支持属性撤销的多授权方完全适应性安全访问控制方案

左雨庭[1,2]，许力[1,3*]，李继国[1,3]，田俊峰[2]
（1. 福建师范大学计算机与网络空间安全学院，福州 350117；
2. 闽江大学网络空间安全学院，福州 350108；
3. 福建省网络安全与密码技术重点实验室，福州 350117）

摘要：多授权方属性基加密是多组织跨域协作场景下，实现数据安全访问控制的极具潜力的核心技术。然而，该类场景的动态特性易引发授权方适应性泄密风险，同时要求对用户部分属性与密钥进行定期更新。针对上述问题与挑战，本文提出一种支持属性撤销的多授权方完全适应性安全访问控制方案。该方案在保障完全适应性安全的前提下，通过设计专属撤销参数、构建高效的用户密钥更新算法，实现了多授权方场景下灵活、高效的属性撤销功能。为保障系统前后向安全，本文设计密文更新组件并提出轻量级密文更新方案，该方案仅需执行一次配对运算与一次乘法运算即可完成更新操作。安全性证明结果表明，该方案可满足完全适应性安全要求。本文通过理论分析与实验对比完成性能评估，测试结果表明，该方案在属性撤销阶段具有明显性能优势：密钥撤销计算开销仅为对比方案的 25%，密文更新计算开销为常数级。

关键词：多授权方属性基加密；访问控制；完全适应性安全；属性撤销；轻量级密文更新
中图分类号：TP309；TP393.08 **文献标志码：**A
DOI: 10.20172/j.issn.2097-3136.260625

Multi-authority fully-adaptively secure access control scheme with attribute revocation

Zuo Yuting[1,2], Xu Li[1,3*], Li Jiguo[1,3], Tian Junfeng[2]
（1. College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350117, China；
2. School of Cyber Science and Engineering, Minjiang University, Fuzhou 350108, China；
3. Fujian Provincial Key Laboratory of Network Security and Cryptology, Fuzhou 350117, China）

Abstract: Multi-authority attribute-based encryption (MA-ABE) schemes serve as a promising core technology for secure data access control in multi-organizational cross-domain collaboration scenarios. However, the dynamic nature of such scenarios may cause adaptive leakage risks of authorities and require the periodic update of partial user attributes and secret keys. To address the above problems and challenges, this paper proposes a fully adaptive secure access control scheme supporting attribute revocation. While guaranteeing full adaptive security, the scheme designs dedicated revocation parameters and adopts an efficient user key update algorithm to realize flexible and efficient multi-authority attribute revocation. To guarantee forward and backward security, this paper constructs a ciphertext update component and proposes a lightweight ciphertext update mechanism, which only requires one pairing operation and one multiplication operation. Rigorous security proofs demonstrate that the proposed scheme satisfies full adaptive security. Theoretical analysis

*通信作者：E-mail：xuli@fjnu.edu.cn
基金项目：国家自然科学基金（62471139，62072104，U21A20465）；福建省科技兴警科研专项（2025YZ040003）；福建省产业技术开发和应用计划项目（2025H0043）

引用格式：左雨庭，许力，李继国，等. 支持属性撤销的多授权方完全适应性安全访问控制方案 [J]. 网络空间安全科学学报，2026，4（3）：92 - 104.

Citation Format: Zuo Y T，Xu L，Li J G，et al. Multi-authority fully-adaptively secure access control scheme with attribute revocation[J]. Journal of Cybersecurity，2026，4（3）：92 - 104.

and experimental comparisons are performed for performance evaluation. The results show that the scheme achieves high efficiency in the attribute revocation phase. Specifically, the computational cost of key revocation is only 25% of that of comparable schemes, and the ciphertext update computational cost is at a constant level.

Keywords: MA-ABE; access control; fully-adaptive security; attribute revocation; lightweight ciphertext update

0　引言

近年来，随着人工智能、区块链、云计算等高新技术的快速发展，人类社会正加速迈入智能互联的数字化时代，基于数据的服务逐渐成为推动科技发展与社会进步的重要支撑[1-3]。然而，此类服务高度依赖用户共享数据，尤其在跨域协作场景中，不同组织机构间的数据流通与价值释放面临严峻挑战，数据共享的安全访问控制问题日益凸显[4]。密文策略属性基加密（ciphertext-policy attribute-based encryption，CP-ABE）是实现细粒度数据访问控制的核心机制[5]。在 CP-ABE 体制下[6]，数据拥有者（data owner， DO）可自定义数据访问控制策略，并依据该策略完成数据加密。属性授权方（attribute authority，AA）为数据请求者（data requester，DR）分发对应属性集 S 与属性密钥。当且仅当数据请求者的属性集合匹配预设访问策略时，才可通过对应属性密钥解密密文，获取原始数据。近年来，通过拓展属性撤销[7]、策略隐藏[8]、计算外包[9]、抗量子攻击[10]等功能并融合区块链技术[11]，CP-ABE 展现出广阔的应用场景。

在动态变化的跨域数据共享现实场景中，分属不同地域、不同管理主体机构的用户属性，通常由多个权威机构联合颁发认证[12]。因此，多授权方属性基加密（multi-authority attribute-based encryption，MA-ABE）方案成为支撑多机构协作、实现细粒度数据安全访问控制的最优方案[13]。然而，动态多变的现实应用环境带来两大核心挑战：授权方动态适应密钥泄露风险、用户属性的动态变更问题。

一方面，受企业利益驱使、黑客攻击等现实因素影响，部分 AA 会与 DR 合谋泄露密钥，或存在 DR 窃取 AA 合谋泄露私钥，会严重威胁方案的整体安全性。但现有多数 MA-ABE 方案[14-15]均默认 AA 为绝对可信实体、不存在被攻击风险，未考虑 AA 密钥泄露引发的安全隐患。Lewko 等[16]提出静态适应性安全模型，该模型限定发生密钥泄露的授权方为固定对象。在真实攻击场景中，敌手的攻击能力具备动态演化特征，静态安全模型无法抵御动态适应性密钥泄露威胁。为此，Datta 等[17]首次提出完全适应性安全（fully adaptive security）模型，允许攻击者在方案全生命周期内自适应获取部分授权方密钥。另一方面，真实场景中数据请求者的属性会随场景与时间动态变更。如何在保障完全适应性安全的前提下，实现用户属性与密钥的安全定期更新，是亟待解决的关键问题。相较于全局用户撤销，属性撤销属于细粒度的动态撤销机制，灵活性更强。Zuo 等[18]设计了支持属性更新的撤销机制，但该方案基于单授权方架构构建，无法适配多授权方场景下的属性撤销需求。Yang 等[19]基于复合阶双线性群构建多授权方属性撤销方案，但该方案未实现完全适应性安全，且存在密钥与密文更新计算开销过高的问题。综上，设计一种兼顾完全适应性安全与高效属性撤销性能的 MA-ABE 方案具有重要研究价值与现实意义。

因此，本文提出支持属性撤销的多授权方完全适应性安全访问控制（multi-authority fully adaptive secure access control with attribute revocation, MFSAC-AR）方案。本文主要贡献如下。

1）提出多授权方协作的撤销参数管理机制。首先在系统设置阶段设计撤销参数，并在撤销阶段设计系统更新算法实现多授权方协作的撤销参数更新，可有效防止存在泄密风险的授权方私自篡改参数、更新用户属性密钥，保障系统安全可控。

2）提出基于撤销参数的高效灵活的用户密钥更新算法。该算法按功能拆分密钥撤销组件，避免撤销过程中全量密钥更新，有效提升密钥更新效率；同时支持属性新增、更新与删除操作，可在完全适应性安全的前提下实现灵活的精细化属性撤销。

3）为保障数据访问的前后向安全，本文将撤销参数嵌入密文更新流程。为降低密文更新引发的计算开销，提出轻量级密文更新机制，该机制仅需执行一次配对运算与一次乘法运算即可完成密文更新，且更新开销与访问策略复杂度无关，实现轻量化更新。

4）安全性证明结果证实，MFSAC-AR 可满足完全适应性安全要求。本文通过理论分析与实验对比完成性能评估，结果表明该方案在属性撤销阶段具有明显性能优势：密钥撤销计算开销仅为对比方案的 25%，密文更新计算开销为常数级。

1 相关工作

1.1 多授权方 ABE 方案

针对现实场景的多机构协作需求，MA-ABE 方案已得到学术界广泛关注。Chase 等[20]首次提出 MA-ABE 方案，该方案依托多个独立授权方管理用户属性并分发密钥，同时兼容阈值与访问树两种访问策略。Lin 等[21]指出 Chase 等[20]方案存在依赖可信中央授权方的局限性，并提出一种基于访问树的无中心安全阈值 MA-ABE 方案。张学旺等[22]融合联盟链与哈希算法，实现密文安全防护与访问策略隐藏。为提升访问策略的表达能力，线性秘密共享机制被引入 CP-ABE 体系中[23]。王静怡等[24]聚焦医疗场景下的患者隐私保护与密钥泄露抵御需求，提出一种支持细粒度策略隐藏的可追踪去中心化访问控制方案。Li 等[25]针对属性管理层面的单点故障与性能瓶颈问题，提出了一种阈值 MA-ABE 访问控制方案。Qin 等[26]实现了更细粒度的多授权方协作管理，可针对单一属性独立配置授权方管理数量与阈值参数。

上述研究[20-26]虽从不同维度推动 MA-ABE 发展，但均未考虑授权方密钥泄露的安全风险。Lewko 等[16]针对静态适应性安全场景，基于复合阶双线性群构建多授权方属性基加密方案，并依托子群判定假设完成安全性证明。Ambrona 等[27]基于素数阶双线性群搭建加密系统，并在通用群模型下验证了方案的安全性。上述方案虽提升了系统安全等级，但仍无法完全适配复杂的现实应用安全需求。Datta 等[17]分别基于复合阶、素数阶双线性群，构造了两类具备完全适应性安全的 MA-ABE 方案。Chen 等[28]对 Datta 等[17]提出的素数阶方案进行优化，实现了参数轻量化与属性复用功能。

1.2 支持属性撤销的 ABE 方案

属性撤销可实现对用户属性的动态化、精细化管理。Hur 等[29]提出一种基于属性群与用户二叉树的属性撤销机制，但王鹏翩等[30]研究发现，该机制存在灵活性弱、密钥维护成本高、无法抵御合谋攻击等缺陷，并基于合数阶双线性群构建了一种直接属性撤销方案。Li 等[31]针对树形访问策略场景，提出基于属性群的轻量化属性撤销机制。为保障访问策略的表达能力，Zuo 等[18]基于线性秘密共享机制构建支持属性更新的访问控制方案，通过周期性更新用户属性与密钥，实现了具备前后向安全的属性撤销功能。肖浩等[32]进一步兼顾非单调策略适配与属性复用需求，构建双层密钥撤销机制，实现了安全高效的属性撤销。

