

可支持公平验证的数据群组灵活协同共享方案

王树兰^{1,2}, 甘竞宏^{1,2}, 赵陈斌³, 邹家须¹, 赵国英^{1,2}, 张应辉⁴

(1. 深圳技术大学大数据与互联网学院, 广东 深圳 518118; 2. 深圳大学应用技术学院, 广东 深圳 518060;

3. 空天信息安全与可信计算教育部重点实验室, 武汉大学国家网络安全学院, 湖北 武汉 430072;

4. 西安邮电大学网络空间安全学院, 陕西 西安 710121)

摘要: 随着云服务的快速发展, 数据的共享和传播受到了广泛关注。基于属性的条件代理重加密 (attribute-based conditional proxy re-encryption, AB-CPRE) 能安全且高效地实现数据群组共享和传播而被广泛应用。然而, 现有的AB-CPRE方案无法实现同一组内多个数据传播者协同将数据传播给新的用户组, 仅支持单一数据传播者, 灵活性差。此外, 现有方案中用户生成验证值所需的计算开销随着密文中属性数增加而增加, 导致云服务器和用户之间的公平验证效率低。针对上述不足, 提出一种可支持公平验证的数据群组灵活协同共享方案。该方案构造了一种协同传播树结构, 以实现同一组内多个数据传播者协同将数据传播给新用户组。此外, 该方案设计了计算开销恒定的验证值生成机制, 使用户仅需常数级计算开销即可生成验证值, 从而高效实现云服务器和用户间的公平可验证。最后, 形式化安全性证明和性能评估表明, 所提方案是可证明安全和公平可验证的, 并在验证值生成、重加密密钥生成和解密阶段比现有方案高效。

关键词: 协同数据传播; 基于属性的条件代理重加密; 协同传播树; 公平可验证

中图分类号: TP309

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2025005

Flexible and collaborative data group sharing scheme with fairness and verifiability

WANG Shulan^{1,2}, GAN Jinghong^{1,2}, ZHAO Chenbin³, ZOU Jiaxu¹,
ZHAO Guoying^{1,2}, ZHANG Yinghui⁴

1. School of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China

2. School of Application and Technology, Shenzhen University, Shenzhen 518060, China

3. Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education,
School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China

4. School of Cyberspace Security, Xian University of Posts & Telecommunications, Xi'an 710121, China

收稿日期: 2024-04-29; **修回日期:** 2024-09-11

通信作者: 赵陈斌, chenbinzhao96@163.com

基金项目: 国家自然科学基金 (62072369, 61702341); 深圳技术大学深圳市高等院校稳定支持项目 (SZWD2021012); 陕西省技术创新引导计划 (基金) (2023-YD-CGZH-31)

Foundation Items: The National Natural Science Foundation of China (62072369, 61702341), The Shenzhen Technology University of Shenzhen Higher Education Stability Support Program (SZWD2021012), The Technology Innovation Leading Program of Shaanxi (2023-YD-CGZH-31)

引用格式: 王树兰, 甘竞宏, 赵陈斌, 等. 可支持公平验证的数据群组灵活协同共享方案[J]. 网络与信息安全学报, 2025, 11(1): 40-53.

Citation Format: WANG S L, GAN J H, ZHAO C B, et al. Flexible and collaborative data group sharing scheme with fairness and verifiability[J]. Chinese Journal of Network and Information Security, 2025, 11(1): 40-53.

Abstract: With the rapid development of cloud services, data sharing and dissemination have received widespread attention. Attribute-based conditional proxy re-encryption (AB-CPRE) has been widely used as a secure and efficient method to achieve group data sharing and dissemination. However, existing AB-CPRE schemes were unable to support multiple data disseminators within the same group to collaboratively disseminate data to new user groups, as only a single data disseminator was permitted, which limited flexibility. Additionally, in these schemes, the computation cost for users to generate proofs increased with the number of attributes in the ciphertext, resulting in low efficiency in ensuring fairness and verifiability between the cloud server and the user. To address these shortcomings, a flexible and collaborative data group sharing scheme with fairness and verifiability was proposed. A collaborative dissemination tree was designed to enable multiple data disseminators within the same group to collaboratively disseminate data to new user groups. Furthermore, the proposed scheme incorporated a constant-cost proof generation mechanism, allowing users to generate proofs with constant computation cost, thereby ensuring efficient fairness and verifiability between the cloud server and the user. Formal security proofs and performance evaluations demonstrate that the proposed scheme is provably secure, fair, and verifiable. It is shown to offer greater efficiency in proof generation, re-encryption key generation, and decryption phases compared to existing schemes.

Keywords: collaborative data dissemination, attribute-based conditional proxy re-encryption, collaborative dissemination tree, fairness and verifiability

0 引言

随着云服务的快速发展, 用户逐渐选择在云上共享数据资源。在现实应用场景中, 除数据共享^[1]之外, 数据协同传播需求日益增加。为了保证数据的隐私性^[2]和可用性, 进行协同传播的数据通常需要被重新加密成新用户可以解密的密文。由于生成重加密密文需要大量计算成本, 执行重加密的实体可能会返回不正确的重加密密文给用户, 以节省自身的计算资源。相反, 用户即使收到正确的重加密密文, 也可能恶意指责该密文是错误的以拒绝支付上述实体执行重加密操作的成本。因此, 实现可支持公平验证的数据群组灵活协同共享是一个迫切需要解决的问题。

条件代理重加密^[3-5] (conditional proxy re-encryption, CPRE) 是常用的一种技术, 这是因为它能够安全高效地实现数据群组共享和传播。通过该技术, 数据传播者只需生成与特定条件相关联的重加密密钥并上传至云服务器, 即可完成数据传播任务, 而无须像其他加密技术^[6]一样下载密文, 解密密文, 最终重新加密数据。然而, 传统的CPRE方案仅基于关键词条件, 难以适应云环境中数据所有者可能具有的多种传播需求。例如, 家庭照片只允许家人传播, 而工作文件则只允许同事传播。基于属性的CPRE

(attribute-based conditional proxy re-encryption, AB-CPRE) 能够实现细粒度的重加密操作, 因此, 是一种非常适用于云环境的加密机制。数据传播者只需生成满足细粒度条件的重加密密钥, 并将其上传至云服务器即可。随后, 云服务器利用接收到的重加密密钥对原始密文进行重加密, 并将其传播给可以解密的新用户组。此外, 为验证重加密密文的正确性, 确保云服务器免受用户的恶意指责, 即验证云服务器和用户间的公平交易, 有学者提出公平可验证的AB-CPRE方案^[7-8]。

尽管AB-CPRE方案在数据群组共享和细粒度传播方面比其他加密方案^[9]更高效灵活, 但它仍然存在以下不足。首先, 该方案无法实现同一组内多个数据传播者协同传播数据至新的用户组, 仅支持单一数据传播者的传播操作, 灵活性差, 因此难以适用于医疗或者机关单位等高度协同需求的场景。其次, 现有的公平可验证AB-CPRE方案^[7-8]中用户生成验证值所需的计算开销随着原始密文的属性数量增加而增加, 无法高效实现云服务器和用户之间的高效验证。因此, 设计可支持公平验证的数据群组灵活协同共享方案是一项具有挑战性的任务。

针对上述不足, 本文提出一种可支持公平验证的数据群组灵活协同共享方案。该方案设计了一种协同传播树结构作为基础模块, 以实现多数

据传播者协同传播的权限控制。即数据所有者可以为叶子节点赋予随机陷门特征,只有当多个数据传播者同时满足预设的随机陷门,数据才能够被传播到新用户组。此外,为高效实现云服务器和用户之间的公平验证,本文设计了一个常数级开销的验证值生成机制,大大降低了用户生成验证值所需的计算开销。

本文的主要贡献如下。

1) 提出一种可支持公平验证的数据群组灵活协同共享方案,该方案设计了协同传播树结构以实现同一组内多个数据传播者的协同传播。

2) 设计一种计算开销恒定的验证值生成机制,大大降低了用户生成验证值所需的计算开销,进而高效实现云服务器和用户间的公平可验证。

3) 给出正式的安全性证明和全面的性能评估。安全性证明表明本文方案是可证明安全以及公平可验证的。此外,实验评估表明本文验证值生成的计算开销是常数级的,且重加密密钥生成阶段和解密等阶段的计算开销更高效。

1 相关工作

1.1 属性基加密

Ge等^[10]设计了一种基于密文策略的属性基加密(ciphertext policy attribute-based encryption, CP-ABE)的协同访问控制方案,实现密文大小不会随着属性数量的增加而增加。He等^[11]提出一种基于区块链的协同解密CP-ABE方案,利用线性秘密共享方案来实现协同解密。Xue等^[12]提出一种改进的属性基加密方案,实现同一组用户能够互相共享特定的属性。Chen等^[13]提出基于文件分层和协同解密的CP-ABE方案,将文件层次的思想融入协同解密中。Madani等^[14]提出一种基于属性的访问控制模型的扩展,实现了用户可在多云的环境下进行协同解密。Zhang等^[15]提出一种可验证机制来验证多参与方是否满足密文中的访问树结构,有效防止不必要的协作而浪费计算资源,此外,该方案考虑了多协同者合作访问共享数据。Li等^[16]将权重的ABE与协同访问机制结合,实现高效的协同共享方案。然而,上述方案并没有考虑加密信息可能需要由多个参与者协同地传播到一个新的用户组中。

