 设计适用于分布式 DHR 系统的安全、高效的认证方案, 是本文研究的主要目标。

本文的贡献包括两个方面:

1) 将 DAA 方案应用于分布式 DHR 系统, 提出了一种适用于分布式 DHR 系统的可追溯直接匿名认证 (Traceable Direct Anonymous Authentication Scheme, Tra-DAA)方案, 解决了 DAA 方案的完全匿名特性与 DHR 的动态反馈需求之间的矛盾。通过将一致性裁决匿名化, 使得针对裁决器的信息窃取无效; 同时本方案在签名中加入了仅策略调度器可解析的追溯参数, 使得策略调度器可以动态追溯到异常的异构执行体, 实现了 DHR 架构可追溯的功能; 通过引入委托计算技术, 提出的方案将 DAA 方案中的伪名计算从 TPM 转移至计算能力更强的 Host 中, 实现了整体运行效率的提升。

2) 在理想现实模型和随机预言模型下, 本文证明了 Tra-DAA 满足匿名、证书不可伪造和签名不可陷害的安全需求; 综合理论分析与实验验证, 通过与其他方案进行对比说明了 Tra-DAA 在运行效率上的优越性。

2 DAA 方案的相关研究

Brickell 等人在对文献[10]的修改中提出了 DAA 的两大安全需求, 即用户可控匿名性和用户可控可追踪性。同时, 他还提出了 DAA 的主要优化方向: 针对计算能力最为薄弱的 TPM 芯片进行效率优化。而后, 基于 DAA 的 BCC-DAA[11]方案被首次提出, 其安全性基于 CL 签名[12]。然而, BCC-DAA 仍然存在许多问题: Sign 阶段并未生成 TPM 密钥的签名, 没有给出可控匿名的具体方案[13], 基于 RSA 的 CL 签名严重降低了协议的运行效率等。椭圆曲线密码 ECC 算法在密钥长度及计算量上较 RSA 具备优越性[14], Brickell 等人提出的 BCL-DAA[10]首次改用 ECC 算法, 在不降低安全性的前提下极大提升了计算效率, 大幅缩短了私钥和签名长度。然而, 由于没有改变协议的交互机制, 该方案仍存在类似 BCC-DAA 方案的安全问题。

为此, Chen 等人提出的 CMS-DAA[15-16]采用非对称双线性对生成的 CL 签名以提高安全性, 但文献[17]随后指出该方案无法在主机被攻破的情况下保证匿名可控关联性。针对 DAA 方案中一直存在的安全模型的问题, Chen 等人[18]对 CMS-DAA 中的安全模型进行了修改, 将 Sign/Verify 协议区分开并给出了在随机预言模型下完整的形式化证明。然而, 该方案违背了 DAA 的性能优化思路。随后, Chen 在文献[18]的基础上提出了被选为新国际标准[19]的改进方案: Issuer 使用 TPM 的承诺值来生成 DAA 证书, 去除 TPM 的重复计算, 并提供了一种新的平台假冒检测思路[16]。然而, 在该方案中的敌手可以使用特殊的证书来通过安全验证。

TPMv2.0 规范[20]对安全问题进行了修正, 舍弃了不安全的 SHA-1 算法, 增加了对 AES、ECC、SHA-256 等多种算法的支持, 更加安全、高效、灵活, 但仍然存在部分安全性证明不完善等问题。Camenisch 等人给出了 TPMv2.0 的代表性修正方案[21], 该方案以较小的改动解决了已发现的安全缺陷, 因此被 TCG 组织采用。

在运行效率的改进上, Chen 等人[22]在文献[18]的基础上提出了效率更高、签名长度更短的方案: 通过将批量处理技术与多次双线性运算合并, 减少了主机的计算量; 在 Sign 阶段, 使用合并计算将 TPM 在 Sign 阶段的计算量降低到两次指数运算; 在不需要关联性签名时置空伪名, 进一步降低 TPM 的计算负担。此后, Canard 等人[23]首次提出了委托计算的优化思路, 即在 TPM 与主机合作生成零知识签名时把 TPM 的一些计算委托给主机, 并将这种思路应用到了文献[13]方案中的伪名计算环节, 将 TPM 在线签名的计算量降低至一次复指数运算。

Chen 和 Feng[24]首次提出了基于 q-SDH 假设的 SDH-DAA 方案, 使用 q-SDH 假设构建 BBS+签名来生成 DAA 证书。随后, 部分学者据此进行了 TPM 签名效率的优化, 比如 Chen 等人[25]提出了减少证书中的一个元素的新方案, Brickell 等人[26]提出改变了 TPM 和主机之间的委托计算方式的新机制, 以及 Camenisch 等人[27]基于文献[28]使用的 BBS+签名, 提出了一种高效的零知识证明方法。该方案是目前效率最高的 SDH-DAA 方案, 但仍存在优化空间。

Chen 等人[29]提出了一种基于配对的 ECC-DAA 方案, 用公共系统参数来代替根密钥, 极大地降低了 Join 阶段的 TPM 运算量。Chen 和 Urian[30]基于文献[29]的 ECC-DAA 方案和文献[26]的 SDH 方案进行

拓展, 提出了带有属性的 DAA 方案, 允许选择性的属性披露, 即 TPM 可保护多个属性。最终, Camenisch 等人[27]在此工作基础上, 将所有属性存储在主机上, 实现了 SDH-DAA 方案的效率优化。Yang 等人[31]首次给出了具备最优 TPM 签名效率的 DAA_{OPT} 方案, 在两种签名模式下 TPM 均只需进行一次指数运算即可完成签名, 并支持选择性属性公开。

除上述采用标准 DAA 安全模型的协议外, pre-DAA 方案[32]在 TPM 上进行平台所需的全部计算, 适用于 TPM 与主机资源相当的应用场景, 然而该方案忽视了主机被攻破情况下 TPM 的安全。Zhu 等人[33]对这一问题进行了补全, 给出了针对 M2M 的安全 DAA 方案。

此外, 为了实现 M2M 系统中设备间的安全通信, Yu 等人[34]将拟态防御与可信计算相结合, 提出了一种基于拟态防御原理的 M2M 网络匿名认证方法, 在颁发方、签名方和验证方之间引入两方证明机制并添加验证后的模拟防御机制, 实现了安全性能的提升, 但具有较高的算力需求。Yang 等人[35]提出了一种可追溯匿名认证协议, 将 Mimic-Defense 中的调度器和仲裁器与 DAA 方案中的 Issuer 和 Verifier 相结合, 并增加了新的 Track 流程, 实现了 Issuer 对特定平台身份的跟踪。该方案初步实现了 DAA 方案在分布式 DHR 系统中的应用, 但 Track 流程引入了不可忽视的额外开销。

Camenisch 等人[36]提出了通用可组合(Universal Composable, UC)安全模型, 将主机和 TPM 建模为可能处于不同损坏状态的个体, 包含了所有的预期安全属性, 且各协议可以单独分析, 模型具有良好的可组合性。Kim 等人[37]重新定义了 DAA 协议的安全概念, 提出了基于博弈的安全模型, 该模型覆盖了更为广泛的现实攻击场景, Camenisch 等人[21,27]所提出的方案在该模型下被证明是安全的。

Bansarkhani 等人[38]首次提出了基于格的后量子 DAA 方案。El Kassem[39]对其进行改进, 提出了在运行效率和存储成本方面均更为优越的 LDAA 方案。Chen 等人[40]给出了应用于车载互联网平台的 V-LDAA 方案, 并进一步提升了签名速度。

综上所述, 目前对于改进 DAA 方案的研究大多聚焦于 TPM 签名效率优化, 许多改进方案已经实现了 TPM 计算量的大幅度降低。在 DAA 方案与分布式 DHR 系统结合的研究中, ECC 类 DAA 方案拥有匿名、高效、安全等特性, 符合分布式 DHR 系统所需安全认证协议的部分需求, 可以据此进行改进。

3 Tra-DAA 方案

3.1 总体设计

Tra-DAA 共包含 3 个角色模块:

1) 异构执行体集合 HE_i: 包含多个嵌入 TPM 芯片的异构执行体 HE; 单个异构执行体又命名为平台, 包含主机 Host 和芯片 TPM 两部分;

2) 裁决器 Arbitrator: 嵌入了验证方 Verifier;

