

支持属性立即撤销的可复用属性基签名及其密钥协商方案

张智烁¹, 杨会喜², 黄文³, 廖永建¹, 周世杰¹

(1. 电子科技大学信息与软件工程学院, 四川 成都 610057; 2. 沧州市职业技术学院机电工程系, 河北 沧州 061001;

3. 四川大学计算机学院, 四川 成都 610065)

摘要: 为了解决后台分布式微服务一前端反向代理网络架构中统一细粒度访问控制和数据安全通信的问题, 提出了可撤销属性的认证策略属性基签名, 基于 Type-3 配对的属性基签名及其互认证密钥协商方案。在所提方案签名算法中, 无须关联访问策略, 生成的签名只和用户部分属性有关, 访问策略在验证算法中输入, 来校验与签名相关联的用户属性是否真实合法并满足访问策略, 所以签名与访问策略实现了解耦, 使得一个签名可复用于多个访问策略的认证过程。另外, 提出了属性哈希过滤算法, 并基于该算法实现了属性的立即撤销机制, 使得签名者不能用过期无效的属性继续用于策略认证。并在选择策略模型下严格地验证了所提属性基签名方案的存在不可伪造性。进一步地, 为了实现数据的安全通信, 提出了与所提属性基签名方案配套的满足扩展 Canetti-Krawczyk 安全模型的认证密钥协商方案。最后, 通过理论和实验对比分析得知, 在安全级别为 128 bit 高级加密标准的要求下, 所提方案中的微服务器端的认证算法比其他属性认证算法消耗更少的时间, 因此所提方案更加适用于复合微服务请求的场景。

关键词: 细粒度访问控制; 属性哈希过滤; 属性撤销; 解耦式属性基签名; 互认证密钥协商; 不可伪造性

中图分类号: TP393

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2024023

Reusable attribute-based signature with attribute revocation and authenticated key agreement

ZHANG Zhishuo¹, YANG Huixi², HUANG Wen³, LIAO Yongjian¹, ZHOU Shijie¹

1. School of Information and Software Engineering, University of Electronic Science and Technology, Chengdu 610057, China

2. Department of Mechanical and Electrical Engineering, Cangzhou Vocational and Technical College, Cangzhou 061001, China

3. School of Computer Science, Sichuan University, Chengdu 610065, China

Abstract: In the pursuit of establishing a unified model for fine-grained access control and secure data communication within a distributed microservices architecture, a verifier-policy attributed-based signature (VP-ABS) scheme,

收稿日期: 2023-02-09; 修回日期: 2024-03-10

通信作者: 廖永建, liaoyj@uestc.edu.cn

基金项目: 中央高校国产移动终端操作系统高安全性关键技术研究及方案设计 (No.ZYGX2020ZB019)

Foundation Item: Research and Solution Design of High Security Key Technologies for Domestic Mobile Terminal Operating Systems (No.ZYGX2020ZB019)

引用格式: 张智烁, 杨会喜, 黄文, 等. 支持属性立即撤销的可复用属性基签名及其密钥协商方案[J]. 网络与信息安全学报, 2024, 10(02): 81-94.

Citation Format: ZHANG Z S, YANG H X, HUANG W, et al. Reusable attribute-based signature with attribute revocation and authenticated key agreement[J]. Chinese Journal of Network and Information Security, 2024, 10(02): 81-94.

augmented with attribute revocation and an authenticated key agreement protocol, was proposed. This scheme was underpinned by Type-3 pairing. In this scheme, signatures generated by signers were linked solely to a subset of the signer's attributes and were disassociated from the access policy. This decoupling allowed for the reusability of the signer's signature across multiple access policies. Additionally, an attribute-Hash filter algorithm was introduced to facilitate a direct attribute revocation mechanism within the proposed VP-ABS scheme. This mechanism was designed to prevent users from authenticating with expired attributes. To further secure data communication, a mutually authenticated key agreement protocol was also proposed. This protocol was secure within the framework of the extended Canetti-Krawczyk (eCK) model and was built upon the foundation of the VP-ABS scheme. A rigorous unforgeability proof for the VP-ABS scheme was provided. Ultimately, theoretical comparisons and simulation experiments conducted at a 128-bit advanced encryption standard (AES) security level demonstrated that the proposed attribute-based authentication and secure communication scheme outperforms other contemporary schemes in terms of efficiency.

Keywords: fine-grained access control, attributed Hash filter, attributed direct revocation, policy-decoupled attribute-based signature, mutually authenticated key agreement protocol, existentially unforgeable

0 引言

随着云计算和分布式技术的发展, 分布式微服务架构已经成为云服务提供商采用的主流服务端架构。分布式微服务架构如图 1 所示。在分布式微服务架构^[1]中, 每个应用包含了多个微服务, 每个微服务架设在 1 个小型分布式服务器上, 用户通过前端代理服务器来请求访问后端的微服务, 后端服务器通过前端 (反向) 代理服务器来向用户提供云服务。每个微服务都有相应的访问策略来决定用户的访问权限。

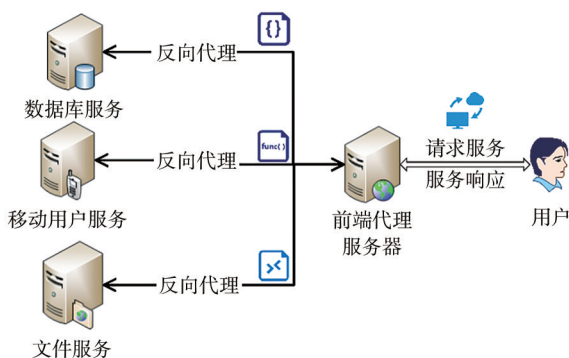


图 1 分布式微服务架构
Figure 1 The framework of distributed micro-services

目前, 主流的基于用户名密码或短消息验证码的微服务认证方案中, 需要至少 1 个存储和管理用户信息的认证服务器为其他微服务提供认证服务。显然, 一旦认证服务器无法正常工作, 所有微服务无法验证用户的合法性, 也就无法为用户提供云服务。此外, 在认证过程中, 需要提前

通过安全传输层 (TLS1.3^[2]/HTTPS^[3]) 搭建一条安全信道来传输认证消息。而在 TLS 搭建过程中, 认证服务器需要公钥基础设施 (public key infrastructure, PKI) (如证书权威机构 (certificate authority, CA) 颁发的公钥证书) 来认可其服务的合法性。随着云服务提供商的指数级增长, CA 管理证书的压力也越来越大, 经常出现证书无法及时更新和过期证书无法立即撤销问题。为了解决证书管理难题, 研究者提出了身份基密码 (identity-based cryptography, IBC)。在 IBC 的密钥生成过程中, 通过将身份信息嵌入密钥中来摆脱对公钥证书的依赖。身份基签名 (identity-based signature, IBS) 作为 IBC 的签名原语, 可实现公开信道中无须公钥证书的身份认证过程。但由于在 IBS 密钥中, 仅嵌入了身份标识信息, 所以无法实现基于访问策略的细粒度访问控制功能, 为分布式微服务架构搭建一个无须公钥证书且无须安全信道的细粒度访问控制及其密钥协商方案。

属性基签名 (attribute-based signature, ABS)^[4]是新型的细粒度消息认证原语, 是 IBS 的继承和发展, 可提供属性级别的访问控制。目前, ABS 可分为密钥策略属性基签名 (key-policy ABS, KP-ABS)、签名策略属性基签名 (signature-policy ABS, SP-ABS) 和认证方策略属性基签名 (verifier-policy ABS, VP-ABS) 3 种类别。

(1) 密钥策略属性基签名

受文献[5]启发, Guo 等^[6]提出了在 Type-1 配对

上的KP-ABS方案,但这个方案会遭受用户公钥替换攻击^[7]。2010年, Li等^[8-9]提出2个在Type-1配对上的支持隐藏用户真实访问策略的门限KP-ABS方案。为了将KP-ABS方案适用于轻量级移动设备中, 2017年, Ren等^[10]提出可验证的外包KP-ABS方案,但KP-ABS方案存在安全漏洞^[11]。Cui等^[12]构造了完全服务器辅助的KP-ABS (server assisted KP-ABS, SA-KP-ABS) 方案。Xiong等^[13]指出上述SA-KP-ABS方案会遭受攻击者和不可信服务器发起的合谋伪造攻击,并提出可抵抗合谋伪造攻击的SA-KP-ABS方案。Rao等^[14]提出了一个KP-ABS方案和相应的密钥策略属性基密签(key-policy attribute-based signcryption, KP-ABSC)方案。

在KP-ABS属性访问策略嵌入用户的密钥中,运行签名算法时,只能对嵌入的访问策略进行签名,可拓展性较差,不适用于多访问策略的分布式微服务云环境中。

(2) 签名策略属性基签名

2008年, Maji等^[15]构造了使用Type-3配对的SP-ABS方案。Cui等^[16]和Liu等^[17]分别构造了基于Maji方案的SP-ABS方案。Zhang等^[18]指出, Maji方案面临致命的签名伪造攻击,这使得基于Maji方案的所有方案会面临这个安全威胁。2014年, Ding等^[19]构造了使用Type-1配对的访问策略为线性秘密共享方案 (linear secret sharing scheme, LSSS) 的可追溯SP-ABS方案,并在q-ADHE困难问题下证明其签名方案的不可伪造性。Rao等^[20]构造了基于SP-ABS原语的属性基签名方案。Yu等^[21]以Rao等^[20]方案为基础,构造了可外包的签名方案。