1.2 基于属性的条件代理重加密

Huang等^[17]提出一种改进的AB-CPRE方案,支持具有属性和时间条件的数据群组共享和传播。Huang等^[18]基于AB-CPRE机制进一步实现了对涉及多个数据所有者共同数据的隐私保护。Deng等^[19]提出AB-CPRE机制与身份广播加密机制的组合方案,完善了“多对多”访问模型。Ge等^[8]设计了一种公平可验证的AB-CPRE方案,验证重加密密文正确性的同时,可保护云服务器免受恶意指控。Jiang等^[20]增强重加密密文在不可信云环境下的可验证性和可信性。Xie等^[21]提出了基于ABE和聚合数字签名的可验证外包方案,其使用聚合数字签名来检测重加密密文的正确性。Sun等^[7]提出公平可验证的广播授权框架用于安全的灵活数据共享。但上述方案在生成验证阶段所需的验证值时,用户需要消耗大量的计算开销。

2 预备知识

2.1 决策双线性Diffie-Hellman假设

$a, b, c \in \mathbb{Z}_p^*$ 和 $R \in G_T$ 被假设是随机数, g 是 G 的生成元, $e: G \times G \rightarrow G_T$ 是双线性对运算。决策双线性Diffie-Hellman假设意味着不存在概率多项式的时间算法以可忽略的概率来区分以下两个元组 $(A = g^a, B = g^b, C = g^c, D = e(g, g)^{abc})$ 和 $(A = g^a, B = g^b, C = g^c, Z = R)$ 。

2.2 离散对数假设

给出一个元组 (e, G, G_T, g, p, g^e) , 其中, $e: G \times G \rightarrow G_T$ 是双线性对运算, g 是 G 的生成元, $\varepsilon \in \mathbb{Z}_p^*$ 。离散对数假设意味着概率多项式时间敌手 A 找到整数 ε 的优势是可忽略的。形式上, 一个概率多项式时间敌手 A 的优势 $\text{Adv}_{\text{DL}} = \Pr[A(e, G, G_T, g, p, g^e) = \varepsilon]$ 是可忽略的。

3 问题表述

3.1 系统模型

系统模型如图1所示,由6个实体组成:可信机构、仲裁机构、云服务器、数据所有者、数据传播者和用户。可信机构用于生成系统参数,并为系统中的用户分发身份密钥和属性密钥;仲裁机构负责响应用户和云服务器的公平可验证请

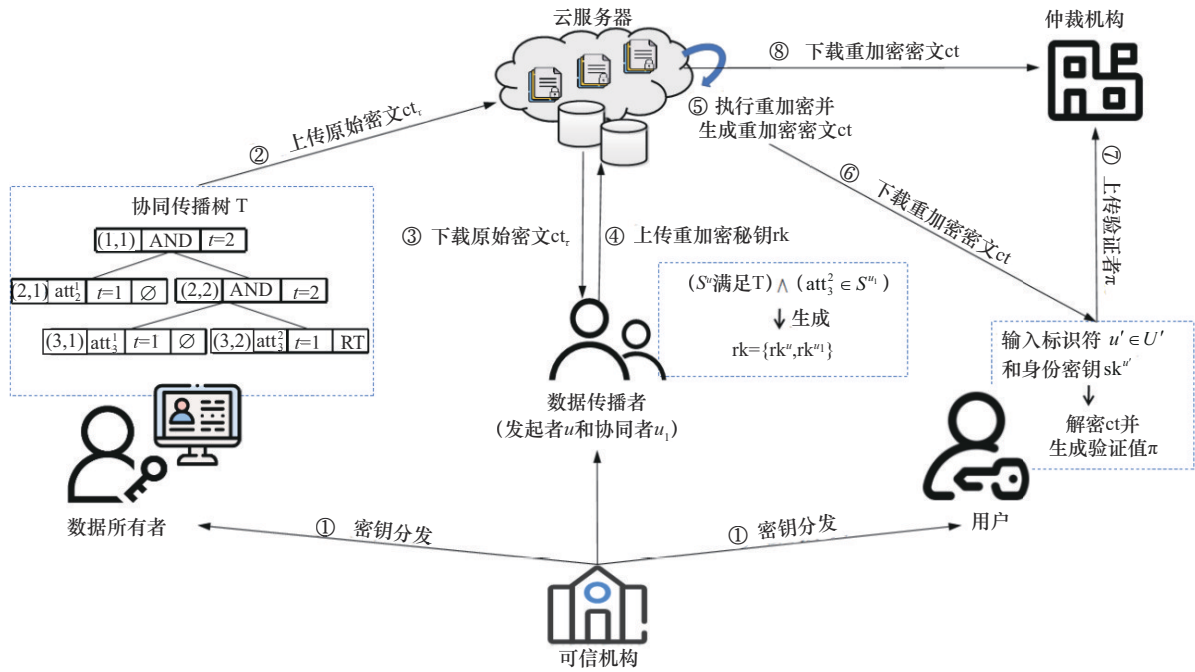


图1 系统模型

Figure 1 System model

求，防止用户对云服务器的恶意指控；云服务器提供一个存储和共享加密数据的平台，此外可以使用数据传播者上传的重加密密钥生成重加密密文；数据所有者能够安全地将数据外包到云服务器中，同时为数据传播定义一个协同传播树对数据进行加密，从而限制数据由特定的授权用户传播；数据传播者分为发起者和协同者，两者都具备生成重加密密钥的能力。此外，数据传播者是数据所有者定义的接收者之一；用户对云服务器中数据发出访问请求，并解密密文。此外，用户还可向仲裁机构提出验证请求来验证重加密密文的正确性。

3.2 形式化算法

可支持公平验证的数据群组灵活协同共享方案包括8个概率多项式时间算法，详细描述如下。

1) $\text{Setup}(\lambda, n) \rightarrow (\text{pk}, \text{msk})$ 。可信机构输入安全参数 $\lambda \in \mathbb{N}$ 和系统中允许存在的最大用户数量 $n \in \mathbb{N}$ ，输出公钥元组 pk 和主密钥元组 msk 。

2) $\text{KeyGen}(u, \text{pk}, \text{msk}, S^u) \rightarrow (\text{sk}^u, \text{ak}^u)$ 。可信机构输入用户标识符 u ，公钥元组 pk ，主密钥元组 msk ，以及用户的属性集 S^u ，输出用户的身份密钥 sk^u 和属性密钥 ak^u 。

3) $\text{Encrypt}(m, \text{pk}, U, T) \rightarrow \text{ct}_t$ 。数据所有者输入明文数据 m ，公钥元组 pk ，接收者标识符集合 U ，以及协同传播树 T ，输出初始密文 ct_t 并上传至云服务器。

4) $\text{RKeyGen}(u, \text{pk}, \text{sk}^u, \text{ak}^u, \text{ct}_t, U') \rightarrow \text{rk}^u$ 。数据传播者输入用户标识符 u ，公钥元组 pk ，用户的身份密钥 sk^u 以及属性密钥 ak^u ，新接收者标识符集合 U' ，以及初始密文 ct_t ，输出重加密密钥 rk^u 并发送给云服务器。

5) $\text{REncrypt}(u, \text{pk}, \text{rk}^u, \text{ct}_t) \rightarrow \text{ct}$ 。云服务器输入用户标识符 u ，公钥元组 pk ，重加密密钥 rk^u 以及原始密文 ct_t ，输出重加密密文 ct 并将其传播给指定的用户组。

6) $\text{Decrypt}(u, \text{pk}, \text{sk}^u, \text{ct}) \rightarrow m$ 。用户输入标识符 u ，公钥元组 pk ，身份密钥 sk^u 和原始密文 ct_t ，解密后输出明文数据 m 。

7) $\text{ProofGen}(u', \text{pk}, \text{sk}^u, \text{ct}) \rightarrow \pi$ 。用户输入标识符 u' ，公钥元组 pk ，身份密钥 sk^u 和重加密密文 ct ，输出验证过程所需的验证值 π 并上传至仲裁机构。

8) $\text{Verify}(\text{ct}_t, \text{rk}^u, \text{ct}, \pi) \rightarrow \text{Ture} / \text{False}$ 。仲裁机构输入原始密文 ct_t ，重加密密钥 rk^u ，重加密

密文 ct 以及用户提交的验证值 π ，若验证通过，则输出 **Ture**，否则输出 **False**。

3.3 安全性定义

本文方案的安全性定义由概率多项式时间敌手 A 和挑战者 C 共同完成如下游戏，具体过程如下。

初始化：敌手 A 选择挑战标识集 U^* 以及协同传播树 T^* ，并将它们发送给挑战者 C 。然后，挑战者 C 运行 **Setup** 算法输出一个公钥元组 pk 并将其发送给敌手 A 。

阶段 1：敌手 A 请求标识符 $u_i \notin U^*$ 的身份密钥 sk^{u_i} ，与属性集 T^* 的属性密钥 ak^{u_i} 。然后，挑战者 C 执行 **KeyGen** 算法返回对应的密钥。此外，敌手 A 还可以请求对于标识符 $u_i \notin U^*$ 和属性集 $S^{u_i} \notin T^*$ 的重加密密钥 rk^{u_i} 。挑战者 C 通过执行 **RKenGen** 算法返回重加密密钥。

挑战：敌手 A 挑战两个等长的消息 m_0 和 m_1 。挑战者 C 选择随机比特 $b \in \{0,1\}$ 并加密明文 m_b ，返回原始密文 ct_b^* 给敌手 A 。