3) 策略调度器 Dispatcher: 嵌入了证书颁发者 Issuer。

Tra-DAA 方案的整个过程分为认证与追溯两个阶段。由于分布式 DHR 系统需要进行动态反馈, 且具有冗余等特性, 适用于分布式 DHR 系统的改进 DAA 方案需要具备匿名、可追溯、高效的特点。为了实现匿名可追溯与效率优化, Tra-DAA 在认证过程与追溯过程中分别做出如下改进:

1) 在认证过程中, Tra-DAA 方案保留 DAA 方案的匿名特性, 使得一致性裁决匿名化, 实现了匿名认证的功能; 同时, 设置追溯参数, 为后续追溯功能的实现提供基础; 此外, 引入了委托计算技术以实现效率优化;

2) 增加了追溯过程, 通过 Dispatcher 和 Arbitrator 间的数据交互, 实现对异常平台的定位追溯。

如图 1 所示, Tra-DAA 在认证过程中共部署了五项协议。其中, Setup 协议用于进行参数的初始化; Join 协议用于完成原始 DAA 证书的申请与颁发, 包括证书申请、平台身份验证、证书生成、证书颁发, 以及证书的正确性检验; Sign 协议用于生成 DAA 签名; Verify 协议用于验证签名是否合法; Link 协议用于在匿名性前提下实现用户可控的关联性。

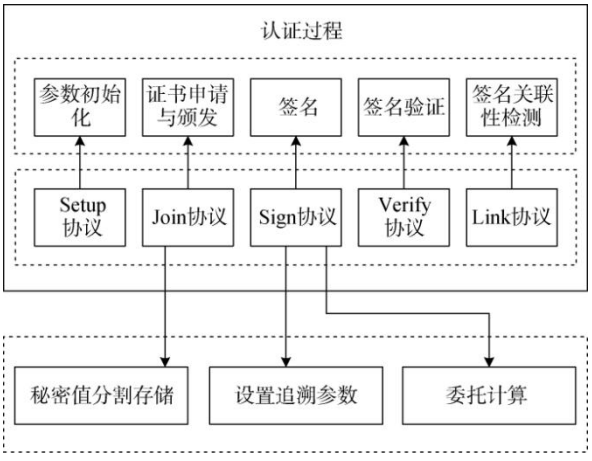

图 1 认证过程协议部署示意图
Figure 1 Schematic diagram of Tra-DAA

Tra-DAA 在认证过程的改进主要体现在 Join 协

议和 Sign 协议中: 在 Join 协议中, 平台对秘密值进行分割存储, 为后续的委托计算提供基础; 在 Sign 协议中, 设置了追溯参数, 并采用高效的委托计算技术来进行追溯参数、伪名和不可链接标签的计算。

综合认证过程与追溯过程, 一次完整的 Tra-DAA 方案流程如下:

1) DAA 证书获取: 平台完成可信度度量, 运行 Setup 协议与 Join 协议, 生成秘密值, 进行签名并发送给 Issuer; Issuer 进行证书颁发, 之后平台将证书信息保留至 Host 中;

2) 可信接入认证: 平台内部的 Host 与 TPM 运行 Sign 协议, 生成关于平台自身的消息 m、追溯参数(T,I)和相应的签名(m,δ), 并将其发送给 Verifier;

3) 平台合法性检验: Verifier 接收到平台传来的签名后运行 Verify 协议, 验证平台的合法性, 将通过验证的平台消息传送至 Arbitrator;

4) 异常反馈: Arbitrator 对接收到的多组数据 m 进行一致性裁决, 将异常的数据对应的签名封装至集合 $Set_{anomalous}$ 中, 之后将 $Set_{anomalous}$ 传输至 Dispatcher;

5) 异常平台追溯: Dispatcher 解析 $Set_{anomalous}$ 中各组签名信息包含的追溯参数(T,I), 定位异常平台的身份信息。

其中, (1)～(3)为认证过程, (4) (5)为追溯过程, Tra-DAA 的详细方案流程见 3.2。

在一次 Tra-DAA 流程中, 角色间的交互关系如图 2 所示。Dispatcher 以可信第三方的身份参与, 并公开其系统参数; Arbitrator 用于接收 Verifier 输出、进行一致性裁决以及向 Dispatcher 发送反馈信息; Verifier 将通过验证的平台的数据和签名传输至 Arbitrator, Arbitrator 会将判定为异常的平台的输出封装至集合 $Set_{anomalous}$并最终反馈至 Dispatcher。

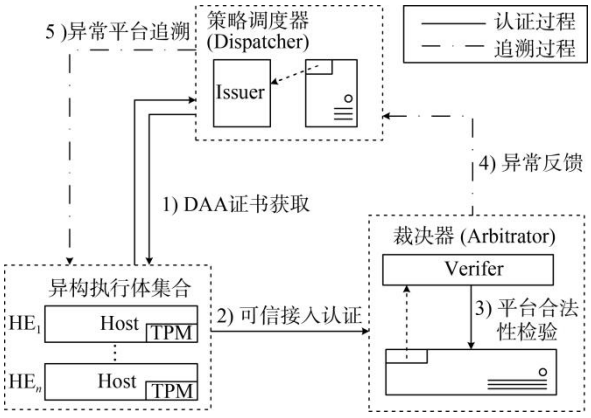

图 2 Tra-DAA 中的角色交互示意图

Figure 2 Schematic diagram of role interaction in Tra-DAA

3.2 方案流程

表 1 给出了 Tra-DAA 方案中的符号释义。ECC 中的标量乘法运算表示为 $Q=[k]P$。

表 1 Tra-DAA 中的符号释义

Table 1 Symbol definition in Tra-DAA

符号	含义
$P_i(i\in\{1,2,T\})$	Issuer 相应循环域的生成元
$e(\cdot)$	非对称的双线性映射
isk:(x,y)	Issuer 的私钥对
ipk:(X,Y)	Issuer 在 ECC 上的公钥对
dsk:(x_d,y_d)	策略调度器的私钥对
dpk:(X_d,Y_d)	Dispatcher 在 ECC 上的公钥对
H_1,H_2	散列函数
H_j	平台的 Host 协议方身份标识
M_i	平台的 TPM 协议方身份标识
tsk	TPM 的秘密值
hsk	Host 的秘密值
λ	Arbitrator 的秘密值
$RougeList$	记录已泄露秘密值的表格
L_{JOINED}	记录已获得证书的平台信息的表格
L_{Trace}	Dispatcher 用于追溯的表格
m	平台自身的可信度度量和执行数据
bsn	用于实现匿名可控功能的伪名
δ	平台生成的 Tra-DAA 签名

3.2.1～3.2.5 分别对认证过程中部署的五个协议进行介绍, 3.2.6 介绍了追溯过程。

3.2.1 Setup 协议

Setup 协议进行参数的初始化。

pac_c: $(G_1,G_2,G_T,e,P_1,P_2,q)$。其中, G_1,G_2,G_T是阶为素数 $q\approx 2^t$的循环群, $G_1=\langle P_1\rangle,G_2=\langle P_2\rangle$, 双线性对 e:$G_1\times G_2\to G_T$。

par_I: (ipk,isk,L_{JOINED})。其中, ipk 和 isk 分别为 Issuer 的公钥和私钥, isk:$x,y\leftarrow Z_q$, ipk: (X,Y), $X=[x]P_2\in G_2$, $Y=[y]P_2\in G_2$; $L_{JOINED}=\phi$为已完成 Join 协议的的平台列表。

par_{Hash}: (H_1,H_2)。为 Tra-DAA 使用的 Hash 函数, $H_1=\{0,1\}\to\{0,1\}^t$, $H_2=\{0,1\}\to G_2$。

$par_{dispatcher}$: $(\lambda, dpk, dsk, L_{Trace})$。其中, dpk 和 dsk 分别为 Dispatcher 的公钥私钥, dsk:$x_d,y_d\leftarrow Z_q$, dpk:(X_d,Y_d), $X_d=[x_d]P_2\in G_2, Y_d=[y_d]P_2\in G_2$; λ为诚实 Arbitrator 的安全信息; $L_{Trace}\neq\phi$为追溯异常身份信息的列表。

3.2.2 Join 协议

Join 协议运行于 Issuer 和 HE 之间。通过 TPM

和 Host 合作生成的关于秘密值的零知识签名, Issuer 可以检验平台身份是否合法。同时在进行证书的存储之前, Host 需要对证书的正确性进行检验。

协议的基本过程如图 3 所示, 详细过程如下所述:

1) Host 向 Issuer 发起 JOIN 请求, 同时向 TPM 发送 TPM.Create 请求;

2) TPM 收到 TPM.Create 请求后随机选择 $tsk \leftarrow Z_q$,计算 $tpk = [tsk]P_1$, 将 tpk 传输至 Host;

3) Host 随机选择 hsk, $u \leftarrow Z_q$, 计算 $hpk = [hsk]P_1$; 选择 Issuer 的公钥 X 对 hpk 进行加密处理得到 $\Omega=hpk+[u]X$; 并计算 $gpk = tpk + hpk$, 作为 Sign 流程的签注公钥;

4) Issuer 收到来自 Host 的 JOIN 请求后, 随机选择一个序列号$n_1 \leftarrow \{0,1\}^t$, 并将n_1发送给 Host;

5) Host 收到n_1后向 TPM 发送 TPM.Commit 请求, 同时本地随机选取$\hat{r} \leftarrow Z_q$, 计算 $R = [\hat{r}]P_1$, $\gamma = [u]P_2$;

6) TPM 接受到 TPM.Commit 请求后随机选择 $r \leftarrow Z_q$, 计算 $E = [r]P_1$, 并将 E 传给 Host;

7) Host 收到转发的消息后, 计算 $c_h = H_2(P_1,tpk,E,\gamma,n_1)$, 将$c_h$发送给 TPM; 同时计算 $z = H_2(P_1, X, \Omega, R, n_1)$, $\hat{s} = \hat{r} + z \cdot hsk \bmod q$, 生成关于 hsk 的零知识签名;

8) TPM 收到c_h后, 在本地随机选取随机数$n_2 \leftarrow \{0,1\}^t$, 计算 $c = H_1(n_2,c_h)$, 以及关于 tsk 的零知识签名 $s = r + c \cdot tsk \bmod q$, 将$(n_2,s)$传给 Host;

9) Host 最后计算 $c = H_1(n_2,c_h)$, 整理 $\pi_t = (c,s,n_2)$为 tsk 的零知识签名, $\pi_h = (z,\hat{s})$为(hsk,u)的零知识签名, 将(tpk,Ω,π_t,π_h)传给 Issuer;

10) Issuer 收到(tpk,Ω,π_t,π_h)后, 计算 $E'=[s]P_1 - [c]tpk$, $R'=[\hat{s}]P_1 - [z]\Omega+[zx]\gamma$, $c_h'=H_2(P_1,tpk,E',\gamma,n_1)$, $c'=H_1(n_2, c_h')$, $z'=H_2(P_1,X,\Omega,R',n_1)$, 如果 $c' \neq c$ 或者 $z' \neq z$, 则放弃这次处理。若均相等, 在确认本次对话的$M_i \notin L_{JOINED}$ 后, Issuer 随机选取 $n \leftarrow Z_q$, 计算 $gpk=\Omega+tpk-[x]\gamma$, $A = [n]P_1$, $B = [y]A$, $C = [x]A + [rxy]gpk$, $D = [ry]gpk$, 最后将生成的证书 $cre:(A,B,C,D)$传至 Host, 并保存(M_i,H_j,gpk)至可信第三方 Dispatcher;

11) Host 选择$e_1,e_2 \leftarrow Z_q$, 在 $A \neq 1_{G_1}, B = 1_{G_1}$条件下验证: $e([e_1]A,Y) \cdot e([-e_1]B,P_2) \cdot e([-e_2]C,P_2) \cdot e([e_2](A+D),X) = 1$。若验证通过, Host 存储证书 $cre = (A,B,C,D,gpk,hsk)$, 否则将其丢弃。

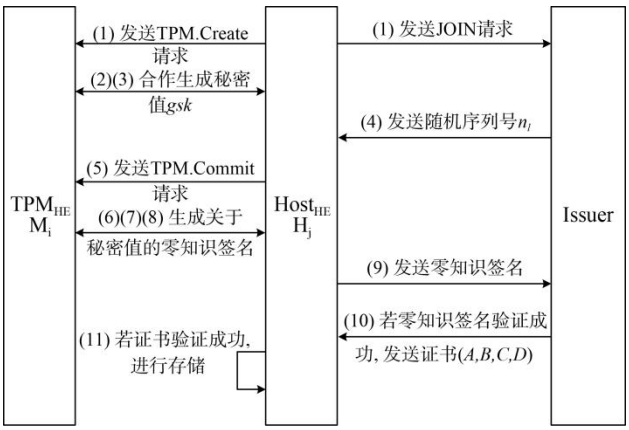

图 3 Join 协议的基本过程

Figure 3 The basic process of the Join protocol

3.2.3 Sign 协议

Sign 协议运行于 HE 中, 使用一个伪名 bsn, TPM 和 Host 可以合作生成一个消息 m 的签名δ。协议的详细过程如下所述:

1) Host 随机选择 $t \leftarrow Z_q$, 计算$A' = [t]A$, $B' = [t]B$, $C' = [t]C$, $D' = [t]D$ 来盲化本地证书;

2) TPM 随机选择 $r \leftarrow Z_q$, 计算用于后续构建零知识签名的元素 $E = [r]P_1$, 将 E 传至 Host;

3) Host 随机选择$\hat{r},\tilde{t} \leftarrow Z_q$, 计算$\widehat{E} = [\hat{r}]P_1$, $\widetilde{E} = E + \widehat{E}$, $T = gpk + [\tilde{t}]X_d$, $I = [\tilde{t}]P_2$, 其中(T,I)为提供给可信第三方 Dispatcher 的追溯参数; 接着 Host 检查伪名 bsn 是否为空, 若伪名为空, 则随机选择 $v \leftarrow Z_q$, 计算$V = [v]P_1$, $K = [v]gpk$, $L = [v]\widetilde{E}$, 以及$c_h = H_2\left(P_1,A',B',C',D',V,K,L,T,I\right)$; 若伪名不为空, 将 V 置空, 计算 $K = e(gpk,H_2(bsn))$, $L = e\left(\widetilde{E},H_2(bsn)\right)$, $c_h = H_2\left(P_1,A',B',C',D',V,K,L,T,I\right)$; 完成判断后 Host 将$(m,bsn,c_h)$传至 TPM;

4) TPM 收到(m,bsn,c_h)后准备自身签名信息: 随机选择随机数 $n_t \leftarrow \{0,1\}^t$, 计算 $c = H_1(n_t,m,bsn,c_h)$以及关于 tsk 的零知识签名 $s = r + c \cdot tsk \bmod q$, 将$(n_t,s)$发给 Host。

5) Host 收到签名后计算 $c = H_1(n_t,m,bsn,c_h)$, 生成包含 tsk 与 hsk 信息的零知识签名 $\bar{s} = s + \hat{r} + c \cdot hsk \bmod q$, $\pi_2=(c, \bar{s}, n_t)$, 并生成最终的 DAA 签名 $\delta = \left(A',B',C',D',V,K,T,I,\pi_2\right)$。

3.2.4 Verify 协议

Verify 协议用以检测一个关于(m,bsn)的 DAA 签名δ是否由合法 TPM 生成。当一个 TPM 被攻破时, 其秘密值 gsk 被加入到假冒列表 *RogueList* 中。在运行 Verify 协议之前, Verifier 需要向平台表明自身的合

法性。由于 Verifier 不需要匿名, 其身份认证采用传统的证明方式。

通过平台检测的 Verifier 在收到请求检验指令 $\delta=(A',B',C',D',V,K,T,I,\pi_2)$后, 执行如下操作:

1) 当 $bsn=\perp$时, 确认 $V\neq 1_{G_1}$; 当 $bsn\neq\perp$时, 确认 $V=\perp$, 计算 $V=e(P_1,H_2(bsn))$;

2) 选择 $e_1,e_2\leftarrow Z_q$, 确认等式 $e([e_1]A',Y)\cdot e([-e_1]B',P_2)\cdot e([-e_2]C',P_2)\cdot e([e_2]A'+D'),X)=1$ 是否成立;

3) 计算, $L'=[\bar{s}]V-[c]K$ $c_h=H_2(P_1,A',B',C',D',V',K',L',T,I)$; 在 $bsn=\perp$时, $V'=V$; 否则$V'=\perp$; 确认$L'=L$ 是否成立;