在SP-ABS中,属性密钥与一个属性集合相绑定,签名者可随意选取其属性集合所满足的访问策略进行签名,灵活性较KP-ABS大幅提高。但每个SP-ABS签名均需要关联1个访问策略,所以对于多访问策略的多微服务云环境来说,用户需要对不同微服务的不同访问策略来生成多个SP-ABS签名供云服务器进行权限验证,这会增大用户端的计算开销。

(3) 认证方策略属性基签名

为了解决SP-ABS对于多个访问策略需要签

名多次的问题, Zhang等^[22]借鉴了外包属性密钥校验 (outsourced key verification, OKV) 思想,在2022年提出了ABS另一种变体——验证方策略属性基签名概念,并使用Type-3配对构造了可证明不可伪造性的VP-ABS^[23]。在VP-ABS概念中,签名算法无须关联访问策略,生成的签名只和用户部分属性有关。访问策略在验证算法中输入,来校验与签名相关联的用户属性是否真实合法并满足访问策略。因为VP-ABS只和用户部分属性相关联,所以VP-ABS实现了签名与访问策略的解耦且实现签名对不同访问策略的可复用性。文献[22]方案只关注VP-ABS用于云服务的认证过程,虽然VP-ABS可在公开信道完成权限认证,但认证过程结束后,云服务器和用户在进行数据交换时,需要另外搭建一条安全信道来完成数据的安全互通。这使得整个系统变得复杂和难以维护。而且,文献[22]方案无法解决用户过期属性的立即撤销问题,这使得用户可使用过期的属性继续访问云服务。

本文主要贡献有以下5点。

(1) 使用Type-3双线性配对构造了更加高效和轻量级持LSSS访问策略的VP-ABS方案,并对其不可伪造性给出了详细证明。本文方案与文献[23]方案不同,认证者无须为每个策略生成1个策略验证公钥就可完成签名校验,节约了前端校验服务器的存储空间。

(2) 为本文VP-ABS方案设计基于可用属性哈希 (Hash) 的过滤算法,可使认证者过滤掉签名者的过期属性。使用本文设计的属性哈希过滤算法,可快捷地实现属性的立即撤销功能。由于过滤算法仅使用可用属性的哈希值,所以在一定程度上可保护签名者 (用户) 的隐私。

(3) 为本文VP-ABS方案设计配套的双向认证密钥协商协议。所协商出的密钥是Type-3类型的SCK-1 (smart-Chen-Kudla-1)^[24]会话密钥,由文献[24]可知,会话密钥能抵抗扩展Canetti-Krawczyk (extended Canetti-Krawczyk, eCK)^[25]安全模型中所有类型的攻击。

(4) 利用可撤销的VP-ABS方案和配套的密钥协商方案。为分布式微服务架构设计了高可用性的认证和数据安全交互方案。

(5) 将本文方案与最新的属性认证方案进行分析和实验仿真对比, 验证了本文方案的完备性和可用性。

1 预备知识

1.1 线性秘密共享方案

设 p 是素数, U 是属性域。对于任意一个在 U 上单调的访问表达式, 可在多项式时间内构造出相对应的 LSSS^[26] 访问控制矩阵 $M_{l \times n}$, $M_{l \times n}$ 的每行都映射在访问表达式中的 1 个属性, 定义这个映射函数为 $\rho(i) \rightarrow \text{attr}_i$, 其中 $1 \leq i \leq l$, $\text{attr}_i \in U$ 。定义待分享的秘密值为 $s \in Z_p^*$, 通过 $s \in Z_p^*$, 构造 1 个元素在 Z_p^* 上的随机向量 $\vec{v} = (s, r_2, r_3, \dots, r_n)$, 其中 $r_2, \dots, r_n \leftarrow_R Z_p^*$, $\leftarrow_R$ 表示从集合中按照均匀分布随机取出的操作, Z_p^* 表示集合 $Z_p = \{0, 1, \dots, p-1\}$ 除 0 外的其他元素组成的集合, 即 $Z_p^* = \{0, 1, \dots, p-1\}$, 则向量 $\vec{\lambda} = M_{l \times n} \vec{v} = (\lambda_1, \lambda_2, \dots, \lambda_l)$ 是访问 $M_{l \times n}$ 对秘密 $s \in Z_p^*$ 的分享向量。根据文献[26], LSSS 有如下性质: 对于 1 个可满足访问结构的属性集 S , 定义行集 $I = \{i | \rho(i) \in S\}_{i=1}^l$, 可在多项式时间内找到常数向量 $\vec{\omega} = (\omega_1, \omega_2, \dots, \omega_{|I|})$, 使其满足 $\sum_{i \in I} \omega_i M_i = (1, 0, 0, \dots)$, 其中 M_i 是矩阵 $M_{l \times n}$ 的第 i 行, 通过表达式 $\sum_{i \in I} \omega_i M_i = (1, 0, 0, \dots)$, 可得表达式 $\sum_{i \in I} \omega_i \lambda_i = s$ 。从而恢复出秘密 $s \in Z_p^*$, 对于任意不能满足访问控制 $M_{l \times n}$ 的属性集合, 无论属性集合所对应 $M_{l \times n}$ 的行向量经过什么线性变化, 恢复向量 $(1, 0, 0, \dots)$ 都无法出现, 即无法恢复出秘密。文献[27]首先提出了可将任意 AND 和 OR 访问表达式编码为 LSSS 访问控制矩阵的高效算法, 在 Lewko 算法^[27]中, 编码出的 $M_{l \times n}$ 元素为集合 $\{-1, 0, 1\}$ 中的一个, 计算出的常数值 $\{\omega_i\}_{\rho(i) \in S}$ 也为集合 $\{-1, 0, 1\}$ 中的一个。进一步地, 定义 $\rho_h(i) \rightarrow \text{Hash}(\text{attr}_i)$, 称 $(M_{l \times n}, \rho_h)$ 为哈希值 LSSS 的访问结构。

1.2 双线性配对

G_1 、 G_2 、 $(G_T)_3$ 是 3 个可高效计算的阶数均为 p 的循环群, 定义双线性配对运算 $e: G_1 \times G_2 \rightarrow (G_T)_3$, 该运算有如下性质。

$$(1) \forall \eta, \zeta \in Z_p^*, e(g^\eta, h^\zeta) = e(g, h)^{\eta\zeta}.$$

(2) $e(g, h) \neq 1$, 当且仅当 $\eta = 0$ 或 $\zeta = 0$ 时, $e(g^\eta, h^\zeta) = 1$ 。其中, g 是 G_1 的生成元, h 是 G_2 的生成元, $e(g, h)$ 是 $(G_T)_3$ 的生成元。

双线性配对有 2 种应用广泛的类型如下。

(1) Type-1 (对称型) 配对: G_1 和 G_2 是同一群, 即 $G_1 = G_2 = G$, $(G_T)_3 = G_T$, $e: G \times G \rightarrow G_T$ 。

(2) Type-3 (非对称型) 配对: G_1 和 G_2 不同群且无法找到映射实现 G_1 和 G_2 中群元素的相互转化, 即 $G_1 = G_2$, $G_1 \not\leftrightarrow G_2$, $e: G_1 \times G_2 \rightarrow (G_T)_3$ 。

由于计算能力的飞速提升, 国际上逐步弃用 80 bit 高级加密标准 (advanced encryption standard, AES)、1 024 bit-RSA、2 048 bit-RSA 安全量级的密码。目前, 国际密码标准需要构造大于或等于 128 bit AES、3 072 bit RSA 安全量级的密码^[28]。而 Type-1 (对称型) 配对由于其底层选取的椭圆曲线群域的扩张性问题, 还无法找到一个高效的、大于或等于 128 bit AES 安全量级的 Type-1 配对。所以, 现在基于双线性配对的密码方案全部是基于 Type-3 (非对称型) 配对。

1.3 数学困难问题及其假设

Type-3 配对上的拓展计算性 Diffie-Hellman 困难问题 (extended computational Diffie-Hellman problem in Type-3 pairing, Type-3 eCDHP): 给定 $g, g^x, g^y \in G_1$ 和 $h, h^x, h^y \in G_2$, 其中 $x, y \in Z_p^*$ 是随机选取的。 g 是 G_1 的生成元, h 是 G_2 的生成元, $e(g, h)$ 是 $(G_T)_3$ 的生成元, $e: G_1 \times G_2 \rightarrow (G_T)_3$ 在多项式时间计算出 $g^{xy} \in G_1$ 或 $h^{xy} \in G_2$ 。

Type-3 配对上的拓展计算性 Diffie-Hellman 假设: 对于任意概率多项式时间的攻击者, 都无法以一个不可忽略的概率破解 Type-3 eCDHP。

2 模型定义

2.1 算法模型

支持立即撤销属性的 VP-ABS 方案及配套的密钥协商方案的 8 个算法如下。

(1) ASetup(λ) $\rightarrow$ (mk, PK): 系统初始化算法, 由属性权威机构 (attribute authority, AA) 运行。输入安全参数 λ , 输出主密钥 mk 和公开参数 PK。

(2) $\text{SigKeyExtract}(\text{mk}, S_u) \rightarrow (\text{SK}_u, \text{EncKey}_u)$:

签名者密钥生成算法, 由AA运行, 输入系统主密钥mk和用户属性 S_u , 输出签名者属性密钥 SK_u 和导出密钥 EncKey_u 。

(3) $\text{VerKeyExtract}(\text{mk}, \text{ID}_v) \rightarrow \text{EncKey}_v$: 认证者密钥生成算法, 由AA运行, 输入主密钥mk和签名者身份 ID_v , 输出认证者和导出密钥 EncKey_v 。