阶段 2：该阶段与阶段 1 相同。

猜测：敌手 A 输出猜测 $b' \in \{0,1\}$ ，如果 $b' = b$ 意味着敌手 A 赢得游戏。

3.4 设计目标

假设可信机构和仲裁机构都是可信实体，云服务器则是半可信实体。本文方案将满足如下 3 个设计目标。

1) 数据机密性。用户如果不在数据所有者或数据传播者授权集合中，则无法获取任何数据。此外，云服务器无法获取任何隐私信息。

2) 公平可验证。用户在云服务器传播重加密密文时可验证重加密密文的正确性。此外，当收到正确的重加密密文时，用户不能恶意地指责云服务器。

3) 抗合谋。未经授权的数据传播者无法合谋产生有效的重加密密钥。此外，未经授权的数据传播者不能与云服务器合谋获取任何隐私信息。

4 协同传播树

为实现协同传播，本文设计了协同传播树（如图 2 所示）作为后续方案的基础模块。其中，

非叶子节点包含了三元组 $\{(x,y), TG, t\}$ ，叶子节点则包含了四元组 $\{(x,y), att_x^y, t, RT\}$ 。详细说明如下。

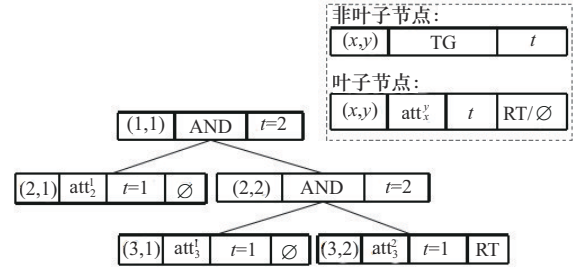


图 2 协同传播树

Figure 2 Collaborative dissemination tree

1) 非叶子节点：元素 (x,y) 表示每个节点位置，其中 x 表示行， y 表示列。TG 表示限门，其包含的类型为 AND, OR, $n'-of-m$ ($n' < m$)。当限门为 AND 时，门限值 t 为该节点所拥有的孩子节点总数量；当限门为 OR 时，门限值 t 为 1。当限门为 $n'-of-m$ ($n' < m$) 时，门限值 t 为 n' 。

2) 叶子节点：元素 (x,y) 表示每个节点位置。不同之处在于叶子节点包含 4 个元素。其中， att_x^y 表示该节点属性。RT 表示随机陷门，其被定义两种状态：暴露或未暴露。如果云服务器能够获取到该节点的秘密值，则该随机陷门状态为暴露；否则，该随机陷门状态为不暴露。此外，叶子节点的限门值 t 为 1。

5 方案构造

假设 $e:G \times G \rightarrow G_T$ 是一个双线性映射，其中 G 和 G_T 是素数阶为 p 的乘法群， $H_1: \{0,1\}^* \rightarrow Z_p^*$ ， $H_2: \{0,1\}^* \rightarrow G$ 和 $H_3: G_T \rightarrow G$ 是 3 个抗碰撞哈希函数。方案详细说明如下。

1) **Setup**。该算法由可信机构执行，生成公钥元组 pk 与主密钥元组 msk 。详细如算法 1 所示。

2) **KeyGen**。该算法生成用户的身份密钥 sk^u ，以及属性密钥 ak^u ，其中用户分为两个类型（发起者和协同者）：属性集 S^u 被分为两个不同的子集： S_1^u 包含不允许协同的属性； S_2^u 包含允许协同的属性。详细如算法 2 所示。

3) **Encrypt**。该算法首先基于消息锁加密算

法^[8]的密钥生成算法生成密钥 ck 。随后，该密钥以及明文数据根据协同传播树 T 和接收者标识符集合 U 进行加密。详细如算法3所示。

4) RKenGen。数据传播者向新用户组传播数据时，使用身份密钥 sk^u 和属性密钥 ak^u 生成重加密密钥并上传云服务器。详细如算法4所示。

5) REncrypt。该算法输入用户标识符 u ，公钥元组 $pk \leftarrow (h, h^\gamma, \dots, h^{\gamma^n}, v, v^\gamma, \dots, v^{\gamma^n}, h^\beta, v^\beta, \phi, \varphi, g^{\beta_1 \beta_2}, e(g, h)^\gamma, e(g, h), g^\gamma, h^{\beta_1 - \beta_2})$ ，重加密密钥 $rk^u \leftarrow (r_1^u, [ru2, ru3, ru4, ru5, ru6, ru7, ru8])$ 以及原始密文 $ct_\tau \leftarrow (c_x^\gamma, c_x^{\gamma'}, c)$ 。

算法1 $Setup(\lambda, n) \rightarrow (pk, msk)$

- 1) 选择随机数 $g, h, v, \phi, \varphi \leftarrow G$
- 2) 选择随机数 $\beta, \beta_1, \beta_2, \gamma, \theta_1, \dots, \theta_n \leftarrow Z_p^*$
- 3) 计算 $h^\beta, v^\beta, g^\gamma, h^{\beta_1 - \beta_2}, g^{\beta_1 \beta_2}, e(g, h), e(g, h)^\gamma$
- 4) **for** $i \in [1, n]$ **do**
- 5) 计算 h^{γ^i}
- 6) 计算 v^{γ^i}
- 7) **end**
- 8) 设置 $pk \leftarrow (h, h^\gamma, \dots, h^{\gamma^n}, v, v^\gamma, v^{\gamma^2}, \dots, v^{\gamma^n}, h^\beta, v^\beta, e(g, h)^\gamma, e(g, h), g^\gamma, h^{\beta_1 - \beta_2}, g^{\beta_1 \beta_2}, \phi, \varphi)$
- 9) 设置 $msk \leftarrow (g, \gamma, \beta, \beta_1, \beta_2, \theta_1, \dots, \theta_n)$
- 10) 返回 pk, msk

算法2 $KeyGen(u, pk, msk, S^u) \rightarrow (sk^u, ak^u)$

- 1) 解析 $pk \leftarrow (h, h^\gamma, \dots, h^{\gamma^n}, v, v^\gamma, v^{\gamma^2}, \dots, v^{\gamma^n}, h^\beta, v^\beta, e(g, h)^\gamma, e(g, h), g^\gamma, h^{\beta_1 - \beta_2}, g^{\beta_1 \beta_2}, \phi, \varphi)$
- 2) 解析 $msk \leftarrow (g, \gamma, \beta, \beta_1, \beta_2, \theta_1, \dots, \theta_n)$
- 3) 选择随机数 $r^u \leftarrow Z_p^*$
- 4) 计算 $sk^u = g^{1/(\gamma + H_1(u))}$, $d_0^u = g^{(\gamma + r^u(\beta_1 - \beta_2))/\beta}$
- 5) **for** 每个属性 $i \in S_1^u$ **do**
- 6) 选择随机数 $r_i^u \leftarrow Z_p^*$
- 7) 计算 $d_i^u = g^{r^u} H_2(i)^{r_i^u}$, $d_i^u = h^{r_i^u(\beta_1 - \beta_2)}$
- 8) **if** 这个用户是数据传播的发起者 **then**
- 9) **for** 每个属性 $j \in S_2^u$ **do**
- 10) 选择随机数 $r_j^u \leftarrow Z_p^*$
- 11) 计算 $d_j^{u''} = g^{r^u} H_2(j)^{r_j^u}$
- 12) 计算 $d_j^{u'''} = h^{r_j^u}$
- 13) 计算 $e_{tr}^u = h^{(\theta_m + r^u)/\beta_1}$
- 14) **else**

15) **for** 每个属性 $j \in S_2^u$ **do**

16) 选择随机数 $r_j^u \leftarrow Z_p^*$

17) 计算 $d_j^{u''} = g^{r^u \beta_2} H_2(j)^{r_j^u}$

18) 计算 $d_j^{u'''} = h^{r_j^u}$

19) 计算 $e_{tr}^u = h^{(r^u - \theta_m)/\beta_1}$

20) 设置 $ak^u \leftarrow (d_0^u, \{d_i^u, d_i^{u''}\}_{i \in S_1^u}, \{d_j^u, d_j^{u''}\}_{j \in S_2^u}, e_{tr}^u)$

21) 返回 sk^u, ak^u

算法3 $Encrypt(m, pk, U, T) \rightarrow ct_\tau$

- 1) 解析 $pk \leftarrow (h, h^\gamma, \dots, h^{\gamma^n}, v, v^\gamma, v^{\gamma^2}, \dots, v^{\gamma^n}, h^\beta, v^\beta, e(g, h)^\gamma, e(g, h), g^\gamma, h^{\beta_1 - \beta_2}, g^{\beta_1 \beta_2}, \phi, \varphi)$
- 2) 生成一个密钥 $ck \leftarrow MLE.KeyGen(m)$
- 3) 计算根节点的 $d_1^1 = t - 1$
- 4) 选择随机数 $a_1, a_2, \dots, a_{d_1^1} \leftarrow Z_p^*$
- 5) 选择随机数 $\ell \leftarrow Z_p^*$ ，并设置 $q_1^1(0) = \ell$
- 6) 设置 $q_1^1(x) \leftarrow \ell + a_1 x + \dots + a_{d_1^1} x^{d_1^1}$
- 7) **for** 其他节点在位置 $(x, y) \in T$ **do**
- 8) 计算 $d_x^y = t - 1$ ，选择随机数 $b_1, \dots, b_{d_x^y} \leftarrow Z_p^*$
- 9) 计算 $q_x^y(0) = q_{parent(x, y)}(index(x, y))$ ，其中， $q_{parent(x, y)}$ 表示父节点坐标， $index(x, y)$ 指标签
- 10) 设置 $q_x^y(x_1) \leftarrow q_x^y(0) + b_1 x_1 + \dots + b_{d_x^y} x_1^{d_x^y}$
- 11) 选择随机数 $k \leftarrow Z_p^*$
- 12) 计算 $\delta = H_1(MLE.Encrypt(m, ck))$
- 13) 计算 $c_1 = (m || ck) \oplus e(g, h)^{k\delta}$
- 14) 计算 $c_2 = h^{k\delta \cdot \prod_{u_i \in U} (\gamma + H_1(u_i))}$, $c_{3,1} = v^{\beta \ell \delta}$
- 15) 计算 $c_{3,2} = v^{k\delta \cdot \prod_{u_i \in U} [(\gamma + H_1(u_i))/H_1(u_i)]}$, $c_{3,3} = v^{\beta \ell}$
- 16) 计算 $c_{3,4} = v^{k \cdot \prod_{u_i \in U} [(\gamma + H_1(u_i))/H_1(u_i)]}$
- 17) 计算 $c_4 = g^{-\gamma k \delta}$, $c_5 = h^{\beta \ell \delta}$
- 18) 计算 $c_6 = e(g, h)^{\gamma \ell \delta}$
- 19) **for** 每个无RT的叶子节点在 $(x, y) \in Y$ **do**
- 20) 计算 $c_{(x, y)} = h^{q_x^y(0)(\beta_1 - \beta_2)\delta}$
- 21) 计算 $c'_{(x, y)} = H_2(at_{(x, y)})^{q_x^y(0)\delta}$ ，其中 $at_{(x, y)}$ 表示节点在 (x, y) 位置的属性
- 22) **for** 每个有RT的叶子节点在 $(x, y) \in X$ **do**
- 23) 计算 $c'_{(x, y)} = H_2(att_{(x, y)})^{q_x^y(0)\delta}$