4) Verifier 查询 *RougeList* 确保当前签名不是来自于被攻破的平台: 对于 $\forall gsk_i\in RL$, 确认 $[gsk_i]V\neq K$ 成立, 否则检测到假冒;

5) 若 1)～4)全部通过, 验证成功并输出 1; 否则, 验证失败并输出 0。

3.2.5 Link 协议

Link 协议的作用是确认两个签名$(\delta_1,m_1,bsn_1\neq\perp)$, $(\delta_2,m_2,bsn_2\neq\perp)$是否是由同一个 TPM 使用同一个 bsn 生成的。具体描述如下:

1) Verifier 收到输入(δ,m), (δ',m')后, 首先确认这两组签名的合法性, 若合法则解析(δ,δ')为 $((A_1',B_1',C_1',D_1',V_1,K_1,T_1,I_1,\pi_{2,1}),(A_2',B_2',C_2',D_2',V_2,K_2,T_2,I_2,\pi_{2,2}))$;

2) 检测(δ,δ')中的K_1与K_2是否相等, 若相等则关联成功令 $f=1$, 否则关联失败令 $f=0$。

3.2.6 追溯过程

Arbitrator 和 Dispatcher 在认证完成后仍需进行交互以完成一次完整的 DHR 运作, 考虑到 Arbitrator 和 Dispatcher 通信内容的短时效性, 两者的通信内容使用对称加密算法 AES 进行加密, 具体流程如下所述:

1) Verify 最终输出为 $f=1$ 或者 $f=0$, 当 $f=0$ 时, 说明本次接收到的消息不合法, Arbitrator 放弃本次对话。若 $f=1$, Arbitrator 保存(δ,m)并进行数据的一致性裁决, 输出异常数据的 HE 会在一致性裁决中被 Arbitrator 标记, 随后 Arbitrator 将异常 HE 的签名 δ保存至异常集合 $Set_{anomalous}$中;

2) Arbitrator 向 Dispatcher 发起 feedback 请求, Dispatcher 接收到请求后随机选择$n_d\leftarrow Z_q$, 生成本次通话密钥$\varepsilon_d=H(n_d,\lambda)$, 并将$n_d$发送至 Arbitrator;

3) Arbitrator 计算$\varepsilon_d=H(n_d,\lambda)$作为本次通话的对话密钥, 对 $Set_{anomalous}$中δ字段进行 AES 加密, 生成最终的 $Set'_{anomalous}$并发送至 Dispatcher;

4) Dispatcher 使用 $\varepsilon_d=H(n_d,\lambda)$ 解密得到 $Set_{anomalous}$, 从相应的签名信息 $\delta=(A',B',C',D',V,K,T,I,\pi_2)$中解析出$(T,I)$, 使用 Dispatcher 的私钥$x_d$计算: $Q'=T-[x_d]I$, 对比L_{Trace}中存储的 (M_i,H_j,Q), 从而追溯到具体的平台身份信息(M_i,H_j);

5) Arbitrator 和 Dispatcher 废弃本次对话密钥ε。

通过使用随机数n_d来生成对话密钥ε, Arbitrator 的秘密值λ的具备安全性。

3.3 委托计算技术

Tra-DAA 的效率优化通过委托计算技术实现。在椭圆密码体系下, 追溯参数、伪名以及不可链接标签的计算[31]均由 TPM 内部委托至算力更强的 Host 中进行, 从而减少了 TPM 芯片在 Sign 协议中的运算量, 进而实现了 Tra-DAA 整体运行效率的提升。该技术的具体实现如下所述:

首先, 在 Join 阶段, 秘密值 gsk 被分割存储在 TPM 和 Host 中。TPM 生成并存储 tsk, Host 生成并存储 hsk, 之后 tsk 和 hsk 分别以 $tpk=[tsk]P_1$和 $hpk=[hsk]P_1$的形式参与到协议的具体交互中, 以保证两者的隐秘性, 具体实现为: TPM 将 tpk 传递给 Host, 由 Host 完成对 $gpk=tpk+hpk$ 的计算与存储。

在随后的 Sign 阶段, TPM 仅进行对 tsk 的零知识签名的计算, 其他需要使用秘密值进行的计算则全部被委托至 Host 中进行, 具体包括追溯参数计算、伪名计算以及不可链接标签计算。

1) 追溯参数计算:

追溯参数设置为$(T=gpk+[\tilde{t}]X_d,I=[\tilde{t}]P_2)$。由于 gpk 的计算已经在 Join 阶段完成, 在 Sign 阶段追溯参数的计算全部在 Host 中进行。

2) 伪名计算:

当 bsn 为空时, 伪名设置为 $K=[v]gpk$, 其承诺为 $L=[v]\tilde{E}$; 当 bsn 不为空时, 伪名设置为 $K=e(gpk,H_2(bsn))$, 其承诺为 $L=e(\tilde{E},H_2(bsn))$。其中, v 为由 Host 选择的随机数, $\tilde{E}$的计算需要使用 TPM 给出的承诺值 E。因此, 伪名计算仅需要由 TPM 完成承诺值计算 $E=[r]P_1$。

3) 不可链接标签计算:

当 bsn 为空时, 不可链接标签设置为$(V=[v]P_1,K=[gsk]V)$, 其承诺值为 $L=[r]V+[\hat{r}]V$。其中, r 为由 TPM 选择的随机数, v、$\hat{r}$为由 Host 选择的随机数。因此, 不可链接标签计算仅需要由 TPM 完成随机数 r 的选取。

由上可见, 通过引入委托计算技术, 在 Sign 阶

段 TPM 只需负责包括随机数选取、承诺值计算以及签名计算在内的 tsk 零知识签名生成, 在保证安全性的前提下, 尽可能地降低了 TPM 的运算量。

4 方案分析

4.1 安全性证明

一个安全的 DAA 方案需要满足匿名、证书不可伪造和签名不可陷害这三个基本安全需求, 本文采用文献[36]中定义的安全概念, 故安全需求描述如下:

1) 匿名: 给定两个关于不同的 $bsn \neq \perp$或 $bsn = \perp$的签名, 攻击者无法判断生成这两个签名的平台相同或不同。

2) 证书不可伪造: 若所有 TPM 均为诚实设备, 且平台均没有使用伪名 bsn 对消息 m 签名, 攻击者无法使用该伪名伪造一个消息 m 的签名; 攻击者仅能以被攻破的 TPM 的方式生成 DAA 签名。

3) 签名不可陷害: 若诚实 Host 没有使用伪名 bsn 对消息 m 签名, 攻击者不能伪造签名来关联此诚实 Host。

本节将在理想现实模型和随机预言模型下论证 Tra-DAA 满足这三个需求。

对于本文提及的分布式 DHR 系统, 给出包含以下假设的威胁模型:

1) 对于三个安全需求, TPM 在攻击者的控制范围内; 除嵌入的 Issuer 之外, Dispatcher 的其他部分完全可信;

2) 针对匿名性需求与签名不可陷害需求, 攻击者有能力加载可以攻破 Issuer 的有效荷载, Host 不在攻击者的控制范围内;

3) 针对证书不可伪造需求, 攻击者有能力加载可以攻破大部分 Host 的有效荷载, Issuer 不在攻击者的控制范围内。

基于上述假设, 我们将构建一个可信第三方$\mathrm{T_p}$和一个模拟器 S, 并证明 S 可以模拟攻击者 A 的一切行为, 使得环境不可区分。可信第三方$\mathrm{T_p}$完成了协议运行的全过程, 同时存储了一些辅助协议运行的列表, 列表的功能和存储格式如表 2 所示。通过这些列表, $\mathrm{T_p}$可以执行协议中理想函数的功能。

Tra-DAA 中的理想函数构造如下所述:

Join 阶段: Host 生成关于自身信息的参数(M_i,H_j,tsk_i,hsk_i), 将(M_i,H_j,tsk_i,hsk_i)传输至可信第三方$\mathrm{T_p}$, $\mathrm{T_p}$从L_{Joined}表中查看当前通讯平台的信息是否已经存在, 若存在则丢弃本次对话, 否则查询 *RogueList* 并确认 $gsk_i = tsk_i + hsk_i$ 是否存在于 *RogueList* 中, 若存在则放弃本次对话。

表 2 可信第三方所存列表的功能和存储格式

Table 2 Functions and storage format of lists stored by trusted third parties

列表名称	列表功能	存储格式
RogueList	记录所有被攻破的平台秘密值	(M_i,H_j,gsk_i)
L_{Signed}	记录不同平台关于消息和伪名的签名	$(\delta,m,bsn,M_i,H_j,gsk_i)$
L_{Joined}	记录获得 DAA 证书的平台G_1	(M_i,H_j,tsk_i,hsk_i)