上述研究均未对多授权方场景下的属性撤销问题开展针对性设计。Yang 等[33]提出支持属性撤销的多授权方访问控制方案，但 Hong 等[34]验证表明，该方案无法满足后向安全要求。此外，部分研究[35-36]通过属性群密钥机制实现了多授权方场景下的高效属性撤销。Yang 等[19]基于复合阶双线性群，提出周期性属性撤销与即时属性撤销两类多授权方属性撤销机制。上述方案虽实现了多授权方架构与属性撤销功能的深度融合，但兼具完全适应性安全与高效属性撤销性能的 CP-ABE 方案，仍是当前亟待解决的研究难点。

2 预备知识

2.1 合数阶对称双线性映射

定义 1 合数阶对称双线性映射[17]：给定一个程序 $\mathcal{G}$，输入安全参数 λ，输出五元组 $\{N=p_1p_2p_3,\mathbb{G},\mathbb{G}_T,g,e\}$。其中，$N=p_1p_2p_3$ 是 3 个不小于 1^λ 的大素数 p_1,p_2,p_3 的乘积，$\mathbb{G}$ 和 $\mathbb{G}_T$ 都是阶为 N 的循环群，g 是 $\mathbb{G}$ 的一个生成元。合数阶对称双线性映射 $e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$ 满足如下性质。

双线性：对于任何元素 $g_1,g_2\in\mathbb{G}$ 以及 $a,b\in\mathbb{Z}_N$，都有 $e(g_1^a,g_2^b)=e(g_1,g_2)^{ab}$；

非退化性：$e(g,g)\neq 1$；

可计算性：对于任意元素 $g_1,g_2\in\mathbb{G}$，存在多项式时间的算法计算 $e(g_1,g_2)$。

令 $\mathbb{G}_{p_i}(i=1,2,3)$ 是 $\mathbb{G}$ 的 p_i 阶子循环群。那么，子群 $\mathbb{G}_{p_i}(i=1,2,3)$ 其中一个生成元可以表示为 $g_i=g^{N/p_i}$，且有子群 $\mathbb{G}_{p_i}(i=1,2,3)$ 生成元的乘积 $\prod_{i=1}^{3}g_i$ 是群 $\mathbb{G}$ 的一个生成元。对于 $\forall h\in\mathbb{G}$，存在唯一的一组数：$\{x_i\in\mathbb{Z}_{g_i}\}_{i\in\{1,2,3\}}$，使得 $h=\prod_{i=1}^{3}g_i^{x_i}$。

合数阶对称双线性群的子群满足正交性：对于任意两个子群 $\mathbb{G}_{p_i},\mathbb{G}_{p_j}$ 以及对应的生成元 $g_i\in\mathbb{G}_{p_i},g_j\in\mathbb{G}_{p_j}$，如果 $i\neq j$，那么 $e(g_i,g_j)=1$。

2.2 线性秘密分享方案

定义 2 线性秘密分享方案（linear secret-sharing schemes，LSSS）[17,23]：设 $\mathcal{U}$ 是一个属性全集。访问策略 $(\boldsymbol{M},\rho)$ 在属性全集 $\mathcal{U}$ 上的线性秘密分享方案包括两个阶段：分享和重构。

分享：该阶段基于访问策略 $(\boldsymbol{M},\rho)$ 分享一个秘密值 $s\in\mathbb{Z}_N$。随机选择 $y_2,\cdots,y_n\in\mathbb{Z}_N$，并设置一个向量 $\vec{\boldsymbol{v}}=(s,y_2,\cdots,y_n)\in\mathbb{Z}_N^d$。计算 $\lambda_{\rho(i)}=\boldsymbol{M}_i\cdot\vec{\boldsymbol{v}}$ 作为属性 $\rho(i)$

的秘密份额，其中，$i\in[1,l]$。

重构：该阶段从所有秘密份额中重构秘密值s，对任意授权集$I=\{i\in[1,l]:\rho(i)\in S\}$，$S\in\mathcal{U}$，存在$\{\omega_i\in\mathbb{Z}_p:i\in I\}$，使得$\sum_{i\in I}\boldsymbol{M}_i\cdot\omega_i=(1,0,\cdots,0)$，即$\sum_{i\in I}\lambda_{\rho(i)}\omega_i=s$。其中，$\boldsymbol{M}$是大小为$l\times d$的秘密分享矩阵，且$\rho$映射$\boldsymbol{M}$的第$i$行$\rho(i)\in\mathcal{U}$。

2.3　常用符号表

本文常用符号见表 1。

表 1　常用符号

Table 1　Frequently used symbols

符号	含义
$\{N,\mathbb{G},\mathbb{G}_T,g,e\}$	双线性映射五元组
$\mathbb{G}_{p_i},g_i$	$\mathbb{G}$的p_i阶子群及其生成元
$I,S,\mathcal{U}$	授权集/属性集/属性全集
$\mathrm{FP},\mathrm{AP},\mathrm{RP}_k$	固定/属性/撤销参数
$H:\{0,1\}\to\mathbb{G}$	Hash函数
$\{\mathrm{ASK}_u\}_{u\in\mathcal{U}}$	全局属性私钥
$\{\mathrm{AK}_{x,k}\}_{x\in S}$	数据请求者的属性密钥
msg	用于加密的数据及密文
CUC	密文更新组件
CT_k	第k次更新后的密文

3　MFSAC-AR 方案架构与形式化定义

3.1　方案架构

本文提出支持属性撤销的多授权方完全适应性安全访问控制方案，其架构如图 1 所示。

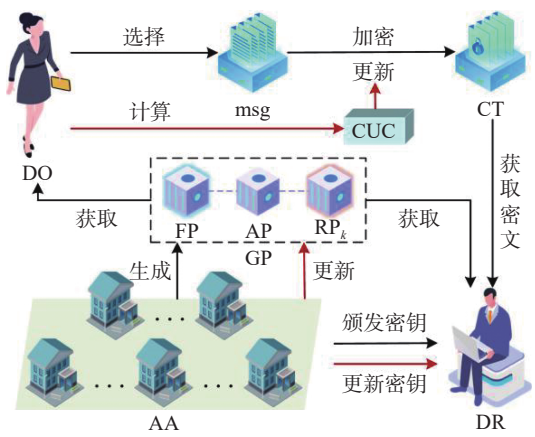

图 1　MFSAC-AR 方案架构

Fig. 1　Architecture of MFSAC-AR

方案架构中涉及 3 类实体：属性授权方、数据拥有者和数据请求者。

属性授权方：有多个属性授权方，在系统设置阶段生成全局参数GP，在密钥生成阶段为 DR 颁发属性和密钥；在撤销阶段更新撤销参数RP_k，为 DR 更新属性和密钥。

数据拥有者：设置访问策略$(\boldsymbol{M},\rho)$并加密数据msg，生成并秘密保存密文更新组件CUC，在撤销阶段使用CUC更新密文。

数据请求者：提供自己的身份信息并获得由 AA 颁发和更新的属性密钥。在解密阶段利用自己的属性密钥解密 DO 加密的数据。

3.2　形式化定义

MFSAC-AR 方案由 5 个阶段构成：系统设置阶段、用户注册阶段、加密阶段、撤销阶段和解密阶段。

3.2.1　系统设置阶段

所有 AA 根据安全参数λ和属性全集$\mathcal{U}$，产生全局参数GP和全局属性私钥$\{\mathrm{ASK}_u\}_{u\in\mathcal{U}}$。该阶段包括如下 3 个算法。

固定参数设置算法$\mathrm{FP}\leftarrow\mathrm{FPSetup}(\lambda)$：由一个 AA 执行，输入安全参数$\lambda$，输出固定参数FP。

属性参数设置算法$(\{a_i\}_{i\in[1,n]},\mathrm{AP},\{\mathrm{ASK}_u\}_{u\in\mathcal{U}})\leftarrow\mathrm{APSetup}(FP,\mathcal{U})$：由所有 AA 执行，输入固定参数FP和属性全集$\mathcal{U}$，输出 AA 的全局身份标识$\{a_i\}_{i\in[1,n]}$，属性参数AP和每个属性的全局属性私钥$\{\mathrm{ASK}_u\}_{u\in\mathcal{U}}$。

全局参数设置算法$\mathrm{GP}\leftarrow\mathrm{GPSetup}(\lambda,\mathrm{FP},\mathrm{AP})$：由 AA 执行，输入固定参数FP，属性参数AP，AA 产生初始撤销参数RP_0，最终生成全局参数GP。

3.2.2　用户注册阶段

AA 根据 DR 的身份信息RID为其分配属性集S，并产生初始属性密钥$\{\mathrm{AK}_{x,0}\}_{x\in S}$。

密钥生成算法$(\{\mathrm{AK}_{x,0}\}_{x\in S})\leftarrow\mathrm{KeyGen}(\mathrm{GP},\mathrm{RID},\{\mathrm{ASK}_u\})$：DR 提交自己的身份信息RID，相关的 AA 根据实际情况为 DR 分配初始属性集S，并基于全局属性私钥$\{\mathrm{ASK}_u\}$产生 DR 的初始属性密钥$\{\mathrm{AK}_{x,0}\}_{x\in S}$。

3.2.3　加密阶段

DO 设置 LSSS 访问策略$(\boldsymbol{M},\rho)$，并加密消息msg产生初始密文CT_0和密文更新组件CUC。

加密算法$(\mathrm{CT}_0,\mathrm{CUC})\leftarrow\mathrm{Enc}(\mathrm{GP},\mathrm{msg},(\boldsymbol{M},\rho))$：DO 输入全局参数GP，消息msg和 LSSS 访问策略$(\boldsymbol{M},\rho)$，输出初始密文CT_0和密文更新组件CUC。

3.2.4　撤销阶段

AA 更新撤销参数并根据实际情况更新 DR 的属

性及密钥。DO 同步更新密文。该阶段包括如下 3 个算法。

系统更新算法 $RP_{k+1} \leftarrow \mathrm{SysUpd}(\mathrm{RP}_k)$：对于第 k 次撤销阶段，所有 AA 协作将撤销参数 RP_k 更新为新的撤销参数 RP_{k+1}。

用户密钥更新算法 $(S', \{\mathrm{AK}_{x,k}\}_{x\in S'}) \leftarrow \mathrm{KeyUpd}(\mathrm{GP}, \mathrm{RID}, \{\mathrm{ASK}_u\})$：由 AA 和 DR 执行，输入全局参数 GP，全局属性私钥 $\{\mathrm{ASK}_u\}$，DR 的身份标识 RID，输出最新属性集 S' 和更新后的密钥 $\{\mathrm{AK}_{x,k}\}_{x\in S'}$。