- 24) 计算 $c''_{(x,y)} = h^{q_x^y(0)\delta}$, $\hat{c}_{(x,y)} = g^{q_x^y(0)\beta_1\beta_2\delta}$
- 25) 计算 $c_8 = h^{\beta\ell}$
- 26) 计算 $\bar{c} = \phi^{H_1(m)}\phi^{H_1(\text{ck})}$
- 27) 设置 $c_x^y \leftarrow \{c_{(x,y)}, c'_{(x,y)}\}_{(x,y) \in Y}$
- 28) 设置 $c_x^{y'} \leftarrow \{c'_{(x,y)}, c''_{(x,y)}, \hat{c}_{(x,y)}\}_{(x,y) \in X}$
- 29) 设置 $c \leftarrow (c_0, c_1, c_2, c_{3,1}, c_{3,2}, c_{3,3}, c_{3,4}, c_4, c_5, c_6, c_8, \bar{c})$
- 30) 返回 $\text{ct}_\tau \leftarrow (c_x^y, c_x^{y'}, c)$
- 算法 4** $\text{RKenGen}(u, \text{pk}, \text{sk}^u, \text{ak}^u, \text{ct}_\tau, U') \rightarrow \text{rk}^u$
- 1) 解析 $\text{pk} \leftarrow (h, h^\gamma, \dots, h^{\gamma^n}, v, v^\gamma, v^{\gamma^2}, \dots, v^{\gamma^n}, h^\beta, v^\beta, e(g, h)^\gamma, e(g, h), g^\gamma, h^{\beta_1 - \beta_2}, g^{\beta_1\beta_2}, \phi, \varphi)$
- 2) 解析 $\text{ak}^u \leftarrow (d_0^u, \{d_i^u, d_i^{u'}\}_{i \in S_1^u}, \{d_j^u, d_j^{u''}\}_{j \in S_2^u}, e_{tr}^u)$
- 3) 解析 $\text{ct}_\tau \leftarrow (c_x^y, c_x^{y'}, c)$
- 4) 选择随机数 $k', s \leftarrow Z_p^*$
- 5) 计算 $r_1^u = \text{sk}^u \cdot v^{s/H_1(u)} = g^{1/(\gamma + H_1(u))} \cdot v^{s/H_1(u)}$
- 6) 计算 $r_2^u = h^{k' \cdot \prod_{u_i \in U} (\gamma + H_1(u_i))}$, $r_3^u = H_3(e(g, h)^{k'}) \cdot h^s$
- 7) 计算 $r_4^u = g^{-\gamma k'}$, $r_5^u = d_0^u \cdot v^s = g^{[\gamma + r^u(\beta_1 - \beta_2)]/\beta} \cdot v^s$
- 8) **for** 每个属性 $i \in S_1^u$ **do**
- 9) 设置 $r_i^u \leftarrow d_i^u$, $r_i^{u'} \leftarrow d_i^{u'}$
- 10) **for** 每个属性 $j \in S_2^u$ **do**
- 11) 设置 $r_j^{u''} \leftarrow d_j^{u''}$, $r_j^{u'''} \leftarrow d_j^{u'''}$
- 12) 设置 $r^{u''''} \leftarrow e_{tr}^u, r_6^u \leftarrow (\{r_i^u, r_i^{u'}\}_{i \in S_1^u}, \{r_j^{u''}, r_j^{u'''}\}_{j \in S_2^u}, r^{u''''})$
- 13) 计算 $r_7^u = e(h^s, v^{\beta\ell})$
- 14) 计算 $r_8^u = e(v^{-s}, h)^{k' \cdot \prod_{u_i \in U} (\gamma + H_1(u_i))/H_1(u_i)}$
- 15) 返回 $\text{rk}^u \leftarrow (r_1^u, r_2^u, r_3^u, r_4^u, r_5^u, r_6^u, r_7^u, r_8^u)$

云服务器利用标识符假设为 u 的发起者上传的 rk^u 运行递归算法 $\text{DecryptNode}(\text{ct}_\tau, \text{rk}^u, (x, y))$, 该算法输入原始密文 ct_τ , 与一组属性 S^u 相关联的重加密密钥 rk^u 和协同传播树 T 中一个节点的位置信息 (x, y) , 详细过程如下。

如果 (x, y) 是一个叶子节点, 且该节点的属性 $z = \text{att}_{(x,y)}$ 定义如下: 当该节点不满足随机陷门 RT 特征, 如果 $z \notin S_1^u$, 则设置算法 $\text{DecryptNode}(\text{ct}_\tau, \text{rk}^u, (x, y)) = \perp$ 。否则, 需要进行计算

$$\text{DecryptNode}(\text{ct}_\tau, \text{rk}^u, (x, y)) = \frac{e(r_z^u, c_{(x,y)})}{e(r_z^{u'}, c'_{(x,y)})} =$$

$$e(g, h)^{r^u q_x^y(0)(\beta_1 - \beta_2)\delta}$$

当该节点满足随机陷门特征时, 算法 DecryptNode 则不仅需要涉及发起者的 rk^u 而且需涉及协同者 u_1 上传的 rk^{u_1} , 如果 $z \notin S_2^u$ 或 $z \notin S_2^{u_1}$, 则算法 $\text{DecryptNode} = \perp$ 。否则, 计算

$$\text{DecryptNode}(\text{ct}_\tau, \text{rk}^u, (x, y)) = \frac{e(r_z^{u''}, c''_{(x,y)})}{e(r_z^{u'''}, c'_{(x,y)})} =$$

$e(g, h)^{r^{u''} q_x^y(0)\beta_1\delta}$, 以及

$$\text{DecryptNode}(\text{ct}_\tau, \text{rk}^{u_1}, (x, y)) = \frac{e(r_z^{u_1''}, c''_{(x,y)})}{e(r_z^{u_1'''}, c'_{(x,y)})} =$$

$$e(g, h)^{r^{u_1} q_x^y(0)\beta_2\delta}。$$

如果 (x, y) 是非叶子节点时, 则先选择该节点大小为 t 的孩子节点 n' 集合, 并将它们存储在集合 $S_{(x,y)}$ 中, 如果不存在该集合, 则算法 DecryptNode 返回 $\perp$ 。其中, 孩子节点 n' 设定如下。如果孩子节点不满足随机陷门特征, 则 n' 只需与发起者 u 关联; 如果 n' 满足随机陷门特征, 则 n' 必须与 u 和协同者 u_1 相关联, 其中, $u \neq u_1$ 。

对于不具备 RT 的节点 $n' \in S_{(x,y)}$, 调用 $\text{DecryptNode}(\text{ct}_\tau, \text{rk}^u, (x, y))$ 算法, 并将该算法的输出存储在 $f_{n'}$ 。这意味着云服务器利用发起者上传的重加密密钥 rk^u 解密得到该孩子节点 n' 所关联的秘密值。对于具备 RT 特征的孩子节点 $n' \in S_{(x,y)}$, 调用 DecryptNode , 并将该算法的输出存储在 $f_{n''}$ 和 $f_{n'''}^u$ 中。这意味云服务器可以分别利用发起者和协同者上传的重加密密钥解密得到该孩子节点 n' 所对应的准秘密值。最后, 云服务器需要将这两个输出 $f_{n''}$ 和 $f_{n'''}^u$ 合并为 $f_{n'}$, 该具备 RT 特征的节点才最终被暴露, 合并公式如下。

$$f_{n'} = f_{n''}^u \cdot f_{n'''}^{u_1} \cdot e(\hat{c}_{(x,y)}, (e_{tr}^u)^{-1}/e_{tr}^{u_1}) =$$

$$e(g, h)^{r^u q_{n'}(0)\delta(\beta_1 - \beta_2)}$$

云服务器可以进一步使用多项式插值公式来计

算 $f_{(x,y)} = \prod_{n' \in S_{(x,y)}} f_{n'}^{\triangleright_{j, S'_{(x,y)}(0)}}$, 其中 $j = \text{index}(n')$,

$S'_{(x,y)} = \text{index}(n') : n' \in S_{(x,y)}$ 。通过解密节点 (1,1) 得到 $A = f(1,1) = e(g, h)^{r^u q_1^1(0)(\beta_1 - \beta_2)\delta} = e(g, h)^{r^{u\ell}(\beta_1 - \beta_2)\delta}$ 。

随后，云服务器可以定义以下等式 $\triangleright_\gamma(u, U) = \gamma^{-1}(\prod_{u_i \in U \cap u_i \neq u} (\gamma + H_1(u_i)) - \prod_{u_i \in U \cap u_i = u} H_1(u_i))$,