Sign 阶段: Host 与 TPM 合作运行 Sign 协议, 签发关于消息 m 的签名δ, 并将$(\delta,m,bsn,M_i,H_j,tsk_i,hsk_i)$发送至$\mathrm{T_p}$, $\mathrm{T_p}$查看该签名对应的平台信息是否存在于L_{Joined}表中, 若存在, 将该签名保存至L_{Signed}表中; 否则, 认为该签名是由未执行 Join 协议的平台生成的野签名, 丢弃本次对话;

Verify 阶段: Verifier 向$\mathrm{T_p}$发送(δ,m)发起签名的验证, $\mathrm{T_p}$向L_{Joined}中查询与(δ,m)相关的所有记录, 确认(δ,m)有且仅有一条记录, 否则说明该签名是伪造签名或者野签名, 向 Verifier 输出 0。接着$\mathrm{T_p}$向 *RogueList* 查询(δ,m)对应的(M_i,H_j,tsk_i,hsk_i)是否存在, 若存在则说明当前平台已被攻破, 向 Verifier 输出 0, 否则输出 1。

模拟器 S 在各阶段的模拟行为如下所述:

Setup 阶段: 现实系统中, S 通过与可信第三方$\mathrm{T_p}$的交互模拟 Issuer 生成系统初始化参数, 并将参数告知 A。

Join 阶段: S 首先扮演 Issuer, 将整数 n 发送给被攻击者 A 控制的实体 Host, A 将n'转发给 TPM。然后, S 对 n 和n'的值进行比较, 若不一致则放弃此次模拟; 若一致, 检测是否为重放攻击, 即询问可信第三方$\mathrm{T_p}$以查询是否已存在关于平台的证书记录。若存在, 直接计算证书并发送给 A; 若不存在, 在理想系统中 S 模拟被攻击者 A 控制的实体 Host, 向$\mathrm{T_p}$申请证书。$\mathrm{T_p}$调用理想函数 Join 协议, 将是否颁发证书的决定告知 S。若同意颁发, S 模拟 Issuer 正常计算证书发给 A; 若不同意, 则放弃此次 Join 连接。

Sign 阶段: S 收到 A 的签名申请后, 首先模拟 A 的行为向$\mathrm{T_p}$发出签名申请。$\mathrm{T_p}$调用理想函数 Sign 协议, 将随机生成的签名发给 S。S 在现实系统中模拟 TPM 将签名和追溯参数发送给 A。

Verify 阶段: S 模拟协议中的诚实实体 Verifier 和 Issuer, 接收 A 的签名验证申请。S 通过$\mathrm{T_p}$的理想函数

Verify 协议分别确认证书、平台以及签名的合法性;

因此, Tra-DAA 的安全性证明如下所述:

定理 1. 在随机预言模型中,Tra-DAA 方案在 DL 假设、DBDH 假设、DDH 假设以及双线性 LRSW 假设下是安全的。

1) 匿名性证明:

在 Sign 阶段, 由 TPM 和 Host 合作生成的零知识签名π_2是通过运算两个随机预言函数H_1和H_2生成的。在随机预言环境中存在着一个模拟器 S 可以对任何语句进行一次模拟运算得到零知识签名π_2。

推论 1. 随机选取一个新的 $gsk_i \leftarrow Z_q$ 和 $gsk_{i-1} \leftarrow Z_q$, 并依次计算(V_i,K_i)和(V_{i-1},K_{i-1}), 在 $bsn = \perp$和 $bsn \neq \perp$场景下, (V_i,K_i)和(V_{i-1},K_{i-1})均是不可区分的。

在 $bsn = \perp$场景下, 区分(V_i,K_i)和(V_{i-1},K_{i-1})可以视作一个符合 DDH 假设的问题。给定一个 DDH 构造 $(G_1,G_2,P_1,P_2,X,Y,Z,q),X \leftarrow xP_1,Y \leftarrow yP_1,Z \leftarrow zP_1$, 将 X 设为 TPM 中秘密值公钥 tpk, 模拟出零知识签名π_t, 并选取 $hsk \leftarrow Z_q$作为一个诚实 Host 的秘密值。对于(V_i,K_i)而言, 当 $bsn_i = \perp$时, 可将 DDH 构造中的 Y 置为 $V = yP_1$, 则 $K = Z + [hsk]V$, 若 $z = xy$, 则可以成功区分 (V_i,K_i)和(V_{i-1},K_{i-1}), 与 DDH 假设不符。

在 $bsn \neq \perp$ 场景下, 区分(V_i,K_i)和(V_{i-1},K_{i-1})可以视作一个符合 DBDH 假设的问题。给定一个 DBDH 构造$(G_1, G_2, P_1, P_2, \Delta_1, \Delta_2, X, Y, Z, q),X \leftarrow xP_1$, $Y \leftarrow yP_2, \Delta_1 \leftarrow \delta P_1, \Delta_2 \leftarrow \delta P_2, Z \leftarrow e(P_1,P_2)^z$, 模拟器随机选择 $tsk \leftarrow Z_q$作为 TPM 的秘密值, 并随机选取 Join 阶段的 V 来生成零知识签名π_h; 同时, 模拟器使用随机预言机使得$H_2(bsn_i) = Y$。在计算零知识签名π_2时, 模拟器模拟计算 $K = [z]e(P_1,P_2)$, 若 $z = xy\delta$, 则可以成功区分(V_i,K_i)和(V_{i-1},K_{i-1}), 与 DBDH 假设不符。

综合推论 1, 在 $bsn = \perp$和 $bsn \neq \perp$的场景下, 敌手 A 在 DDH 假设与 DBDH 假设下无法区分由诚实平台签发的两组签名信息(V_i,K_i)和(V_{i-1},K_{i-1}), 无法将由诚实平台签发的两个签名进行关联。因此, Tra-DAA 在随机预言模型下保证了匿名性。

2) 签名不可陷害证明:

推论 2. 在 Host 被攻破, 存储的 cre 参数全部被敌手 A 获取的情况下, A 无法伪造出合法的签名δ。

在 Tra-DAA 中, Sign 阶段生成的δ包含了对秘密值 $gsk = tsk + hsk$ 的零知识签名π_2: $\bar{s} = s + \hat{r} + c \cdot hsk \ mod \ q$, 其中 $s = r + c \cdot tsk \, mod \, q$, 敌手 A 需要获取 TPM 的秘密值 tsk 来生成合法的π_2。然而, 平台的秘密值 gsk 以 gpk 的形式储存在 Host 中, 给定一组椭圆曲线的 DL 构造$(G,P,X,x),X \leftarrow xP$, 其中, $gpk = [gsk]P$, $Z = [z]P$。模拟器计算 $s = r + c \cdot tsk \, mod \, q$, 生成合法的零知识签名$\pi_2$。若 $z = gsk$, 则敌手成功伪造出合法签名δ。然而, 区分 gpk 和 Z 在 DL 假设下无法实现。因此, 敌手在攻破 Host 并获取存储的全部 cre 参数的情况下, 依旧无法伪造出合法的签名δ。

推论 3. 敌手 A 在未获知任何 cre 参数的情况下, 无法伪造出合法的签名δ。

在 Tra-DAA 中, δ的合法性将在 Verify 环节验证, δ中包含了被盲化处理后的(A,B,C,D)。给定一组 B-bLRSW 构造$(G_1,G_2,P_1,P_2,A,B,C,D,q)$, r, x, y, $f \leftarrow Z_q$, $B \leftarrow yA$, $C \leftarrow [x]A + [rxy]gpk$, $D \leftarrow [ry]gpk$, 并给定 (A,X,Y,Z), $A \leftarrow G_1$, $Y = [a]AC = [x]A + [rxy]F$, $Z = [ry]F$。若敌手 A 成功伪造出合法的(A,B,C,D), 即表明敌手在未获得合法 gpk 字段的前提下完成了证书信息的构造, 与 LRSW 假设不符。因此, 敌手 A 在 LRSW 假设下无法区分(A,X,Y,Z)与(A,B,C,D), 即在未获取任何 cre 证书字段和 Issuer 的私钥的情况下, A 无法生成一个可以通过 Verifier 验证的 DAA 签名。