密文更新算法 $\mathrm{MC}_k \leftarrow \mathrm{CTUpd}(\mathrm{RP}_k, \mathrm{msg}, \mathrm{CUC})$：DO 输入全局参数 GP、要更新密文对应的消息 msg 和对应的密文更新组件 CUC，输出更新后的消息相关密文 MC_k。

3.2.5 解密阶段

DR 用自己的最新属性密钥 $\{\mathrm{AK}_{x,k}\}_{x\in S'}$ 解密密文 CT_k 得到数据 msg′，如果 DR 的属性集 S' 满足 CT_k 访问策略，那么存在授权集 $I \subset S'$，使得 $\mathrm{msg}' = \mathrm{msg}$。

解密算法 $\mathrm{msg}' \leftarrow \mathrm{Dec}(\mathrm{GP}, \{\mathrm{AK}_{x,k}\}_{x\in I'}, \mathrm{CT}_k)$：DR 输入全局参数 GP，DR 的最新属性密钥 $\{\mathrm{AK}_{x,k}\}_{x\in S'}$，密文 CT_k，输出解密后的消息 msg′。

3.3 正确性要求

如果对于任意由安全参数 $\lambda \in \mathbb{Z}_N$ 产生的全局参数 GP，任意消息 msg 及对应的访问策略 $(\boldsymbol{M}, \rho)$，任意具有满足访问策略的授权集 I 的 DR（身份标识为 RID），都有等式 (1) 成立，那么 MFSAC-AR 是正确的。

$$\Pr\left[\begin{array}{c} \mathrm{FP} \leftarrow \mathrm{FPSetup}(\lambda) \\ (\{a_i\}_{i\in[1,n]}, \mathrm{AP}, \{\mathrm{ASK}_u\}_{u\in\mathcal{U}}) \leftarrow \mathrm{APSetup}(\mathrm{FP}, \mathcal{U}) \\ \mathrm{GP} \leftarrow \mathrm{GPSetup}(\lambda, \mathrm{FP}, \mathrm{AP}) \\ (\{\mathrm{AK}_{x,0}\}_{x\in S}) \leftarrow \mathrm{KeyGen}(\mathrm{GP}, \mathrm{RID}, \{\mathrm{ASK}_u\}) \\ (\mathrm{CT}_0, \mathrm{CUC}) \leftarrow \mathrm{Enc}(\mathrm{GP}, \mathrm{msg}, (\boldsymbol{M}, \rho)) \\ \mathrm{RP}_k \leftarrow \mathrm{SysUpd}(\mathrm{RP}_{k-1}) \\ (S', \{\mathrm{AK}_{x,k}\}_{x\in S'}) \leftarrow \mathrm{KeyUpd}(\mathrm{GP}, \mathrm{RID}, \{\mathrm{ASK}_u\}) \\ \mathrm{MC}_k \leftarrow \mathrm{CTUpd}(\mathrm{RP}_k, \mathrm{msg}, \mathrm{CUC}) \\ \mathrm{msg}' \leftarrow \mathrm{Dec}(\mathrm{GP}, \mathrm{CT}_k, \{\mathrm{AK}_{x,k}\}_{x\in I'}) \end{array} \Rightarrow \mathrm{msg}' = \mathrm{msg}\right] = 1 \quad (1)$$

3.4 安全模型

部分 AA 会与 DR、其他 AA 合谋，或遭 DR 攻击，致使其管理的全局属性私钥泄露。AA 合谋上限不超过撤销参数 RP_k 的阈值。属性不满足访问策略的 DR 可能尝试与其他属性不满足访问策略的 DR 合谋，破坏方案的机密性，可能窃取部分 AA 的密钥。此外，MFSAC-AR 方案支持用户属性动态撤销，这意味着完全适应性安全额外考虑撤销阶段涉及的前后向安全和抗合谋攻击（撤销前后的密钥合谋）。

总的来说，MFSAC-AR 方案的安全模型表示为敌手 $\mathcal{A}$ 和挑战者 $\mathcal{C}$ 之间的交互游戏，其中，$\mathcal{A}$ 的目标是攻破方案的安全性获得原始数据，$\mathcal{C}$ 的目标是攻破完全适应性安全分布式 MA-ABE 方案[17]。敌手 $\mathcal{A}$ 赢得下述游戏的优势定义为 $\mathrm{Adv}_{\mathcal{A}}(\lambda)$。

系统设置：挑战者 $\mathcal{C}$ 执行系统设置阶段的算法，生成全局参数 GP 和全局属性私钥 $\{\mathrm{ASK}_u\}_{u\in\mathcal{U}}$，并将全局参数 GP 发送给敌手 $\mathcal{A}$。同时秘密保存全局属性私钥 $\{\mathrm{ASK}_u\}_{u\in\mathcal{U}}$。

询问阶段 I：敌手 $\mathcal{A}$ 适应性地进行多项式次询问操作。

1）初始属性密钥询问：敌手 $\mathcal{A}$ 提交身份属性对 (RID, x)，其中，$x \in S \subset \mathcal{U}$。然后挑战者 $\mathcal{C}$ 执行 KeyGen 算法，生成属性 x 对应的初始属性密钥 $\mathrm{AK}_{x,0}$。

2）全局属性私钥询问：敌手 $\mathcal{A}$ 询问全局属性 $u \in \mathcal{U}$ 的全局属性私钥。挑战者 $\mathcal{C}$ 发送对应的全局属性私钥 ASK_u 给 $\mathcal{A}$。

3）属性密钥更新询问：敌手 $\mathcal{A}$ 询问身份属性对 (RID, x) 的第 k 次属性密钥更新。挑战者 $\mathcal{C}$ 执行 SysUpd 算法更新撤销参数，然后执行 KeyUpd 算法，将更新的属性密钥 $\mathrm{AK}_{x,k}$ 输出给 $\mathcal{A}$。

挑战阶段：敌手 $\mathcal{A}$ 提交两个等长的消息 msg_0, msg_1 和 LSSS 访问策略 $(\boldsymbol{M}, \rho)$。设定的 LSSS 访问策略需要满足约束如下：令 U_{ASK} 表示 LSSS 访问策略中敌手询问过全局属性私钥 $\{\mathrm{ASK}_u\}$ 对应的属性集合，U_{RID} 表示敌手询问过的有关于身份 RID 的所有属性构成的集合。要求 $U_{\mathrm{ASK}} \cup U_{\mathrm{RID}}$ 不能线性表出向量 $(1, 0, \cdots, 0)$。然后挑战者 $\mathcal{C}$ 投掷随机硬币 $c \leftarrow \{0,1\}$，并执行 Encrypt 算法，在敌手给定的 LSSS 访问策略 $(\boldsymbol{M}, \rho)$ 上加密消息 msg_c。当敌手执行密钥更新询问时，挑战者 $\mathcal{C}$ 执行 LightCTUpd 算法更新密文。

询问阶段 II：敌手 $\mathcal{A}$ 可以在满足挑战阶段的约束的基础上执行多次询问阶段 I 中的所有询问。

猜测阶段：敌手 $\mathcal{A}$ 提交对于 c 的猜测 c'，如果 $c = c'$，那么敌手 $\mathcal{A}$ 赢得游戏。

敌手 $\mathcal{A}$ 在这个游戏中的优势定义为：$\mathrm{Adv}_{\mathcal{A}}(\lambda) = |\Pr[c = c'] - 1/2|$。

定义 3　完全适应性安全：如果在概率多项式时间敌手$\mathcal{A}$以不超过ε的概率赢得上述游戏，即$\mathrm{Adv}_{\mathcal{A}}(\lambda)\leqslant\varepsilon$，则称 MFSAC-AR 方案是完全适应性安全的。

4　MFSAC-AR 方案的具体构造

4.1　系统设置阶段

固定参数设置算法$\mathrm{FP}\leftarrow\mathrm{FPSetup}(\lambda)$：首先，根据给定的安全参数$\lambda$，AA 选择素数$p_1,p_2,p_3$，并计算$N=p_1p_2p_3$。接下来，AA 产生$N$阶的双线性群$\mathrm{G}=(N,\mathbb{G},\mathbb{G}_T,e)$，其中，$e:\mathbb{G}\times\mathbb{G}\rightarrow\mathbb{G}_T$；选择$\mathbb{G}$的$p_1$阶子群$\mathbb{G}_{p_1}$的生成元$g_1$。然后，AA 构造安全的 hash 函数$H:\{0,1\}\rightarrow\mathbb{G}$。最后，AA 设置固定参数$\mathrm{FP}=(\mathrm{G},g_1,H)$。

属性参数设置算法$(\{a_i\}_{i\in[1,n]},\mathrm{AP},\{\mathrm{ASK}_u\}_{u\in\mathcal{U}})\leftarrow\mathrm{APSetup}(\mathrm{FP},\mathcal{U})$：首先，所有属性授权方根据实际情况分配管理属性集，第i个属性授权方AA_i管理的属性集记为$\mathcal{U}_i\subset\mathcal{U}$。接下来，$\mathrm{AA}_i$设置自己的全局身份标识$a_i\leftarrow\mathbb{Z}_N$。$\mathrm{AA}_i$为属性$u\in\mathcal{U}_i$选择两个随机值$\alpha_u,\beta_u\in\mathbb{Z}_N$作为该属性的全局属性私钥，即$\mathrm{ASK}_u=(\alpha_u,\beta_u)$。然后，$\mathrm{AA}_i$计算$\mathrm{APK}_{A,u}=g_1^{\alpha_u},\mathrm{APK}_{B,u}=g_1^{\beta_u}$作为该属性的全局属性公钥，即$\mathrm{APK}_u=(\mathrm{APK}_{A,u},\mathrm{APK}_{B,u})$。最后，$\mathrm{AA}_i$秘密保存$\{\mathrm{ASK}_u\}_{u\in\mathcal{U}_i}$，所有全局属性的全局属性私钥记为$\{\mathrm{ASK}_u\}_{u\in\mathcal{U}}$，公钥集合为属性参数$\mathrm{AP}=\{\mathrm{ASK}_u\}_{u\in\mathcal{U}}$。

全局参数设置算法$\mathrm{GP}\leftarrow\mathrm{GPSetup}(\lambda,\mathrm{FP},\mathrm{AP})$：AA 设置初始撤销参数$\mathrm{RP}_0\in\mathbb{G}_{p_1}$，并构造全局参数$\mathrm{GP}=(\mathrm{FP},\mathrm{RP}_0,\mathrm{AP})$。