并计算

$$c'_5 = e(c_5, r_5^u) / (c_6 \cdot A) = e(h^{\beta \ell \delta}, v^s) \\ c'_1 = c_1 \cdot (e(c_4, h^{\triangleright_\gamma(u, U)}) \cdot e(r_1^u, c_2))^{-1/\prod_{u_i \in U \cap u_i \neq u} H_1(u_i)} = \\ (m \| ck) \oplus e(v^s, h^{-k\delta})^{\prod_{u_i \in U} (\gamma + H_1(u_i)) / H_1(u_i)}$$

最后，云服务器设置 $c'_2 \leftarrow r_2^u$, $c'_3 \leftarrow r_3^u$, $c'_4 \leftarrow r_4^u$, $\bar{c}' \leftarrow \bar{c}$, $c'_6 \leftarrow c_{3,1} \cdot c_{3,2}$, 并传播重加密密文 ct 给新用户组 U' , 其中, $ct = (c'_0, c'_1, c'_2, c'_3, c'_4, c'_5, c'_6, \bar{c}')$ 。

6) Decrypt。算法输入用户标识符 u , 公钥元组 $pk \leftarrow (h, h^\gamma, \dots, h^{\gamma^n}, v, v^\gamma, \dots, v^{\gamma^n}, h^\beta, v^\beta, \phi, g^\gamma, \phi, e(g, h)^\gamma, e(g, h), g^{\beta_1 \beta_2}, h^{\beta_1 - \beta_2})$, 身份密钥 sk^u 和原始密文 $ct_\tau \leftarrow (c_x^y, c_x^{y'}, c)$ 。如果 $u \in U$, 用户定义 $\triangleright_\gamma(u, U) = \gamma^{-1}(\prod_{u_i \in U \cap u_i \neq u} (\gamma + H_1(u_i)) - \prod_{u_i \in U \cap u_i = u} H_1(u_i))$, 并计算

$$\zeta = (e(c_4, h^{\triangleright_\gamma(u, U)}) \cdot e(sk^u, c_2))^{1/\prod_{u_i \in U \cap u_i \neq u} H_1(u_i)} = e(g, h)^{k\delta}$$

最后，用户获得 $(m \| ck) = c_1 \oplus \zeta$ 。如果 $\bar{c}' = \phi^{H_1(m)} \phi^{H_1(ck)}$, 则输出明文数据 m 。

7) ProofGen。输入用户标识符 u' , 公钥元组 $pk \leftarrow (h, h^\gamma, \dots, h^{\gamma^n}, v, v^\gamma, \dots, v^{\gamma^n}, h^\beta, v^\beta, \phi, g^\gamma, \phi, e(g, h)^\gamma, e(g, h), g^{\beta_1 \beta_2}, h^{\beta_1 - \beta_2})$, 用户的身份密钥 $sk^{u'}$ 和重加密密文 $ct = (c'_0, c'_1, c'_2, c'_3, c'_4, c'_5, c'_6, \bar{c}')$ 。如果 $u' \in U'$, 用户可以定义以下等式 $\triangleright_\gamma(u', U') = \gamma^{-1}(\prod_{u_i \in U' \cap u_i \neq u'} (\gamma + H_1(u_i)) - \prod_{u_i \in U' \cap u_i = u'} H_1(u_i))$, 并计算

$$\zeta' = (e(c'_4, h^{\triangleright_\gamma(u', U')}) \cdot e(sk^{u'}, c'_2))^{1/\prod_{u_i \in U' \cap u_i \neq u'} H_1(u_i)} = e(g, h)^{k'}$$

然后，用户计算 $z' = c'_3 / H_3(\zeta') = h^s$, 并可以获得 $(m \| ck) = (c'_1 \cdot e(z', c'_6)) \oplus c'_5$ 。当 $\bar{c}' = c' = \phi^{H_1(m)} \phi^{H_1(ck)}$, 用户得到 m 和 ck 。最后，用户生成验证阶段所需的验证值 $\pi = (m, ck)$ 并上传至云服务器。

8) Verify。输入原始密文 $ct_\tau \leftarrow (c_x^y, c_x^{y'}, c)$, 重加密密钥 $rk^u \leftarrow (r_1^u, r_2^u, r_3^u, r_4^u, r_5^u, r_6^u, r_7^u, r_8^u)$, 重加密密文 $ct = (c'_0, c'_1, c'_2, c'_3, c'_4, c'_5, c'_6, \bar{c}')$ 以及用户上传的验证值 $\pi = (m, ck)$ 。仲裁机构首先会验证以下公

式 $\bar{c}' = \bar{c}$, $c'_2 = r_2^u$, $c'_3 = r_3^u$, $c'_4 = r_4^u$, $c'_6 = c_{3,1} \cdot c_{3,2}$ 是否成立。如果上述公式中存在某一个公式不成立，则输出 False，表明云服务器未正确生成重加密密文。否则，仲裁机构将会继续计算 $\delta = H_1(\text{MLE}.\text{Encrypt}(m, ck))$ 并进一步验证以下等式 $\bar{c}' = \bar{c} = \phi^{H_1(m)} \phi^{H_1(ck)}$, $c'_1 \neq (m \| ck) \oplus (r_8^u)^\delta$ 和 $c'_5 \neq (r_7^u)^\delta$ 是否成立。如果上述公式成立，表明云服务器未正确生成重加密密文，并输出 False，否则，该算法将会输出 True。

6 安全性证明

定理1 假设决策双线性 Diffie-Hellman 在群 G 下成立，本文方案满足选择明文攻击下不可区分性 (indistinguishability under chosen plaintext attack, IND-CPA)。

证明 本文通过描述敌手 A 和挑战者 C 之间的博弈来证明本文方案是 IND-CPA 安全的。

初始化：敌手 A 选择挑战身份集合 U^* 和挑战传播策略 T^* 。挑战者 C 随机选择 $x \in \mathbb{Z}_p^*$ 并运行 Setup 算法生成系统所需的公钥元组 $pk \leftarrow (h, h^\gamma, \dots, h^{\gamma^n}, v, v^\gamma, \dots, v^{\gamma^n}, h^\beta, v^\beta, e(g, h)^\gamma, e(g, h), g^\gamma, h^{\beta_1 - \beta_2}, g^{\beta_1 \beta_2}, \phi, \phi)$, 哈希函数 H_1 、 H_2 和 H_3 以及主密钥元组 msk , 其中 $v^{\gamma_i} = h^{\gamma_i \cdot x} (i \in [0, n])$ 。

阶段1：挑战者 C 保存映射表 L_k , 其存储敌手 A 的查询元组。敌手 A 可查询具有标识符 u_i 的身份密钥 sk^{u_i} 和带有属性集 S^{u_i} 的属性密钥 ak^{u_i} 。如果 $u_i \in U^*$ 且 $S^{u_i} \in T^*$, 挑战者 C 将会返回 $\perp$ 。否则，挑战者 C 运行 KeyGen 算法生成以下用户密钥对 $sk^{u_i} = g^{1/\gamma + H_1(u_i)}$, $ak^{u_i} = (d_0^u, \{d_j^{u_i}, d_j^{u_i'}\}_{j \in S_1^{u_i}}, \{d_j^{u_i''}, d_j^{u_i'''}\}_{j \in S_2^{u_i}}, e_{tr}^{u_i} = h^{(r^{u_i} - \theta_m)/\beta_1})$ 。此外，敌手 A 还可以对重加密密文 rk^{u_i} 发起查询，其中，用户标识符 $u_i \notin U^*$ 以及属性集 $S^{u_i} \notin T^*$ 。挑战者 C 运行 RKenGen 算法，随后，随机选择 $g^y \in G$ 并生成符合上述相应条件的重加密密钥 $rk^{u_i} = (g^y, h^{k' \prod_{u_j \in U} (\gamma + H_1(u_j))}, H_3(e(g, h)^{k'}) \cdot h^s, r_4^{u_i}, r_5^{u_i}, r_7^{u_i}, r_8^{u_i}, \{r_j^{u_i}, r_j^{u_i'}\}_{j \in S_1^{u_i}}, r_j^{u_i''} = e_{tr}^{u_i}, \{r_j^{u_i''}, r_j^{u_i'''}\}_{j \in S_2^{u_i}})$ 。最后，C 将其提供给敌手 A 并记录到 L_k 。

挑战：敌手 A 挑战等长消息 m_0 和 m_1 。挑战

者 C 选择 $b \in \{0,1\}$, 运行 Encrypt 算法来生成挑战密文 $ct_\tau^* = (\delta = H_1(\text{MLE}.\text{Encrypt}(m, ck)), c_1, c_{3,1}, \{h^{q_x^{(0)}(\beta_1 - \beta_2)\delta}, H_2(\text{att}_{(x,y)})^{q_x^{(0)}\delta}\}_{(x,y) \in Y}, c_{3,2}, c_{3,3}, c_{3,4}, c_{4,1}, \{g^{q_x^{(0)}\beta_1\beta_2\delta}, H_2(\text{att}_{(x,y)})^{q_x^{(0)}\delta}, h^{q_x^{(0)}\delta}\}_{(x,y) \in X}, \bar{c}, c_5, c_6, c_8, v^{k\delta \cdot \prod_{u_j \in U} \gamma + H_1(u_j)/H_1(u_j)}, h^{k\delta \cdot \prod_{u_j \in U} (\gamma + H_1(u_j))})$ 并给敌手 A。

