软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007647] [CSTR: 32375.14.jos.007647]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于变色龙哈希的可撤销注册属性加密方案*

王经纬[1], 夏志华[1], 宁建廷[2,3,4], 许胜民[3,4], 李朋林[5]

[1](暨南大学 网络空间安全学院, 广东 广州 510632)

[2](武汉大学 国家网络安全学院, 湖北 武汉 430072)

[3](福建师范大学 计算机与网络空间安全学院, 福建 福州 350117)

[4](分析数学及应用教育部重点实验室 (福建师范大学), 福建 福州 350117)

[5](上海同态信息技术有限责任公司, 上海 200030)

通信作者: 宁建廷, E-mail: jtning88@gmail.com

摘　要: 属性加密方案能够实现基于用户属性的细粒度访问控制, 但密钥的生成依赖于一个授权中心来完成, 容易带来密钥泄露的安全隐患. 注册加密方案通过引入用户注册、公钥聚合等机制, 缓解了由授权中心带来的安全风险. 然而, 目前基于多 sRABE 子方案并行构建的注册属性加密方案在用户动态增减时需要频繁执行公钥重聚合操作, 其计算与存储开销随子方案数量线性增长, 难以适用于用户频繁变化的场景. 针对上述不足, 提出一种基于变色龙哈希的可撤销注册属性加密方案, 在单一 sRABE 框架下实现了高效的用户注册与撤销. 方案引入陷门可控的变色龙哈希函数, 通过公钥聚合方对公钥一致性的维护, 新用户加入时无需执行系统级主公钥聚合, 用户撤销仅需对注册槽状态进行标记, 从而避免了多子方案结构带来的计算开销. 在效率方面, 方案对新用户加入阶段的计算开销进行了分析, 结果表明该过程的计算复杂度仅与用户属性数量相关, 与系统用户规模无关. 安全性分析表明, 方案在不可区分模型下可以抵抗选择明文攻击.

关键词: 注册加密; 属性加密; 用户撤销; 变色龙哈希; 数据共享

中图法分类号: TP309

中文引用格式: 王经纬, 夏志华, 宁建廷, 许胜民, 李朋林. 基于变色龙哈希的可撤销注册属性加密方案. 软件学报. http://www.jos.org.cn/1000-9825/7647.htm

英文引用格式: Wang JW, Xia ZH, Ning JT, Xu SM, Li PL. Revocable Registered Attribute-based Encryption Scheme Using Chameleon Hash. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7647.htm

Revocable Registered Attribute-based Encryption Scheme Using Chameleon Hash

WANG Jing-Wei[1], XIA Zhi-Hua[1], NING Jian-Ting[2,3,4], XU Sheng-Min[3,4], LI Peng-Lin[5]

[1](College of Cyber Security, Jinan University, Guangzhou 510632, China)

[2](School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China)

[3](College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350117, China)

[4](Key Laboratory of Analytical Mathematics and Applications (Fujian Normal University), Fuzhou 350117, China)

[5](Shanghai Tongtai Information Technology Co. Ltd., Shanghai 200030, China)

Abstract: Attribute-based encryption (ABE) enables fine-grained access control based on user attributes, but it relies on a centralized authority for key generation, which may cause security risks such as key leakage. Registered-based encryption schemes mitigate this issue using mechanisms such as user registration and public key aggregation. However, existing constructions based on multiple parallel sRABE sub-schemes require frequent public key re-aggregation when users dynamically join or leave, leading to computational and storage overhead that grows linearly with the number of sub-schemes. Therefore, these constructions are difficult to apply in scenarios with

* 基金项目: 国家自然科学基金 (12441101, 62372108, 62402109, 62425205, 62572123)
收稿时间: 2025-10-17; 修改时间: 2025-11-24; 采用时间: 2026-01-30; jos 在线出版时间: 2026-06-10

frequent user changes. To overcome this limitation, this study proposes a chameleon-hash-based revocable registered attribute-based encryption scheme that achieves efficient user registration and revocation within a single sRABE framework. By introducing a trapdoor-controllable chameleon hash function and allowing the public key aggregator to maintain public key consistency, new users can be registered without system-level public key re-aggregation. Meanwhile, user revocation only requires marking the status of registration slots, thus avoiding the computational overhead caused by the multi-sub-scheme structure. In terms of efficiency, the computational overhead of the new-user registration phase is analyzed, and the results show that the computational complexity of this process depends only on the number of user attributes and is independent of the number of users in the system. Security analysis shows that the proposed scheme is secure against chosen-plaintext attacks under the indistinguishability model.

Key words: registered-based encryption; attribute-based encryption (ABE); user revocation; chameleon hash; data sharing

随着云计算与大数据技术的迅速发展, 如何在开放环境中实现安全、高效的数据共享已成为学术界与工业界的研究重点. 属性加密 (attribute-based encryption, ABE) 方案因能够在密文层面实现基于用户属性的细粒度访问控制, 被广泛应用于电子健康记录、云存储、物联网等多种场景[1−4]. ABE 允许数据拥有者在加密时定义访问策略, 只有满足策略要求的用户才能解密, 这为分布式环境下的数据隐私保护提供了优雅的解决方案[5,6]. 然而, 传统 ABE 方案仍然面临密钥安全方面的问题: 在大多数现有方案中, 属性授权中心必须生成系统主密钥并负责向用户分发私钥, 一旦属性授权中心被攻击或存在内部不当行为, 可能带来主密钥泄露、密钥托管以及用户隐私暴露等问题, 从而严重威胁系统的整体安全与稳定[7−9].

为了解决上述隐患, 有研究者提出了注册加密 (registered-based encryption) 的概念[10]. 该机制在密钥生成过程中引入了用户注册环节, 使得系统主公钥的生成不依赖于可信中心, 仅需要一个公钥聚合方来聚合所有合法用户的公钥. 通过这种设计, 注册加密有效避免了密钥托管的风险. 将注册加密机制与属性加密结合后, 形成的注册属性加密 (registered attribute-based encryption, RABE) 方案, 能够在保留属性加密灵活访问控制能力的同时, 显著提升密钥的安全性, 因而在近年的学术研究中引起了广泛关注.

尽管注册属性加密方案在理论与实践中均展现出巨大潜力, 但现有研究仍存在明显不足. 尤其在用户动态管理方面, 现实应用中的用户群体往往处于频繁变动之中, 新增或撤销用户时有发生. 然而, 大多数现有的 RABE 方案假设所有用户在系统初始化时已经确定, 并要求所有用户完成注册后才能生成系统主公钥. 这一前提在实际部署中难以满足, 严重限制了相关方案的可扩展性与灵活性. 为应对这一问题, 已有部分工作尝试设计支持用户动态变更的 RABE 方案, 但这些方案仍存在 2 类主要缺陷: 一方面, 部分方案无法支持密码原语的黑盒使用, 限制了与现有密码库和标准的兼容性[11]; 另一方面, 有些方案需要依赖于多个槽注册属性加密 (slotted registered attribute-based encryption, sRABE) 子方案来完成构造, 导致计算与存储开销较大, 难以在大规模应用中保持高效[12−15].

针对上述挑战, 本文提出一种基于变色龙哈希函数的可撤销注册属性加密方案. 变色龙哈希函数[16]是一类具备陷门可控性的哈希函数, 在已知陷门信息的情况下某个输入对应的哈希值可高效地被构造为另一个输入的碰撞, 非常适合用于实现灵活的用户身份更新与密钥管理. 我们在单个 sRABE 方案[17]中引入变色龙哈希函数, 使得方案在保证安全性的同时, 可以支持高效的用户撤销与注册更新, 避免了重复部署多个子方案所带来的冗余开销. 与现有方案相比, 本文方案在系统构造简洁性、运行效率及密钥安全性方面均取得了更好的平衡, 为动态环境下的安全数据共享提供了切实可行的解决思路. 本文的主要贡献如下.

(1) 提出一种基于变色龙哈希函数的可撤销注册属性加密方案, 基于变色龙哈希函数陷门可控的特性实现了数据用户的灵活动态管理, 扩展了注册属性加密方案的可用性.

(2) 在撤销机制方面, 结合注册加密方案需要公钥聚合的特点以及变色龙哈希函数可以有效制造碰撞的特性, 设计一种高效的用户撤销机制, 实现了动态用户撤销与注册, 新用户加入时无需重新生成或大规模修改系统主公钥, 仅需执行与对应注册槽及用户属性相关的局部密钥计算.

(3) 在计算开销方面, 本文方案基于单个 sRABE 方案[17]构建, 无需并行运行 l+1 个 sRABE 方案, 同时满足属性加密方案的基本要求, 有效降低了方案执行过程中的计算开销, 适用于大规模、多变动的应用环境.

(4) 在安全性方面, 证明本文方案在不可区分模型下可以抵抗选择明文攻击.

本文第 1 节介绍注册加密方案以及属性加密方案的研究现状. 第 2 节介绍本文所需的基础知识. 第 3 节介绍

本文方案的形式化定义及安全模型. 第 4 节详细地介绍本文方案的具体构造. 第 5 节分别对本文方案的安全性和计算开销展开分析. 最后在第 6 节总结全文.

1 相关工作

为解决密钥托管问题, 研究人员提出了许多方案. 首先, 采用门限密码学是一种较为经典的解决方案[18,19]. 这类方案通过秘密分享的方法将系统主密钥分割为多个秘密份额并分发给多个相互独立的授权中心. 任何机构都无法单独恢复主密钥或获得密文解密权限, 从而降低中心化带来的风险. 例如, Duan 等人[20]使用代理重加密技术构建了一个支持跨区块链数据共享的多中心属性加密方案, 可以对存储在云中的密文进行有效的访问策略更新操作. Ghopur 等人[21]在医疗物联网场景下构建了一个多中心的分布式可搜索属性加密方案, 其中每个授权中心都可以独立为用户生成属性密钥. 虽然这些方案缓解了单点故障问题, 在一定程度上提升了系统的可扩展性与可靠性. 但在这些方案中, 私钥的最终生成仍依赖于用户之外的第三方实体. 当足够数量的密钥生成方遭到腐化或互相勾结时, 系统的机密性依旧无法保障. 除此之外, 还有其他一些方案试图从责任溯源的角度加强中央授权机构对恶意方的管控力度[22−24], 包括在中心化体系中引入审计与可追责机制等. 这些方法在增强密钥管理透明度与可验证性方面具有一定效果, 但无法从根本上消除密钥托管的风险.

为彻底解决密钥托管问题, Garg 等人[10]于 2018 年首次提出注册加密的概念并给出了两个具体的方案构造, 这两个方案的注册执行时间分别与注册用户数量呈多项式和对数多项式关系. 随后, Cong 等人[11]进一步优化了该方案的效率. 在 2020 年, Goyal 等人在文献 [25] 中提出一种针对恶意密钥管理者的防护机制, 进一步提高了注册加密方案的安全性. 然而, 上述方案的构建都依赖于密码原语的非黑盒使用. 针对这个问题, Hohenberger 等人[17]在 EUROCRYPT 2023 中基于复合阶双线性群构建了第 1 个基于密码原语黑盒使用的注册加密方案. 在此基础上, 文献 [26] 提出了动态去中心化的函数加密方案, 允许加密者独立生成自己的主密钥, 并用其加密数据. 当用户希望访问数据时, 由加密者充当权威机构并提供部分密钥. Datta 等人[27]结合函数加密构建了面向线性函数的注册函数加密方案和注册属性内积函数加密方案, 将注册加密从谓词计算拓展到线性函数类别. 文献 [28] 以无信任环境为研究背景, 提出了基于多中心的注册属性加密方案, 缓解了注册加密方案可能面临的单点故障问题.