4.2　用户注册阶段

密钥生成算法$(\{\mathrm{AK}_{x,0}\}_{x\in S})\leftarrow\mathrm{KeyGen}(\mathrm{GP},\mathrm{RID},\{\mathrm{ASK}_u\})$：首先，所有的 AA 根据 DR 身份信息分配属性集S。然后，AA 对属性$x\in S\cap\mathcal{U}_i$分别计算：初始密钥撤销组件（即嵌入了撤销参数的组件）$\mathrm{KRC}_{x,0}=(H(\mathrm{RID})\cdot\mathrm{RP}_0)^{\alpha_x}$和初始密钥辅助组件（即辅助解密的组件）$\mathrm{KAC}_{x,0}=(H(\mathrm{RID}))^{\beta_x}$。最后，DR 的初始属性密钥为$\{\mathrm{AK}_{x,0}=(\mathrm{KRC}_{x,0},\mathrm{KAC}_{x,0})\}_{x\in S}$。

4.3　加密阶段

加密算法$(\mathrm{CT}_0,\mathrm{CUC})\leftarrow\mathrm{Enc}(\mathrm{GP},\mathrm{msg},(\boldsymbol{M},\rho))$：首先，DO 获取全局参数 GP，并设置 LSSS 访问策略$(\boldsymbol{M},\rho)$。其中，$\boldsymbol{M}\in\mathbb{Z}_N^{l\times d}$，$\rho:[1,l]\rightarrow\mathcal{U}$。$\boldsymbol{M}_i(i\in[1,l])$表示访问矩阵$\boldsymbol{M}$的第$i$行元素构成的行向量。然后，DO 执行算法 1 加密要共享的数据 msg，并输出初始密文$\mathrm{CT}_0=\{\mathrm{MC}_0,(\boldsymbol{M},\rho),\mathrm{APC}\}$和密文更新组件 CUC，其中，$\mathrm{MC}_0$表示初始消息相关密文，APC表示访问策略相关密文。密文更新组件CUC由 DO 秘密保存，并用于后续密文更新。

加密算法的伪代码如算法 1 所示。

算法1　加密算法

输入：全局参数GP，访问策略$(\boldsymbol{M},\rho)$，加密数据msg；

输出：初始密文CT_0和密文更新组件CUC。

1. 选择$s,y_2,\cdots,y_d,y_2',\cdots,y_d'\xleftarrow{R}\mathbb{Z}_N$；
2. 构造列向量$\vec{\boldsymbol{v}}_A=(s,y_2,\cdots,y_d)\in\mathbb{Z}_N^d$；
3. 构造列向量$\vec{\boldsymbol{v}}_B=(-s,y_2',\cdots,y_d')\in\mathbb{Z}_N^d$；
4. **for** $i\in[1,l]$ **do**
5. 　选择$r_{\alpha,i},r_{\beta,i}\xleftarrow{R}\mathbb{Z}_N$；
6. 　计算$\mathrm{AC}_{A,i,1}=g_1^{r_{\alpha,i}},\mathrm{AC}_{A,i,2}=g_1^{\boldsymbol{M}_i\cdot\vec{\boldsymbol{v}}_A}P_{A,\rho(i)}^{r_{\alpha,i}}$；
7. 　计算$\mathrm{AC}_{B,i,1}=g_1^{r_{\beta,i}},\mathrm{AC}_{B,i,2}=g_1^{\boldsymbol{M}_i\cdot\vec{\boldsymbol{v}}_B}P_{B,\rho(i)}^{r_{\beta,i}}$；
8. $\mathrm{APC}=\{\mathrm{AC}_{A,i,1},\mathrm{AC}_{A,i,2},\mathrm{AC}_{B,i,1},\mathrm{AC}_{B,i,2}\}_{i\in[1,l]}$；
9. 计算$\mathrm{MC}_0=\mathrm{msg}\cdot e(g_1,\mathrm{RP}_0)^s$和$\mathrm{CUC}=g_1^s$；
10. $\mathrm{CT}_0=\{\mathrm{MC}_0,(\boldsymbol{M},\rho),\mathrm{APC}\},\mathrm{CUC}$

4.4　撤销阶段

系统更新算法$\mathrm{RP}_k\leftarrow\mathrm{SysUpd}(\mathrm{RP}_{k-1})$：对于第$k$次撤销阶段，AA 基于拉格朗日差值技术协作执行算法 2，将撤销参数RP_{k-1}更新为新的撤销参数RP_k，用于后续密钥和密文更新。

用户密钥更新算法$(S',\{\mathrm{AK}_{x,k}\}_{x\in S'})\leftarrow\mathrm{KeyUpd}(\mathrm{GP},\mathrm{RID},\{\mathrm{ASK}_u\})$：首先，所有属性授权方 AA 根据实际情况为 DR 更新最新属性集S'。然后，AA_i对属性$x\in S'\cap\mathcal{U}_i$更新初始密钥撤销组件$\mathrm{KRC}_{x,k}=(h(\mathrm{RID})\cdot\mathrm{RP}_k)^{\alpha_x}$。DR 更新后的属性密钥为$\{\mathrm{AK}_{x,k}=(\mathrm{KRC}_{x,k},\mathrm{KAC}_{x,k}=\mathrm{KAC}_{x,k-1})\}_{x\in S'}$。

密文更新算法$\mathrm{MC}_k\leftarrow\mathrm{CTUpd}(\mathrm{RP}_k,\mathrm{msg},\mathrm{CUC})$：DO 获取最新的撤销参数$\mathrm{RP}_k$和msg对应的密文更新组件CUC，并更新消息相关密文$\mathrm{MC}_k=\mathrm{msg}\cdot e(\mathrm{CUC},\mathrm{RP}_k)$。

算法2　系统更新算法

输入：撤销参数RP_{k-1}；

输出：更新后的撤销参数RP_k。

1. 根据实际情况预设的安全阈值t，选择t个AA构成撤销参数委员会$\mathbb{AA}_t$；
2. **for** $\mathrm{AA}_i\in\mathbb{AA}_t$ **do**
3. 　AA_i选择随机数$\tau_i\in\mathbb{Z}_N$；
4. 　AA_i计算$\mathrm{RP}_{\{k-1,i\}}=\mathrm{RP}_{k-1}^{\tau_i}$给其他AA；
5. **for** $i\in[1,n]$ **do**
6. 　AA_i计算$\mathrm{RP}_k'=\prod_{\mathrm{AA}_j\in\mathbb{AA}_t}\mathrm{RP}_{k-1,j}^{\prod_{\mathrm{AA}_j\in\mathbb{AA}_t\setminus\mathrm{AA}_j}\frac{a_i}{a_i-a_j}}$；
7. **if** 超过t个AA_i计算的RP_k'相同 **then**
8. 　RP_k更新为该相同的RP_k'；

4.5 解密阶段

解密算法 $\mathrm{msg}' \leftarrow \mathrm{Dec}(\mathrm{GP},\{\mathrm{AK}_{x,k}\}_{x\in I'},\mathrm{CT}_k)$：首先，DR 获取密文 CT_k 并选择满足访问策略的授权集 $I \in S'$。然后，DR 根据授权集 $I \in S'$ 构造 $\boldsymbol{M}$ 的子矩阵 $\boldsymbol{M}_I$。最后，令 $\vec{\omega} = (\omega_1,\cdots,\omega_l) \in \mathbb{Z}_N^I$，DR 计算 $\vec{\omega} = \boldsymbol{M}_I^{-1}(1,0,\cdots,0)^T$，代入式 (2) 恢复数据。

$$\mathrm{msg}' = \frac{\mathrm{MC}_k}{\prod_{i\in I}\left(\dfrac{e(\mathrm{AC}_{A,i,2},H(\mathrm{RID})\cdot\mathrm{RP}_k)\cdot e(\mathrm{AC}_{B,i,2},H(\mathrm{RID}))}{e(\mathrm{AC}_{A,i,1},\mathrm{KRC}_{\rho(i),k})\cdot e(\mathrm{AC}_{B,i,1},\mathrm{KAC}_{\rho(i),k})}\right)^{\omega_i}} \tag{2}$$

5 正确性分析与安全性证明

5.1 正确性分析

本节分析提出的 MFSAC-AR 方案的正确性。首先令 $D = D_R \cdot D_A$，其中:

$$\begin{cases} D_R = \prod_{i\in I}\left(\dfrac{e(\mathrm{AC}_{A,i,2},H(\mathrm{RID})\cdot\mathrm{RP}_k)}{e(\mathrm{AC}_{A,i,1},\mathrm{KRC}_{\rho(i),k})}\right)^{\omega_i} \\ D_A = \prod_{i\in I}\left(\dfrac{e(\mathrm{AC}_{B,i,2},H(\mathrm{RID}))}{e(\mathrm{AC}_{B,i,1},\mathrm{KAC}_{\rho(i),k})}\right)^{\omega_i} \end{cases}$$

D_R 的推导如式 (3) 所示：

$$\begin{aligned} D_R &= \prod_{i\in I}\left(\frac{e(\mathrm{APC}_{A,i,2},H(\mathrm{RID})\cdot\mathrm{RP}_k)}{e(\mathrm{APC}_{A,i,1},\mathrm{KRC}_{\rho(i),k})}\right)^{\omega_i} \\ &= \prod_{i\in I}\left(\frac{e(g_1^{\boldsymbol{M}_i\cdot\vec{\boldsymbol{v}}_A}\cdot P_{A,\rho(i)}^{r_{\alpha,i}},H(\mathrm{RID})\cdot\mathrm{RP}_k)}{e(g_1^{r_{\alpha,i}},(H(\mathrm{RID})\cdot\mathrm{RP}_k)^{\alpha_{\rho(i)}})}\right)^{\omega_i} \\ &= \prod_{i\in I}\left(\frac{e(g_1^{\boldsymbol{M}_i\cdot\vec{\boldsymbol{v}}_A}\cdot g_1^{\alpha_{\rho(i)}r_{\alpha,i}},H(\mathrm{RID})\cdot\mathrm{RP}_k)}{e(g_1^{r_{\alpha,i}},(H(\mathrm{RID})\cdot\mathrm{RP}_k)^{\alpha_{\rho(i)}})}\right)^{\omega_i} \\ &= \prod_{i\in I}\left(\frac{e(g_1^{\boldsymbol{M}_i\cdot\vec{\boldsymbol{v}}_A},H(\mathrm{RID})\cdot\mathrm{RP}_k)\cdot e(g_1^{\alpha_{\rho(i)}r_{\alpha,i}},H(\mathrm{RID})\cdot\mathrm{RP}_k)}{e(g_1^{r_{\alpha,i}},(H(\mathrm{RID})\cdot\mathrm{RP}_k)^{\alpha_{\rho(i)}})}\right)^{\omega_i} \\ &= \prod_{i\in I} e(g_1^{\boldsymbol{M}_i\cdot\vec{\boldsymbol{v}}_A},H(\mathrm{RID})\cdot\mathrm{RP}_k)^{\omega_i} \\ &= \prod_{i\in I} e(g_1,H(\mathrm{RID})\cdot\mathrm{RP}_k)^{\sum_{i\in I}\boldsymbol{M}_i\cdot\vec{\boldsymbol{v}}_A\cdot\omega_i} \\ &= e(g_1,H(\mathrm{RID})\cdot\mathrm{RP}_k)^s \end{aligned} \tag{3}$$