(4) $\text{AttrSign}(\text{SK}_u, S_u, m, T) \rightarrow \sigma(\text{HS}_u, m, T)$: 属性签名算法, 由签名者运行, 输入签名者属性密钥 SK_u , 属性集合 S_u , 待签名消息 m 和时间戳 T , 输出签名 $\sigma(\text{HS}_u, m, T)$ 。

(5) $\text{HFilter}(\text{HashDict}(\text{ID}_u), \sigma(\text{HS}_u, m, T)) \rightarrow \sigma(\text{HSF}, m, T)$: 对于 HS_u 中的每个属性哈希 $H_2(\text{attr}_i)$, 判断是否在 $\text{HashDict}(\text{ID}_u)$ 中, 即 $H_2(\text{attr}_i) \in \text{HashDict}(\text{ID}_u)$, 将通过判断的属性哈希集合输出为HSF, 将签名 $\sigma(\text{HS}_u, m, T)$ 过滤为 $\sigma(\text{HSF}, m, T)$ 。

(6) $\text{Verify}((M_{l \times n}, \rho_h), \sigma(\text{HSF}, m, T)) \rightarrow 0/1$: 签名验证算法由认证者运行, 输入过滤后的签名 $\sigma(\text{HSF}, m, T)$, 待匹配的Hash-LSSS访问策略 $(M_{l \times n}, \rho_h)$, 当前时间戳 T_{cur} , 输出0(认证失败)或1(认证成功)。

(7) **SessionKeyAgreement**: 会话密钥协商利用签名者VP-ABS中的认证项和由认证者发回的不可篡改和伪造信息, 签名者和认证者可协商出满足eCK安全模型的SCK-1^[24]会话密钥。

(8) $\text{AttrRevoke}(\text{ID}_u) \rightarrow (\text{AttrHash}(\text{ID}_u))_{\text{up}}$: 属性撤销算法, 输入待更新属性的用户 ID_u , 输出更新后的 $(\text{AttrHash}(\text{ID}_u))_{\text{up}}$ 。

2.2 安全模型及安全要求

2.2.1 VP-ABS方案安全模型

本文VP-ABS方案在选择策略模型(selective policy model, sP)下, 对自适应选择消息攻击(chosen-message attack, CMA)存在不可伪造性(existentially unforgeable, EUF), 安全模型命名为sP-EUF-CMA, 由敌手A和挑战者C交互的安全游戏所定义。

(1) **Init**阶段: 敌手A向挑战者C定义挑战

访问策略 $(M_{l \times n}^*, \rho^*)$ 。

(2) **Setup**阶段: 挑战者C运行 $\text{ASetup}(\lambda) \rightarrow (\text{mk}, \text{PK})$ 算法, 并将公开参数PK发送给敌手A。

(3) **SigKeyExtractQuery**阶段: 敌手A向挑战者C提交属性集合 S_u , C运行 $\text{SigKeyExtract}(\text{mk}, S_u)$ 算法, 向敌手A导出密钥 $(\text{SK}_u, \text{EncKey}_u)$ 。其中, S_u 无法满足挑战访问策略。这个过程可运行多次。

(4) **VerKeyExtractQuery**阶段: 敌手A自适应地模仿认证者, 向挑战者C提交属性集合 S_v , 挑战者C运行 $\text{VerKeyExtract}(\text{mk}, \text{ID}_v)$ 算法, 返回导出密钥 EncKey_v 给敌手A。这个过程可运行多次。

(5) **AttrSignQuery**阶段: 敌手A向挑战者C提交待签名的属性集合 $S'_u \subset S_u$ 、待签名消息 m 和时间戳 T 。挑战者C运行 $\text{AttrSign}(\text{SK}_u, S'_u, m, T)$, 并返回签名 $\sigma(S'_u, m, T)$ 给敌手A。在这个阶段, 询问的 S'_u 可满足挑战访问策略。这个过程可运行多次。

(6) **Forgery**阶段: 敌手A输出满足挑战访问策略的签名 $\sigma(S^*, m^*, T^*)$, 签名中的 (m^*, S^*, T^*) 不能同时出现在同一签名询问AttrSignQuery中, 且 S^* 不能是任意一个签名询问属性集合 S_i 的子集。

如果敌手A能在Forgery阶段输出合法满足挑战访问结构的签名, 就判定敌手A赢得这个安全游戏; 如果敌手A在多项式时间内, 不能以不可忽略的概率赢得这个安全游戏, 那么就认为本文VP-ABS方案在sP-EUF-CMA安全模型下是安全的。

2.2.2 会话密钥安全性要求

所协商出的会话密钥要满足eCK安全模型, 应具有如下安全性。

(1) 已知会话密钥安全性: 攻击者不能利用已知的会话密钥推断出其他会话密钥。

(2) 强前向安全性: 即使攻击者能窃取通信双方的长私钥和系统主私钥, 已经协商的会话密钥的私密性不受影响。

(3) 可认证安全性和抵抗模拟身份攻击的安全性: 会话密钥中嵌入了通信双方的身份信息, 通信双方均在明确对方身份前提下, 才能协商出会话密钥。攻击者在没有窃取到双方长私钥前提

下,不能模拟任何一方身份。即使攻击者窃取了一方的长私钥,攻击者只能模拟窃取到的长私钥身份,不能模拟其他身份,即攻击者不能从已知的长私钥中推导出其他长私钥,更不能计算出主私钥。

3 方案构造

(1) ASetup(λ) $\rightarrow$ (mk, PK): 由 AA 运行。

1) 使用安全参数 λ 运行 Type-3 配对导出算法, 导出安全的 Type-3 配对实例: $pp_3 = \{G_1, G_2, g, h, p, e: G_1 \times G_2 \rightarrow (G_T)_3\}$;

2) 选择单向抗碰撞的哈希函数为: $H_1: \{0,1\}^* \rightarrow G_1, H_2: \{0,1\}^* \rightarrow G_2, H_3: \{0,1\}^* \rightarrow Z_p^*, h_{sha256}: \{0,1\}^* \rightarrow \{0,1\}^{256}$;

3) 随机选择 $\alpha, \beta, b, c \xleftarrow{R} Z_p^*$ 并计算 h^a, h^b 和 g^c 。将 α, β, b 和 c 设置为系统主私钥 mk。公开参数 $PK = pp_3, H_1, H_2, H_3, H_v, h^a, h^b, g^c$ 。

(2) SigKeyExtract(mk, S_u) $\rightarrow$ ($SK_u, EncKey_u$): 签名者向 AA 提交它的属性集合 S_u , 由 AA 运行以下 3 个步骤。

步骤 1 导出 $ID_u = h_{sha256}(S_u)$ 。AA 随机在 Z_p^* 中取 $t \xleftarrow{R} Z_p^*$ 计算 $SK_u = \{SK_1, SK_2, \{SK_{attr_i}\}_{attr_i \in S_u}\}$ 为:

$$SK_1 = H_1(ID_u)^\alpha g^{bct} \quad (1)$$

$$SK_2 = g^{ct} \quad (2)$$

$$\{SK_{attr_i} = H_2(attr_i)^t\}_{attr_i \in S_u} \quad (3)$$

并导出协商密钥 $EncKey_u = H_1(ID_u)^\beta$ 。

步骤 2 将用户属性的哈希值以 ID_u 为索引, 记录在文件 AttrHash 中, 即 $HashDict(ID_u) = \{H_2(attr_i)\}_{attr_i \in S_{alive}}$, 并将 AttrHash 文件同步给所有认证者, 认证者使用文件 AttrHash 运行 HashFilter 算法。

步骤 3 将 $\{SK_u, EncKey_u\}$ 通过安全信道返回给用户。用户运行如式 (4)~(6) 所示的测试, 来验证返回密钥的合法性。

$$e(SK_1, h) \stackrel{?}{=} e(H_1(ID_u), h^a) e(SK_2, h^b) \quad (4)$$

$$\left\{ e(g^c, SK_{attr_i}) \stackrel{?}{=} e(SK_2, H_2(attr_i)) \right\}_{attr_i \in S_u} \quad (5)$$

$$e(EncKey_u, h) \stackrel{?}{=} e(H_1(h_{sha256}(S_u)), h^\beta) \quad (6)$$

(3) VerKeyExtract(mk, ID_v) $\rightarrow$ $EncKey_v$: 认证者向 AA 提交资质 S_v , AA 首先验证认证者的资质, 然后导出 $ID_v = h_{sha256}(S_v)$, 并计算 $EncKey_v = H_1(ID_v)^\beta$ 。认证者(微服务器)验证密钥的合法性, 如式 (7) 所示。

$$e(EncKey_v, h) \stackrel{?}{=} e(H_1(h_{sha256}(S_v)), h^\beta) \quad (7)$$

(4) AttrSign(SK_u, S_u, m, T) $\rightarrow$ $\sigma(HS_u, m, T)$: 对消息 m 和时间戳 T 在属性集合 S_u 进行签名。

1) Offline 离线计算: $ID_u = h_{sha256}(S_u)$, $HS_u = \{H_2(attr_i)\}_{attr_i \in S_u}$