用户撤销作为用户管理的功能同样值得关注. 在注册加密方案中, 密钥在完成注册后即成为系统主公钥的一部分, 随系统运行长期使用. 然而, 无论是出于功能性需求还是安全性考虑, 密钥撤销都是至关重要的. 在注册加密之前, 关于撤销机制的研究主要分为两类: 通过更新非撤销用户的私钥实现撤销的间接撤销机制与通过管理撤销列表实现的直接撤销机制. Asano 等人[29]基于文献 [17] 的研究, 结合“二次幂”构造与完全二叉树结构首次在注册加密方案中实现了用户撤销. 该方案以增加注册加密方案的计算及存储开销为代价, 实现了撤销功能, 但其算法执行时间与撤销用户数量线性相关. Li 等人[13]提出通过“注销”用户的方式实现用户撤销. 然而, 每次用户撤销都需要维护用户的辅助密钥以及更新云服务器中的密文. 此外, 这些方案都依赖于文献 [17] 中提出的“基于 l+1 个 sRABE 方案构建注册属性加密方案”的架构.

外包解密最早被引入属性加密时, 主要用于将计算开销较大的解密操作外包给半可信服务器以减轻终端的计算负担. Green 等人在文献 [30] 中首次提出了支持外包解密的属性加密方案, 证明了在不泄露明文内容的前提下可以由服务器协助完成大部分解密计算. 随后, 相关工作[31−33]在可撤销属性加密和代理重加密等场景中进一步发展, 通过引入转换密钥或辅助解密密钥实现了对被撤销用户的访问控制与后向安全性保障. 本文方案借鉴了外包解密的基本思想, 但其目的并非降低用户解密开销, 而是通过引入辅助解密密钥机制确保被撤销用户即使持有历史私钥也无法解密后续生成的密文, 从而增强系统在动态用户场景下的后向安全性.

2 基础知识

2.1 常用符号说明

表 1 列出了在本文中使用符号的说明.

表 1 符号说明

符号	含义
$\mathbb{G}$, $\mathbb{G}_T$	以合数N为阶的双线性乘法循环群
$\mathbb{Z}_N$	以合数N为阶的整数群
U	系统属性空间
S	用户属性集合
L	注册加密方案槽数量
$(\boldsymbol{M}, \rho)$	访问策略
λ	安全参数
e	双线性映射

2.2 复合阶双线性映射

本文所提方案底层构建依赖于复合阶双线性群, 具体定义如下.

令 $(p_1, p_2, p_3, \mathbb{G}, \mathbb{G}_T, e, g)$ 表示一个复合阶双线性群, 其中 p_1, p_2, p_3 是 3 个不同的素数. $\mathbb{G}$ 和 $\mathbb{G}_T$ 是阶为 $N=p_1p_2p_3$ 的乘法循环群, g 是群 $\mathbb{G}$ 的生成元. 如果 e 满足下列条件, 则我们称 e: $\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$ 是一个双线性映射.

(1) 双线性: 对于任意 $u, v\in\mathbb{G}$ 以及 $a, b\in\mathbb{Z}_N$, $e(u^a, v^b)=e(u, v)^{ab}$.

(2) 非退化性: 存在 $u, v\in\mathbb{G}$ 使得 $e(u, v)\neq 1$.

(3) 可计算性: 对于任意 $u, v\in\mathbb{G}$, 都可以有效地计算 $e(u, v)$.

2.3 子群判定假设

令 $(p_1, p_2, p_3, \mathbb{G}, \mathbb{G}_T, e, g)$ 表示一个复合阶双线性群, 其中 p_1, p_2, p_3 是 3 个不同的素数. $\mathbb{G}$ 和 $\mathbb{G}_T$ 是阶为 $N=p_1p_2p_3$ 的乘法循环群, 且 $g_1\in\mathbb{G}_1, g_2\in\mathbb{G}_2, g_3\in\mathbb{G}_3$. 我们定义一系列分布 D_0, D_1, 其中每个分布 $D_b=(D, T_b)$ 都由一个集合 D 和一个挑战元素 T_b 组成. 如果对于所有的多项式时间攻击者 A 都存在一个可忽略函数 $negl()$ 使得对所有 $\lambda\in N$ 满足下列条件, 则称以下每个假设对于元组 $(p_1, p_2, p_3, \mathbb{G}, \mathbb{G}_T, e, g)$ 成立:

$$|\Pr[A(D, T_0)=1]-\Pr[A(D, T_1)=1]|=negl(\lambda).$$

假设 1: 随机选择 $r\in\mathbb{Z}_N$, 令 $D=((\mathbb{G}, \mathbb{G}_T, e, g, N), g_1, g_3)$, $T_0=g_1^r$, $T_1=(g_1g_2)^r$.

假设 2: 随机选择 $s_{12}, s_{23}\in\mathbb{Z}_N$, 令 $D=((\mathbb{G}, \mathbb{G}_T, e, g, N), (g_1g_2)^{s_{12}}, (g_2g_3)^{s_{23}})$, $T_0=(g_1g_3)^r$, $T_1=g^r$.

假设 3: 随机选择 $\alpha, s, t_1, t_2, r\in\mathbb{Z}_N$, 令 $D=((\mathbb{G}, \mathbb{G}_T, e, g, N), g_1, g_2, g_3, g_1^\alpha g_2^{t_1}, g_1^s g_2^{t_2})$, $T_0=e(g_1, g_1)^{\alpha s}$, $T_1=e(g, g)^r$.

2.4 访问结构

令 $U=\{A_1, A_2, \ldots, A_n\}$表示属性空间, 则称一个非空的属性集合 $A\subseteq 2^U$ 为访问结构. 此外, 对于任意 $B, C\in U$, 如果 $B\in A$ 并且 $B\subseteq C$, 那么 $C\in A$, 则 A 是单调的. 对于任意 $D\in A$, D 是授权集合.

2.5 变色龙哈希函数

一个变色龙哈希函数包括 KeyGen、Hash、Verify 以及 Adapt 这 4 个算法, 具体定义如下.

KeyGen$(1^\lambda)\to\{pk, sk\}$: 算法输入安全参数 λ, 输出公钥 pk 以及私钥 sk.

Hash$(pk, m)\to\{hash, r\}$: 算法输入公钥 pk 以及消息 m, 输出一个哈希值 $hash$ 以及随机数 r.

Verify$(pk, m, hash, r)\to 0$ or 1: 算法输入公钥 pk、消息 m、哈希值 $hash$ 以及随机数 r, 如果验证通过, 算法输出 1, 否则算法输出 0.

Adapt$(sk, m, m', hash, r)\to r'$: 算法输入私钥 sk、原消息 m、新消息 m'、与原消息 m 对应的哈希值 $hash$ 以及随机数 r, 输出与新消息相对应的随机数 r'.

为了便于理解, 我们还提供了一个变色龙哈希函数的实例.

KeyGen(1^λ): 根据安全参数 λ 生成群参数 $D=(\mathbb{G}, \mathbb{G}_T, p, e)$, 令 g 为群 $\mathbb{G}$ 的生成元, 选择随机数 $x\in\mathbb{Z}_p$, 计算 $Y=g^x$, 则 $pk=\{g, Y\}$, $sk=x$.

Hash(pk, m): 对于给定的消息 m, 选择随机数 $r\in\mathbb{Z}_p$, 计算消息 m 的哈希 $hash=g^mY^r$.

Verify(pk, m, $hash$, r): 验证等式 $hash=g^mY^r$ 是否成立, 如果成立输出 1, 否则输出 0.

Adapt(sk, m, m', $hash$, r): 根据 $m+xr=m'+xr'$来计算新的随机值 r', 并验证 $hash=g^{m'}Y^{r'}$.

2.6 基于槽的注册加密 (sRABE) 方案

(1) Setup(1^λ, $1^{|U|}$, 1^L)→crs: 算法的输入为安全参数 λ、属性空间 U 的最大属性数量以及系统最大用户数量 L (即注册加密方案中槽的数量), 输出公共参考字符串 crs.

(2) KeyGen(crs, i)→(pk_i, sk_i): 算法输入为公共参考字符串 crs 和槽下标 i, 输出公钥 pk_i 和私钥 sk_i.

(3) IsValid(crs, i, pk_i)→{0, 1}: 算法输入为公共参考字符串 crs、槽下标 i、用户公钥 pk_i, 如果 pk_i 是对应于槽 i 的合法公钥, 则输出 1, 否则输出 0, 表示公钥 pk_i 不合法.

(4) Aggregate(crs, (pk_1, S_1),…, (pk_L, S_L))→(mpk, hsk_1,…, hsk_L): 算法输入为公共参考字符串 crs 和一系列公钥及对应属性集 (pk_1, S_1),…, (pk_L, S_L), 输出主公钥 mpk 以及一系列对应的辅助解密密钥 hsk_1,…, hsk_L.

(5) Encrypt(mpk, ($\boldsymbol{M}$, ρ), m)→ct: 算法输入为主公钥 mpk、访问策略 ($\boldsymbol{M}$, ρ) 以及消息 m, 输出密文 ct.

(6) Decrypt(sk, hsk, ct)→res: 算法输入为密钥 sk、辅助解密密钥 hsk 以及密文 ct, 输出结果 res.

3 形式化定义及安全模型

本文所提的方案中共包含 5 个实体, 分别是: 系统建立方、公钥聚合方、数据拥有方、数据用户和云服务器. 本节首先介绍方案的系统模型, 包括各个参与方以及其职责; 其次, 定义构成本文方案的 8 个算法; 最后介绍本文方案的安全模型.

3.1 系统模型

本文方案的系统模型图如图 1 所示.

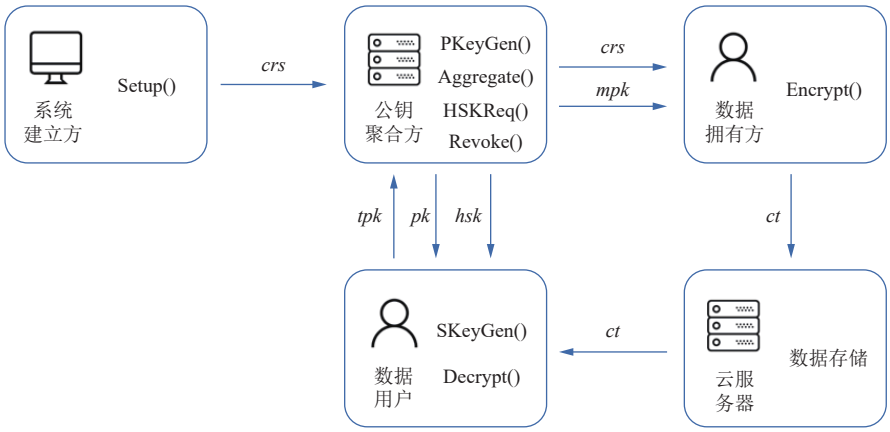

图 1 方案系统模型图

(1) 系统建立方是完全可信的, 负责系统的初始化并提供公共参考字符串 (common reference string, CRS), 在后续的系统运行过程中不参与任何流程与计算.

(2) 公钥聚合方是半可信实体, 负责公钥聚合、维护系统主公钥的一致性并辅助用户完成注册与撤销操作. 公钥聚合方不掌握任何用户私钥成分, 也不参与最终明文恢复过程. 公钥聚合方所持有的变色龙哈希函数陷门仅用于构造公钥层面的可控哈希碰撞, 不会影响系统的数据机密性.

(3) 数据拥有方在系统中可以通过制定不同的访问策略来向不同的用户群体共享自己的数据. 在完成数据加密的工作后, 数据拥有方将加密数据上传至云服务器存储以便数据用户访问.

(4) 数据用户可以在系统中注册并获取对应的解密密钥. 根据其所拥有的属性的不同, 各个数据用户的解密权限也不尽相同.