D_A 的推导如式 (4) 所示：

$$\begin{aligned} D_A &= \prod_{i\in I}\left(\frac{e(\mathrm{APC}_{B,i,2},H(\mathrm{RID}))}{e(\mathrm{APC}_{B,i,1},\mathrm{KAC}_{\rho(i),k})}\right)^{\omega_i} \\ &= \prod_{i\in I}\left(\frac{e\left(g_1^{\boldsymbol{M}_i\cdot\vec{\boldsymbol{v}}_B}\cdot P_{B,\rho(i)}^{r_{\beta,i}},H(\mathrm{RID})\right)}{e\left(g_1^{r_{\beta,i}},(H(\mathrm{RID}))^{\beta_{\rho(i)}}\right)}\right)^{\omega_i} \\ &= \prod_{i\in I}\left(\frac{e\left(g_1^{\boldsymbol{M}_i\cdot\vec{\boldsymbol{v}}_B}\cdot g_1^{\beta_{\rho(i)}r_{\beta,i}},H(\mathrm{RID})\right)}{e\left(g_1^{r_{\beta,i}},(H(\mathrm{RID}))^{\beta_{\rho(i)}}\right)}\right)^{\omega_i} \\ &= \prod_{i\in I} e\left(g_1^{\boldsymbol{M}_i\cdot\vec{\boldsymbol{v}}_B},H(\mathrm{RID})\right)^{\omega_i} \\ &= \prod_{i\in I} e(g_1,H(\mathrm{RID}))^{\sum_{i\in I}\boldsymbol{M}_i\cdot\vec{\boldsymbol{v}}_B\cdot\omega_i} \\ &= e(g_1,H(\mathrm{RID}))^{-s} \end{aligned} \tag{4}$$

最后，在等式 (5) 中验证 $\mathrm{msg}' = \mathrm{msg}$。

$$\begin{aligned} \mathrm{msg}' &= \mathrm{MC}_k/D = \mathrm{MC}_k/(D_R\cdot D_A) \\ &= \frac{\mathrm{MC}_k}{e(g_1,H(\mathrm{RID})\mathrm{RP}_k)^s e(g_1,H(\mathrm{RID}))^{-s}} \\ &= \mathrm{msg}\cdot e(g_1^s,\mathrm{RP}_k)/e(g_1,\mathrm{RP}_k)^s \\ &= \mathrm{msg} \end{aligned} \tag{5}$$

因此，MFSAC-AR 方案满足正确性。

5.2 安全性证明

本节证明本文提出的 MFSAC-AR 方案满足完全适应性安全。

定理 1 如果文献 [17] 提出的合数阶完全适应性安全 MA-ABE 方案是安全的，那么 MFSAC-AR 方案满足完全适应性安全。

证明 假设存在一个 PPT 敌手 $\mathcal{A}$ 以不可忽略的优势攻破 MFSAC-AR 方案，以敌手 $\mathcal{A}$ 作为子程序构造的 PPT 模拟者 $\mathcal{B}$ 能够在和挑战者 $\mathcal{C}$ 的交互过程中，以不可忽略的优势攻破方案[17]。

首先，挑战者 $\mathcal{C}$ 产生复合阶群 $\mathrm{G} = (N = p_1p_2p_3, \mathbb{G},\mathbb{G}_T,e) \leftarrow \mathcal{G}(1^\lambda)$，并选择子群 $\mathbb{G}_{p_1}$ 的一个生成元 g_1。$\mathcal{C}$ 投掷随机硬币 β，若 $\beta = 0$，令群元素 $T_\beta \leftarrow \mathbb{G}$，若 $\beta = 1$，令群元素 $T_\beta \leftarrow \mathbb{G}_{p_1}$。然后，模拟者 $\mathcal{B}$ 接收到来自挑战者 $\mathcal{C}$ 发送的 (G,g_1,T_β)，并执行如下流程。

系统设置阶段：$\mathcal{B}$ 向挑战者 $\mathcal{C}$ 发起询问，并生成全局参数 GP 和全局属性私钥 $\{\mathrm{ASK}_u\}_{u\in\mathcal{U}}$。

1）固定参数 FP 生成：$\mathcal{B}$ 向挑战者 $\mathcal{C}$ 发起全局设置询问，$\mathcal{C}$ 随机选择 $h \leftarrow \mathbb{G}_{p_1}$ 并发送 (G,g_1,h) 给 $\mathcal{B}$，$\mathcal{B}$ 设置固定参数 $\mathrm{FP} = (G,g_1)$。

2）属性参数 AP 生成：$\mathcal{B}$ 发起授权方公钥询问。挑战者 $\mathcal{C}$ 为属性 $u \in \mathcal{U}$ 选择属性主密钥 $\{\mathrm{MSK}_u = (y_{A,u}, y_{B,u} \in \mathbb{Z}_N)\}_{u\in\mathcal{U}}$，并秘密保存，计算属性公钥 $P_{A,u} = g_1^{y_{A,u}}, P_{B,u} = g_1^{y_{B,u}}$，挑战者 $\mathcal{C}$ 发送属性公钥 $\{(P_{A,u},P_{B,u})\}_{u\in\mathcal{U}}$ 给 $\mathcal{B}$。$\mathcal{B}$ 为属性 $u \in \mathcal{U}$ 选择随机数 $a_u,b_u \in \mathbb{Z}_N$，并隐含设置属性 u 的全局属性私钥为 $\mathrm{ASK}_u = (\alpha_u = y_{A,u} + a_u, \beta_u = y_{B,u} + b_u)$，计算 $\mathrm{APK}_{A,u} = P_{A,u}g_1^{a_u}, \mathrm{APK}_{B,u} = P_{B,u}g_1^{b_u}$。属性参数记为 $\mathrm{AP} = \{\mathrm{APK}_{A,u},\mathrm{APK}_{B,u}\}_{u\in\mathcal{U}}$。

3）全局参数 GP 生成：$\mathcal{B}$ 选择参数 $\tau_0 \in \mathbb{Z}_N$，并计算初始撤销参数 $\mathrm{RP}_0 = h^{\tau_0}$。$\mathcal{B}$ 构造全局参数为 $\mathrm{GP} = (\mathrm{FP},\mathrm{RP}_0,\mathrm{AP})$。

询问阶段 I：敌手 $\mathcal{A}$ 适应性地进行多项式次询问操作。

1）Hash 询问：$\mathcal{B}$ 构造 hash 列表 HL 用于存储身份-hash 对。$\mathcal{A}$ 提交身份信息 RID 给 $\mathcal{B}$。如果没有询问过 RID 的 hash 值，那么 $\mathcal{B}$ 将 RID 发送给挑战者 $\mathcal{C}$，$\mathcal{C}$ 随机选择 $H(\mathrm{RID}) \leftarrow \mathbb{G}$，并发送给 $\mathcal{B}$。$\mathcal{B}$ 将 $\{\mathrm{RID},H(\mathrm{RID})\}$ 存入 HL，并将 $H(\mathrm{RID})$ 发送给 $\mathcal{A}$。如果已经询问过

RID的 hash 值，$\mathcal{B}$将{RID,H(RID)}从 HL 取出，并发送H(RID)给$\mathcal{A}$。

2）全局属性私钥询问：敌手$\mathcal{A}$询问全局属性$u\in\mathcal{U}$的全局属性私钥。$\mathcal{B}$向挑战者$\mathcal{C}$询问u对应的属性主密钥$\mathrm{MSK}_u=(y_{A,u},y_{B,u})$。然后，$\mathcal{B}$计算$u$的全局属性私钥为$\mathrm{ASK}_u=(\alpha_u=y_{A,u}+a_u,\beta_u=y_{B,u}+b_u)$，并发送给$\mathcal{A}$。

3）初始属性密钥更新询问：首先，敌手$\mathcal{A}$提交身份RID和询问属性集$S\subset\mathcal{U}$给$\mathcal{B}$。如果$\mathcal{A}$已经询问过属性$x\in S$对应的全局属性私钥，则$\mathcal{B}$计算$\mathrm{KRC}_{x,0}=(H(\mathrm{RID})\cdot RP_0)^{\alpha_x}$，$\mathrm{KAC}_{x,0}=(H(\mathrm{RID}))^{\beta_x}$。对于未询问过全局属性私钥的属性$x\in S$，$\mathcal{B}$将$H(\mathrm{GID})=H(\mathrm{RID})^{1/\tau_0}$和$x$发送给$\mathcal{C}$。$\mathcal{C}$对属性$x$计算属性密钥$K_{\mathrm{GID},A,x}=(H(\mathrm{GID})\cdot h)^{y_{A,x}}$,$K_{\mathrm{GID},B,x}=(H(\mathrm{GID}))^{y_{B,x}}$。然后，$\mathcal{C}$发送$\mathrm{SK}_{\mathrm{GID},,x}=(K_{\mathrm{GID},A,x},K_{\mathrm{GID},B,x})$给$\mathcal{B}$。最后，$\mathcal{B}$为$x$计算初始属性密钥为$\mathrm{AK}_{x,0}=(\mathrm{KRC}_{x,0},\mathrm{KAC}_{x,0})$，其中：

$$\begin{cases}\mathrm{KRC}_{x,0}=K_{\mathrm{GID},A,x}^{\tau_0}(H(\mathrm{RID})\cdot\mathrm{RP}_0)^{a_x}\\ \mathrm{KAC}_{x,0}=K_{\mathrm{GID},B,x}^{\tau_0}(H(\mathrm{RID}))^{b_x}\end{cases}$$

4）密钥更新询问：敌手$\mathcal{A}$询问身份属性集对(RID,S')的第k次更新的属性密钥。$\mathcal{B}$选择随机数$\tau_k\in\mathbb{Z}_N$，并更新撤销参数$\mathrm{RP}_k=h^{\tau_k}$。