综合推论 2 和推论 3, 在 LRSW 假设和 DL 假设下, Tra-DAA 满足 1 型和 2 型不可伪造性。

3) 不可陷害性证明:

推论 4. 在 Issuer 被攻破的场景下, 一个诚实的平台无法被陷害。

给定一个椭圆曲线密码下的 DL 构造$(P,[x]P)$,将 $[x]P$ 置为 TPM 的秘密值公钥 $tpk = [tsk]P_1$, 随机选择 $hsk \leftarrow Z_q$作为 Host 的秘密值。随后, S 在与诚实 Host 的 Sign 和 Join 交互中通过随机预言机模拟出π_2。敌手 A 需要知晓秘密值 gsk 以实现可关联诚实平台的签名的伪造。若敌手 A 实现了成功伪造, 即说明 A 通过 $tpk = [tsk]P_1$成功反推出 tsk, 完成了椭圆曲线密码下的离散对数问题求解, 与 DL 假设不符。

综合推论 4, Tra-DAA 在随机预言模型下具有不可陷害性。

综上所述, 当攻击者能够做出伪造证书或伪造签名等不合法行为时, 模拟器会输出模拟失败的结果。每一个模拟失败的结果都建立在攻击者已攻破协议中数学难题的基础上, 而这些数学难题经证明不可破解, 因此, 模拟器的成功模拟具有必然性。模拟器扮演诚实 TPM 的行为, 使得攻击者无法区分所

处环境, 并在与攻击者的交互过程中获取了有用信息。此外, 模拟器在理想环境中模拟攻击者的行为, 借助随机预言机和可信第三方的理想函数, 可以实现在现实和理想系统中协议运行的一致, 使得环境无法区分是处于理想还是现实系统中。

综合推论 1、2、3、4, Tra-DAA 在 DL 假设、DBDH 假设、DDH 假设以及双线性 LRSW 假设下是安全的, 故定理一成立。根据密码学协议的安全性定义, Tra-DAA 在理想现实模型和随机预言机模型下是安全的。

4.2 运行效率分析

本节将 Tra-DAA 与近几年的 DAA 方案进行效率分析和对比。具体来说, 通过定量分析不同协议实体在 Join 和 Sign/Verify 协议中承担的计算工作, 将各方案的计算效率进行直观的对比和展示。

对基于 ECC 的 DAA 方案, 通常使用$G_i(i=\{1,2,T\})$来表示在群G_i上的一次指数运算; $G_i{}^r$表示在群G_i上的r 指数运算; P 表示一次双线性映射的计算; 而P^r表示批量证明 r 个双线性映射的计算量。

从表 3 中可以看出, Tra-DAA 在 Join 阶段的 TPM 计算量较方案[28, 22, 21]相比具有优势, 与方案[35]持平。在证书的合法性验证中, Tra-DAA 吸收了文献[27]的思想, 采用批量证明双线性映射, 在不改变安全性的前提下, 效率比单独计算四个双线性对提高了 40%。在保证安全性的基础上, Tra-DAA 相较于 DAA TPMv2.0[21]在 TPM 上减少了一次指数运算。由于使用了委托计算技术, Tra-DAA 需要额外证明 Host 的秘密值 hsk 的有效性, Host 计算量有所增长。然而, DHR 环境中的 Host 是承载拟态防御服务的提供者, 具有一定计算能力与存储能力, Host 计算量的少量提升不会引发协议的整体运行效率的降低。方案[32]虽然具备最优的 TPM 计算量, 但该方案将 TPM 和 Host 统一视为一个诚实实体, 零知识证明的对象是 Issuer 的私钥, 其方案的安全场景过于理想, 没有太多的实际应用参考价值。综上所述, 在 Join 阶段, Tra-DAA 在保证协议实用安全性的前提下尽可能降低了 TPM 的计算量。

在 Sign/Verify 阶段, 由于 Tra-DAA 需要为策略调度服务器提供可追溯参数, Sign 每次签名时需要额外计算 $T=gpk+[\tilde{t}]X_d, I=[\tilde{t}]P_2$, 即对 Host 而言, Sign 阶段需要额外进行两次指数运算, 故 Host 承载的计算量高于其他方案。但 Host 的计算能力远强于 TPM, 该程度的 Host 计算量增长不会降低协议的整体效率。对于 TPM 而言, Tra-DAA 将部分计算委托至 Host 中进行, 在 $bsn=\perp$和 $bsn\neq\perp$的场景下, TPM 的运算量恒定为一次指数运算, 在不改变协议安全性的前提下, 达到了 TPM 在理论上的最低运算量, 实现了 TPM 效率优化的极限。关于委托计算实现的效率改进, 其具体分析如下:

在前面的 Join 协议中, 秘密值 gsk 被分解为存储于 TPM 芯片内的 tsk 和 Host 内部的 hsk, TPM 会将计算出的 tpk 传至 Host, Host 则结合其生成的 hsk 计算 $gpk=tpk+hpk$, 并将 gpk 储存在本地。在随后的 Sign 阶段中, 在伪名为空以及伪名不为空的两种情况下, TPM 需要进行的全部运算均为随机数 r 的选取、承诺值计算 $E=[r]P_1$以及签名(n_t,s)的计算, 整个过程恒定为一次指数运算。而 Sign 阶段需要完成的追溯参数计算、伪名计算以及不可链接标签计算全部被委托至算力更强的 Host 中进行, 使得整体效率得到优化。

表 3 Join 协议计算量对比

Table 3 Comparison of computation costs in Join protocol

方案	TPM	Host	Issuer
ECC-DAA[28]	$3G_1$	P^4	$2G_1+2G_1^2$
batch-DAA[22]	$3G_1$	P^4	$2G_1+2G_1^2$
pre-DAA[32]	G_1	$2G_1^2+4P$	$5G_1+G_1^2$
DAA TPMv2.0[21]	$3G_1$	$3G_1+4P$	$G_1+G_1^2$
Track-DAA[35]	$2G_1$	P^4	$3G_1+2G_1^2$
ra-DAA	$2G_1$	$3G_1+G_1^2+P^4$	$G_1^3+2G_1^2+4G_1$

表 4 Sign/Verify 协议计算量对比

Table 4 Comparison of computation costs in Sign/Verify protocol

方案	TPM	Host	Verifer
ECC-DAA[28]	$3G_1$	$4G_1$	$2G_1^2+4P+nG_1$
batch-DAA[22]	$G_1/2G_1$	$4G_1$	$G_1^2+P^4+nG_1$
pre-DAA[32]	$3G_1$	$4G_1$	$2G_1^2+4P+nG_1$
DAA TPMv2.0[21]	$2G_1/3G_1$	$6G_1/7G_1$	$2G_1^2+4P+nG_1$
Track-DAA[35]	$2G_1/3G_1$	$4G_1$	$G_1^2+P^4+nG_1$
Tra-DAA	G_1	$8G_1+2G_2/5G_1+2G_2+2P$	$G_1^2+P^4+nG_1/G_1^2+P^4+P+nG_1$

在 Verify 阶段, 对于 $bsn\neq\perp$的签名, Verifier 需要自行计算 $V=e(P_1,H_2(bsn))$, 因此额外多出了一次双线性对的运算 P。其中, nG_1 是依次代入 *RogueList* 中已公开的假冒平台秘密值的计算量, P^4 是验证证书合法性的计算量, ${G_1}^2$则是利用零知识技术验证平台身份的计算量。

由表 4 可见, 除 batch-DAA[22]外, Tra-DAA 在

Verify 阶段的效率优于绝大部分 DAA 方案; 对比 batch-DAA, Tra-DAA 具有更高的最优 TPM 效率, 且额外兼顾了 TPMv2.0 规范。对比同样具备可追溯功能的 Track-DAA[35], Tra-DAA 具有更高的 TPM 运行效率, 降低了增加可追溯功能带来的额外开销。

此外, 各方案的 Link 协议计算量没有明显差别。

综上所述, 相比于已有的 DAA 方案, Tra-DAA 方案在 TPM 计算效率上具有最优表现。

4.3 与现有方案的实验性能对比

本实验不考虑 DAA 各个协议方之间的通信开销与异构执行体进行可信度度量的时间, 仅针对 DAA 方案本身的开销, 将 Tra-DAA 与现有的其他 DAA 方案进行比较。为简化实验, DAA 中签名中所使用的消息 m 和 bsn 均使用随机字符串代替。此外, 由于假冒 TPM 检测机制需要额外的 *RogueList* 辅助, 且不存在定义 *RogueList* 大小的相关标准, 因此也不考虑假冒 TPM 检测的开销。