(5) 云服务器在系统中仅负责密文的存储与传输.

3.2 算法定义

我们在图 1 中展示了本文方案的具体执行流程. 该方案由 8 个算法组成, 对各个算法的描述如下.

(1) Setup(1^{λ}, $1^{|U|}$, 1^{L})→*crs*: 该算法由系统建立方执行, 负责系统构建并生成 CRS. 算法输入为安全参数 λ、属性空间 U 的最大属性数量以及系统最大用户数量 L (即注册加密方案中槽的数量), 输出公共参考字符串 *crs*.

(2) SKeyGen(*crs*, *i*)→(tpk_i, sk_i): 该算法由数据用户执行, 负责为自己生成私钥和一个临时公钥. 算法输入为公共参考字符串 *crs* 和用户对应的槽下标 *i*, 算法输出该用户的临时公钥 tpk_i 和私钥 sk_i.

(3) PKeyGen(*crs*, *i*, tpk_i, *L*, *D*, S_i)→(pk_i, *D*) or ⊥: 该算法由公钥聚合方执行, 负责将用户的临时公钥转换为用户公钥. 算法输入为公共参考字符串 *crs*、槽下标 *i*、临时公钥 tpk_i、系统用户数量上限 *L*、一个将槽下标映射到与用户 *i* 相关的身份信息的字典 *D* 以及用户属性 S_i, 算法输出该用户的公钥 pk_i 并更新字典 *D*.

(4) Aggregate(*crs*, (pk_1, S_1),…, (pk_L, S_L), *D*)→(*mpk*, *D*): 该算法由公钥聚合方执行, 负责将所有注册用户的公钥聚合为系统主公钥. 算法输入为公共参考字符串 *crs*、一系列公钥及对应属性集 (pk_1, S_1),…, (pk_L, S_L) 以及字典 *D*, 算法输出主公钥 *mpk* 并更新字典 *D* 的相关信息.

(5) Encrypt(*mpk*, (***M***, ρ), *m*)→*ct*: 该算法由数据拥有方执行, 负责加密待共享的消息. 算法输入主公钥 *mpk*、访问策略 (***M***, ρ) 以及消息 *m*, 算法输出加密后的密文 *ct*.

(6) HSKReq(*i*, *D*, *ct*)→*hsk*: 该算法由公钥聚合方执行, 负责为请求用户生成辅助解密密钥. 算法输入为用户对应的槽下标 *i*、字典 *D* 以及密文 *ct*, 算法输出一个针对用户 *i* 与密文 *ct* 的辅助解密密钥 *hsk*.

(7) Decrypt(*sk*, *hsk*, *ct*)→*res* or ⊥: 该算法由数据用户执行, 负责将加密后的消息恢复为明文. 算法输入为密钥 *sk*、辅助解密密钥 *hsk* 以及密文 *ct*, 算法输出解密结果 *res*.

(8) Revoke(*i*, *D*)→*D*: 该算法由公钥聚合方执行, 负责撤销选定用户的数据解密能力. 算法输入为待撤销用户对应的槽下标 *i*、字典 *D*, 输出更新后的字典 *D*.

3.3 安全模型

本文考虑选择明文攻击下不可区分安全模型. 在该模型中, 攻击者在系统初始化后固定其要挑战的访问策略, 并且能够自适应地请求任意未撤销用户的私钥, 但不允许获取与挑战密文对应的私钥或辅助解密密钥. 获胜目标是判定攻击者能否区分给定挑战密文所对应的明文信息, 具体的安全模型可以由以下挑战者 C 和攻击者 A 之间的安全游戏表示.

初始化阶段: 攻击者 A 承诺其将发起的注册查询次数, 说明每个查询是针对腐化密钥还是诚实密钥并准备挑战访问策略 *P**. 挑战者 C 运行 Setup() 算法生成 *crs* 并返回给攻击者 A.

查询阶段 1: 攻击者 A 在多项式时间内自适应地向挑战者 C 发起以下查询.

● 查询 1: 攻击者 A 选择用户 *i* 发起密钥查询. 挑战者 C 运行 SKeyGen() 和 PKeyGen() 算法获取相应的用户私钥 sk_i 和用户公钥 pk_i. 如果 *i* 在初始化时被声明为腐化, 则挑战者 C 将 sk_i 和 pk_i 返回给攻击者 A, 否则挑战者 C 只将 pk_i 返回给攻击者 A.

● 查询 2: 攻击者 A 选择用户 *i* 发起撤销查询. 挑战者 C 运行 Revoke() 算法撤销相应用户.

挑战阶段: 攻击者 A 准备 2 个消息 m_0, m_1, 挑战者 C 随机选择 $b\in\{0, 1\}$运行 Encrypt() 算法生成挑战密文 *ct** 并返回给攻击者 A.

查询阶段 2: 与查询阶段 1 相同, 但攻击者 A 禁止查询能解密挑战密文 *ct**的密钥.

猜测阶段: 攻击者 A 输出猜测结果 $b'\in\{0, 1\}$. 如果 $b=b'$, 则称攻击者 A 以 Adv=|Pr[$b=b'$]–1/2|的优势获胜.

定义 1. 如果不存在多项式时间的攻击者可以以不可忽略的优势赢得这个游戏, 那么本文方案具备选择明文攻击不可区分安全性.

4 方案设计

4.1 技术概述

目前大多数的注册属性加密方案都是基于 Hohenberger 等人在文献 [17] 中采用 l+1 个 sRABE 子方案并行的结构 (RABE 方案) 来实现的. 尽管该类方案通过巧妙的构造保证了有限次数的公钥聚合操作, 但依然对计算和存储开销有不小的要求. 本文考虑在单一 sRABE 框架下构建注册属性加密方案的可行性. 虽然 sRABE 方案与 RABE 方案有许多共同点, 包括用户数量存在上限, 支持属性加密操作等, 但由于 sRABE 方案要求先完成公钥聚合才能执行后续的加密等操作, 因此我们认为 sRABE 方案与 RABE 方案的主要差距在于用户管理的灵活性. 如果能在保持系统主公钥不变的情况下, 允许系统中的用户随时加入和退出系统, 那么即使是基于单 sRABE 方案也可以构建功能完备的注册属性加密方案.

在本文中, 我们基于变色龙哈希函数来实现这样的功能. 具体来说, 在系统构建时期, 我们并不需要所有用户都立刻加入系统, 而是按照设置的系统用户数量上限 L, 在系统中添加 L 个虚拟用户, 仅作为占位使用. 需要注意的是, 这些虚拟用户的公钥通过变色龙哈希函数生成, 其陷门信息由公钥聚合方持有. 随后, 公钥聚合方基于这些虚拟用户的公钥完成聚合生成系统主公钥. 当有新用户加入系统时, 只需逐一撤销虚拟用户并基于对应变色龙哈希函数的陷门为新用户构造既有用户公钥的碰撞. 我们采用公钥聚合方持有变色龙哈希函数陷门的设计, 是为了解决注册加密方案在用户撤销后公钥需要重新聚合导致的计算复杂、系统维护开销大的问题. 通过允许公钥聚合方针对被撤销的槽构造可控哈希碰撞, 系统可以在不重新聚合所有用户公钥的情况下, 为新用户生成一致性公钥, 从而显著降低聚合操作的计算成本. 在安全性方面, 变色龙哈希函数陷门并不用于用户私钥或密文解密的计算. 此外, 新用户注册过程中仍需根据其属性集合执行与单一槽相关的局部属性密钥更新, 但上述操作不涉及系统主公钥的重新聚合.

在用户撤销方面, 虽然我们采用状态标记的方法来对撤销用户进行标记, 但与采用直接撤销机制不同的是, 我们的标记并不是针对撤销用户, 而是针对注册加密方案中的槽进行标记, 因此只需维护一个常数规模的槽列表即可. 此外, 在安全性方面, 为了保证方案的后向安全性, 我们借鉴了外包解密的思想. 系统中的用户在解密前, 需要针对解密密文生成一个对应的辅助解密密钥, 因此即使撤销用户手中仍掌握着曾经的私钥, 也无法获得后续密文对应的辅助解密密钥, 从而保证无法解密后续密文. 公钥聚合方虽然能够生成辅助解密密钥, 但因不掌握用户私钥成分且不参与明文恢复, 其所持有的陷门与辅助解密能力均不会破坏系统的机密性与后向安全性. 此外, 辅助解密密钥的生成仅需要 2 次群上的指数运算和 1 次双线性映射计算, 因此该设计在保证安全性的同时保持了较高的计算效率.

4.2 基于变色龙哈希的可撤销注册属性加密方案介绍

(1) Setup(1^{λ}, $1^{|U|}$, 1^{L})$\rightarrow crs$

该算法由系统建立方执行. 算法的输入为安全参数 λ、属性空间 U 的最大属性数量以及系统最大用户数量 L (即注册加密方案中槽的数量). 系统建立方首先生成一个复合阶双线性群 $\Delta=(p_1, p_2, p_3, \mathbb{G}, \mathbb{G}_T, e, g)$. 令 $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_3$ 表示群 $\mathbb{G}$ 的子群, 其阶分别为 p_1, p_2, p_3, 令 $N=p_1p_2p_3$. 系统建立方随机选择 $g_1\in\mathbb{G}_1$, $g_3\in\mathbb{G}_3$, $\alpha, \beta\in\mathbb{Z}_N$, 计算 $h=g_1^{\beta}$, $Z=e(g_1, g_1)^{\alpha}$. 对于所有槽下标 $i\in[L]$, 系统建立方随机选择 $t_i, \delta_i, \tau_i\in\mathbb{Z}_N$, 其中 τ_i 为盲化因子. 系统建立方计算 CRS 中的槽组件如下:

$$A_i=(g_1g_3)^{t_i},\ B_i=g_1A_i^{\beta}g_3^{\tau_i},\ P_i=(g_1g_3)^{\delta_i}.$$

随后, 对于属性 $\omega\in U$ 以及槽下标 $i\in[L]$, 系统建立方随机选择 $u_{i,\omega}\in\mathbb{Z}_N$, 对任意 $j\in[L]$ 且 $j\neq i$, 随机选择盲化因子 $\gamma_{i,j,\omega}\in\mathbb{Z}_N$. 系统建立方计算 CRS 中的属性组件如下:

$$U_{i,\omega}=g_1^{u_{i,\omega}},\ W_{i,j,\omega}=A_i^{u_{j,\omega}}g_3^{\gamma_{i,j,\omega}}.$$

系统建立方输出公共参考字符串 crs:

$$crs=\left(\Delta,Z,g_1,h,g_3,\{A_i,B_i,P_i\}_{i\in[L]},\{U_{i,\omega},W_{i,j,\omega}\}_{i\neq j,\omega\in U}\right).$$

(2) SKeyGen(*crs*, *i*)→(*tpk*$_i$, *sk*$_i$)

该算法由数据用户执行. 算法输入为公共参考字符串 *crs* 和槽下标 *i*. 数据用户随机选择 r_i, $r_{i,1}$, $r_{i,2}\in\mathbb{Z}_N$, 令 $r_i=r_{i,1}r_{i,2}$ 并计算:

$$T'_i=g_1^{r_i},\ Q'_i=P_i^{r_{i,2}},\ R'_i=g_3^{r_{i,2}},\ F'_i=g_3^{r_{i,1}},\ E'_i=g_1^{r_{i,1}}.$$

对于所有 $j\neq i$, 数据用户计算交叉项 $V'_{j,i}=A_j^{r_{i,2}}$. 数据用户生成临时公钥 $tpk_i=(r_i, T'_i, Q'_i, R'_i, F'_i, E'_i, \{V'_{j,i}\}_{j\neq i})$, 秘密保存自己的私钥 $sk_i=(r_{i,1}, r_{i,2})$.