对于新属性集S'，如果$\mathcal{B}$计算过属性$x\in S'$的全局属性私钥，则更新$\mathrm{KRC}_{x,k}=(H(\mathrm{RID})\cdot RP_k)^{\alpha_x}$，$\mathrm{KAC}_{x,k}=(H(\mathrm{RID}))^{\beta_x}$。否则$\mathcal{B}$将$H(\mathrm{GID})=H(\mathrm{RID})^{1/\tau_k}$和$x$发送给$\mathcal{C}$。$\mathcal{C}$计算$K_{\mathrm{GID},A,x}=(H(\mathrm{GID})\cdot h)^{y_{A,x}}$,$K_{\mathrm{GID},B,x}=(H(\mathrm{GID}))^{y_{B,x}}$。$\mathcal{B}$更新最新的属性密钥为$\mathrm{AK}_{x,k}=(\mathrm{KRC}_{x,k},\mathrm{KAC}_{x,k})$，其中：

$$\begin{cases}\mathrm{KRC}_{x,k}=K_{\mathrm{GID},A,x}^{\tau_k}(H(\mathrm{RID})\cdot\mathrm{RP}_k)^{a_x}\\ \mathrm{KAC}_{x,k}=K_{\mathrm{GID},B,x}^{\tau_k}(H(\mathrm{RID}))^{b_x}\end{cases}$$

挑战阶段：敌手$\mathcal{A}$提交两个等长的消息msg_0、msg_1和 LSSS 访问策略$(\boldsymbol{M},\rho)$。此外，$\mathcal{A}$提交访问策略中涉及的部分属性对应的全局属性公钥{APK_u}（满足安全模型定义的约束）。$\mathcal{B}$将数据$\mathrm{msg}'_0=\mathrm{msg}_0^{1/\tau_k}$，$\mathrm{msg}'_1=\mathrm{msg}_1^{1/\tau_k}$和$(\boldsymbol{M},\rho)$发送给$\mathcal{C}$。挑战者$\mathcal{C}$投掷随机硬币$c\leftarrow\{0,1\}$，并按照如下流程加密消息$\mathrm{msg}'_c$。

$\mathcal{C}$从$\mathbb{Z}_N$中随机选择$s,y_2,\cdots,y_d,y'_2,\cdots,y'_d$用于构造列向量$\vec{\boldsymbol{v}}_A=(s,y_2,\cdots,y_d)$,$\vec{\boldsymbol{v}}_B=(-s,y'_2,\cdots,y'_d)$，并计算$C=\mathrm{msg}'_c e(g_1,h)^s$。对于$i\in[1,l]$，$\mathcal{C}$随机选择$r_{A,i},r_{B,i}\leftarrow\mathbb{Z}_N$，计算$C_{1,A,i}=g_1^{r_{A,i}}$，$C_{2,A,i}=g_1^{\boldsymbol{M}_i\cdot\vec{\boldsymbol{v}}_A}\cdot P_{A,\rho(i)}^{r_{A,i}}$，$C_{1,B,i}=g_1^{r_{B,i}}$和$C_{2,B,i}=g_1^{\boldsymbol{M}_i\cdot\vec{\boldsymbol{v}}_B}\cdot P_{B,\rho(i)}^{r_{B,i}}$。

$\mathcal{B}$构造列向量$\vec{\boldsymbol{v}}'_A=(s',z_2,\cdots,z_n)\in\mathbb{Z}_N^d$和$\vec{\boldsymbol{v}}'_B=(-s',z'_2,\cdots,z'_d)\in\mathbb{Z}_N^d$，计算$\mathrm{APC}=\{\mathrm{AC}_{A,i,1},\mathrm{AC}_{A,i,2},\mathrm{AC}_{B,i,1},\mathrm{AC}_{B,i,2}\}_{i\in[1,l]}$，其中：

$$\begin{cases}\mathrm{AC}_{A,i,1}=C_{1,A,i},\mathrm{AC}_{A,i,2}=C_{2,A,i}\cdot g_1^{\boldsymbol{M}_i\cdot\vec{\boldsymbol{v}}_A}\cdot C_{1,A,i}^{a_x}\\ \mathrm{AC}_{B,i,1}=C_{1,B,i},\mathrm{AC}_{B,i,2}=C_{2,B,i}\cdot g_1^{\boldsymbol{M}_i\cdot\vec{\boldsymbol{v}}_B}\cdot C_{1,B,i}^{b_x}\end{cases}$$

当敌手$\mathcal{A}$进行密钥更新询问时，$\mathcal{B}$更新消息相关密文$\mathrm{MC}_k=C^{\tau_k}\cdot e(g_1,h)^{s'\cdot\tau_k}$。

询问阶段 II：敌手$\mathcal{A}$可以在满足挑战阶段约束的基础上，执行多次询问阶段 I 中的所有询问。

猜测阶段：敌手$\mathcal{A}$提交对于c的猜测c'，模拟者$\mathcal{B}$提交c'给$\mathcal{C}$，如果$c=c'$，那么敌手$\mathcal{A}$获胜。

显然，$\mathcal{B}$可以利用$\mathcal{A}$在攻破完全适应性安全分布式 MA-ABE 方案[17]中获得不可忽略的优势。由于文献 [17] 是完全适应性安全的，因此本方案满足完全适应性安全。

6　性能评估

6.1　环境配置与实验设置

实验选用操作系统为 64 位 Windows 10 计算机，处理器为 Intel(R) Core(TM) i5-7300HQ (2.50GHz)，内存为 16GB。在 VSCode 中配置 Java 环境，版本为 OpenJDK 11.0.24，并配置 JPBC 库[37]。选取 Type A1 类型的椭圆曲线并构造三素数阶复合群，每个素数的位长为 256 位。

基于上述配置，基本操作的执行时长如表 2 所示，所有结果均为测试 1 000 次所得的平均值。由于四则运算时间较短，计算方法为统计 1 000 次运算的总时间除以 1 000。$\mathbb{Z}_N$中元素的四则运算计算 1 000 次的总时间不足 1 ms，因此不统计在表 2 中。

为公平对比，设置每个 AA 管理一个属性，为 3 个方案选择相同的子群$\mathbb{G}_{p_1}$的生成元g_1，选取相同的msg。为了更好地突出方案的性能和访问策略的关系，选择相同的与门访问策略转换的 LSSS。转换算法采用文献 [38] 提出的高效生成方案。

表 2　基本操作执行时长

Table 2　Duration of basic operations

群类型	运算类型	时间/ms
$\mathbb{G}$群及子群元素	四则运算	0.1
	指数	122
配对操作	—	66
$\mathbb{G}_T$群元素	四则运算	≈0
	指数	8

6.2　理论分析

本节对 Datta 等[17]方案、Yang 等[19]方案和本文

方案进行理论分析。基础阶段的计算开销和撤销阶段的计算开销对比如表 3 和表 4 所示。其中，U表示属性全集中的属性总数；$|S|$和$|S'|$分别表示 DR 初始和更新后的属性数量；l表示访问策略行数，I表示用于解密的授权集的属性数量。$|H|$表示 hash 函数的时间开销。形式如$T_{\mathbb{X},X}$表示群$\mathbb{X}$（或其子群）的元素执行X运算的时间开销，$X=p$代表指数运算，$X=F$表示四则运算。$T_{\mathbb{G},p}$表示群$\mathbb{G}$及其子群元素执行指数运算的时间开销，T_{Pair}表示配对操作的计算开销。

首先，对比基础阶段（包括属性参数设置、密钥生成、加密和解密）的计算开销。由于选择元素的时间和$\mathbb{Z}_N$群元素四则运算时间几乎为 0，则加密阶段构造 LSSS 和计算秘密共享份额总时间开销$T_L=5l^2T_{\mathbb{Z},F}$，解密阶段$\vec{\omega}$计算的时间开销$T_\omega=\left(2I^2+\frac{4(I-1)I(I+1)}{3}\right)T_{\mathbb{Z},F}$都可以忽略不计。因此，在理论分析小节，属性参数设置阶段只统计计算全局属性公钥的时间开销。密钥生成阶段统计计算属性密钥的时间开销。加密阶段统计产生密文的时间开销，解密阶段统计D和msg′的时间开销。

其次，分析撤销阶段的计算开销。对于 Yang 等[19]方案是否留存密文计算过程中的部分随机值，会影响撤销阶段密文更新开销，因此对两种不同情况均进行分析。其中，Yang 等-1[19]方案考虑留存计算密文时的部分随机值，这种方式无需更新密文中与撤销无关的部分。Yang 等-2[19]方案在计算密文后不保存随机值，这需要重新计算密文。表 4 展示了这两个方案撤销阶段的计算开销对比。

6.3 实验对比

根据前面配置的实验环境和设置，对 3 个方案不同阶段的计算开销和存储开销进行对比。

属性参数设置阶段的计算开销和存储开销分别如图 2 和图 3 所示，由于复合阶群中$\mathbb{G}$群及其子群元素的指数操作时间更长，因此，Yang 等[19]方案计算开销最低。在存储开销方面，为保障属性参数及方案安全性，本文为每个 AA 分配全局唯一身份标识。该设计引入了额外的存储开销，每个身份标识为$\mathbb{Z}_N$群元素。因此，本文方案在全局身份管理上的额外存储开销为$n|\mathbb{Z}_N|$，其中，n表示 AA 的总数，$|\mathbb{Z}_N|$是每个身份标识的存储开销。显然，该开销仅与 AA 数量呈严格的线性关系。

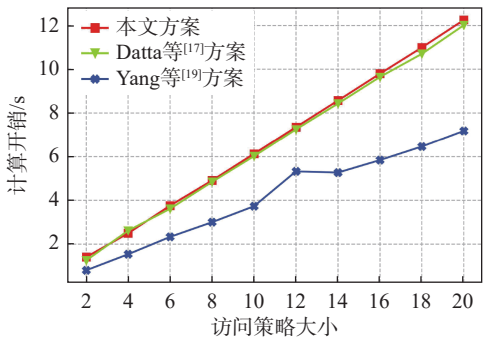

图 2 属性参数设置计算开销

Fig. 2 Computing cost of AP setup

表 3 基础阶段的计算开销对比

Table 3 Comparison of computing cost in the foundation phase

方案	属性参数设置	密钥生成	加密	解密
Datta等[17]	$2UT_{\mathbb{G},p}$	$\|H\|+\|S\|(2T_{\mathbb{G}_T,p}+T_{\mathbb{G},F})$	$T_{\mathbb{G}_T,F}+T_{\mathbb{G}_T,p}+T_{\text{Pair}}+l(6T_{\mathbb{G},p}+2T_{\mathbb{G},F})$	$\|H\|+T_{\mathbb{G}_T,F}+I(T_{\mathbb{G},F}+2T_{\mathbb{G}_T,F}+T_{\mathbb{G}_T,p}+4T_{\text{Pair}})$
Yang等[19]	$U(T_{\mathbb{G},p}+T_{\mathbb{G}_T,p}+T_{\text{Pair}})$	$2\|H\|+\|S\|(4T_{\mathbb{G},p}+2T_{\mathbb{G},F})$	$T_{\mathbb{G}_T,F}+T_{\mathbb{G}_T,p}+T_{\text{Pair}}+l(4T_{\mathbb{G},p}+T_{\mathbb{G},F}+2T_{\mathbb{G}_T,p}+T_{\mathbb{G}_T,F})$	$\|H\|+T_{\mathbb{G}_T,F}+I(4T_{\mathbb{G},F}+T_{\mathbb{G}_T,F}+3T_{\text{Pair}})$
本文	$2UT_{\mathbb{G},p}$	$\|H\|+\|S\|(2T_{\mathbb{G},p}+T_{\mathbb{G},F})$	$T_{\mathbb{G}_T,F}+T_{\mathbb{G}_T,p}+T_{\text{Pair}}+l(6T_{\mathbb{G},p}+2T_{\mathbb{G},F})$	$\|H\|+T_{\mathbb{G}_T,F}+I(T_{\mathbb{G},F}+2T_{\mathbb{G}_T,F}+T_{\mathbb{G}_T,p}+4T_{\text{Pair}})$