2) Online 在线计算: 签名者随机选择 r , $r_k \xleftarrow{R} Z_p^*$ 计算签名随机项 Y_1 和瞬时协商密钥项 K_1 为 $Y_1 = H_1(ID_u)^r$, $K_1 = h^{r_t}$; 计算签名消息哈希值 $h_m = H_3(m \| T \| ID_u \| Y_1 \| K_1) \in Z_p^*$ 。输出签名如式 (8) 所示。

$$\sigma(HS_u, m, T) = \left(T, m, ID_u, HS_u, Y_1, K_1, \sigma_1, \sigma_2, \{\sigma_{attr_i}\}_{H_2(attr_i) \in HS_u} \right) \quad (8)$$

$$\sigma_1 = (SK_1)^{r+h_m} \quad (9)$$

$$\sigma_2 = (SK_2)^{r+h_m} \quad (10)$$

$$\sigma_{attr_i} = (SK_{attr_i})^{r+h_m} \quad (11)$$

(5) HFilter($HashDict(ID_u), \sigma(HS_u, m, T)$) $\rightarrow$ $\sigma(HSF, m, T)$: 判断 HS_u 中的每个属性哈希值 $H_2(attr_i)$ 是否在 $HashDict(ID_u)$ 中, 即 $H_2(attr_i) \in HashDict(ID_u)$, 通过判断的哈希集合输出为 HSF, 将签名 $\sigma(HS_u, m, T)$ 过滤为 $\sigma(HSF, m, T)$ 。签名构成如式 (12) 所示。

$$\sigma(HSF, m, T) = \left(T, m, ID_u, HSF, Y_1, K_1, \sigma_1, \sigma_2, \{\sigma_{attr_i}\}_{H_2(attr_i) \in HSF} \right) \quad (12)$$

$$\sigma_1 = (SK_1)^{r+h_m} \quad (13)$$

$$\sigma_2 = (SK_2)^{r+h_m} \quad (14)$$

$$\left\{ \sigma_{attr_i} = (SK_{attr_i})^{r+h_m} \right\}_{H_2(attr_i) \in HSF} \quad (15)$$

(6) Verify($(M_{l \times n}, \rho_h), \sigma(HSF, m, T)$) $\rightarrow$ 0/1: 由认证者运行签名验证算法。

1) 判断签名 $\sigma(\text{HSF}, m, T)$ 中的时间戳 T 是否在许可时限内, 当且仅当, T 在时限内, 进行步骤2; 否则, 算法返回0。

2) 验证过滤后的属性子集是否满足访问策略: 访问策略 $(M_{l \times n}, \rho_h)$ 与 HSF 相匹配的属性集合所对应的行号 $I = \{i | \rho_h(i) \in \text{HSF}\}_{i=1}^l$, 如果能用高斯消元法在多项式时间内找到常数 $\{\omega_i\}_{i \in I}$, 使得 $\sum_{i \in I} \omega_i M_i = \overbrace{(1, 0, \dots, 0)}^n$ 表达式成立, 就说明 HSF 满足访问策略 $(M_{l \times n}, \rho_h)$ 。如果找不到, 则算法返回0; 如果能找到, 进行步骤3。

3) 验证匹配属性行号 $I = \{i | \rho_h(i) \in \text{HSF}\}_{i=1}^l$ 对应签名的合法性和正确性。计算校验哈希值 $\hat{h}_m = H_3(m \| T \| \text{ID}_u \| Y_1 \| K_1) \in Z_p^*$, 判断 $E_1 \stackrel{?}{=} E_2$, $\{E_{\text{attr},1} \stackrel{?}{=} E_{\text{attr},2}\}_{i \in I}$ 是否成立, 其中:

$$\begin{aligned} E_1 &= e(\sigma_1, h) = e(H_1(\text{ID}_u)^{a(r+h_m)} g^{\text{bct}(r+h_m)}, h) = \\ &= e(H_1(\text{ID}_u), h)^{a(r+h_m)} \cdot e(g, h)^{\text{bct}(r+h_m)} \quad (16) \\ E_2 &= e(Y_1 \cdot H_1(\text{ID}_u)^{\hat{h}_m}, h^a) \cdot e(\sigma_2, h^b) = \end{aligned}$$

$$\begin{aligned} &= e(H_1(\text{ID}_u)^r \cdot H_1(\text{ID}_u)^{\hat{h}_m}, h^a) \cdot e(\sigma_2, h^b) = \\ &= e(H_1(\text{ID}_u), h)^{a(r+\hat{h}_m)} \cdot e(g^{\text{ct}(r+h_m)}, h^b) = \\ &= e(H_1(\text{ID}_u), h)^{a(r+\hat{h}_m)} \cdot e(g, h)^{\text{bct}(r+h_m)} \quad (17) \end{aligned}$$

$$E_{\text{attr},1} = e(g^c, \sigma_{\text{attr}}) = e(g^c, H_2(\text{attr}_i)^{t(r+h_m)}) \quad (18)$$

$$E_{\text{attr},2} = e(\sigma_2, H_2(\text{attr}_i)) e(g^{\text{ct}(r+h_m)}, H_2(\text{attr}_i)) \quad (19)$$

如果所有判别式 (16)~(19) 成立, 则签名在验证匹配属性行号 I 中合法, 算法返回1; 否则, 返回0。

(7) SessionKeyAgreement: 会话密钥协商。

1) 签名者 (用户) 使用属性密钥, 在本地运行签名算法 $\text{AttrSign}(\text{SK}_u, S'_u, m, T)$, 将输出的签名 $\sigma(S'_u, m, T)$ 发送给认证者。

2) 认证者端整体验证逻辑如图2所示。首先, 运行哈希属性过滤算法将撤销的属性过滤掉 $\text{HFilter}(\text{HashDict}(\text{ID}_u), \sigma(\text{HS}_u, m, T)) \rightarrow \sigma(\text{HSF}, m, T)$; 然后, 将过滤后的签名运行认证算法 $\text{Verify}((M_{l \times n}, \rho_h), \sigma(\text{HSF}, m, T)) \rightarrow 0/1$, 判断过滤后签名的权限和合法性。

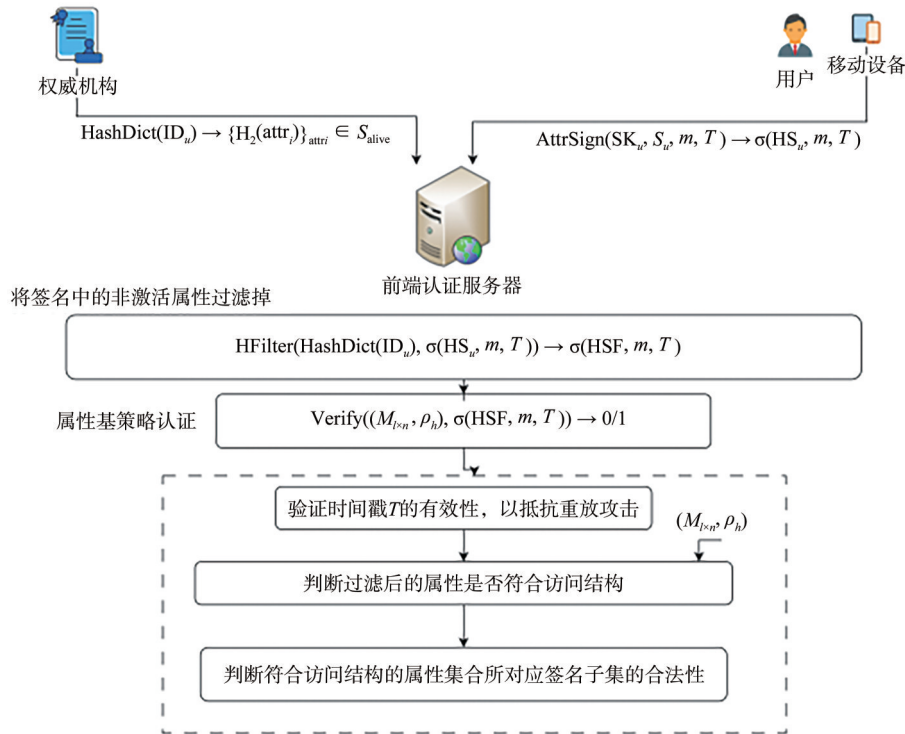


图2 认证者端整体验证逻辑
Figure 2 The global verification logic of verifier side

如果验证成功, 认证者随机取 $r_v \xleftarrow{R} Z_p^*$, 使用密钥导出函数 (KDF) 导出嵌入身份信息 (ID_u, ID_v) 的临时会话密钥如式 (20) 所示。

$$\begin{aligned} K_{vu} &= \text{KDF}(e(H_1(ID_u)^{r_v}, h^\beta) e(\text{EncKey}_v, K_1), K_1^{r_v}) \\ &= \text{KDF}(e(H_1(ID_u), h)^{r_v \beta} e(H_1(ID_v), h)^{r_v \beta}, h^{r_v \beta}, h^{r_v \beta}) \end{aligned} \quad (20)$$

设置 $\text{msg}_v = \text{success}$ 并将会话密钥协商项报文及其 HMAC_{vu} 发送给签名者; 否则, 直接将 fail 发送给签名者, 如式 (21) 所示。

$$(\text{HMAC}_{vu} = h_{\text{sha256}}(K_{vu}, h^{r_v}, \text{msg}_v \| ID_v), h^{r_v}, \text{msg}_v \| ID_v) \quad (21)$$

3) 签名者收到认证者发来的成功消息 $\text{sg}_v = \text{success}$ 后, 计算临时密钥如式 (22) 所示。

$$\begin{aligned} K_{uv} &= \text{KDF}(e(\text{EncKey}_u, h^{r_v}) e(H_1(ID_v)^{r_i}, h^\beta), h^{r_v \beta}, h^{r_v \beta}) \\ &= \text{KDF}(e(H_1(ID_u), h)^{r_i \beta} e(H_1(ID_v), h)^{r_i \beta}, h^{r_i \beta}, h^{r_i \beta}) \end{aligned} \quad (22)$$