(3) PKeyGen(*crs*, *i, tpk*$_i$, *L*, *D*, *S*$_i$)→(*pk*$_i$, *D*) or ⊥

该算法由公钥聚合方执行. 算法输入为公共参考字符串 *crs*、一个用户对应的槽下标 *i* 及其临时公钥 *tpk*$_i$、系统用户数量上限 *L* 以及一个将槽下标映射到与用户 *i* 相关的身份信息的字典 *D* (初始为 (Null, 0, 0, 0, 0)). 公钥聚合方解析 $tpk_i=(r_i, T'_i, Q'_i, R'_i, \{V'_{j,i}\}_{j\neq i})$, 验证 *D*[*i*][0] 的值.

● 如果值为 False, 表示下标为 *i* 的槽已有用户注册, 算法直接输出⊥.

● 如果值为 Null, 表示下标为 *i* 的槽尚未被注册过, 算法继续验证临时公钥的合法性. 通过验证 $i\leqslant L$, $e(g_3, T'_i)=1=e(E'_i, R'_i)$, $e(g_3, F'_i)=1=e(g_1, E'_i)$, $e(T'_i, P_i)=e(E'_i, Q'_i)$, $e(R'_i, P_i)=e(g_3, Q'_i)$ 以及对于所有 $j\neq i$, $e(E'_i, V'_{j,i})=e(T'_i, A'_j)$, $e(g_3, V'_{j,i})=e(R'_i, A'_j)$. 如果上述任一验证不通过, 算法输出⊥. 公钥聚合方随机选择 *x*, $id\in\mathbb{Z}_p$, 计算 $Y_1=g_1^x$, $Y_{2,j}=A_j^x$, $T_i=Y_1^{id}T'_i$, $V_{j,i}=Y_{2,j}^{id}V'_{j,i}$, 公开用户 *i* 公钥 $pk_i=(T_i, \{V_{j,i}\}_{j\neq i})$. 同时, 公钥聚合方秘密更新 *D*[*i*]=(*revo*=False, *id*, *x*, r_i, 0).

● 如果值为 True, 表示下标为 *i* 的槽处于"空缺"状态, 算法继续验证临时公钥的合法性. 通过验证 $i\leqslant L$, $e(g_3, T'_i)=1=e(E'_i, R'_i)$, $e(g_3, F'_i)=1=e(g_1, E'_i)$, $e(T'_i, P_i)=e(E'_i, Q'_i)$, $e(R'_i, P_i)=e(g_3, Q'_i)$ 以及对于所有 $j\neq i$, $e(E'_i, V'_{j,i})=e(T'_i, A'_j)$, $e(g_3, V'_{j,i})=e(R'_i, A'_j)$. 如果上述任一验证不通过, 算法输出⊥. 公钥聚合方解析 *D*[*i*]=(*revo*=True, *id'*, *x*, r'_i, hsk_i), $hsk_i=(mpk, i, S'_i, A_i, B_i, \hat{V}_i, \{\hat{W}_{i,\omega}\}_{\omega\in U})$, $mpk=(\Delta, g_1, h, Z, \hat{T}, \{\hat{U}'_\omega\}_{\omega\in U})$, 选择 $id\in\mathbb{Z}_p$ 使得 $x\cdot id'+r'_i=x\cdot id+r_i$, 计算 $Y_1=g_1^x$, $Y_{2,j}=A_j^x$. 随后计算 $T_i=Y_1^{id}T'_i$, $V_{j,i}=Y_{2,j}^{id}V'_{j,i}$, $\forall j\in[L]$ 计算 $\hat{V}_i=\prod_{j\neq i}V_{i,j}$, 公开用户 *i* 公钥 $pk_i=(T_i, \{V_{j,i}\}_{j\neq i})$. 同时更新属性密钥, 对于 $\omega\in U$, 计算属性公钥 $\hat{U}_\omega=\prod_{j\in[L]:\ \omega\notin S_j}U_{j,\omega}$ 以及属性-槽相关辅助密钥 $\hat{W}_{i,\omega}=\prod_{j\neq i:\ \omega\notin S_j}W_{i,j,\omega}$, $\forall j\in[L]$. 算法更新 $mpk=(\Delta, g_1, h, Z, \hat{T}, \{\hat{U}_\omega\}_{\omega\in U})$, $hsk_i=(mpk, i, S_i, A_i, B_i, \hat{V}_i, \{\hat{W}_{i,\omega}\}_{\omega\in U})$, *D*[*i*]=(*revo*=False, *id*, *x*, r_i, hsk_i). 需要强调的是, 当 *D*[*i*][0]=True 时, PKeyGen() 算法中包含与 Aggregate() 算法结构相似的属性级更新操作, 但该操作仅限于单一注册槽.

(4) Aggregate(*crs*, (pk_1, S_1),…, (pk_L, S_L), *D*)→(*mpk*, *D*)

该算法由公钥聚合方执行. 算法输入为公共参考字符串 *crs* 和一系列公钥及对应属性集 (pk_1, S_1),…, (pk_L, S_L), 字典 *D*. 公钥聚合方计算公钥 $\hat{T}$ 以及槽相关辅助密钥 $\hat{V}_i$:

$$\hat{T}=\prod\nolimits_{j\in[L]}T_j,\ \forall i\in[L]\text{计算}\hat{V}_i=\prod\nolimits_{j\neq i}V_{i,j}.$$

在属性密钥方面, 对于任意 $\omega\in U$, 公钥聚合方计算属性公钥 $\hat{U}_\omega$ 以及属性-槽相关辅助密钥 $\hat{W}_{i,\omega}$:

$$\hat{U}_\omega=\prod\nolimits_{j\in[L]:\omega\notin S_j}U_{j,\omega},\ \forall i\in[L]\text{计算}\hat{W}_{i,\omega}=\prod\nolimits_{j\neq i:\omega\notin S_j}W_{i,j,\omega}.$$

最后, 公钥聚合方输出主公钥 *mpk*, 将辅助解密密钥 hsk_i 存储到字典 *D* 中:

$$mpk=\left(\Delta,g_1,h,Z,\hat{T},\{\hat{U}_\omega\}_{\omega\in U}\right),\ D[i][4]=hsk_i=\left(mpk,i,S_i,A_i,B_i,\hat{V}_i,\{\hat{W}_{i,\omega}\}_{\omega\in U}\right).$$

(5) Encrypt(*mpk*, (***M***, ρ), *m*)→*ct*

该算法由数据拥有方执行. 算法输入为主公钥 *mpk*, 访问策略 (***M***, ρ) 以及待加密的消息 *m*, 其中 ***M*** 是一个 $K\times N$ 的矩阵, $\rho:[K]\to U$ 是一个将矩阵行映射到属性的映射. 数据拥有方选择随机数 $s, v_2,\ldots,v_n\in\mathbb{Z}_N$, h_1, $h_2\in\mathbb{G}_1$, 使得 $h=h_1h_2$, 令 *s* 为秘密值, $\boldsymbol{M}_k$ 为矩阵 ***M*** 的第 *k* 行, 设置向量 $\vec{v}=[s, v_2,\ldots, v_n]^{\mathrm{T}}$, 对任意 $k\in[K]$, 计算 $\lambda_k=\boldsymbol{M}_k\times\vec{v}$. 数据拥有方计算 $C_1=m\cdot Z^s$, $C_2=g_1^s$, 对任意 $k\in[K]$, $C_{3,k}=h_2^{\lambda_k}\hat{U}_{\rho(k)}^{-s}$, $C_4=h_1^s\hat{T}^{-s}$. 最终, 数据拥有方生成密文:

$$ct=\left((\boldsymbol{M},\rho),C_1,C_2,\{C_{3,k}\}_{k\in[K]},C_4\right).$$

(6) HSKGen(i, D, ct)→hsk

该算法由公钥聚合方执行. 算法输入为用户对应的槽下标 i, 字典 D, 密文 ct. 当收到数据用户的辅助密钥申请请求后, 公钥聚合方解析 $D[i]$=($revo$=False, id, x, r_i, hsk_i), ct=(($\boldsymbol{M}$, ρ), C_1, C_2, $\{C_{3,k}\}_{k\in[K]}$, C_4), 其中 hsk_i=(mpk, i, S_i, A_i, B_i, $\hat{V}_i$, $\{\hat{W}_{i,\omega}\}_{\omega\in U}$). 然后, 公钥聚合方验证 $D[i][0]$ 是否为 False, 如果不是, 则直接输出 $\perp$. 否则, 公钥聚合方计算 $tct_1=A_i^{-x\cdot id}$, $tct_2=e(C_2,A_i^{r_i}V_i)$, 生成辅助解密密钥并将其作为结果返回给数据用户.

$$hsk=\left(mpk,i,S_i,A_i,B_i,\{\hat{W}_{i,\omega}\}_{\omega\in U},tct_1,tct_2\right).$$

(7) Decrypt(sk, hsk, ct)→res or $\perp$

该算法由数据用户执行. 算法输入为用户的私钥 sk, 辅助解密密钥 hsk 以及密文 ct. 数据用户解析 sk=($r_{i,1}$, $r_{i,2}$), hsk=(mpk, i, S_i, A_i, B_i, $\hat{V}_i$, $\{\hat{W}_{i,\omega}\}_{\omega\in U}$, tct_1, tct_2), mpk=(Δ, g_1, h, Z, $\hat{T}$, $\{\hat{U}_\omega\}_{\omega\in U}$), ct=(($\boldsymbol{M}$, ρ), C_1, C_2, $\{C_{3,k}\}_{k\in[K]}$, C_4). 随后, 数据用户验证自身属性集合 S_i 是否满足访问策略 ($\boldsymbol{M}$, ρ), 如果不满足则算法直接输出 $\perp$. 否则, 令 I=$\{k\in[K]:\rho(k)\in S_i\}$ 表示矩阵 $\boldsymbol{M}$ 中与属性集 S_i 相关联的矩阵行集合, 记为 I=$\{k_1,\ldots,k_{|I|}\}$, 同时抽取这些矩阵行组成一个新矩阵 $\boldsymbol{M}_{S_i}$. 数据用户生成一个常数向量 $\boldsymbol{c}_{S_i}\in\mathbb{Z}_N^{|I|}$ 使得 $\boldsymbol{c}_{S_i}\boldsymbol{M}_{S_i}$=(1, 0,…, 0). 随后, 数据用户计算:

$$D_{\text{slot}}=e(C_4,A_i)\cdot tct_2,\ D_{\text{attr}}=\prod\nolimits_{1\leqslant j\leqslant|I|}\left(e\left(C_{3,k_j},A_i\right)\cdot e\left(C_2,W_{i,\rho(k_j)}\right)\right)^{c_{S_i,S_{i,j}}},\ res=C_1\cdot D_{\text{slot}}\cdot D_{\text{attr}}/e\left(C_2,B_i\cdot tct_1\cdot g_1^{-r_{i,1}}\right).$$

(8) Revoke(i, D)→D

该算法由公钥聚合方执行. 算法输入为待撤销用户对应的槽下标 i、字典 D. 在收到数据拥有方的撤销请求后, 公钥聚合方随机选择 $id'\in\mathbb{Z}_N$ 并更新 $D[i][0]$=($revo$=True), $D[i][1]=id'$.

5 方案分析

5.1 选择明文攻击安全性分析

在本节中, 我们证明本文方案在选择明文攻击下是安全的. 该安全游戏由挑战者 C 和攻击者 A 之间的互动完成. 我们在证明中引入了半功能密文和半功能槽, 注册到半功能槽的密钥可以用来解密正常密文, 注册到正常槽的密钥可以用来解密半功能密文, 而注册到半功能槽的密钥无法解密半功能密文. 与 sRABE[17]类似, 我们逐渐将挑战密文与槽参数替换为半功能的版本. 由于注册在半功能槽的密钥无法解密半功能密文, 因此在最终的游戏中可以证明本文方案的安全性.