阶段 2: 该阶段与阶段 1 保持一致。

猜测: 敌手 A 输出一个猜测 b' 。

在以下生成接收者标识符集 U' 的重加密密钥查询的情况下, 证明发现 A 成功破解本文方案。

(1) 如果 $U' \cap L_k \neq \emptyset$, A 解密 h^s 来破坏本文方案安全性, 然后则有 $A_{\text{Dis}}(g^y, sk^{u_i} \cdot v^{s/H_1(u_i)})$ 成立。

由于敌手 A 可以计算 $h^{\gamma + H_1(u_i)}$ 和 $v = h^x$, 则 $A_{\text{Dis}}(e(g^y, h^{\gamma + H_1(u_i)}), e(g^{1/\gamma + H_1(u_i)} \cdot h^{x \cdot s/H_1(u_i)}, h^{\gamma + H_1(u_i)}))$ 成立, 其中, 由于 $e(g, h)$ 是公共参数, 则会有 $A_{\text{Dis}}(e(g^y, h^{\gamma + H_1(u_i)}) \cdot e(g, h)^{-1}, e(h, h)^{x \cdot s \cdot (\gamma + H_1(u_i))/H_1(u_i)})$ 成立。因 g^y 是随机选择的, 则存在 $c \in Z_p^*$ 使 $e(g^y, h^{\gamma + H_1(u_i)})e(g, h)^{-1} = e(g, h)^{y(\gamma + H_1(u_i)) - 1} = e(h, h)^c$ 。显然, 敌手 A 已知 h^x 、 h^s 和 $h^{(\gamma + H_1(u_i))/H_1(u_i)}$ 。因此, 当 $A_{\text{Dis}}(e(h, h)^c, e(h, h)^{x \cdot s \cdot (\gamma + H_1(u_i))/H_1(u_i)})$ 成立, 敌手 A 可通过区分 $e(h, h)^c$ 和 $e(h, h)^{x \cdot s \cdot (\gamma + H_1(u_i))/H_1(u_i)}$ 来打破定理 1 中的假设。

(2) 如果 $U' \cap L_k = \emptyset$, 则 A 可在不知道任何 $u_i \in U'$ 情况下破坏本文方案的 IND-CPA 安全性。

因此, 由反证法思路, 定理 1 可以被证明。

定理 2 假设 H_1 是抗碰撞哈希函数且离散对数成立, 本文方案是公平可验证的。

证明 假设存在一个能够以不可忽略的优势 ϵ 破解本文方案公平可验证性的敌手 A。该证明可构建挑战者 B, B 能够以优势 ϵ 解决离散对数问题, 并以概率 ϵ' 打破哈希函数 H_1 的抗碰撞性, 满足 $\epsilon' \geq \epsilon - \text{Adv}_{\text{DL}} - \text{Adv}_{\text{MLE}}$, 其中, Adv_{DL} 为打破离散对数假设的优势, Adv_{MLE} 为打破 MLE 算法的优势。

初始化: 挑战者 B 选择挑战身份集合 U^* 和挑战传播策略 T^* 。挑战者 B 随机选择 ϵ 、 η 并运行 Setup 算法来生成系统所需的公钥元组 $pk \leftarrow (h, h^\gamma, \dots, h^{\gamma^n}, v, v^\gamma, \dots, v^{\gamma^n}, h^\beta, v^\beta, \phi, g^\gamma, h^{\beta_1 - \beta_2}, e(g, h)^\gamma, e(g, h), g^{\beta_1 \beta_2}, \phi)$, 最后生成 H_1 、 H_2 和 H_3 。

阶段 1: 当敌手 A 发出 $O_{\text{ak}^u}(S^u)$ 、 $O_{\text{rk}^u}(S^u, \bar{U})$ 和 $O_{\text{Verify}}(ct_\tau, rk^u, ct, \pi)$ 查询时, 由于挑战者 B 知道 msk, 则挑战者 B 通过 KeyGen、RKeyGen 和 Verify 算法生成密钥和重加密密钥, 然后返回结果给 A。

挑战: 敌手 A 输出消息 m^* 和传播策略 T^* , 并将其发给 B。B 计算 $ct_\tau^* = \text{Encrypt}(m^*, T^*, U^*)$, $ct^* = \text{REncrypt}(ct_\tau^*, rk^{u^*})$, 其中 rk^{u^*} 是符合 T^* 以及用户组 $\bar{U}$ 的重加密密钥, 并将 ct_τ^* 、 ct^* 发送给 A。

阶段 2: 该阶段与阶段 1 保持一致。

猜测: 敌手 A 输出一个验证值 $\pi = (m, \rho)$ 和一个重加密密文 $ct^*(\bar{U})$ 。挑战者 B 选择一个符合 $\bar{U}$ 的用户标识符 u , 并使用 msk 生成一个身份密钥 sk^u 。然后, 挑战者 B 可以解密 $ct^*(\bar{U})$ 以获取 m^* 和 δ^* 。如果 A 赢了游戏, 这意味 $m^* \notin \{m^*, \perp\}$, 即 $\bar{c}^* = c^*$ 。

$$\bar{c}^* = c^* \Leftrightarrow$$

$$\phi^{H_1(m^*)} \phi^{H_1(ck^*)} = \phi^{H_1(m^*)} \phi^{H_1(ck^*)} \Leftrightarrow$$

$$g^{\epsilon \cdot H_1(m^*) + \eta \cdot H_1(ck^*)} = g^{\epsilon \cdot H_1(m^*) + \eta \cdot H_1(ck^*)} \Leftrightarrow \epsilon \cdot (H_1(m^*) - H_1(m^*)) = \eta \cdot (H_1(ck^*) - H_1(ck^*))$$

最后, 挑战者 B 可以从上述公式计算出 $\epsilon = [\eta \cdot (H_1(ck^*) - H_1(ck^*))] / [(H_1(m^*) - H_1(m^*))]$ 。

分析: 如果敌手 A 赢得了上述游戏, 则意味着 $\text{Verify}(ct_\tau^*, rk^{u^*}, ct^*, \pi) = \text{False}$, 并可得到 $\delta = H_1(m \| ck)$ $c^* = \phi^{H_1(m)} \phi^{H_1(ck)}$, $\bar{c}^* = \phi^{H_1(m)} \phi^{H_1(ck)}$, $\bar{c}^* = c_1^* \neq (m \| ck) \oplus (r_8^{u^*})^\delta$ 。由于式 $c_5^* \neq (r_7^{u^*})^\delta$ 成立, 有 $(m \| ck) \neq (m^* \| ck^*)$ 。此外, 由于存在等式 $\bar{c}^* = c^* = \phi^{H_1(m)} \phi^{H_1(ck)}$, 则将会进一步存在等式 $\phi^{H_1(m^*)} \phi^{H_1(ck^*)} = \phi^{H_1(m)} \phi^{H_1(ck)}$ 。将 $\phi = \phi^\eta$ 表示为未知的随机值, 然后有 $H_1(m^*) = H_1(m)$ 或者 $H_1(ck^*) = H_1(ck)$ 。否则, 任何人都可使用得到的值来计算 $\eta = (H_1(m^*) - H_1(m)) / (H_1(m^*) - H_1(m))$ 并解决离散对数假设。同时, 挑战者 B 可找到抗碰撞哈希函数 H_1 的一个碰撞元组, 其中, $m \neq m^*$, $H_1(m^*) = H_1(m)$, 或者 $ck \neq ck^*$, $H_1(ck^*) = H_1(ck)$ 。挑战者 B 找到这样一个碰撞元组的优势是 $\epsilon - \text{Adv}_{\text{DL}} - \text{Adv}_{\text{MLE}}$ 。

因此, 假设 H_1 是一个抗碰撞的哈希函数并且离散对数假设成立, 则本文方案是公平可验证的。

定理3 本文方案是抗合谋的。

证明 首先, 考虑未经授权的数据传播者合谋的情况, 在Encrypt算法中, 原始密文包含了一个随机秘密值 ℓ 。因此, 为了成功执行重加密操作, 云服务器必须使用重加密密钥恢复 $A = e(g, h)^{r^\ell(\beta_1 - \beta_2)\delta}$ 。然而, 当数据传播者上传的重加密密钥不符合原始密文中的协同传播树时, 则云服务器无法获取 $A = e(g, h)^{r^\ell(\beta_1 - \beta_2)\delta}$ 。其次, 同一组内的数据传播者可以使用 e_{tr}^u 协同解密满足随机陷门特征的叶子节点, 但不在同一组中的数据传播者则无法协同解密这些叶子节点。这是因为数据传播者所拥有的 e_{tr}^u 与唯一的随机值 θ_m 相关联。最后, 即使数据传播者与云服务器相互勾结以访问密文, 这样的合谋攻击也不会生效, 这是因为云服务器在任何阶段都无法获得任何隐私信息。

7 性能评估

仿真实验基于Java语言和椭圆双曲线系统算法库, 硬件平台是Intel Core i5 CPU@2.20 Ghz, 内存24 GB。符号描述如表1所示。本节将现有方案与本文方案的计算开销、通信开销进行比较, 如表2、表3所示。

表1 符号描述

符号	描述
P	一次双线性配对操作
E	G 中的一个指数运算
E_t	G_t 中的一个指数运算
$ A_t $	ct_t 中的属性数
$ A $	ct 中的属性数
$ A_X $	t_t 中有RT特征的节点数
$ G $	G 中元素的位长度
$ G_t $	G_t 中元素的位长度
$ Z_p $	Z_p 中元素的位长度