利用式 (23) 判断临时密钥的正确性一致性。

$$\begin{aligned} h_{\text{sha256}}(K_{uv}, h^{r_v}, \text{msg}_v \| ID_v) &= \\ h_{\text{sha256}}(K_{vu}, h^{r_v}, \text{msg}_v \| ID_v) \end{aligned} \quad (23)$$

如果式 (23) 成立, 发送 $\text{msg}_u = \text{success}$ 给认证者, 完成密钥协商过程。

当密钥导出函数是安全时, 临时密钥 K_{vu} (K_{uv}) 的私密性就决定了整个密钥协商协议的安全性。而本文方案所导出的临时密钥是 Type-3 配对类型的 SCK-1 临时密钥, 文献[25]被验证在 eCK 模型下是安全的。

(8) $\text{AttrRevoke}(ID_u) \rightarrow (\text{AttrHash}(ID_u))_{\text{up}}$: 属性撤销算法由 AA 运行, 输入用户 ID_u , 根据用户 ID_u 查询在时效范围内的属性 $\text{attr}_i \in (S_u)_{\text{alive}}$, 输出更新后的属性哈希集合 $(\text{AttrHash}(ID_u))_{\text{up}} = \{H_2(\text{attr}_i)\}_{\text{attr}_i \in (S_u)_{\text{alive}}}$, 并将更新后的属性哈希集合 $(\text{AttrHash}(ID_u))_{\text{up}}$ 同步给所有认证者。

4 安全性证明及分析

4.1 VP-ABS 方案安全性证明

本节将对本文 VP-ABS 方案在 sP-EUF-CMA 安全模型下进行不可伪造性证明。

证明: 假如存在敌手 A 可自适应选择签名消

息, 在给定访问结构, 问询 H_1 最多 q_{H_1} 次, 问询 H_3 最多 q_{H_3} 次, 问询 $\text{SigKeyExtractQuery}$ 最多 q_E 次, 问询 AttrSignQuery 最多 q_S 次, 以概率 $\epsilon_1 \geq \frac{10(q_S + 1)(q_S + q_{H_3})}{p}$ 可伪造满足挑战访问策略的 VP-ABS 合法签名, 则可构造挑战算法挑战者 C 在多项式时间内, 以不可忽略的概率 $\epsilon_2 \geq \frac{1}{9}$ 解决 Type-3 eCDHP。

Init 阶段: 敌手 A 向挑战者 C 提交挑战访问策略 $(M_{I \times n}^*, \rho^*)$ 。挑战者 C 从 Type-3 eCDHP 获取的挑战实例 $g, g^a, g^\gamma \in G_1$ 和 $h, h^a, h^\gamma \in G_2$, 其中 $a, \gamma \in Z_p^*$ 是困难问题随机选取的对挑战者 C 未知的值。 g 是 G_1 的生成元, h 是 G_2 的生成元, $e: G_1 \times G_2 \rightarrow (G_T)_3$ 。

Setup 阶段: 挑战者 C 随机选取 $c, b, \beta \xleftarrow{R} Z_p^*$, 公开参数为 $\text{PK} = h^a, h^b, g^c$ 。

问询阶段: 在这个阶段, 敌手 A 可自适应地向挑战者 C 发起问询。

(1) 哈希问询: 挑战者 C 存有 4 个哈希问询 $L_{H_1}, L_{H_2}, L_{H_3}$ 和 L_{H_4} , 如果有相应问询输入, 挑战者 C 将列表中存有的值经过计算后, 作为输出; 否则, 挑战者 C 将进行以下步骤。

步骤 1 (H_5 问询): 输入 $S_i \in \{0, 1\}^*$, 挑战者 C 随机选择 $ID_i \in \{0, 1\}^m$ 作为输出, 并将 $\{S_i: ID_i\}$ 记录在 L_{H_5} 中。

步骤 2 (H_1 问询): 输入 $ID_i \in \{0, 1\}^m$, 挑战者 C 查询 $\{S_i: ID_i\}$ 是否记录在 L_{H_5} 中, 如果不存在, 则中止这次问询; 如果存在, 当 S_i 满足挑战访问策略 $(M_{I \times n}^*, \rho_h^*)$, 挑战者 C 设置 $\eta_i = 1$, 随机选取 $\delta_i \xleftarrow{R} Z_p^*$ 计算 $Q_i = g^{\gamma \delta_i}$ 为输出, 并将 $\{ID_i: \eta_i, \delta_i\}$ 记录在 L_{H_1} 中; 如果 S_i 不满足挑战访问策略 $(M_{I \times n}^*, \rho_h^*)$, 挑战者 C 设置 $\eta_i = 0$, 随机选取 $\delta_i \xleftarrow{R} Z_p^*$ 计算 $Q_i = g^{\delta_i}$ 为输出, 并将 $\{ID_i: \eta_i, \delta_i\}$ 记录在 L_{H_1} 中。

步骤 3 (H_2 问询): 输入 $\text{attr}_i \in \{0, 1\}^*$, 挑战者 C 随机选择 $a_i \xleftarrow{R} Z_p^*$ 计算 $A_i = h^{a_i}$ 作为输出, 并将 $\{\text{attr}_i: a_i\}$ 记录在 L_{H_2} 中。

步骤 4 (H_3 问询): 输入 $Y_1 \in G_1, K_1 \in G_2$

和 $m_i \| T_i \| ID_i \| HS_i \in \{0, 1\}^*$, 挑战者 C 查询 $\{S_i: ID_i\}$ 是否记录在 L_{H_3} 中且 $HS_i = \{H_2(attr_i)\}_{attr_i \in S_i}$, 如果不存在, 中止此次问询; 如果存在, 挑战者 C 随机选择 $h_i \xleftarrow{R} Z_p^*$ 输出, 并将 $\{(m_i \| T_i \| ID_i \| HS_i, Y_i, K_i): h_i\}$ 记录在 L_{H_3} 中。

(2) SigKeyExtractQuery: 敌手 A 向挑战者 C 提交性集 S_i , 当 S_i 满足挑战访问策略 $(M_{l \times n}^*, \rho^*)$ 时, 挑战者 C 仅生成协商密钥 $EncKey_i = H_1(H_5(S_i))^\beta = g^{\gamma\delta, \beta}$ 给敌手 A ; 当 S_i 不满足挑战访问策略 $(M_{l \times n}^*, \rho^*)$ 时, 挑战者 C 随机取 $t_i \xleftarrow{R} Z_p^*$ 生成属性密钥如式 (24)~(26) 所示。

$$SK_i = H_1(H_5(S_i))^\alpha g^{bct_i} = g^{\alpha\delta, i} g^{bct_i} \quad (24)$$

$$SK_2 = g^{ct_i} \quad (25)$$

$$\{SK_{attr_k} = H_2(attr_k)^{t_i} = h^{a, t_i}\}_{attr_k \in S_i} \quad (26)$$

并导出协商密钥 $EncKey_i = H_1(H_5(S_i))^\beta = g^{\delta, \beta}$ 。

(3) VerKeyExtractQuery 阶段: 敌手 A 模拟认证者向挑战者 C 提交 S_v , 当 S_i 满足挑战访问策略 $(M_{l \times n}^*, \rho^*)$ 时, 挑战者 C 生成 $EncKey_v = H_1(H_5(S_v))^\beta = g^{\gamma\delta, \beta}$ 给敌手 A ; 当 S_i 不满足挑战访问策略 $(M_{l \times n}^*, \rho^*)$ 时, 挑战者 C 生成 $EncKey_v = H_1(H_5(S_v))^\beta = g^{\delta, \beta}$ 给 A 。

(4) AttrSignQuery 阶段: 敌手 A 向挑战者 C 提交待签名属性集合 S_i , 待签名消息 m_i 和时间戳 T_i 。挑战者 C 查询 $\{S_i: ID_i\}$ 是否记录在 L_{H_3} 中, 如果不存在, 中止这次问询; 如果存在, 挑战者 C 模拟生成问询签名。

挑战者 C 首先随机选取 $h_i, r, t_h \xleftarrow{R} Z_p^*, K_1 \xleftarrow{R} G_2$, 然后计算并隐喻设置 $Y_i = g^r \cdot (H_1(ID_i)^{h_i})^{-1}$, 最后生成模拟签名如式 (27)~(29) 所示。

$$\sigma_1 = g^{ar} g^{bct_h} \quad (27)$$

$$\sigma_2 = g^{ct_h} \quad (28)$$

$$\{\sigma_{attr_k} = H_2(attr_k)^{t_h}\}_{attr_k \in S_i} \quad (29)$$

在这种隐喻的设置下, 存在式 (30)。

$$Y_i = H_1(ID_i)^{r^{\oplus}} \leftrightarrow g^r \cdot (H_1(ID_i)^{h_i})^{-1} \quad (30)$$

无论 S_i 是否满足挑战访问策略 $(M_{l \times n}^*, \rho^*)$, 所有签名项均可成功模拟。最后, 挑战者 C 将 $\{(m_i \| T_i \| ID_i \| S_i, Y_i, K_i): h_i\}$ 记录在 L_{H_3} 中。