(1) Game$_0$: 在这个游戏中, 挑战者加密挑战消息 m_b^*, 其主要步骤如下.

初始化阶段: 攻击者 A 承诺其将发起的注册查询次数, 说明每个查询是针对腐化密钥还是诚实密钥, 准备挑战访问策略 P^*=($\boldsymbol{M}$, ρ). 挑战者维护一个字典 D 并遵循 Setup() 算法首先生成复合阶双线性群 Δ=(p_1, p_2, p_3, $\mathbb{G}$, $\mathbb{G}_T$, e, g). 令 $N=p_1p_2p_3$, 算法随机选择 $g_1\in\mathbb{G}_1$, $g_3\in\mathbb{G}_3$, α, $\beta\in\mathbb{Z}_N$, 计算 $h=g_1^\beta$, $Z=e(g_1,g_1)^\alpha$. 对于 $i\in[L]$, 算法随机选择 t_i, δ_i, $\tau_i\in\mathbb{Z}_N$. 算法槽组件 $A_i=(g_1g_3)^{t_i}$, $B_i=g_1A_i^\beta g_3^{\tau_i}$, $P_i=(g_1g_3)^{\delta_i}$. 随后, 对属性 $\omega\in U$ 以及槽下标 $i\in[L]$, 随机选择 $u_{i,\omega}\in\mathbb{Z}_N$, 对 $j\in[L]$ 且 $j\neq i$, 选择盲化因子 $\gamma_{i,j,\omega}\in\mathbb{Z}_N$ 并计算属性组件 $U_{i,\omega}=g_1^{u_{i,\omega}}$, $W_{i,j,\omega}=A_i^{u_{j,\omega}}g_3^{\gamma_{i,j,\omega}}$. 最终输出公共参考字符串 crs=(Δ, Z, g_1, h, g_3, $\{A_i,B_i,P_i\}_{i\in[L]}$, $\{U_{i,\omega},W_{i,j,\omega}\}_{i\neq j,\omega\in U}$).

查询阶段 1: 攻击者 A 向挑战者发起以下查询.

查询 1: 攻击者 A 选择用户 i 发起密钥查询. 挑战者执行 SKeyGen() 算法, 随机选择 r_i, $r_{i,1}$, $r_{i,2}\in\mathbb{Z}_N$, 令 $r_i=r_{i,1}r_{i,2}$ 并计算 $T_i'=g_1^{r_i}$, $Q_i'=P_i^{r_{i,2}}$, $R_i'=g_3^{r_{i,2}}$, $F_i'=g_3^{r_{i,1}}$, $E_i'=g_1^{r_{i,1}}$. 对于所有 $j\neq i$, 计算交叉项 $V_{j,i}'=A_j^{r_{i,2}}$ 并生成临时公钥 tpk_i=(r_i, T_i', Q_i', R_i', $\{V_{j,i}'\}_{j\neq i}$), 和对应私钥 sk_i=($r_{i,1}$, $r_{i,2}$), 随后挑战者验证 $D[i][0]$.

● 如果值为 False, 表示下标为 i 的槽已有用户注册, 直接输出 $\perp$.

● 如果值为 Null, 表示下标为 i 的槽尚未被注册过, 继续验证临时公钥的合法性. 随后挑战者选择 x, $id\in\mathbb{Z}_N$, 计算 $Y_1=g_1^x$, $Y_{2,j}=A_j^x$, $T_i=Y_1^{id}T_i'$, $V_{j,i}=Y_{2,j}^{id}V_{j,i}'$, 令 pk_i=(T_i, $\{V_{j,i}\}_{j\neq i}$). 挑战者更新 $D[i]$=(False, id, x, r_i, 0, pk_i, sk_i).

● 如果值为 True, 表示下标为 i 的槽处于“空缺”状态, 挑战者验证临时公钥的合法性. 挑战者选择 $id\in\mathbb{Z}_N$ 使得 $x\cdot id'+r_i'=x\cdot id+r_i$, 计算 $Y_1=g_1^x$, $Y_{2,j}=A_j^x$. 随后计算 $T_i=Y_1^{id}T_i'$, $V_{j,i}=Y_{2,j}^{id}V_{j,i}'$, $\forall i\in[L]$, 计算 $\hat{V}_i=\prod_{j\neq i}V_{i,j}$, 令 pk_i=(T_i, $\{V_{j,i}\}_{j\neq i}$).

对于 $\omega\in U$, 计算 $\hat{U}_\omega=\prod_{j\in[L]:\,\omega\notin S_j}U_{j,\omega}$ 以及 $\hat{W}_{i,\omega}=\prod_{j\neq i:\,\omega\notin S_j}W_{i,j,\omega}$, $\forall\, i\in[L]$. 挑战者更新 $mpk=(\Delta, g_1, h, Z, \hat{T}, \{\hat{U}_\omega\}_{\omega\in U})$, $hsk_i=(mpk, i, S_i, A_i, B_i, \hat{V}_i, \{\hat{W}_{i,\omega}\}_{\omega\in U})$, $D[i]=(\text{True}, id', x, r'_i, hsk_i\, pk_i, sk_i)$.

如果 i 在初始化时被声明为腐化, 则挑战者 C 将 $sk_i=(r_{i,1}, r_{i,2})$ 和 $pk_i=(T_i, \{V_{j,i}\}_{j\neq i})$ 返回给攻击者 A, 否则挑战者 C 只将 $pk_i=(T_i, \{V_{j,i}\}_{j\neq i})$ 返回给攻击者 A.

查询 2: 攻击者 A 选择用户 i 发起撤销查询. 挑战者运行 Revoke() 算法, 随机选择 $id'\in\mathbb{Z}_N$ 并更新 $D[i][0]=(revo=\text{True})$, $D[i][1]=id'$.

挑战阶段: 攻击者准备 2 个消息 $m_0, m_1\in\mathbb{G}_T$, 对于 $i\in[L]$, 准备 (c_i, S_i, pk_i^*), 挑战者运行 Encrypt() 算法, 选择随机数 $s, v_2,\ldots, v_n\in\mathbb{Z}_N$, $h_1, h_2\in\mathbb{G}_1$, 使得 $h=h_1h_2$, 设置向量 $\vec{v}=[s, v_2,\ldots, v_n]^{\mathrm{T}}$, 对 $k\in[K]$, 计算 $\lambda_k=\boldsymbol{M}_k\times\vec{v}$. 挑战者计算 $C_1=m_b\cdot Z^s$, $C_2=g_1^s$, $C_{3,k}=h_2^{\lambda_k}\hat{U}_{\rho(k)}^{-s}$, $C_4=h_1^s\hat{T}^{-s}$. 最终生成挑战密文 $ct^*=((\boldsymbol{M}, \rho), C_1, C_2, \{C_{3,k}\}_{k\in[K]}, C_4)$.

查询阶段 2: 与查询阶段 1 相同, 但攻击者 A 禁止查询能解密挑战密文 ct^*的密钥.

猜测阶段: 攻击者 A 输出一个猜测结果 $b'\in\{0, 1\}$.

此时, 攻击者在 Game_0 中的优势为 $\text{Adv}(\text{Game}_0)=|\Pr[b'=b]-1/2|$.

(2) Game_1: 与 Game_0 基本相同, 但在下列阶段略有修改.

初始化阶段: 挑战者选择 $\beta_1, \beta_2\in\mathbb{Z}_N$ 并设置 $\beta=\beta_1+\beta_2$. 与 Game_0 相同, 我们同样令 $h=g_1^\beta$. 此外, 我们将选择秘密值 $s\in\mathbb{Z}_N$ 的时间从挑战阶段提前到了初始化阶段并设置 $P_i=(g_1^sg_3)^{\delta_i}$.

挑战阶段: 在生成挑战密文时, 挑战者令 $h_1=g_1^{\beta_1}$, $h_2=g_1^{\beta_2}$. 选择随机数 $v_2,\ldots, v_n\in\mathbb{Z}_N$, 设置向量 $\vec{v}'=[1, v_2,\ldots, v_n]^{\mathrm{T}}$, 对 $k\in[K]$, $C_{3,k}=(g_1{}^s)^{\beta_2m_k^{\mathrm{T}}\vec{v}'-\sum_{i\in[L]:\,\rho(k)\notin S_i}u_{i,\rho(k)}}$, $C_4=(g_1{}^s)^{\beta_1}\left(\prod_{i\in[L]}\frac{R_i}{Q_i^{\delta_i^{-1}}}\right)$.

对于多项式时间攻击者 A, 其获得的公钥如果合法必然来源于 PKeyGen() 算法的合法输出. 对于 $i\in[L]$, 我们可以得到 $T_i=g_1^{r_i}$, $Q_i=P_i^{r_{i,2}}$, $R_i=g_3^{r_{i,2}}$, $F_i=g_3^{r_{i,1}}$, $E_i=g_1^{r_{i,1}}$, $V_{j,i}=A_j^{r_{i,2}}$. 那么在 Game_0 和 Game_1 中, 下列等式成立:

$$\hat{T}=\prod\nolimits_{i\in[L]}T_i=\prod\nolimits_{i\in[L]}g_1{}^{r_i},\ \hat{U}_\omega=\prod\nolimits_{i\in[L]:\,\rho(k)\notin S_i}U_{i,\rho(k)\notin S_i}=g^{\sum_{i\in[L]:\,\rho(k)\notin S_i}u_{i,\rho(k)}}.$$

在两个游戏中, h, h_1, h_2 在 $\mathbb{G}_1$ 上分布均匀且 $h=h_1h_2$. 同时在 Game_1 中, $\beta_1, \beta_2\in\mathbb{Z}_N$ 并且 $\beta=\beta_1+\beta_2$ 在 $\mathbb{Z}_N$ 上分布均匀. 因此, β 的分布与 Game_0 相同.

对于 Game_0 和 Game_1 中 P_i 的分布, 在 Game_0 中:

$$P_i=(g_1g_3)^{\delta_i}=g_1^{\delta_i \bmod p_1}g_3^{\delta_i \bmod p_3}.$$

由于 δ_i 在 $\mathbb{Z}_N$ 上分布均匀, 那么根据中国剩余定理, $\delta_i \bmod p_1$ 和 $\delta_i \bmod p_3$ 也在 $\mathbb{Z}_N$ 上互相独立且分布均匀.

在 Game_1 中:

$$P_i=(g_1^sg_3)^{\delta_i}=g_1^{s\delta_i \bmod p_1}g_3^{\delta_i \bmod p_3}.$$

由于 δ_i 在 $\mathbb{Z}_N$ 上依然分布均匀, 那么只要 $s\neq 0 \bmod p_1$, $s\delta_i \bmod p_1$ 也在 $\mathbb{Z}_N$ 上分布均匀. 因此 P_i 在 Game_0 和 Game_1 中的分布在统计上也同样是不可区分的.

对于 Game_1 中属性相关挑战密文, 令 $k\in[K]$:

$$C_{3,k}=(g_1^s)^{\beta_2m_k^{\mathrm{T}}\vec{v}'-\sum_{i\in[L]:\,\rho(k)\notin S_i}u_{i,\rho(k)}}=h_2^{sm_k^{\mathrm{T}}\vec{v}'}\hat{U}_{\rho(k)}^{-s}=h_2^{m_k^{\mathrm{T}}\vec{v}''}\hat{U}_{\rho(k)}^{-s},$$

其中, $\vec{v}''=s\vec{v}'=[s, sv_2,\ldots, sv_n]^{\mathrm{T}}$. 这与 Game_0 中的分布相一致.