表2 现有方案与本文方案的计算开销比较

Table 2 The comparisons of computation costs in the existing schemes

算法	Encrypt	RKeyGen	Decrypt	ProofGen
VF-CP-ABPRE	$(4 + 3 A_t)E + E_t$	$(5 + A_t + 3 A)E + 2E_t$	$(1 + 2 A_t)P + 2E + A_t E_t$	$(1 + 2 A)P + 2E + (A + 1)E_t$
VF-PPBA	$(4 + 2 A_t)E + E_t$	$(9 + 2 A_t + 2 A)E + 3E_t$	$(3 + A_t)P + 2E + 2E_t$	$4P + (2 + A)E + 3E_t$
本文方案	$(10 + 2 A_t + A_X)E + 2E_t$	$6E + 2P + E_t$	$2P + 3E_t$	$3P + 3E_t$

表3 现有方案与本文方案的通信开销比较

Table 3 The comparisons of communication costs in the existing schemes

算法	Encrypt	REncrypt
VF-CP-ABPRE	$(3 + 2 A_t) G + G_t $	$(3 + 2 A) G + 3 G_t $
VF-PPBA	$(3 + A_t) G + G_t $	$(6 + A) G + 4 G_t $
本文方案	$(2 A_t + A_X + 9) G + 2 G_t $	$5 G + 2 G_t $

7.1 计算开销评估

(1) Encrypt算法计算开销

Encrypt算法的计算开销比较如图3所示, 所有方案在该算法上的计算开销都随着属性数量的增加而线性增加。本文方案比VF-CP-ABPRE方案更高效, 其中当属性数量为25个时, 本文加密时间仅需1 186.2 ms。具体由表2可知, 本文方案需 $(10 + 2|A_t| + |A_X|)E + 2E_t$ 计算开销实现数据传播者进行数据协同传播。其中, VF-PPBA方案的计算开销为 $(4 + 2|A_t|)E + E_t$, 比本文方案高效, 但是该方案并不支持协同传播功能。VF-CP-ABPRE方案需要为同一个属性生成多个密文, 因此, 计算开销为 $(4 + 3|A_t|)E + E_t$ 。

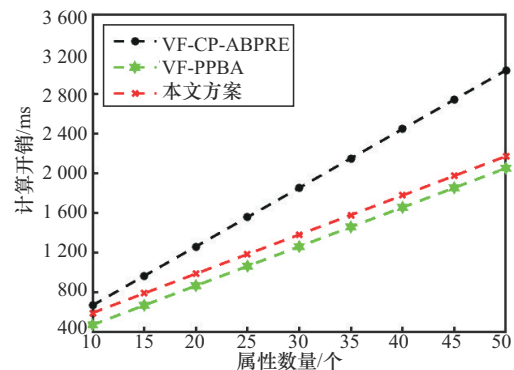


图3 Encrypt算法的计算开销比较

Figure 3 Comparison of computation costs on the Encrypt algorithm

(2) RKeyGen计算开销

RKeyGen算法的计算开销比较如图4所示, 可

见本文方案与属性数量无关, 比其他方案更高效。当属性数量为25个时, VF-CP-ABPRE和VF-PPBA方案分别需要2 072.7 ms、2 153.6 ms, 而本文仅需152.7 ms。具体由表2可知, 本文方案只需根据标识符来定义新用户组而无须生成新的访问策略, 因此在该阶段计算开销仅需 $6E + 2P + 1E_t$ 。其他方案要为用户组生成新访问策略, 故计算开销为 $(5 + |A_t| + 3|A|)E + 2E_t$ 和 $(9 + 2|A_t| + 2|A|)E + 3E_t$ 。

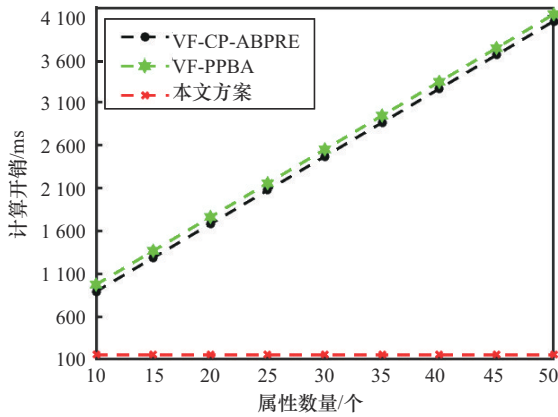


图4 RKeyGen算法的计算开销比较

Figure 4 Comparison of computation costs on the RKeyGen algorithm

(3) Decrypt 计算开销

Decrypt算法的计算开销比较如图5所示, 本文方案在该阶段的效率明显高效于其他两个对比方案。当属性数量为25个时, 本文方案计算开销只需要91.5 ms, 然而, VF-CP-ABPRE和VF-PPBA方案则分别需要918.1 ms和497.2 ms。具体由表2可知, 在解密原始密文的过程中, 本文方案只需要用户解密一个与属性无关的广播密文即可, 因此仅需要 $2P + 3E_t$ 的计算开销。VF-CP-ABPRE和VF-PPBA方案则需要用户解密基于属性加密的密文, 即这两个方案的计算开销都与原始密文的属性数量呈线性关系。因此, 这两个方案计算开销分别为 $(1 + 2|A_t|)P + 2E + |A_t|E_t$ 和 $(3 + |A_t|)P + 2E + 2E_t$ 。

(4) ProofGen 计算开销

ProofGen算法的计算开销比较如图6所示。其中, 本文方案验证值生成算法的计算开销是常量级的。相比所要对比的两个方案, 本文方案在该阶段上所需的计算开销是最高效的。除此之外, 可以从图6中看出其他两个对比方案的计算开销都会随着

属性数量的增加而增加, 而本文方案的计算开销则保持在107.7 ms。具体由表2可知, 本文方案由于只需要用户解密一个与属性数量无关的广播密文, 故仅需要 $3P + 3E_t$ 的计算开销即可生成验证阶段所需要的验证值。然而, VF-CP-ABPRE和VF-PPBA方案都需要用户解密基于属性加密的密文, 因此, 这两个方案计算的开销分别为 $(1 + 2|A_t|)P + 2E + (|A_t| + 1)E_t$ 和 $4P + (2 + |A_t|)E + 3E_t$ 。

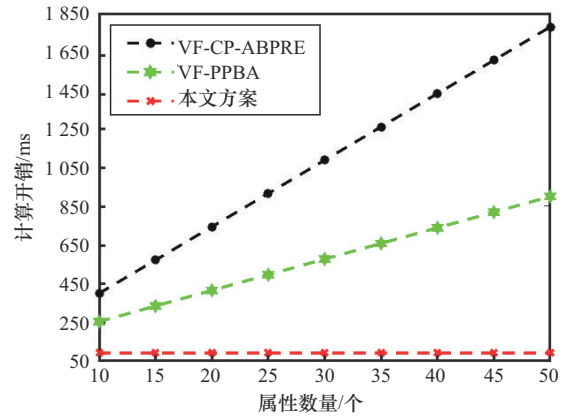


图5 Decrypt算法的计算开销比较

Figure 5 Comparison of computation costs on the Decrypt algorithm

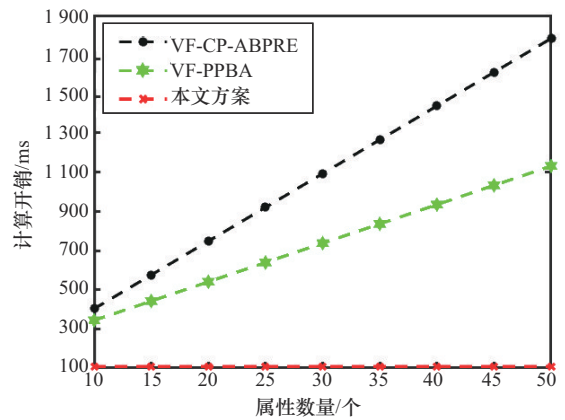


图6 ProofGen算法的计算开销比较

Figure 6 Comparison of computation costs on the ProofGen algorithm

(5) 不同算法的计算开销评估

本文方案的计算开销与数据传播者数量相关性实验测试如图7所示。其中可见RKeyGen、Decrypt和ProofGen算法的计算开销保持恒定, 与数据传播者数量无关, 经测试执行上述算法分别需要152.7 ms, 91.5 ms和107.7 ms。然而, Encrypt算法的计算开销随着数据传播者的数量增加而线性

增加,当数据传播者数量为4时,执行该算法计算开销为674 ms。

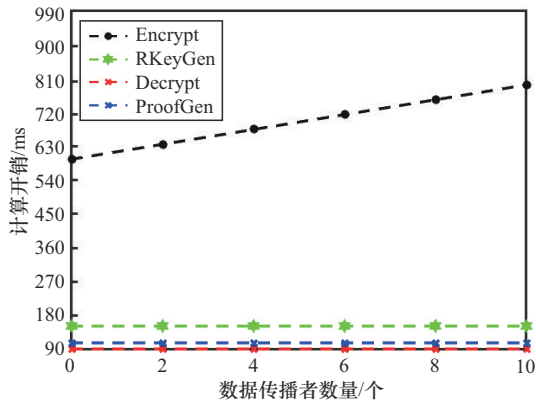


图7 不同算法的计算开销评估

Figure 7 Computation costs evaluation of the different algorithms

7.2 通信开销评估

(1) Encrypt算法通信开销

Encrypt算法的通信开销比较如图8所示,其中,所有方案的通信开销都随着原始密文中的属性数量增加而增加,当属性数量为15个时,本文方案通信开销为5.125 KB。具体由表3可知,为了传播与细粒度的协同传播相关的原始密文,本文方案通信开销为 $(2|A_t| + |A_x| + 9)|G| + 2|G_t|$ 。因不需要为协同传播来传播额外密文,VF-CP-ABPRE和VF-PPBA方案的通信开销为 $(3 + 2|A_t|)|G| + |G_t|$ 和 $(3 + |A_t|)|G| + |G_t|$ 。