表 4 撤销阶段的计算开销对比

Table 4 Comparison of computing cost in the revocation phase

方案	系统更新	密钥更新	密文更新
Yang等-1[19]	-	$2\|H\|+\|S'\|(4T_{\mathbb{G},p}+2T_{\mathbb{G},F})$	$lT_{\mathbb{G},p}+\|H\|$
Yang等-2[19]	-	$2\|H\|+\|S'\|(4T_{\mathbb{G},p}+2T_{\mathbb{G},F})$	$T_{\mathbb{G}_T,F}+T_{\mathbb{G}_T,p}+T_{\text{Pair}}+l(4T_{\mathbb{G},p}+T_{\mathbb{G},F}+2T_{\mathbb{G}_T,p}+T_{\mathbb{G}_T,F})$
本文	$t(T_{\mathbb{G},p}+T_{\mathbb{G},F})$	$\|H\|+\|S'\|(T_{\mathbb{G},p}+T_{\mathbb{G},F})$	$T_{\text{Pair}}+T_{\mathbb{G}_T,F}$

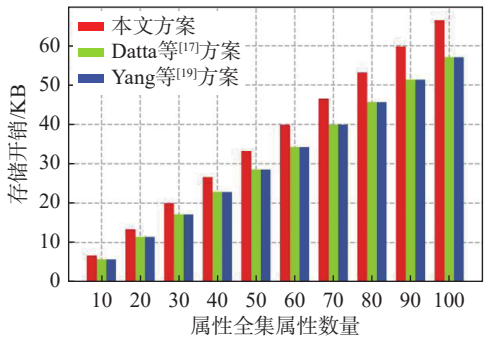

图 3　属性参数设置存储开销

Fig. 3　Storage cost of AP setup

在密钥生成阶段，本文方案与 Datta 等[17]方案构造思想一致，密钥生成算法本质相同，时间开销也相同。Yang 等[19]方案密钥生成算法最复杂，时间开销最大。密钥生成计算开销如图 4 所示。但是 3 个方案的密钥形式是一致的，因此，在图 5 中，3 个方案的密钥存储开销相同。

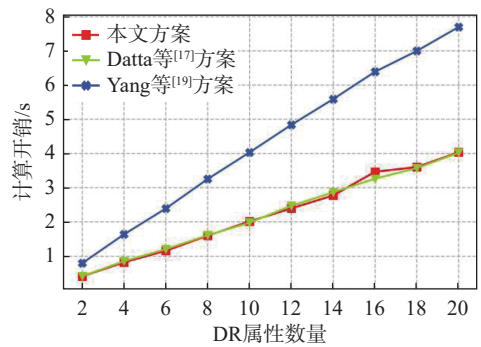

图 4　密钥生成计算开销

Fig. 4　Computing cost of KeyGen

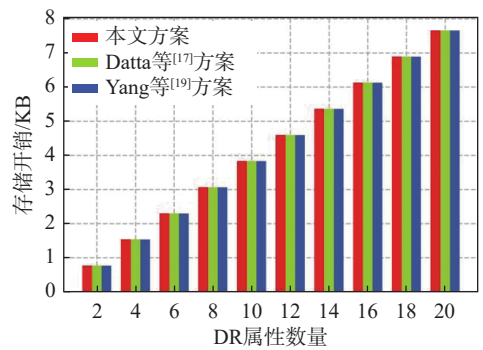

图 5　密钥存储开销

Fig. 5　Storage cost of key

加密计算开销如图 6 所示。由于配对运算的时间开销远小于𝔾群及其子群元素的指数操作时间，Yang 等[19]方案中与访问策略相关的密文部分计算开销小于本文方案和 Datta 等[17]方案。由于 3 个方案的密文结构一致，因此密文存储开销相同，如图 7 所示。

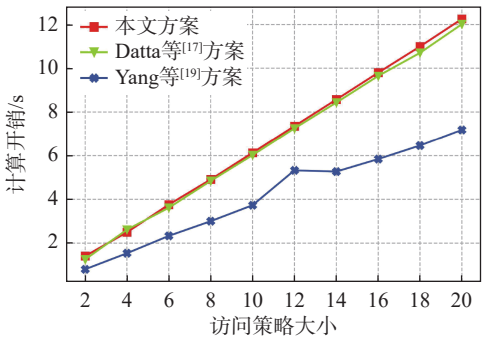

图 6　加密计算开销

Fig. 6　Computing cost of encryption

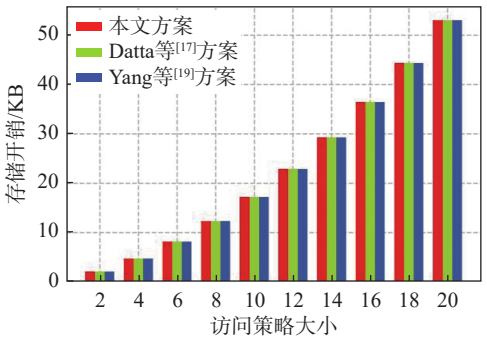

图 7　密文存储开销

Fig. 7　Storage cost of ciphertext

在解密阶段，本文方案与 Datta 等[17]方案解密算法复杂度一致，不存在𝔾群及其子群元素的指数操作，配对操作的计算开销是主体。Yang 等[19]方案解密算法的计算开销约为另外两个方案的 75%，计算开销对比如图 8 所示。

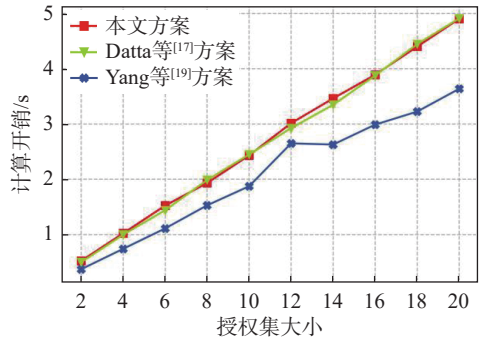

图 8　解密计算开销

Fig. 8　Computing cost of decryption

在撤销阶段，本文额外引入系统更新算法，该算法涉及网络通信延迟和参数共识同步引发的开销。其中，通信开销仅涉及 $RP_{\{k-1,i\}}$ 和 RP'_k 两个群元素。此外，算法 2 允许 AA 在本地异步计算 $RP_{\{k-1,i\}}$，仅在最终经过一次一致性确认，因此，系统更新开销在合理范围内。

在密钥更新阶段，由于不改变密钥的存储形式，仅当更新后的属性总数发生变化时，密钥的存储开销才会按比例变化。因此，密钥更新关注计算开销。如图 9 所示，Yang 等[19]方案的密钥更新意味着重新计算所有的密钥。密钥更新的计算开销与密钥生成的计算开销一致。本文方案基于基础阶段引入的撤销参数，在撤销阶段仅需更新密钥撤销组件，密钥更新开销约为 Yang 等[19]方案的 25%。特别是在高动态大规模撤销场景中，本文方案的密钥更新计算开销远小于 Yang 等[19]方案。

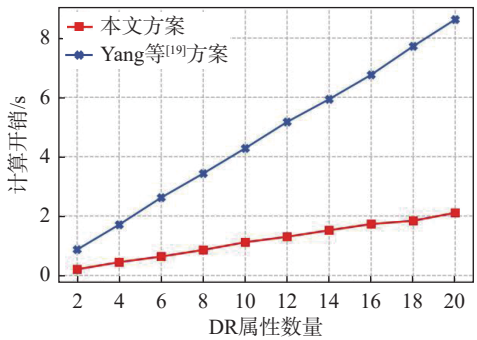

图 9 密钥更新计算开销

Fig. 9 Computing cost of KeyUpd

在密文更新阶段，对于 Yang 等[19]方案，实际场景中不同的密文更新方式会导致不同的计算开销和额外存储开销。如图 10 和图 11 所示，Yang 等−1[19]方案考虑留存计算密文时的部分随机值。这种方式降低了密文更新的计算开销，但引入了额外的随机数存储开销。Yang 等−2[19]方案在计算密文后不保存随机值，这种方式额外存储开销为 0，但是需要完整计算整个密文。本文的密文更新开销为常数级，仅执行一次配对运算和一次乘法运算，计算开销约为 0.065 s。额外存储开销仅包含密文更新组件 CUC。因此，在高动态撤销需求下，本文设计的轻量级密文更新机制不受密文访问策略大小影响，密文更新更高效。

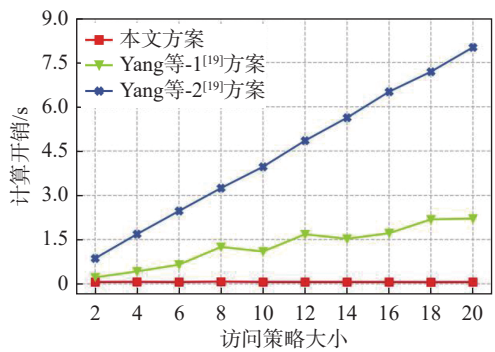

图 10 密文更新计算开销

Fig. 10 Computing cost of CTUpd

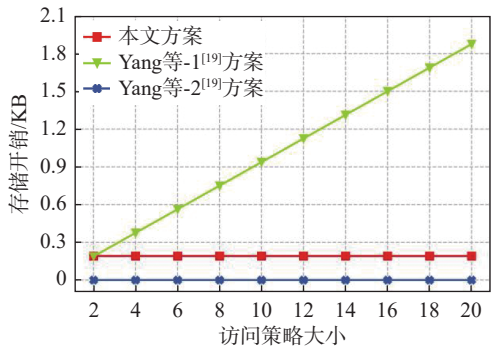

图 11 密文更新额外存储开销

Fig. 11 Addition storage cost of CTUpd

总的来说，由于本文方案和 Datta 等[17]方案设计思想一致，两个方案在基础阶段的开销基本一致，本文方案仅在属性参数设置阶段额外存储全局身份标识。与 Yang 等[19]方案相比，两个方案为确保完全适应性安全，整体上增加了计算开销。在撤销阶段，本文方案考虑撤销的高效性，与同样为复合阶群构造的 Yang 等[19]方案相比，本文方案在密钥更新和密文更新两个阶段性能更高。