对于 Game_1 中属性相关挑战密文:

$$\frac{Q_i^{\delta_i^{-1}}}{R_i}=\frac{P_i^{r_i\delta_r^{-1}}}{g_3^{r_i}}=\frac{g_1^{sr_i}g_3^{r_i}}{g_3^{r_i}}=g_1^{sr_i},\ C_4=(g_1^s)^{\beta_1}\left(\prod\nolimits_{i\in[L]}\frac{R_i}{Q_i^{\delta_i^{-1}}}\right)=h_1^s\prod\nolimits_{i\in[L]}g_1^{-sr_i}=h_1^s\hat{T}^{-s},$$

因此, C_4 的分布与 Game_0 中的分布相一致. 此时, 攻击者在 Game_1 中的优势满足$|\text{Adv}(\text{Game}_0)-\text{Adv}(\text{Game}_1)|\leqslant negl(\lambda)$.

(3) Game_2: 与 Game_1 的区别在于挑战密文被替换为半功能密文, 同时我们修改了 P_i 与挑战密文的构造:

$$P_i=((g_1g_2)^sg_3)^{\delta_i}.$$

令 $T=g_1^s$ 或 $(g_1g_2)^s$, $s\in\mathbb{Z}_N$, 挑战者随机选择 $\alpha, \beta_1, \beta_2\in\mathbb{Z}_N$, 设置 $\beta=\beta_1+\beta_2$, 计算 $h=g_1^\beta$, $Z=e(g_1, g_1)^\alpha$. 对于 $i\in[L]$, 随

机选择 t_i, δ_i, $\tau_i\in\mathbb{Z}_N$. 随后计算槽组件 $A_i=(g_1g_3)^{t_i}$, $B_i=g_1^{\alpha}A_i^{\beta}g_3^{\tau_i}$, $P_i=(Tg_3)^{\delta_i}$. 随后, 对属性 $\omega\in U$ 以及槽下标 $i\in[L]$, 随机选择 $u_{i,\omega}\in\mathbb{Z}_N$, 对 $j\in[L]$ 且 $j\neq i$, 选择盲化因子 $\gamma_{i,j,\omega}\in\mathbb{Z}_N$ 并计算属性组件 $U_{i,\omega}=g_1^{u_{i,\omega}}$, $W_{i,j,\omega}=A_i^{u_{j,\omega}}g_3^{\gamma_{i,j,\omega}}$. 最终输出公共参考字符串:

$$crs=\left(\Delta,Z,g_1,h,g_3,\{A_i,B_i,P_i\}_{i\in[L]},\{U_{i,\omega},W_{i,j,\omega}\}_{i\neq j,\omega\in U}\right).$$

在查询阶段, 挑战者 C 按照 Game_1 响应攻击者的查询, 如果 i 被腐化, 返回公钥 $pk_i=(T_i,\{V_{j,i}\}_{j\neq i})$ 和私钥 $sk_i=(r_{i,1},r_{i,2})$, 否则返回 $pk_i=(T_i,\{V_{j,i}\}_{j\neq i})$.

在挑战阶段, 挑战者生成密文 $C_1=m_b\cdot e(g_1,T)^s$, $C_2=T$. 选择随机数 $v_2,\ldots,v_n\in\mathbb{Z}_N$, 设置向量 $\vec{v}'=[1,v_2,\ldots,v_n]^{\mathrm{T}}$, 对 $k\in[K]$:

$$C_{3,k}=T^{\beta_2 m_k^{\mathrm{T}}\vec{v}'-\sum_{i\in[L]:\,\rho(k)\notin S_i}u_{i,\rho(k)}},\ C_4=T^{\beta_1}\left(\prod\nolimits_{i\in[L]}\frac{R_i}{Q_i^{\delta_i^{-1}}}\right).$$

由于 $e(g_1,T)^s=e(g_1,g_1)^{\alpha s}$, 无论 $T=g_1^s$ 或 $(g_1g_2)^s$, 如果 $T=g_1^s$, 则与 Game_1 完全相同, 如果 $T=(g_1g_2)^s$, 则与 Game_2 完全相同. 此时, 攻击者在 Game_2 中的优势满足$|\text{Adv}(\text{Game}_1)-\text{Adv}(\text{Game}_2)|\leqslant negl(\lambda)$.

(4) Game_3: 与 Game_2 的区别在于槽 l 被替换为半功能槽, 即在系统建立阶段, 挑战者按如下构造生成槽参数:

$$A_l=(g_1g_3)^{t_l},\ B_l=g_1^{\alpha}A_l^{\beta}(g_2g_3)^{\tau_l},\ P_l=((g_1g_2)^s g_3)^{\delta_l}.$$

在 Game_3 开始时, 挑战者接受挑战 (Δ,g_1,g_3,X,Y,T), 其中 $X=(g_1g_2)^{s_{12}}$, $Y=(g_2g_3)^{s_{23}}$, $s_{12},s_{23}\in\mathbb{Z}_N$, $T=g_1^s$ 或 $(g_1g_2)^s$. 挑战者 C 随机选择 $\alpha,\beta_1,\beta_2\in\mathbb{Z}_N$, 设置 $\beta=\beta_1+\beta_2$, 计算 $h=g_1^{\beta}$, $Z=e(g_1,g_1)^{\alpha}$. 对于 $i\in[L]$, 随机选择 t_i, δ_i, $\tau_i\in\mathbb{Z}_N$. 如果 $i<l$, 挑战者设置 $A_i=(g_1g_3)^{t_i}$, $B_i=g_1^{\alpha}A_i^{\beta}Y^{\tau_i}$, $P_i=(Xg_3)^{\delta_i}$. 如果 $i=l$, 挑战者设置 $A_l=T$, $B_l=g_1^{\alpha}A_l^{\beta}g_3^{\tau_l}$, $P_l=(Xg_3)^{\delta_l}$. 如果 $i>l$, 挑战者设置 $A_i=(g_1g_3)^{t_i}$, $B_i=g_1^{\alpha}A_l^{\beta}g_3^{\tau_i}$, $P_i=(Xg_3)^{\delta i}$.

随后, 对属性 $\omega\in U$ 以及槽下标 $i\in[L]$, 随机选择 $u_{i,\omega}\in\mathbb{Z}_N$, 对 $j\in[L]$ 且 $j\neq i$, 选择盲化因子 $\gamma_{i,j,\omega}\in\mathbb{Z}_N$ 并计算属性组件 $U_{i,\omega}=g_1^{u_{i,\omega}}$, $W_{i,j,\omega}=A_i^{u_{j,\omega}}g_3^{\gamma_{i,j,\omega}}$. 最终输出公共参考字符串:

$$crs=\left(\Delta,Z,g_1,h,g_3,\{A_i,B_i,P_i\}_{i\in[L]},\{U_{i,\omega},W_{i,j,\omega}\}_{i\neq j,\omega\in U}\right).$$

在查询阶段, 挑战者 C 按照 Game_2 响应攻击者的查询, 如果 i 被腐化, 返回公钥 $pk_i=(T_i,\{V_{j,i}\}_{j\neq i})$ 和私钥 $sk_i=(r_{i,1},r_{i,2})$, 否则返回 $pk_i=(T_i,\{V_{j,i}\}_{j\neq i})$.

在挑战阶段, 挑战者生成密文 $C_1=m_b\cdot e(g_1,X)^s$, $C_2=X$. 选择随机数 $v_2,\ldots,v_n\in\mathbb{Z}_N$, 设置向量 $\vec{v}'=[1,v_2,\ldots,v_n]^{\mathrm{T}}$, 对 $k\in[K]$:

$$C_{3,k}=X^{\beta_2 m_k^{\mathrm{T}}\vec{v}'-\sum_{i\in[L]:\,\rho(k)\notin S_i}u_{i,\rho(k)}},\ C_4=X^{\beta_1}\left(\prod\nolimits_{i\in[L]}\frac{R_i}{Q_i^{\delta_i^{-1}}}\right).$$

由于 $s_{12}\in\mathbb{Z}_N$ 对应于 Game_2 中的 s, 即 $C_1=m_b\cdot e(g_1,(g_1g_2)^{s_{12}})^{\alpha}=m_b\cdot e(g_1,g_1)^{\alpha s_{12}}=Z^{s_{12}}$, 与 Game_2 中相符, 再考虑 B_i 的分布, 只要 $s_{23}\neq 0 \bmod p_2$ 且 $s_{23}\neq 0 \bmod p_3$, 则 $\{Y^{\tau_i}=(g_2g_3)^{s_{23}\tau_i},\tau_i\in\mathbb{Z}_N\}$ 与 $\{(g_2g_3)^{\tau_i},\tau_i\in\mathbb{Z}_N\}$ 的分布必然相同.

此时, 对于 $t\in\mathbb{Z}_N$, 如果 $T=(g_1g_3)^t$, 则 A_i 的分布与 Game_3 一致. 如果 $T=g_1^t$, 则 A_i 的分布与 Game_2 一致. 那么, 攻击者在 Game_3 中的优势满足$|\text{Adv}(\text{Game}_2)-\text{Adv}(\text{Game}_3)|\leqslant negl(\lambda)$.

(5) Game_4: 与 Game_3 的区别在于挑战者构造挑战密文 ct^*时, 随机选择 $C_1\in\mathbb{G}_T$, 即密文的分布与消息无关.

现在, 对于攻击者来说, 其获胜的优势为 $\text{Adv}(\text{Game}_4)=0$.

综上, 攻击者 A 的优势上界为:

$$\text{Adv}(\text{Game}_0)\leqslant \text{neg1}(\lambda)+\text{Adv}(\text{Game}_1)+\text{Adv}(\text{Game}_2)+\text{Adv}(\text{Game}_3)+\text{Adv}(\text{Game}_4).$$

由于 $\text{Adv}(\text{Game}_4)=0$ 且不同游戏之间的区别是可忽略的, 因此可以得出攻击者的整体优势同样是可忽略的, 即本文方案在不可区分模型下可以抵抗选择明文攻击.

5.2 公钥聚合方安全性分析

在本文方案中, 公钥聚合方承担了 2 项额外的系统功能: 一是持有变色龙哈希函数的陷门以便在用户撤销后

为新用户生成与原公钥一致的碰撞; 二是负责在解密阶段辅助解密密钥生成. 尽管公钥聚合方掌握一定的系统秘密信息, 但这并不会削弱系统的安全性. 下面从机密性、可撤销性与后向安全性这 3 个方面展开分析.

首先在机密性方面, 公钥聚合方所持有的变色龙哈希陷门仅用于在哈希空间内构造公钥碰撞, 不涉及任何属性密钥或明文恢复操作. 由于碰撞公钥仅改变哈希输入, 完全不涉及密钥生成算法, 因此公钥聚合方即便能够为新用户生成有效公钥, 也无法从中推导出任何用户私钥成分. 此外, 用户私钥由用户自己独立生成, 公钥聚合方在该过程中不参与任何秘密信息的分发与计算, 因此无法获取对密文的解密能力.

其次在可撤销性方面, 公钥聚合方的陷门能力仅用于针对“撤销槽”进行可控碰撞构造, 不会影响未撤销用户的密钥安全. 由于系统通过状态标记机制管理槽状态, 公钥聚合方只能根据系统授权对特定槽执行更新操作, 无法绕过撤销标记以恢复被撤销用户的密钥有效性. 该设计保证了撤销操作的可验证性与可控性, 从而维持系统在动态用户管理场景下的安全.

最后在后向安全性方面, 我们借鉴外包解密的思想使得每次解密都依赖于由公钥聚合方动态生成的辅助解密密钥. 尽管公钥聚合方可以访问该密钥, 但其仅能针对特定密文和用户身份计算得到, 并且最终的明文恢复仍需用户私钥参与. 也就是说, 公钥聚合方无法单独利用辅助解密密钥推导出明文, 而被撤销的用户即便保留曾经的私钥, 也无法获取新的辅助解密密钥, 因此无法解密后续密文, 从而有效保证了系统的后向安全性.