挑战者 C 模拟的签名是正确且不可区分的, 即敌手 A 在多项式时间内无法区分挑战者 C 生成的模拟签名与正常签名。即模拟生成的签名可通过判别式验证 $Eq_1: E_1 = E_2, Eq_3: \{E_{attr,1} = E_{attr,2}\}_{i \in I}$, 其中各项计算结果如式 (31)~(34) 所示。

$$E_1 = e(\sigma_1, h) = e(g^{ar} g^{bct_h}, h) = e(g, h)^{ar} e(g, h)^{bct_h} \quad (31)$$

$$E_2 = e(Y_i \cdot H_1(ID_i)^{h_i}, h^a) \cdot e(\sigma_2, h^b) = e(g^r \cdot (H_1(ID_i)^{h_i})^{-1} \cdot H_1(ID_i)^{h_i}, h^a) e(g^{ct_h}, h^b) = e(g, h)^{ar} e(g, h)^{bct_h} \quad (32)$$

$$E_{attr,1} = e(g^c, \sigma_{attr_i}) = e(g^c, H_2(attr_i)^{t_h}) \quad (33)$$

$$E_{attr,2} = e(\sigma_2, H_2(attr_i)) = e(g^{ct_h}, H_2(attr_i)) \quad (34)$$

Forgery 阶段: 敌手 A 输出满足挑战访问策略签名 $\sigma(S^*, m^*, T^*)$, 签名中的 (m^*, S^*, T^*) 不能同时出现在签名问询 AttrSignQuery 中。挑战者 C 通过分叉引理^[29]重放游戏条带模拟方法, 得到敌手 A 提交的 2 个伪造的签名如式 (35)、式 (36) 所示。

$$\sigma(S^*, m^*, T^*) = (T^*, ID^*, S^*, Y_1^*, Y_2^*, \sigma_1, \sigma_2, \{\sigma_{attr_k}\}_{attr_k \in S^*}) \quad (35)$$

$$\sigma_2(S^*, m^*, T^*) = (T^*, ID^*, S^*, Y_1^*, Y_2^*, \sigma_1', \sigma_2', \{\sigma_{attr_k}'\}_{attr_k \in S^*}) \quad (36)$$

在 2 个敌手 A 提交的伪造签名中, 只有 H_3 的输出选择不同随机数, 即 $H_3 \Rightarrow (h_m, h'_m)$, H_1 、 H_2 和 H_5 选择都保持一致且随机数 $r^*, t^* \in Z_p^*$ 。挑战者 C 判断 A 查询 $\{S_i: ID^*\}$ 是否记录在 L_{H_3} 中, S^* 不满足挑战访问策略且 L_{H_3} 的 $\{ID^*: \eta^*, \delta^*\}$ 中 $\eta_i = 1$, 如果条件不成立, 中止游戏; 如果条件成立, 则存在 $H_1(ID^*) = g^{\gamma\delta^*}$ 。挑战者 C 可计算 $g^{\gamma\delta^*}$ 解决 Type-3 eCDHP, 如式 (37)~(39) 所示。

$$\frac{\sigma_2}{\sigma_2'} = \frac{g^{ct^*(r^* + h_m^*)}}{g^{ct^*(r^* + h'_m^*)}} = g^{ct^*(h_m^* - h'_m^*)} \quad (37)$$

$$\frac{\sigma_1}{\sigma_1'} = \frac{(g^{\alpha\gamma\delta^*})^{r^* + h_m^*} g^{\text{bct}(r^* + h_m^*)}}{(g^{\alpha\gamma\delta^*})^{r^* + h_m'^*} g^{\text{bct}(r^* + h_m'^*)}} = \frac{g^{\alpha\gamma\delta^*(h_m^* - h_m'^*)} g^{\text{bct}(h_m^* - h_m'^*)}}{1} \quad (38)$$

$$g^{\gamma\alpha} = \left(\frac{\sigma_1}{\sigma_1' \left(\frac{\sigma_2}{\sigma_2'} \right)^b} \right)^{\frac{1}{\delta^*(h_m^* - h_m'^*)}} \quad (39)$$

根据分叉引理^[29], 条带重放方法成功概率为 $\frac{1}{9}$, 所以挑战者 C 成功解决 eCDHP 的概率为 $\frac{1}{9}$ 。

4.2 会话密钥安全性分析

签名者与认证者协商出的会话密钥正确性如式 (40) 所示。

$$K_{uv} = K_{vu} = \text{KDF}(e(H_1(ID_u), h)^{r_u\beta} \cdot e(H_1(ID_v), h)^{r_v\beta}, h^{r_u, r_v}) \quad (40)$$

本文方案协商出的密钥和文献[25]协商出的密钥一致。文献[25]被严格验证在 eCK 模型下是语义安全的, 具有如下安全性。

(1) 已知会话密钥安全性: 由于每次会话密钥中随机数 r , $r_v \xleftarrow{R} Z_p^*$ 都是在协商过程中随机选取的, 所以各个会话密钥间是相互独立的。攻击者不能利用已知的会话密钥, 推断出其他会话密钥。

(2) 强前向安全性: 即使攻击者能窃取到通信双方的长私钥 $H_1(ID_v)^\beta$, $H_1(ID_u)^\beta$ 和系统主私钥, 且攻击者能监听密钥协商信道来获取 h' 和 h^{r_u} , 则攻击者只能恢复出这个密钥的导出项 $e(H_1(ID_u), h)^{r_u\beta} e(H_1(ID_v), h)^{r_v\beta}$ 。由于 Type-3 eCDHP 攻击者仍无法恢复出另一项 h^{r_u, r_v} , 所以已协商会话密钥的私密性不受影响, 即满足强前向安全性。

(3) 可认证安全性和抵抗模拟身份攻击的安全性: 通信双方均要使用嵌入自己身份的长私钥才能导出会话密钥, 且在计算中也需要在会话密钥中嵌入对方的身份信息。攻击者在没有窃取到双方长私钥前提下, 不能模拟任何一方身份。进一步地, 攻击者不能从已知长私钥推导出其他长私钥, 更不能计算出主私钥。

(4) 安全分析: 因为 G_1 群的阶 p 为素数, 所以任意 $H_1(ID) \in G_1$ 都是 G_1 群的生成元, 由于 G_1 群上离散对数困难问题, 任意概率多项式时间内

的攻击者均无法由 $H_1(ID)^\beta$ 计算出 β 。由于哈希函数 H_1 的单向性和抗碰撞性, $H_1(ID) \in G_1$ 在群 G_1 上是随机的, 所以每个长私钥 $H_1(ID)^\beta$ 在群 G_1 上都是相互独立和随机的, 攻击者不能从已知长私钥推导出其他长私钥, 更不能计算出主私钥。

5 方案部署应用

将本文支持属性过滤的 VP-ABS 方案及其密钥协商方案应用在分布式微服务架构中, 为其构建细粒度用户认证及数据安全传输方案。在微服务架构中, 用户作为签名者, 微服务器作为认证者, 各微服务器间数据不共享、不互通, 前端代理服务器用于数据中转用途和代理验证, 是“诚实—好奇”半可信的, 即会诚实地执行系统指令, 但也会尝试窃听服务器和用户的交互数据。系统整体架构及运行流程如图 3 所示。

(1) 系统初始化: AA 运行 ASetup(λ), 公布公开参数 PK 完成系统初始化。

(2) 用户初始化: 用户提交属性集合 S_u 给 AA, AA 运行 SigKeyExtract(mk, S_u) 为用户生成密钥 (SK_u , EncKey $_u$) 返回给用户。AA 和用户需要安全信道进行交互。

(3) 服务器初始化 (包括后台微服务器和反向代理服务器): 微服务器向 AA 提交资质凭证, AA 运行 VerKeyExtract(mk, ID_v) 为微服务器生成预备协商密钥 EncKey $_v$, 并将用户的可用属性哈希集合同步给反向代理服务器。在此过程中, AA 和服务器需要安全信道交互。各微服务器指定访问策略 $(M_{l \times m}, \rho_h)_i$, 并发送给反向代理服务器 (如 Nginx)。由于访问策略中的值由哈希函数进行盲化, 所以通过公开信道传输即可。

(4) 用户认证: 用户运行签名生成算法 AttrSign(SK_u, S_u, m, T) $\rightarrow \sigma(HS_u, m, T)$, 将生成的签名和将要访问的微服务代码发送给前端反向代理服务器。首先, 前端代理服务器运行哈希属性过滤算法过滤掉用户发送签名中被撤销的属性, 导出只包含可用属性的签名 HFilter(HashDict(ID_u), $\sigma(HS_u, m, T)$) $\rightarrow \sigma(HSF, m, T)$ 。然后, 将过滤后的签名与请求的微服务所对应的访问结构

$(M_{l \times n}, \rho_h)_i$ 输入签名认证算法 $\text{Verify}((M_{l \times n}, \rho_h)_i, \sigma(\text{HSF}, m, T)) \rightarrow 0/1$ ，来判断用户对每个微服务的访问权限；最后，将用户的签名发送给通过验证的微服务器。在这个过程中，用户签名可对应多个微服务器的访问策略的认证过程，所以用户生成的签名具有复用性。

(5) 密钥协商：微服务器和用户均运行基于本文 VP-ABS 认证密钥协商方案，协商出安全信道密钥进行数据交互。本文系统模型中的每个微服务器都与用户搭建一条安全信道，这样保证了每个微服务数据隔离，信道的安全性也相互独立，不会互相影响，增加了系统整体的容错能力。