7 结束语

本文针对 MA-ABE 方案的部分密钥泄露问题和用户属性撤销问题，提出了支持属性撤销的多授权方完全适应性安全访问控制方案，在确保完全适应性安全的基础上，设计撤销参数并构建高效的属性更新算法，实现了灵活的多授权方属性撤销。为保障前后向安全，本文提出了密文更新组件并提出了轻量级密文更新方案，安全性证明表明 MFSAC-AR 方案满足完全适应性安全；最后对方案进行理论分析和实验对比，结果表明 MFSAC-AR 方案在撤销阶段具有高效性。未来，将针对合数阶群方案存在的参数长度长、运算开销大的问题，进一步基于素数阶群构造满足完全适应性安全的可撤销多授权方访问控制方案，并结合可验证外包计算技术，提升方案的实用性。

参考文献：

[1] Zha D，Bhat Z P，Lai K H，et al. Data-centric artificial intelligence：a survey[J]. ACM Computing Surveys，2025，57（5）：1-42.

[2] 牛翔宇，孔兰菊，蒋亚丽，等. 面向多层级区块链架构的轻量级高效验证资产跨链转移方法 [J]. 计算机研究与发展，2025，62（11）：2870-2887.

Niu X Y，Kong L J，Jiang Y L，et al. A lightweight and effi-

ciently verified assets cross-chain transfer method for multi-level blockchains architecture[J]. Journal of Computer Research and Development，2025，62（11）：2870-2887.

[3] Nawrocki P，Osypanka P，Posluszny B. Data-driven adaptive prediction of cloud resource usage[J]. Journal of Grid Computing，2023，21（1）：1-19.

[4] Farayola O A，Olorunfemi O L，Shoetan P O. Data privacy and security in it：a review of techniques and challenges[J]. Computer Science & IT Research Journal，2024，5（3）：606-615.

[5] Zhang Y，Deng R，Xu S，et al. Attribute-based encryption for cloud computing access control：a survey[J]. ACM Computing Surveys（CSUR），2020，53（4）：1-41.

[6] Wang X，Yu M，Wang Y，et al. Attribute-based access control encryption[J]. IEEE Transactions on Dependable and Secure Computing，2025，22（3）：2227-2242.

[7] Ghopur D，Ma J，Ma X，et al. Puncturable ciphertext-policy attribute-based encryption scheme for efficient and flexible user revocation[J]. Science China Information Sciences，2023，66（7）：172104.

[8] Gu C，Li J，Zhang Y，et al. EABE-PUFPH：Efficient attribute-based encryption with reliable policy updating under full policy hiding[J]. IEEE Transactions on Computers，2025，74（11）：3750-3762.

[9] 苏泽林，张文芳，王小敏. 支持策略更新和即时密文验证的外包属性基加密方案 [J]. 计算机研究与发展，2024，61（12）：3088-3097.
Su Z L，Zhang W F，Wang X M. Outsourced attribute-based encryption scheme with policy updating and verifiable ciphertext[J]. Journal of Computer Research and Development，2024，61（12）：3088-3097.

[10] Sravya G，Kumar P S，Padmavathy R. Survey of post-quantum lattice-based ciphertext-policy attribute-based encryption schemes for cloud storage：taxonomy，open issues，and future directions[J]. IEEE Transactions on Services Computing，2024，17（6）：4540-4557.

[11] 周权，卫凯俊，陈民辉，等. 基于区块链的国密轻量级属性基访问控制方案 [J]. 密码学报，2024，11（5）：1126-1138.
Zhou Q，Wei K J，Chen M H，et al. Blockchain-based domestic cryptographic lightweight attribute-based access control scheme[J]. Journal of Cryptologic Research，2024，11（5）：1126-1138.

[12] Chen L，Tai Z，Zhang H，et al. Traceable and revocable multi-authority attribute-based encryption with cloud and fog computing feasible for IoV[J]. IEEE Transactions on Dependable and Secure Computing，2026.

[13] Oberko P S，Obeng V H，Xiong H. A survey on multi-authority and decentralized attribute-based encryption[J]. Journal of Ambient Intelligence and Humanized Computing，2022，13（1）：515-533.

[14] Duan P，Ma Z，Gao H，et al. Multi-authority attribute-based encryption scheme with access delegation for cross blockchain data sharing[J]. IEEE Transactions on Information Forensics and Security，2024，20：323-337.

[15] Lin Y，Xiong H，Su H，et al. Multi-authority CP-ABE scheme with cryptographic reverse firewalls for internet of vehicles[J]. IEEE Transactions on Intelligent Transportation Systems，2025，26（4）：5348-5359.

[16] Lewko A，Waters B. Decentralizing attribute-based encryption[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin，Heidelberg：Springer Berlin Heidelberg，2011：568-588.

[17] Datta P，Komargodski I，Waters B. Fully adaptive decentralized multi-authority ABE[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. Cham：Springer Nature Switzerland，2023：447-478.

[18] Zuo Y，Xu L，Li J，et al. Secure and efficient blockchain-based access control scheme with attribute update[J]. IEEE Transactions on Consumer Electronics，2025，71（1）：1539-1550.

[19] Yang F，Cui H，Jing J. Decentralized attribute-based access control with attribute revocation and outsourced decryption[C]//2023 15th International Conference on Computer Research and Development（ICCRD）. IEEE，2023：246-257.

[20] Chase M. Multi-authority attribute based encryption[C]//Theory of Cryptography：4th Theory of Cryptography Conference，TCC 2007，Amsterdam，the Netherlands，Springer Berlin Heidelberg，2007：515-534.

[21] Lin H，Cao Z，Liang X，et al. Secure threshold multi authority attribute based encryption without a central authority[J]. Information Sciences，2010，180（13）：2618-2632.

[22] 张学旺，姚亚宁，付佳丽，等. 策略隐藏的高效多授权机构 CP-ABE 物联网数据共享方案 [J]. 计算机研究与发展，2023，60（10）：2193-2202.
Zhang X W，Yao Y N，Fu J L，et al. Efficient multi-authority CP-ABE IoT data sharing scheme with hidden policies[J]. Journal of Computer Research and Development，2023，60（10）：2193-2202.

[23] Waters B. Ciphertext-policy attribute-based encryption：An expressive，efficient，and provably secure realization[C]//International Workshop on Public Key Cryptography. Berlin，Hei-

delberg：Springer Berlin Heidelberg，2011：53-70.

[24] 王静怡，阚海斌. mHealth 中细粒度策略隐藏和可追踪去中心访问控制方案 [J]. 计算机研究与发展，2024，61（6）：1525-1535.
Wang J Y，Kan H B. Fine-grained policy-hiding and traceable decentralized access control scheme in mHealth[J]. Journal of Computer Research and Development，2024，61（6）：1525-1535.

[25] Li W，Xue K，Xue Y，et al. TMACS：a robust and verifiable threshold multi-authority access control system in public cloud storage[J]. IEEE Transactions on Parallel and Distributed Systems，2016，27（5）：1484-1496.

[26] Qin X，Huang Y，Yang Z，et al. A blockchain-based access control scheme with multiple attribute authorities for secure cloud data sharing[J]. Journal of Systems Architecture，2021，112：101854.

[27] Ambrona M，Gay R. Multi-authority ABE，revisited[J/OL]. Cryptology ePrint Archive，2021. [2025-10-23] https://ia.cr/2021/1381.

[28] Chen J，Chu Q，Gao Y，et al. Improved fully adaptive decentralized MA-ABE for NC1 from MDDH[C]//International conference on the Theory and Application of Cryptology and Information Security. Singapore：Springer Nature Singapore，2023：3-32.

[29] Hur J，Noh D K. Attribute-based access control with efficient revocation in data outsourcing systems[J]. IEEE Transactions on Parallel and Distributed Systems，2011，22（7）：1214-1221.

[30] 王鹏翩，冯登国，张立武. 一种支持完全细粒度属性撤销的 CP-ABE 方案 [J]. 软件学报，2012，23（10）：2805-2816.
Wang P P，Feng D G，Zhang L W. CP-ABE scheme supporting fully fine-grained attribute revocation[J]. Journal of Software，2012，23（10）：2805-2816.

[31] Li J，Yao W，Han J，et al. User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage[J]. IEEE Systems Journal，2018，12（2）：1767-1777.

[32] 肖浩，徐子慧，姜奇，等. 面向工业物联网高效可撤销的属性基访问控制 [J]. 网络空间安全科学学报，2025，3（2）：84-95.
Xiao H，Xu Z H，Jiang Q，et al. Efficient and revocable attribute-based access control for industrial internet of things[J]. Journal of Cybersecurity，2025，3（2）：84-95.

[33] Yang K，Jia X，Ren K，et al. DAC-MACS：Effective data access control for multiauthority cloud storage systems[J]. IEEE Transactions on Information Forensics and Security，2013，8（11）：1790-1801.

[34] Hong J，Xue K，Li W. Comments on “DAC-MACS：effective data access control for multiauthority cloud storage systems”/security analysis of attribute revocation in multiauthority data access control for cloud storage systems[J]. IEEE Transactions on Information Forensics and Security，2015，10（6）：1315-1317.

[35] Tu S，Waqas M，Huang F，et al. A revocable and outsourced multi-authority attribute-based encryption scheme in fog computing[J]. Computer Networks，2021，195：108196.

[36] Liu Z，Jiang Z L，Wang X，et al. Practical attribute-based encryption：outsourcing decryption，attribute revocation and policy updating[J]. Journal of Network and Computer Applications，2018，108：112-123.

[37] Decaro A，Iovino V. jPBC：Java pairing based cryptography[C]//2011 IEEE Symposium on Computers and Communications（ISCC）. IEEE，2011：850-855.

[38] Liu Z，Cao Z，Wong D S. Efficient generation of linear secret sharing scheme matrices from threshold access trees[DB/OL]. Cryptology ePrint Archive，2010. [2025-10-23] https://ia.cr/2010/374.