综上所述, 公钥聚合方在方案中所持有的陷门与辅助解密能力均受到严格的安全边界约束, 既不具备推导任何用户私钥或明文的能力, 也无法破坏用户撤销与后向安全的性质. 即使变色龙哈希函数的陷门被泄露, 攻击者也只能构造公钥层面的哈希碰撞, 无法推导任何用户私钥成分或恢复密文内容, 因此不会破坏方案在选择明文攻击模型下的机密性. 此外, 公钥聚合方的相关操作均基于公开算法执行, 其输出结果可以被系统中其他实体验证, 从而不影响注册属性加密方案对公钥聚合方透明性与可审计性的基本要求.

5.3 功能对比分析

在本节中, 我们将本文方案与文献 [17,29] 中的方案进行对比, 其中文献 [17] 的两个方案 (sRABE 和 RABE) 分别实现了注册加密和注册属性加密的功能且 sRABE 是本文方案构造的基础方案, 而文献 [29] 中虽然也同样设计了两个方案, 但 DRABE 方案与本文方案类似, 采用密钥更新的方法实现用户撤销, 因此我们选择这些方案进行对比. 首先, 我们在表 2 中列出本文方案与对比方案的一些关键功能信息. 本文方案与 sRABE、RABE、DRABE 都具备注册加密的特性, 且需在构造系统时指定用户数量上限和属性空间, 但只有本文方案与 DRABE 方案支持用户撤销. 本文方案与 DRABE 方案的不同点在于, DRABE 方案的构造依赖于对多个 sRABE 方案的并行使用, 由此带来了较大规模的计算和存储开销, 而本文通过设计, 基于单个 sRABE 方案实现了注册加密与属性加密方案的结合, 允许用户灵活地退出、加入系统.

表 2 方案功能对比

方案	注册加密	可撤销	用户数量	属性空间	子方案 (sRABE) 数量
sRABE[17]	√	不支持	有上限	固定	1
RABE[17]	√	不支持	有上限	固定	l+1
DRABE[29]	√	支持	有上限	固定	l+1
本文方案	√	支持	有上限	固定	1

5.4 新用户加入的计算开销分析

在基于变色龙哈希的可撤销注册属性加密方案中, 新用户加入系统时的计算开销主要来源于 PKeyGen() 算法以及与用户属性相关的局部密钥生成操作. Aggregate() 算法由于仅在系统初始运行时执行一次, 与新用户加入时的计算开销无关. 与基于多个 sRABE 子方案构建的注册属性加密方案不同, 本文方案在用户注册过程中无需重新生成或重新聚合系统主公钥, 而是将计算限制在单一注册槽内, 从而显著降低了动态用户管理场景下的整体开销. 具体而言, 当新用户加入系统时:

(1) 如果对应槽的状态标记 $D[i][0]$=Null, 则 PKeyGen() 算法仅需利用变色龙哈希函数构造一次公钥碰撞, 以生成与既有聚合结构相容的用户公钥, 该过程仅涉及常数次群指数运算.

(2) 如果 $D[i][0]$=False, 系统输出 $\perp$.

(3) 如果 $D[i][0]$=True, 系统在利用变色龙哈希函数构造公钥碰撞的基础上, 还需执行一次针对该槽的属性密钥更新操作. 该操作在形式上等价于 Aggregate() 算法中的属性密钥计算子步骤, 但其作用范围仅限于单一注册槽及当前用户的属性集合, 不涉及系统主公钥的重新聚合或对其他槽的遍历. 因此, 新用户加入系统时的整体计算复杂度与系统中已有用户的数量无关. 与基于多个 sRABE 子方案的注册属性加密方案在用户动态更新时需要执行多次公钥聚合操作不同, 本文方案将用户注册和撤销过程中的计算限制在单一注册槽内, 从而避免了聚合计算开销随子方案数量增长的问题.

为验证本文方案在新用户加入阶段的效率优势, 下面我们将对新用户注册过程中的计算开销进行仿真实验分析. 实验主要统计群乘法运算、群指数运算和双线性对运算的次数, 并与文献 [29] 中的方案进行对比. 令 L 表示系统中的用户数量, $|S_i|$表示用户 i 拥有的属性数量, 对比结果如表 3 所示. 结果表明, 在固定用户属性数量的情况下, 文献 [29] 方案中新用户加入的计算开销随系统用户规模增长, 而本文方案中新用户加入的计算开销仅与用户属性数量相关, 与系统中已有用户数量无关, 证明了本文方案在动态用户管理场景下的效率优势.

表 3　新用户注册计算开销统计

方案	群乘法运算次数	群指数运算次数	双线性对运算次数
文献[29]	$\log L(4L)$	0	$\log L(6+4\|S_i\|)$
本文方案	$2+2L$	$2+2\|S_i\|$	$8+4\|S_i\|$

综上所述, 本文方案通过将用户注册阶段的计算开销限制在单一槽和单一 sRABE 框架内, 在保证功能完备性的同时, 提升了系统在用户变更场景下的可扩展性与实用性.

5.5 模拟结果及分析

在本节中, 我们将本文方案与文献 [17] 中的 sRABE 方案以及文献 [29] 中的 DRABE 方案进行对比, 实验基于 JPBC[34]实现, 其中双线性映射采用 JPBC 中的默认曲线 Type A1. 实验的平台为一台 Macbook Pro 笔记本电脑, 处理器为 Intel Core i7 (2.7 GHz), 内存为 16 GB. 此外, 本文方案中属性加密方案使用的访问策略通过 AND 相互连接, 实验中采用毫秒作为运行时间的统计单位. 在系统建立阶段, 系统中用户数量固定, 槽数量为 2、4、8、16、32. 在加密阶段和解密阶段, 用户属性集中的数量和访问策略中属性的数量按照 20、30、…、80 的规律递增. 在撤销阶段, 由于本文方案与文献 [29] 中 DRABE 方案的撤销操作均只需常数次操作即可完成, 因此我们还额外统计了撤销后补充相同数量用户所需的时间开销.

图 2 展示了 Setup()、Encrypt()、Decrypt()、Revoke() 算法的时间开销对比情况, 其中图 2(a) 是系统建立运行时间对比, 图 2(b) 是加密运行时间对比, 图 2(c) 是解密运行时间对比, 图 2(d) 是用户撤销及加入运行时间对比. 相比于文献 [29], 本文方案在 Setup()、Encrypt()、Revoke() 算法中的时间开销均有明显的优势, 而在 Decrypt() 算法中的时间开销与文献 [29] 中的 DRABE 方案基本一致. 这是由于 DRABE 方案的构建依赖于 l+1 个 sRABE 子方案, 在运行时相当于同时运行了 l+1 个 sRABE 方案, 因此在系统建立、加密、撤销及用户加入等算法中的时间开销较大, 而本文方案是在单个 sRABE 方案的基础上构建的, 因此随着系统中槽数量上限与属性数量的增加, 优势将愈加明显. 在撤销机制方面, 文献 [29] 中的 DRABE 方案在加入用户达到特定数量时, 需要重新执行聚合算法更新系统主公钥, 而本文方案基于变色龙哈希函数易于构造碰撞的能力, 确保新加入的用户同样可以使用之前用户聚合得到的系统主公钥, 避免了多次的聚合操作, 仅需对新用户的公钥进行更新即可, 因此计算上更具优势. 与文献 [17] 中的 sRABE 方案相比, 本文方案在 Setup()、Encrypt() 算法中的时间开销基本相同, 然而本文方案还额外实现了用户撤销的功能, 因此在功能上具备优势.

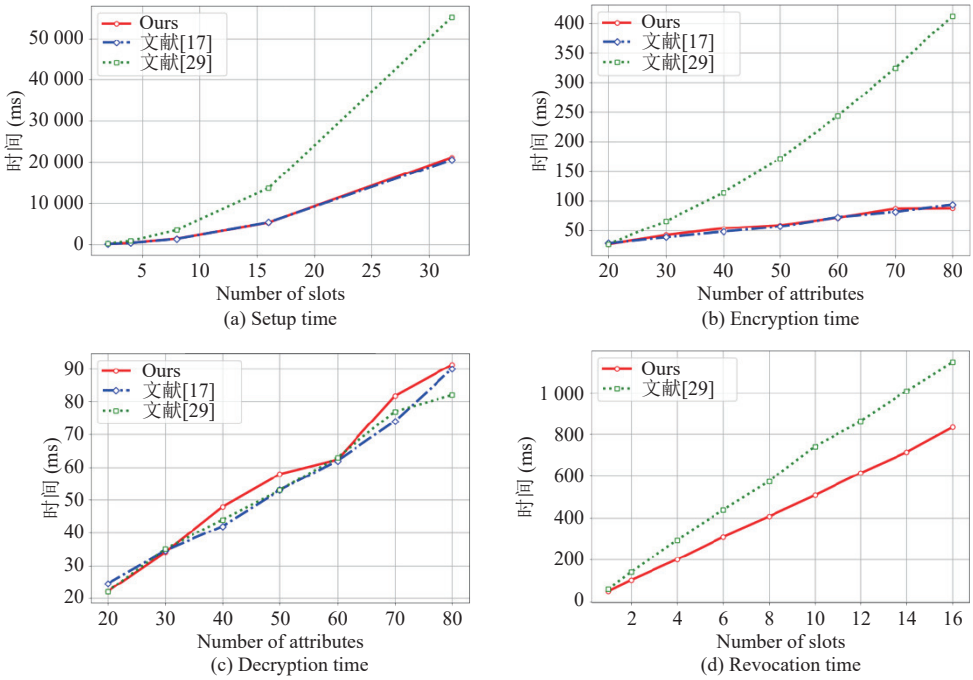

图 2 时间开销对比图

6 总 结

为解决属性加密中的密钥安全问题, 满足注册加密对用户频繁增减场景的支持, 同时降低并行运行多个 sRABE 子方案带来的计算开销, 本文提出一种基于变色龙哈希的可撤销注册属性加密方案. 该方案在注册加密天然免疫密钥托管问题的基础上, 利用变色龙哈希函数易于制造碰撞的特性, 实现用户的动态加入与撤销. 当有新用户加入时, 只需巧妙构造其个人公钥并利用变色龙哈希函数的陷门, 即可高效利用既有的系统主公钥. 在计算开销方面, 该方案基于单个 sRABE 构建, 无需运行多个 sRABE 子方案. 在安全性方面, 本文方案在不可区分模型下可以抵抗选择明文攻击. 实验结果表明, 该方案的计算开销与同类方案相比具有效率优势.