(6) 属性撤销：AA 将更新后的属性哈希集合 $(\text{AttrHash}(\text{ID}_u))_{\text{up}} \rightarrow \{H_2(\text{attr}_i)\}_{\text{attr}_i \in (S_u)_{\text{ave}}}$ 同步给所有前端反向代理服务器。

6 方案理论对比和实验分析

6.1 方案理论对比

本节首先对本文方案与具有属性认证功能的密码学认证方案在特性和效率方面进行理论分析和对比，方案特性对比如表1所示，方案复杂度对比如表2所示。对比涉及的参数如下。

(1) T^{p1} : Type-1 配对运算所需时间。

(2) $T_G^e/T_{G_T}^e$: Type-1 配对 G/G_T 群上指数运算所需时间。

(3) T^{p3} : Type-3 配对运算所需时间。

(4) $T_{G_1}^e/T_{G_2}^e/T_{(G_T)_3}^e$: Type-3 配对 $G_1/G_2/(G_T)_3$ 群上指数运算所需时间。

(5) $T_{G_1}^h/T_{G_2}^h$: 群元素哈希运算 $\{0, 1\}^* \rightarrow G_1/\{0, 1\}^* \rightarrow G_2$ 的所需时间。

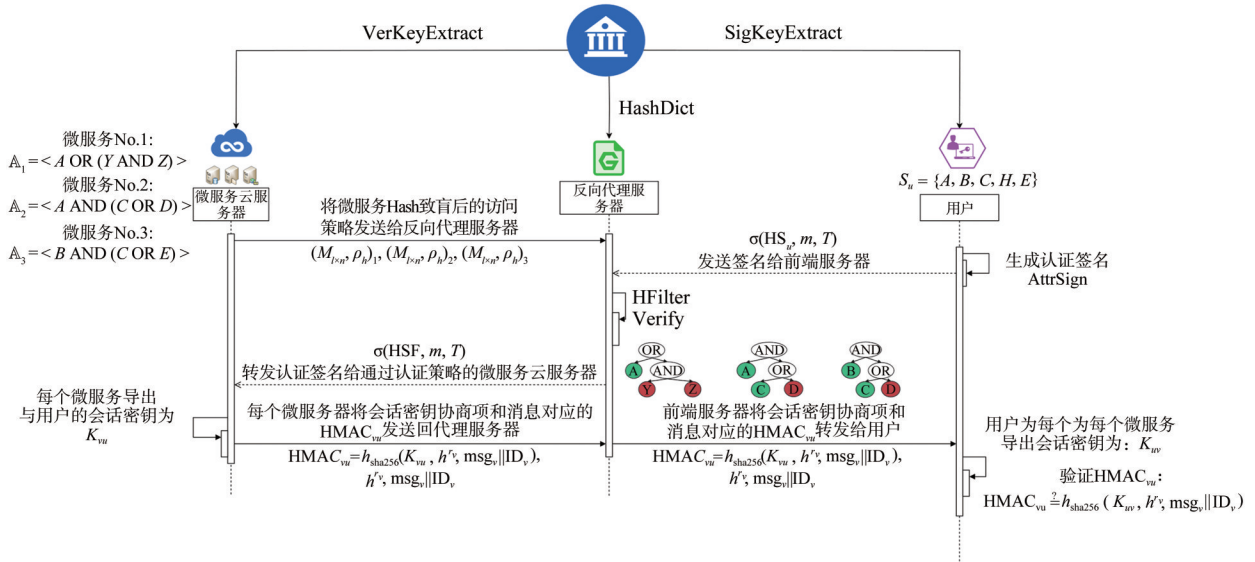


图3 系统整体架构及运行流程
Figure 3 The framework and workflow of overall system

表1 方案特性对比
Table 1 Comparison of scheme features

方案	方案类型	配对类型	无限属性空间	认证后密钥协商	多策略复用	动态属性撤销	不可伪造性证明	属性隐藏
文献[19]	SP-ABS	Type-1	×	×	×	×	√	×
文献[22]	VP-ABS	Type-3	×	×	√	×	×	×
文献[23]	OKV	Type-3	√	×	√	×	√	×
文献[30]	OKV	Type-3	×	×	√	√	×	×
本文	VP-ABS	Type-3	√	√	√	√	√	√

注：

- (6) n_u : 属性集 S_u 中的属性个数。
- (7) l : LSSS 访问矩阵 $M_{l \times n}$ 的行数。
- (8) n_i : 属性匹配的行标集合 I 中包含行标的个数, 存在 $n_i \leq l$ 。
- (9) $|G|/|G_T|/|Z_p^*|$: Type-1 配对 $G/G_T/Z_p^*$ 群中元素所占空间大小。
- (10) $|G_1||G_2||G_T|_3/|Z_p^*|_3$: Type-3 配对 $G_1/G_2/(G_T)_3/Z_p^*$ 群中元素所占空间大小。
- (11) $|T_\sigma|$: 时间戳所占空间大小。
- (12) $|\text{ID}_u|$: $\text{ID}_u = h_{\text{sha256}}(S_u)$ 所占空间大小。
- (13) $|\text{attr}|$: 属性所占空间大小。

由表 1 可以看出, 本文方案在可验证语义安全 (不可伪造性) 前提下, 完成了用户访问控制认证和动态属性撤销, 并能完成签名者和认证者互认证的会话密钥协商, 使数据能通过搭建的安全信道安全互通, 完成了签名的认证复用和用户属性的哈希隐私盲化, 适用于分布式微服务架构中的用户认证场景。

由表 2 可以看出, 本文方案在签名生成算法时间复杂度上与同类型基于 VP-ABS 的面向服务属性认证文献[23]方案相比, 仅多出 $T_{G_2}^e$ 运行时间, 这是因为本文方案具有密钥协商功能, 多出的 $T_{G_2}^e$ 运算是生成密钥协商项所增加的, 增加的较少计算开销对于系统安全性的提升是完全值得的; 另外, 本文方案的在服务端的验证算法与文献[23]方案相比有很大提升, 减少了时间为 $n_i T_{(G_T)_3}^e + n_i T_{G_2}^h$ 的复杂度。随着 n_i 的增加, 减少的时间会更多。

6.2 方案实验分析

为了比较本文方案和同类型 VP-ABS 的文献[23]方案在实际场景中签名和认证算法的运行效

率, 本文在 Ubuntu 20.04 (gcc/g++ 9.4.0) 16 GB 内存 CPU R7-5800H 环境下, 对文献[23]方案和本文方案用 C++ 语言模拟。在 Type-3 配对上, 使用 Core 库, Type-3 配对为 BLS12-383, 是经过国际密码组织认证的 128 bit AES 安全量级的配对曲线。

Sign 算法模拟运行所需时间对比如图 4 所示, 设置 $n_i = n_u$ 。由图 4 可以看出, 在 128 bit (AES) 安全量级下, 文献[23]方案与本文方案的签名算法实际运行所需时间基本相同。

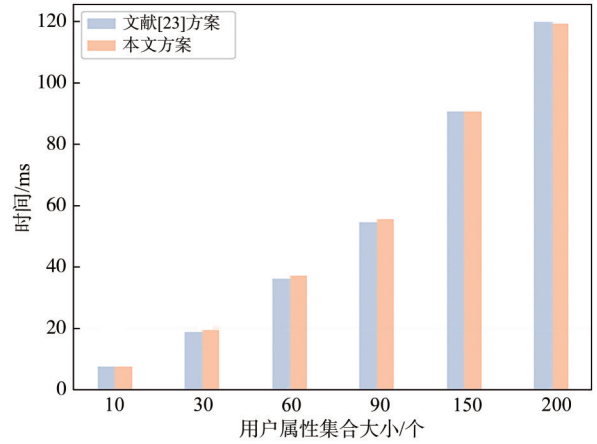


图 4 Sign 算法模拟运行所需时间
Figure 4 Simulation run time of Sign algorithm

另外, 当用户属性集合等于 90 时, 本文方案只需不到 60 ms 即可生成验证签名块, 非常高效和实用。

Verify 算法模拟运行所需时间对比如图 5 所示。由图 5 可以看出, 在 128 bit (AES) 安全量级下, 随着 $n_i = n_u$ 的增大, 文献[23]的 Verify 算法所需时间比本文方案多, 这与理论分析所得的结果一致。本文方案在匹配的访问策略属性集合大小 $n_i = n_u = 60$ 时, 只需不到 500 ms。微服务器在自定义访问策略时, 一般不会超过 20 个叶子

表 2 方案复杂度对比
Table 2 Comparison of scheme complexity

方案	认证生成时间/ms	验证时间/ms	认证项通信负载
文献[19]	$4lT_G^e + 4T_G^e$	$3T^{p1} + lT_G^e + lT^{p1}$	$l G + 3 G $
文献[22]	$(1 + n_u)T_{G_1}^e + 3T_{G_2}^e$	$2T_{G_2}^e + 2T^{p3} + 2n_i T^{p3} + T_{G_1}^h$	$n_u G_1 + 3 G_2 + G_1 + Z_p^* _3 + \text{ID}_u $
文献[23]	$3T_{G_1}^e + n_u T_{G_2}^e$	$T_{G_1}^e + (3 + 2n_i)T^{p3} + n_i T_{(G_T)_3}^e + T_{G_1}^h + n_i T_{G_2}^h$	$3 G_1 + n_u G_2 + \text{ID}_u + T_\sigma + n_u \text{attr} $
文献[30]	$n_u T_{G_1}^e + T_{G_2}^e$	$n_i T^{p3} + n_i T_{(G_T)_3}^e + n_i T^{p3} + T_{G_1}^h$	$n_u G_1 + G_2 + \text{ID}_u $
本文	$3T_{G_1}^e + n_u T_{G_2}^e + T_{G_2}^e$	$(3 + 2n_i)T^{p3} + T_{G_1}^e + T_{G_1}^h$	$3 G_1 + n_u G_2 + G_2 + \text{ID}_u + T_\sigma + n_u G_2 $