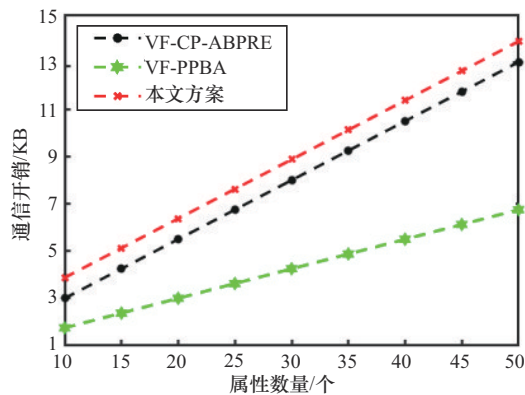


图8 Encrypt算法的通信开销比较

Figure 8 Comparison of communication costs on the Encrypt algorithm

(2) REncrypt通信开销

REncrypt算法的通信开销比较如图9所示,本文方案在该阶段上的通信开销是恒定不变的,并且明

显高效于所要对比的两个方案。从图9可以看出,本文方案的通信开销始终为0.875 KB。具体来说,本文方案由于只需要定义新用户的标识符集而并不需要传播关于新访问策略的密文,故通信开销仅为 $5|G| + 2|G_t|$ 。然而,为了让新用户能够访问到数据,方案VF-CP-ABPRE和VF-PPBA都需要传播新访问策略的密文,因此,这两个方案的通信开销分别为 $(3 + 2|A_t|)|G| + 3|G_t|$ 和 $(6 + |A_t|)|G| + 4|G_t|$ 。

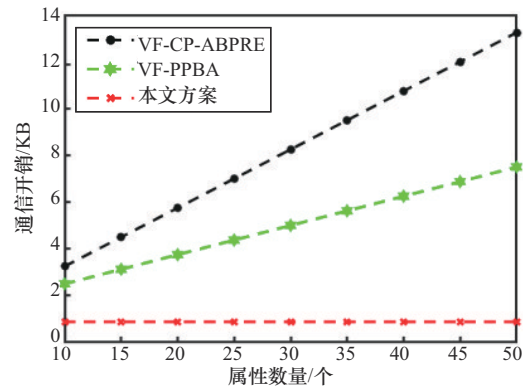


图9 REncrypt算法的通信开销比较

Figure 9 Comparison of communication costs on the REncrypt algorithm

(3) 不同算法的通信开销评估

不同算法的通信开销评估如图10所示,在REncrypt算法中,云服务器传播重加密密文过程中所需的通信开销与数据传播者的数量无关,始终为0.875 KB。但是,在Encrypt算法中,生成的初始密文大小与数据传播者数量呈线性相关,因此其通信开销呈不断递增趋势。当数据传播者数量为4时,传播Encrypt算法的密文通信开销为4.375 KB。

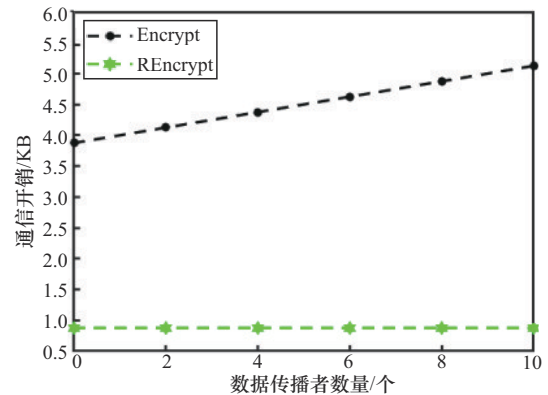


图10 不同算法的通信开销评估

Figure 10 Communication costs evaluation of the different algorithms

8 结束语

本文提出的可支持公平验证的数据群组灵活协同共享方案通过设计协同传播树结构和计算开销恒定的验证值生成机制, 不仅实现了多传播者协作的灵活性, 还降低了验证值生成等阶段的计算开销, 从而确保了云服务器与用户之间的公平验证效率。另外, 理论分析和性能评估表明, 本文方案是IND-CPA安全、高效和公平可验证的。未来的研究将聚焦于海量数据传播者环境下进一步优化方案的性能。

参考文献:

- [1] 葛文婷, 李卫海, 俞能海. 基于属性加密的块级云数据去重方案[J]. 网络与信息安全学报, 2023, 9(5): 106-115.
- GE W T, LI W H, YU N H. Block level cloud data deduplication scheme based on attribute encryption[J]. Chinese Journal of Network and Information Security, 2023, 9(5): 106-115.
- [2] ZHANG J, SU S, ZHONG H, et al. Identity-based broadcast proxy re-encryption for flexible data sharing in VANETs[J]. IEEE Transactions on Information Forensics and Security, 2023, 18: 4830 - 4842.
- [3] QIU J, HWANG G, LEE H. Efficient conditional proxy re-encryption with chosen-ciphertext security[C]//Proceedings of 2014 Ninth Asia Joint Conference on Information Security, Berlin: Springer, 2014: 104-110.
- [4] HU H, ZHOU Y, Z C, et al. Efficient and HRA secure universal conditional proxy re-encryption for cloud-based data sharing[J]. Applied Sciences, 2022, 12(19): 9586-9600.
- [5] XU P, JIAO T, WU Q, et al. Conditional identity-based broadcast proxy re-encryption and its application to cloud email[J]. IEEE Transactions on Computers, 2016, 65(1): 66-79.
- [6] CUI J, LI B, ZHONG H, et al. Achieving revocable attribute group-based encryption for mobile cloud data: a multi-proxy assisted approach in VANETs[J]. IEEE Transactions on Dependable and Secure Computing, 2023, 20(4): 2988-3001.
- [7] SUN J, XU G, ZHANG T, et al. Verifiable, fair and privacy-preserving broadcast authorization for flexible data sharing in clouds[J]. IEEE Transactions on Information Forensics and Security, 2023, 18: 683-698.
- [8] GE C, SUSILO W, BAEK J, et al. A verifiable and fair attribute-based proxy re-encryption scheme for data sharing in clouds[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 19(5): 2907-2919.
- [9] 刘尚, 郭银章. 云计算多授权中心 CP-ABE 代理重加密方案[J]. 网络与信息安全学报, 2022, 8(3): 176-188.
- LIU S, GUO Y Z. Multi-authority based CP-ABE proxy re-encryption scheme for cloud computing[J]. Chinese Journal of Network and Information Security, 2022, 8(3): 176-188.
- [10] GE J, WEN M, WANG L, et al. Attribute-based collaborative access control scheme with constant ciphertext length for smart grid[J]. IEEE International Conference on Communications, 2022: 540-546.
- [11] HE Y, WANG H, LI Y, et al. An efficient ciphertext-policy attribute-based encryption scheme supporting collaborative decryption with blockchain[J]. IEEE Internet of Things Journal, 2022, 9(4): 2722 - 2733.
- [12] XUE Y, XUE K, GAI N, et al. An attribute-based controlled collaborative access control scheme for public cloud storage[J]. IEEE Transactions on Information Forensics and Security, 2019, 14(11): 2927-2942.
- [13] CHEN N, LI J, ZHANG Y, et al. Efficient CP-ABE scheme with shared decryption in cloud storage[J]. IEEE Transactions on Computers, 2022, 71(1): 175-184.
- [14] MADANI M A, KERKRI A, AISSAOUI M. MC-ABAC: an ABAC-based model for collaboration in multi-cloud environment[J]. International Journal of Advanced Computer Science and Applications, 2023, 14(6): 1182-1190.
- [15] ZHANG Y, LI B, LIU B, et al. An attribute-based collaborative access control scheme using blockchain for IoT devices[J]. Electronics, 2020, 9(2): 285-307.
- [16] LI X, WANG H, MA S, et al. Revocable and verifiable weighted attribute-based encryption with collaborative access for electronic health record in cloud[J]. Cybersecurity, 2024, 7(18): 2523-3246.
- [17] HUANG Q, YANG Y, FU J. Secure data group sharing and dissemination with attribute and time conditions in public cloud[J]. IEEE Transactions on Services Computing, 2021, 14(4): 1013-1025.
- [18] HUANG Q, YANG Y, YUE W, et al. Secure data group sharing and conditional dissemination with multi-owner in cloud computing[J]. IEEE Transactions on Cloud Computing, 2021, 9(4): 1607-1618.
- [19] DENG H, ZHANG J, QIN Z, et al. Policy based broadcast access authorization for flexible data sharing in clouds[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 19(5): 3024-3037.
- [20] JIANG L, ALAZAB M, QIN Z. Secure task distribution with verifiable re-encryption in mobile-crowdsensing-assisted emergency IoT system[J]. IEEE Internet of Things Journal, 2024, 11(3): 3896-3908.
- [21] XIE C, SHI R, ZHANG X, et al. Verifiable outsourcing emrs scheme with attribute-based encryption in cloud-edge environments[J]. Journal of Information Security and Applications, 2023, 76: 103526-103538.

[作者简介]



王树兰（1987-），女，河南新乡人，博士，深圳技术大学副教授，主要研究方向为可搜索加密、密码学算法与协议、数据安全。



邹家须（1985-），男，河南商丘人，深圳技术大学实验师，主要研究方向为网络与信息安全、终端设备安全、数据安全。



甘竞宏（1999-），男，广西贵港人，深圳大学硕士生，主要研究方向为密码学、信息安全。



赵国英（1996-），男，河南商丘人，深圳大学硕士生，主要研究方向为信息安全与密文检索。



赵陈斌（1996-），男，安徽六安人，武汉大学博士生，主要研究方向为网络安全、应用密码学。



张应辉（1985-），男，陕西西安人，博士，西安邮电大学教授，主要研究方向为公钥加密、云安全和无线网络安全。