References

[1] Zhang K, Zhang Y, Li YP, Liu XM, Lu LF. A blockchain-based anonymous attribute-based searchable encryption scheme for data sharing. IEEE Internet of Things Journal, 2024, 11(1): 1685–1697. [doi: 10.1109/JIOT.2023.3290975]

[2] Tao JY, Zhang L, Kan HB. Crowdsourcing scheme based on blockchain and decentralized accountable attribute-based authentication. Ruan Jian Xue Bao/Journal of Software, 2025, 36(4): 1844–1858 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/7208.htm [doi: 10.13328/j.cnki.jos.007208]

[3] Ma C, Gao HY, Hu B. Constrained pseudorandom function supporting *t*-DNF and its application to attribute-based encryption. Journal of Cryptologic Research, 2025, 12(3): 662–678 (in Chinese with English abstract). [doi: 10.13868/j.cnki.jcr.000787]

[4] Wang XH, Yu MY, Wang YH, Pi YJ, Xu P, Wang S, Jin H, Han M. Attribute-based access control encryption. IEEE Trans. on Dependable and Secure Computing, 2025, 22(3): 2227–2242. [doi: 10.1109/TDSC.2024.3481497]

[5] Su H, Guo ZZ, Xu MZ. Efficient regulatable identity privacy protection scheme in blockchain. Ruan Jian Xue Bao/Journal of Software, 2026, 37(4): 1777–1800 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/7427.htm [doi: 10.13328/j.cnki.jos.007427]

[6] Guo R, Yang X, Jia CY, Wang JM. A searchable attribute-based encryption scheme supporting policy hiding in cloud-assisted Internet of medical things. Journal of Cryptologic Research, 2025, 12(1): 49–68 (in Chinese with English abstract). [doi: 10.13868/j.cnki.jcr.000748]

[7] Fan K, Li WH, Bai YH, Yang YT, Yang K, Li H. EIV-BT-ABE: Efficient attribute-based encryption with black-box traceability based on encrypted identity vector. IEEE Internet of Things Journal, 2024, 11(9): 15229–15240. [doi: 10.1109/JIOT.2023.3345134]

[8] Singamaneni KK, Muhammad G, Ali Z. A novel multi-qubit quantum key distribution ciphertext-policy attribute-based encryption model to improve cloud security for consumers. IEEE Trans. on Consumer Electronics, 2024, 70(1): 1092–1101. [doi: 10.1109/TCE.2023.3331306]

[9] Zhang KJ, Wang WB, Xu SF, Yu XY, Wang J, Li PC, Qian R. Multi-keyword attribute-based searchable encryption scheme supporting re-encryption for cloud storage. Journal on Communications, 2024, 45(9): 244–257 (in Chinese with English abstract). [doi: 10.11959/j.issn.1000-436x.2024150]

[10] Garg S, Hajiabadi M, Mahmoody M, Rahimi A. Registration-based encryption: Removing private-key generator from IBE. In: Proc. of the 16th Int'l Conf. on Theory of Cryptography (TCC 2018). Panaji: Springer, 2018. 689–718. [doi: 10.1007/978-3-030-03807-6_25]

[11] Cong KL, Eldefrawy K, Smart NP. Optimizing registration based encryption. In: Proc. of the 18th IMA Int'l Conf. on Cryptography and Coding (IMACC 2021). Springer, 2021. 129–157. [doi: 10.1007/978-3-030-92641-0_7]

[12] Li CL, Qiao YN, Bo JG, Yang F. Registered-based revocable attribute-based encryption with keyword search for secure cloud data sharing. IEEE Internet of Things Journal, 2025, 12(23): 50082–50095. [doi: 10.1109/JIOT.2025.3606577]

[13] Li JG, Chen SB, Lu Y, Ning JT, Shen J, Zhang YC. Revocable registered attribute-based encryption with user deregistration. IEEE Internet of Things Journal, 2025, 12(15): 31526–31535. [doi: 10.1109/JIOT.2025.3573050]

[14] Zhang MC, Wan QX, Liu MH, Zhu JL, Wu QT. Efficient searchable attribute encryption scheme supporting dynamic index updates. Ruan Jian Xue Bao/Journal of Software, 2025, 12(15): 31526–31535 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/7428.htm [doi: 10.13328/j.cnki.jos.007428]

[15] Yang XD, Chen AJ, Wang ZS, Liao ZF, Wang CF. Blockchain-based multi-authority ciphertext-policy attribute-based encryption scheme with equality test. Acta Electronica Sinica, 2024, 52(3): 898–908 (in Chinese with English abstract). [doi: 10.12263/DZXB.20220950]

[16] Ateniese G, Magri B, Venturi D, Andrade E. Redactable blockchain-or-rewriting history in bitcoin and friends. In: Proc. of the 2017 IEEE European Symp. on Security and Privacy (EuroS&P). Paris: IEEE, 2017. 111–126. [doi: 10.1109/EUROSP.2017.37]

[17] Hohenberger S, Lu G, Waters B, Wu DJ. Registered attribute-based encryption. In: Proc. of the 42nd Annual Int'l Conf. on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2023). Lyon: Springer, 2023. 511–542. [doi: 10.1007/978-3-031-30620-4_17]

[18] Boneh D, Franklin M. Identity-based encryption from the weil pairing. In: Proc. of the 21st Annual Int'l Cryptology Conf. on Advances in Cryptology (CRYPTO 2001). Santa Barbara: Springer, 2001. 213–229. [doi: 10.1007/3-540-44647-8_13]

[19] Chen L, Harrison K, Soldera D, Smart NP. Applications of multiple trust authorities in pairing based cryptosystems. In: Proc. of the 2002 Int'l Conf. on Infrastructure Security. Bristol: Springer, 2002. 260–275. [doi: 10.1007/3-540-45831-X_18]

[20] Duan PF, Ma ZF, Gao HM, Tian T, Zhang YQ. Multi-authority attribute-based encryption scheme with access delegation for cross blockchain data sharing. IEEE Trans. on Information Forensics and Security, 2025, 20: 323–337. [doi: 10.1109/TIFS.2024.3515812]

[21] Ghopur D, Ma JF, Ma XD, He F, Liu KZ, Jiang T, Wang XY. Decentralized multiauthority attribute-based searchable encryption for E-health cloud. IEEE Internet of Things Journal, 2025, 12(11): 15723–15735. [doi: 10.1109/JIOT.2025.3529480]

[22] Goyal V, Lu S, Sahai A, Waters B. Black-box accountable authority identity-based encryption. In: Proc. of the 2008 ACM Conf. on Computer and Communications Security (CCS 2008). Alexandria, USA: ACM, 2008. 427–436. [doi: 10.1145/1455770.1455824]

[23] Li M, Ding HN, Wang Q, Zhang MW, Meng WZ, Zhu LH, Zhang ZJ, Lin XD. Decentralized threshold signatures with dynamically private accountability. IEEE Trans. on Information Forensics and Security, 2024, 19: 2217–2230. [doi: 10.1109/TIFS.2023.3347968]

[24] Wang JW, Wu JW, Yin XC. Collusion-resistant multi-authority electronic health records sharing scheme. Acta Electronica Sinica, 2023, 51(5): 1179–1186 (in Chinese with English abstract). [doi: 10.12263/DZXB.20220769]

[25] Goyal R, Vusirikala S. Verifiable registration-based encryption. In: Proc. of the 40th Annual Int'l Cryptology Conf. on Advances in Cryptology (CRYPTO 2020). Santa Barbara: Springer, 2020. 621–651. [doi: 10.1007/978-3-030-56784-2_21]

[26] Nguyen K, Pointcheval D, Schädlich R. Dynamic decentralized functional encryption: Generic constructions with strong security. In: Proc. of the 28th IACR Int'l Conf. on Practice and Theory of Public-key Cryptography. Røros: Springer, 2025. 344–379. [doi: 10.1007/978-3-031-91829-2_11]

[27] Datta P, Pal T, Yamada S. Registered FE beyond predicates: (attribute-based) linear functions and more. In: Proc. of the 30th Int'l Conf. on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2024). Kolkata: Springer, 2024. 65–104. [doi: 10.1007/978-981-96-0875-1_3]

[28] Lu G, Waters B, Wu DJ. Multi-authority registered attribute-based encryption. In: Proc. of the 44th Annual Int'l Conf. on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2025). Madrid: Springer, 2025. 3–33. [doi: 10.1007/978-3-031-91131-6_1]

[29] Asano K, Attrapadung N, Hara K, Hashimoto K, Watanabe Y. Key revocation in registered attribute-based encryption. In: Proc. of the 28th IACR Int'l Conf. on Practice and Theory of Public-key Cryptography (PKC 2025). Røros: Springer, 2025. 102–133. [doi: 10.1007/

978-3-031-91826-1_4]

[30] Green M, Hohenberger S, Waters B. Outsourcing the decryption of ABE ciphertexts. In: Proc. of the 20th USENIX Conf. on Security. San Francisco: USENIX Association, 2011. 34.

[31] Lai JZ, Deng RH, Guan CW, Weng J. Attribute-based encryption with verifiable outsourced decryption. IEEE Trans. on Information Forensics and Security, 2013, 8(8): 1343–1354. [doi: 10.1109/TIFS.2013.2271848]

[32] Chen LQ, Xu SG, Zhang H, Weng J. Fair-and-exculpable-attribute-based searchable encryption with revocation and verifiable outsourced decryption using smart contract. IEEE Internet of Things Journal, 2025, 12(4): 4302–4317. [doi: 10.1109/JIOT.2024.3484227]

[33] Xu SM, Yang GM, Li XG, Han XS, Yan XT, Huang XY. Enhancing secure cloud data sharing: Dynamic user groups and outsourced decryption. IEEE Trans. on Dependable and Secure Computing, 2025, 22(6): 7971–7984. [doi: 10.1109/TDSC.2025.3602079]

[34] de Caro A, Iovino V. jPBC: Java pairing based cryptography. In: Proc. of the 16th IEEE Symp. on Computers and Communications (ISCC 2011). Kerkyra: IEEE, 2011. 850–855. [doi: 10.1109/ISCC.2011.5983948]

附中文参考文献

[2] 陶静怡, 张亮, 阚海斌. 基于区块链和去中心化可问责属性认证的众包方案. 软件学报, 2025, 36(4): 1844–1858. http://www.jos.org.cn/1000-9825/7208.htm [doi: 10.13328/j.cnki.jos.007208]

[3] 马超, 高海英, 胡斌. 支持 *t*-DNF 的受限伪随机函数及其在属性加密中的应用. 密码学报 (中英文), 2025, 12(3): 662–678. [doi: 10.13868/j.cnki.jcr.000787]

[5] 苏航, 郭兆中, 徐茂智. 高效的区块链中可监管身份隐私保护方案. 软件学报, 2026, 37(4): 1777–1800. http://www.jos.org.cn/1000-9825/7427.htm [doi: 10.13328/j.cnki.jos.007427]

[6] 郭瑞, 杨鑫, 贾晨阳, 王俊茗. 云辅助医疗物联网中支持策略隐藏的可搜索属性加密方案. 密码学报 (中英文), 2025, 12(1): 49–68. [doi: 10.13868/j.cnki.jcr.000748]

[9] 张克君, 王文彬, 徐少飞, 于新颖, 王钧, 李鹏程, 钱榕. 面向云存储且支持重加密的多关键词属性基可搜索加密方案. 通信学报, 2024, 45(9): 244–257. [doi: 10.11959/j.issn.1000-436x.2024150]

[14] 张明川, 万千雪, 刘牧华, 朱军龙, 吴庆涛. 支持索引动态更新的高效可搜索属性加密方案. 软件学报, 2026, 37(3): 1393–1412. http://www.jos.org.cn/1000-9825/7428.htm [doi: 10.13328/j.cnki.jos.007428]

[15] 杨小东, 陈艾佳, 汪志松, 廖泽帆, 王彩芬. 基于区块链的多授权密文策略属性基等值测试加密方案. 电子学报, 2024, 52(3): 898–908. [doi: 10.12263/DZXB.20220950]

[24] 王经纬, 吴静雯, 殷新春. 抗共谋攻击的多授权电子健康记录共享方案. 电子学报, 2023, 51(5): 1179–1186. [doi: 10.12263/DZXB.20220769]

作者简介

王经纬, 博士, 主要研究领域为属性加密, 注册加密, 云数据安全.

夏志华, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为数字取证, 加密图像处理.

宁建廷, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为公钥密码学, 隐私计算, 人工智能安全.

许胜民, 博士, 教授, 博士生导师, 主要研究领域为密码学, 应用密码学.

李朋林, 硕士, 主要研究领域为高性能同态加密, 数据可信流通, 数据隐私保护.