(属性)节点(即最多500 ms内可对3个微服务的访问策略($60/20 = 3$))进行验证,对验证端也不会造成较大的计算开销,非常适合部署于多微服务的现代网络架构中。

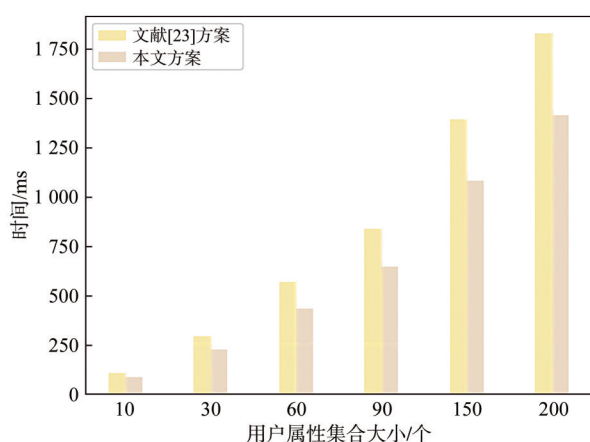


图5 Verify算法模拟运行所需时间
Figure 5 Simulation run time of Verify algorithm

7 结束语

针对分布式微服务架构中统一细粒度访问控制和数据安全通信问题,本文构建了基于Type-3配对的可立即撤销属性策略解耦式属性基签名及其互认证密钥协商方案。在运行签名算法时,无须关联访问策略,生成的签名只和用户部分属性有关,访问策略在验证算法中输入,来校验与签名相关联的用户属性是否真实合法并满足访问策略,所以签名与访问策略实现了解耦,可复用于认证多个访问策略。另外,本文方案实现了属性的立即撤销机制,使签名者不能用过期、无效属性继续用于策略认证,并严格地验证了本文属性基签名方案存在不可伪造性。为了实现数据的安全通信,提出了与本文属性基签名配套的满足eCK安全模型的互认证密钥协商方案。最后,通过理论和实验对比分析可知,在安全级别为128 bit (AES)需求下,本文方案中的认证算法比同类型方案要更加高效。未来会将本文方案应用于格密码上,来抵抗量子攻击算法。

参考文献:

[1] 冯志勇, 徐砚伟, 薛霄, 等. 微服务技术发展的现状与展望[J]. 计算机研究与发展, 2020, 57(5): 1103-1122.
FENG Z Y, XU Y W, XUE X, et al. Review on the develop-

ment of micro-service architecture[J]. Journal of Computer Research and Development, 2020, 57(5): 1103-1122.
[2] LEE H, KIM D, KWON Y. TLS 1.3 in practice: how TLS 1.3 contributes to the Internet[C]//Proceedings of the Web Conference 2021. New York: ACM, 2021: 70-79.
[3] FELT A P, BARNES R, KING A, et al. Measuring HTTPS adoption on the web[C]//Proceedings of the 26th USENIX Security Symposium (USENIX Security 17). Vancouver: 2017:1323-1338.
[4] OBERKO P S K, OBENG V H K S, XIONG H, et al. A survey on attribute-based signatures[J]. Journal of Systems Architecture, 2022(124): Article No.102396.
[5] GOYAL V, PANDEY O, SAHAI A, et al. Attribute-based encryption for fine-grained access control of encrypted data[C]//Proceedings of the 13th ACM Conference on Computer and Communications Security. New York: ACM press, 2006: 89-98.
[6] GUO S Q, ZENG Y P. Attribute-based signature scheme[C]//Proceedings of the International Conference on Information Security and Assurance. Piscataway: IEEE Press, 2008: 509-511.
[7] TAN S Y, HENG S H, GOI B M. On the security of an attribute-based signature scheme[C]//Proceedings of the International Conference on U-and E-Service, Science and Technology. Berlin: Springer, 2009: 161-168.
[8] LI J, AU M H, SUSILO W, et al. Attribute-based signature and its applications[C]//Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security. New York: ACM, 2010: 60-69.
[9] LI J, KIM K. Hidden attribute-based signatures without anonymity revocation[J]. Information Sciences, 2010, 180(9): 1681-1689.
[10] REN Y L, JIANG T J. Verifiable outsourced attribute-based signature scheme[J]. Multimedia Tools and Applications, 2018, 77(14): 18105-18115.
[11] UZUNKOL O. Comments on 'verifiable outsourced attribute-based signature scheme'[J]. Multimedia Tools and Applications, 2019, 78(9): 11735-11742.
[12] CUI H, DENG R H, LIU J K, et al. Server-aided attribute-based signature with revocation for resource-constrained industrial-internet-of-things devices[J]. IEEE Transactions on Industrial Informatics, 2018, 14(8): 3724-3732.
[13] XIONG H, BAO Y Y, NIE X Y, et al. Server-aided attribute-based signature supporting expressive access structures for industrial internet of things[J]. IEEE Transactions on Industrial Informatics, 2020, 16(2): 1013-1023.
[14] RAO Y S, DUTTA R. Efficient attribute-based signature and signcryption realizing expressive access structures[J]. International Journal of Information Security, 2016, 15(1): 81-109.
[15] MAJI H K, PRABHAKARAN M, ROSULEK M. Attribute-based signatures[C]//Proceedings of the Cryptographers Track at the RSA Conference. Berlin: Springer, 2011: 376-392.
[16] CUI H, WANG G L, DENG R H, et al. Escrow free attribute-based signature with self-revealability[J]. Information Sciences: An International Journal, 2016, 367(C): 660-672.
[17] LIU J H, HUANG X Y, LIU J K. Secure sharing of personal health records in cloud computing: ciphertext-policy attribute-

- based signcryption[J]. Future Generation Computer Systems, 2015(52): 67-76.
- [18] ZHANG Y, FENG D G, ZHANG Z F, et al. On the security of an efficient attribute-based signature[C]//Proceedings of the International Conference on Network and System Security. Berlin: Springer, 2013: 381-392.
- [19] DING S L, ZHAO Y M, LIU Y Y. Efficient traceable attribute-based signature[C]//Proceedings of the 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications. Piscataway: IEEE Press, 2014: 582-589.
- [20] RAO Y S. A secure and efficient ciphertext-policy attribute-based signcryption for personal health records sharing in cloud computing[J]. Future Generation Computer Systems, 2017(67): 133-151.
- [21] YU J G, LIU S H, WANG S L, et al. LH-ABSC: a lightweight hybrid attribute-based signcryption scheme for cloud-fog-assisted IoT[J]. IEEE Internet of Things Journal, 2020, 7(9): 7949-7966.
- [22] ZHANG Z S, ZHOU S J. A decentralized strongly secure attribute-based encryption and authentication scheme for distributed internet of mobile things[J]. Computer Networks, 2021 (201): Article No.108553.
- [23] ZHANG Z S, HUANG W, CAI S Y, et al. An efficient reusable attribute-based signature scheme for mobile services with multi access policies in fog computing[J]. Computer Communications, 2022, 196(C): 9-22.
- [24] CHEN L, CHENG Z, SMART N P. Identity-based key agreement protocols from pairings[J]. International Journal of Information Security, 2007, 6(4): 213-241.
- [25] LAMACCHIA B, LAUTER K, MITYAGIN A. Stronger security of authenticated key exchange[C]//Proceedings of the International Conference on Provable Security. Berlin, Heidelberg: Springer, 2007: 1-16.
- [26] BEIMEL A. Secure schemes for secret sharing and key distribution[M]. Technion-Israel Institute of Technology, Faculty of Computer Science, 1996.
- [27] LEWKO A, WATERS B. Decentralizing attribute-based encryption[C]//Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2011: 568-588.
- [28] 李高磊, 李建华, 周志洪, 等. 面向新型关键基础设施的密码应用安全性评估技术综述[J]. 网络与信息安全学报, 2023, 9(6): 1-19.
- LI G L, LI J H, ZHOU Z H, et al. Overview of cryptographic application security assessment technology for new key infrastructure[J]. Chinese Journal of Network and Information Security, 2023, 9(6): 1-19.
- [29] POINTCHEVAL D, STERN J. Security arguments for digital signatures and blind signatures[J]. Journal of Cryptology, 2000, 13(3): 361-396.
- [30] ZHANG Z S, HUANG W, ZHOU S J, et al. A revocable multi-authority fine-grained access control architecture against ciphertext rollback attack for mobile edge computing[J]. Journal of Systems Architecture, 2022(129): Article No.102589.

[作者简介]



张智烁 (1995-), 男, 天津静海人, 电子科技大学博士生, 主要研究方向为新一代公钥密码学。



杨会喜 (1970-), 女, 河北沧州人, 沧州市职业技术学院副教授, 主要研究方向为数字工业物联网系统。



黄文 (1991-), 男, 四川成都人, 四川大学博士后, 主要研究方向为差分隐私、隐私计算和离散对数困难问题高效求解算法。



廖永建 (1976-), 男, 四川成都人, 电子科技大学副教授、硕士生导师, 主要研究方向为传统密码学与格密码学。



周世杰 (1970-), 男, 四川成都人, 电子科技大学教授、博士生导师, 主要研究方向为人工智能安全、信息安全。