采用 Mocha 测试 Tra-DAA 和代表性方案 TPMv2.0 DAA[21]各个流程的运行时间, 考虑到单次运行时间的特殊性, 在运行 1000 次协议后对运行时间取平均值, Join、Sign 以及 Verify 阶段的运行时间见表 5 与表 6(在 Link 阶段, 两方案的运行时间没有明显差别, 故不在表中体现), Join 阶段各协议方运行时间的对比细节见图 4, Sign 以及 Verifier 阶段各协议方运行时间的对比细节图 5 与图 6。

在 Join 阶段, 从图 4 和表 5 可以看出, 相对于 DAA TPMv2.0, Tra-DAA 中 TPM 方的计算量优于 DAA TPMv2.0, 减少了 1097us 的计算时间, 缩短了 33%耗时。为了满足 TPMv2.0 的规范, Tra-DAA 采用了秘密值分割存储的方法, 平台需要提供两部分秘密值的零知识签名, 引入了额外的计算量。然而, 在 DHR 场景中, Host 和 Issuer 均为具备足够计算资源的服务器设备, 其计算能力 10 倍强于 TPM, 因此额外计算量的增加对协议整体效率的影响很小。

从图 5 和图 6 的结果对比可以看出, Tra-DAA 在 Sign 阶段以及 Verifier 阶段有着绝对的计算优势, TPM 的计算耗时在伪名为空和不为空的情况下分别缩短了 1201us 和 2478us, 计算性能分别提升了 50%和 70%, 这是因为 Tra-DAA 采用了委托计算伪名和委托计算不可链接标签的技术, 将 TPM 的计算转移至计算能力更强的 Host 中进行, 从而提升了协议的整体效率。

对于追溯环节, 在现有的拟态防御研究中, DHR 的冗余度一般设置为 3。因此, 假设在最坏情形下, 异常集合 $Set_{anomalous}$的大小为 3, 即 AES 需要加解

表 5 Join 阶段运行时间对比

Table 5 Comparison of the running time in Join phase (μs)

方案	TPM Join	Host Join	Issuer Join
DAA TPMv2.0[21]	3289	9468	5468
Tra-DAA	2192	16890	7897

表 6 Sign/Verify 阶段运行时间对比

Table 6 Comparison of the running time in Sign/Verify phase (μs)

方案		TPM Sign	Host Sign	Verifier	总耗时
DAA TPMv2.0[21]	$bsn=\perp$	2297	10987	27893	41171
	$bsn\neq\perp$	3567	11929	27893	43389
Tra-DAA	$bsn=\perp$	1096	17034	15392	32969
	$bsn\neq\perp$	1089	16481	17832	35955

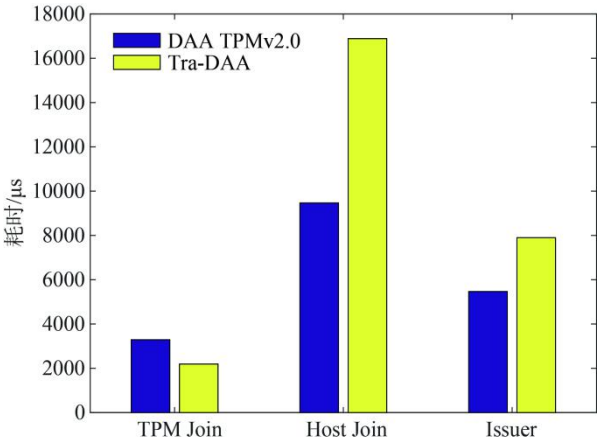

图 4 Join 阶段各协议方的耗时对比

Figure 4 Time-consuming comparison of each role in the Join phase

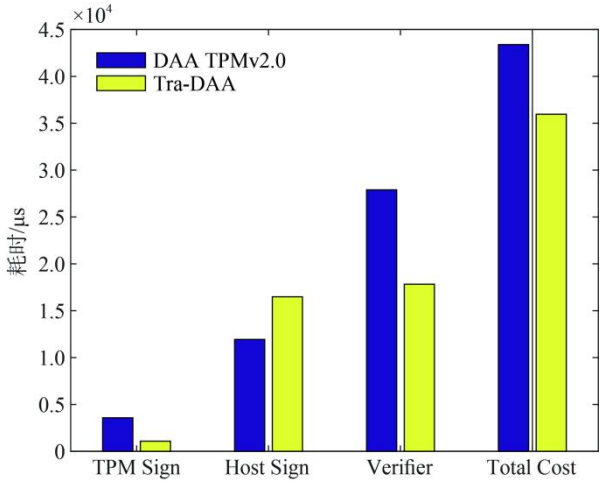

图 5 $bsn\neq\perp$时各协议方 Sign/Verifier 阶段耗时对比

Figure 5 Time-consuming comparison in Sign/Verifier phase when $bsn\neq\perp$

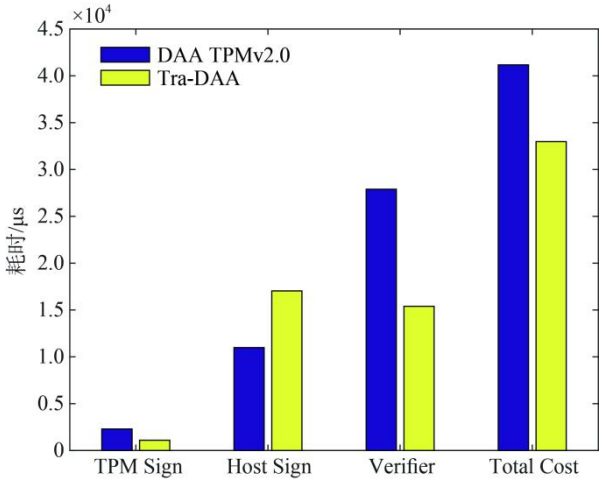

图 6 $bsn=\perp$时各协议方 Sign/Verifier 阶段耗时对比
Figure 6 Time-consuming comparison in Sign/Verifier phase when $bsn=\perp$

密 3 组 DAA 签名大小的数据量。因为本文采用 BN-381 曲线, 在哈希函数输出取 160 比特、q 取 256 比特的情况下, 三组签名的大小为 11160 比特, 采用 ECB 分组的 AES-128, 故加解密时间为 231μs, 结合三次 ECC 加密所需的耗时 3240μs, 追溯环节的总耗时为 3471μs, 仅占 Tra-DAA 总协议时间的 5%, 对整体协议的运行效率的影响极小。此外, 在实际应用中, 追溯环节将在运行速度更快的服务器设备上运行, 得到更低的时间占比。由此可见, 增加的可追溯功能满足了分布式 DHR 系统在接收异构执行体反馈上的需求, 但并未对整体运行效率造成不可忽视的影响。

综合以上分析, Tra-DAA 的整体效率要优于实验参照协议 DAA TPMv2.0, 符合具有冗余特性的分布式 DHR 系统对认证方案的需求。

5 结束语

本文提出了一种适用于分布式 DHR 系统的可追溯直接匿名认证 Tra-DAA 方案, 重点研究面向分布式 DHR 系统应用直接匿名认证机制时的可行性和效率性问题, 通过增加可追溯功能解决了现有 DAA 方案的完全匿名特性与 DHR 的动态反馈需求之间的矛盾。我们对 Tra-DAA 方案进行了安全性证明, 并与具有代表性的 DAA 方案进行理论分析和实验对比, 验证了 Tra-DAA 方案的优越性。实验结果表明, Tra-DAA 方案实现了可追溯功能的同时仅增加了 5%的耗时, 并通过引入委托计算技术将 TPM 的计算量降至理论最低值, 使其整体的运行效率得到显著提升。

参考文献

[1] Wu J X. Mimicry Defense Technology to Build Endogenous Security in National Information Network Space[J]. Information and Communications Technologies, 2019, 13(6): 4-6.
(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, 13(6): 4-6.)

[2] Xu J J. Research on Cyberspace Mimic Defense Based on Dynamic Heterogeneous Redundancy Mechanism[J]. *Journal of Computer and Communications*, 2021, 9(7): 1-7.

[3] Hu H C, Wu J X, Wang Z P, et al. Mimic Defense: A Designed-in Cybersecurity Defense Framework[J]. *IET Information Security*, 2018, 12(3): 226-237.

[4] Wang Z P, Hu H C, Cheng G Z. A DNS Architecture Based on Mimic Security Defense[J]. *Acta Electronica Sinica*, 2017, 45(11): 2705-2714.
(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.)

[5] Wu T, Hu C N, Chen Q N, et al. Defense-Enhanced Dynamic Heterogeneous Redundancy Architecture Based on Executor Partition[J]. *Journal on Communications*, 2021, 42(3): 122-134.
(吴铤, 胡程楠, 陈庆南, 等. 基于执行体划分的防御增强型动态异构冗余架构[J]. *通信学报*, 2021, 42(3): 122-134.)

[6] Chen S, Ma M D, Luo Z X. An Authentication Framework for Multi-Domain Machine-to-Machine Communication in Cyber-Physical Systems[C]. *2015 IEEE Globecom Workshops*, 2016: 1-6.

[7] Chen H C, You I, Weng C E, et al. A Security Gateway Application for End-to-End M2M Communications[J]. *Computer Standards & Interfaces*, 2016, 44: 85-93.

[8] Zhang J X, Pang J M, Zhang Z, et al. Executors Scheduling Algorithm for Web Server with Mimic Structure[J]. *Computer Engineering*, 2019, 45(8): 14-21.
(张杰鑫, 庞建民, 张铮, 等. 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(8): 14-21.)

[9] Shepherd C, Arfaoui G, Gurulian I, et al. Secure and Trusted Execution: Past, Present, and Future - a Critical Review in the Context of the Internet of Things and Cyber-Physical Systems[C]. *2016 IEEE Trustcom/BigDataSE/ISPA*, 2017: 168-177.

[10] Brickell E, Chen Liqun, Li Jiangtao. A new direct anonymous attestation scheme from bilinear maps[C]. *International Conference on Trusted Computing*, 2008: 166-178.

[11] Brickell E, Camenisch J, Chen L Q. Direct Anonymous Attestation[C]. *The 11th ACM conference on Computer and communications security*, 2004: 132-145.

[12] Camenisch J, Lysyanskaya A. A Signature Scheme with Efficient Protocols[M]. Security in Communication Networks. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003: 268-289.

[13] Bernhard D, Fuchsbauer G, Ghadafi E, et al. Anonymous Attestation with User-Controlled Linkability[J]. *International Journal of Information Security*, 2013, 12(3): 219-249.

[14] Isern-Deyà A P, Huguet-Rotger L, Payeras-Capellà M M, et al. On the Practicability of Using Group Signatures on Mobile Devices:

Implementation and Performance Analysis on the Android Platform[J]. *International Journal of Information Security*, 2015, 14(4): 335-345.

[15] Chen L Q, Morrissey P, Smart N P. Pairings in Trusted Computing[C]. *The 2nd international conference on Pairing-Based Cryptography*, 2008: 1-17.

[16] Chen Liqun, Morrissey P, Smart N P. On proofs of security for DAA schemes[C]. *International Conference on Provable Security*, 2008: 156-175.

[17] Chen L Q, Li J T. A Note on the Chen-Morrissey-Smart DAA Scheme[J]. *Information Processing Letters*, 2010, 110(12/13): 485-488.

[18] Chen L Q, Morrissey P, Smart N. DAA: Fixing the Pairing Based Protocols[J]. *IACR Cryptol EPrint Arch*, 2009: 198.

[19] ISO. Information technology - Security techniques - Anonymous digital signatures - Part 2: Mechanisms using a group public key (ISO/IEC 20008-2), 2013.

[20] ISO. Information technology - Trusted Platform Module Library - Part 1: Architecture (ISO/IEC 11889-1), 2015.

[21] Camenisch J, Chen L Q, Drijvers M, et al. One TPM to Bind them All: Fixing TPM 2.0 for Provably Secure Anonymous Attestation[C]. *2017 IEEE Symposium on Security and Privacy*, 2017: 901-920.

[22] Chen Liqun. A DAA scheme using batch proof and verification[C]. *International Conference on Trust and Trustworthy Computing*, 2010: 166-180.

[23] Canard S, Pointcheval D, Sanders O. Efficient delegation of zero-knowledge proofs of knowledge in a pairing-friendly setting[C]. *International Workshop on Public Key Cryptography*, 2014: 167-184.

[24] Chen X F, Feng D G. Direct Anonymous Attestation for Next Generation TPM[J]. *Journal of Computers*, 2008, 3(12): 43-50.

[25] Chen L Q. A DAA Scheme Requiring less TPM Resources[C]. *The 5th international conference on Information security and cryptology*, 2009: 350-365.

[26] Brickell E, Li Jiangtao. A pairing-based DAA scheme further reducing TPM resources[C]. *International Conference on Trust and Trustworthy Computing*, 2010: 181-195.

[27] Camenisch J, Drijvers M, Lehmann A. Anonymous attestation using the strong Diffie-Hellman assumption revisited[C]. *International Conference on Trust and Trustworthy Computing*, 2016: 1-20.

[28] Au M H, Susilo W, Mu Y. Constant-Size Dynamic K-TAA[M]. *Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006: 111-125.

[29] Chen Liqun, Page D, Smart N P. On the design and implementation of an efficient DAA scheme[C]. *International Conference on Smart Card Research and Advanced Applications*, 2010: 223-237.

[30] Chen Liqun, Urian R. DAA-A: Direct anonymous attestation with attributes[C]. *International Conference on Trust and Trustworthy Computing*, 2015: 228-245.

[31] Yang K, Chen L Q, Zhang Z F, et al. Direct Anonymous Attestation with Optimal TPM Signing Efficiency[J]. *IEEE Transactions on Information Forensics and Security*, 2021, 16: 2260-2275.

[32] Bernhard D, Fuchsbauer G, Ghadafi E. Efficient signatures of knowledge and DAA in the standard model[C]. *International Conference on Applied Cryptography and Network Security*, 2013: 518-533.

[33] Zhu Z. Research on M2M trusted direct anonymous attestation technologies based on UC security framework[D]. Nanjing: Southeast University, 2017.
(朱政. 基于 UC 安全框架的 M2M 可信直接匿名认证技术研究[D]. 南京: 东南大学, 2017.)

[34] Yu C, chen L Q, Lu T Y. A Direct Anonymous Attestation Scheme Based on Mimic Defense Mechanism[C]. *2020 International Conference on Internet of Things and Intelligent Applications*, 2021: 1-5.

[35] Yang Z Y, Chen L Q. A Traceable Anonymous Authentication Method for Mimic Defense[C]. *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference*, 2021: 1831-1836.

[36] Camenisch J, Drijvers M, Lehmann A. Universally composable direct anonymous attestation[C]. *Public-Key Cryptography--PKC 2016*, 2016: 234-264.

[37] Kim H, Lee K, Park J H, et al. Improving the Security of Direct Anonymous Attestation under Host Corruptions[J]. *International Journal of Information Security*, 2021, 20(4): 475-492.

[38] El Bansarkhani R, Kaafarani A. Direct Anonymous Attestation from Lattices[J]. *IACR Cryptol EPrint Arch*, 2017: 1022.

[39] El Kassem N. Lattice-based direct anonymous attestation[D]. *University of Surrey*, 2020.

[40] Chen L Q, Tu T Y, Yu K L, et al. V-LDAA: A New Lattice-Based Direct Anonymous Attestation Scheme for VANETs System[J]. *Security and Communication Networks*, 2021, 2021: 1-13.

陈立全 于 2005 年在东南大学获得博士学位, 教授、博士生导师, 现任东南大学网络空间安全学院副院长、东南大学信息安全研究中心副主任, 主要研究方向为密码与安全协议、物联网安全、区块链技术等。Email: Lqchen@seu.edu.cn

羊子煜 于 2021 年在东南大学网络空间安全专业获得硕士学位, 主要研究方向为边缘计算、拟态防御等。Email: 736707148@qq.com

张子燕 现为东南大学网络空间安全专业硕士研究生, 主要研究方向为密码学、区块链技术等。Email: amblyan@163.com

刘苏慧 于 2021 在曲阜师范大学计算机科学与技术学院获得硕士学位, 现为东南大学网络空间安全学院博士研究生。主要研究方向包括云辅助物联网数据安全、功能密码学和区块链技术等。Email: suhliu@ 126.com