格上具有多功能的属性基加密

郭丽峰, 杨杰莹, 马添军, 张夏蕾

引用本文

郭丽峰, 杨杰莹, 马添军, 张夏蕾. 格上具有多功能的属性基加密[J]. 计算机科学, 2025, 52(10): 412-422.

GUO Lifeng, YANG Jieying, MA Tianjun, ZHANG Xialei. Multi-functional Attribute Based Encryption from Lattices [J]. Computer Science, 2025, 52(10): 412-422.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.240600137

格上具有多功能的属性基加密

郭丽峰　杨杰莹　马添军　张夏蕾
山西大学计算机与信息技术学院　太原 030006

摘　要　格上属性基加密具有抗量子攻击的特性,并且将访问控制策略嵌入密文或者密钥,可实现属性的细粒度访问控制。但是由于属性基加密固有的弱点,相同属性的用户可能会泄露密钥。为避免密钥泄露,属性基加密方案需实现追踪并撤销特定用户解密权限的功能。然而,非法用户仍可能通过收集大量密文数据,试图恢复过去会话的密钥。为有效抵御这种攻击,方案必须实现前向安全。针对当前格密码领域的需求与挑战,提出基于判定性误差学习问题(Decisional Learning with Error,DLWE)可证明安全的格上具有多功能的属性基加密(Multi-functional Attribute based Encryption from Lattices)方案。使用完全二叉树追踪解密密钥中与用户相关的身份矩阵(即完全二叉树叶子节点的值),以便追踪恶意用户;引入用户撤销机制,允许属性权威在不重新为用户生成密钥的情况下,及时且有效地撤销用户的权限;采用标签穿刺的方法,确保即使当前密钥泄漏,过去密文仍然保持安全,实现前向安全。此外,由于格上采样算法的不确定性,目前格上的属性基加密实验难以实现,因此通过理论分析验证所提方案的安全性和正确性。该方案不仅优化了空间存储效率,还弥补了格密码中属性基加密方案功能单一导致的不足。

关键词:格密码;属性基加密;可追踪;前向安全;用户撤销
中图分类号　TP309

Multi-functional Attribute Based Encryption from Lattices

GUO Lifeng,YANG Jieying,MA Tianjun and ZHANG Xialei
College of Computer and Information Technology,Shanxi University,Taiyuan 030006,China

Abstract　Attribute based encryption from lattices has the property of resisting quantum attacks,and achieves fine-grained access control of attributes by cleverly embedding access control policies into ciphertext or keys. However,due to the inherent weaknesses of attribute based encryption,users with the same attribute may leak the key. To avoid key leakage,attribute based encryption schemes need to implement the function of tracking and revoking specific user decryption permissions. However,illegal users may still attempt to recover the keys of past sessions by collecting large amounts of encrypted data. To effectively resist such attacks,the scheme must implement forward security. In response to the current demands and challenges in the field of lattices cryptography,this paper proposes a multi-functional attribute based encryption scheme based on the Decisional Learning with Errors(DLWE) problem from lattices that can prove security. The scheme uses a complete binary tree to track the identity matrices related to the users in the decryption key(such as the values of the leaf nodes of the complete binary tree) in order to track malicious users. Introducing a user revocation mechanism that allows attribute authority to revoke user permissions in a timely and effective manner without generating new keys for the users. Using tag puncturing method to ensure that even if the current key is leaked,the past ciphertext remains secure and achieves forward security. In addition,due to the uncertainty of the upsampling algorithm from lattice,it is currently difficult to achieve experiments on attribute based encryption from lattice. Therefore,the security and correctness of the scheme are verified through theoretical analysis. The scheme not only optimizes space storage efficiency,but also compensates for the shortcomings caused by the lack of functions of attribute based encryption schemes on lattice cryptography.

Keywords　Lattice,Attribute based encryption,Traceable,Forward security,User revocation

1　引言

随着大数据的发展,人们对信息的依赖程度不断增加,保护敏感数据和通信安全变得愈发重要。在这一背景下,传统密码学中的公钥密码学在面对量子攻击时显得很无力。格密码为抵抗量子攻击提供了一种解决方案。然而,格上的公钥密码加密体制和传统的公钥密码加密体制一样,都只是实现一对一的加密,不能满足多对一细粒度的访问控制。因此,

到稿日期:2024-06-24　返修日期:2024-08-03
基金项目:山西省自然科学基金(202203021221012)
This work was supported by the Natural Science Foundation of Shanxi Province(202203021221012).
通信作者:郭丽峰(lfguo@sxu.edu.cn)

目前格上的属性基加密研究成为一个热点。

格上 ABE 与传统 ABE 类似，分为密文策略属性加密(Ciphertext Policy-Attribute Based Encryption，CP-ABE)和密钥策略属性加密(Key Policy-Attribute Based Encryption，KP-ABE)两大分支。在 CP-ABE 中，密文与访问策略相关联，密钥与属性相关联，当密钥的属性集满足密文的访问策略时可解密密文。KP-ABE 则相反，在密钥中嵌入访问策略，密文嵌入属性集，当密文属性集满足密钥的访问策略时才可以解密密文。2005 年，Regev[1] 指出了格、Learning with Errors (LWE)问题与随机线性码之间的关系，为基于格的密码学的发展提供了理论基础和启发。在此基础上，Boyen[2] 提出了格上的第一个 KP-ABE 方案，该方案可实现多对一细粒度的访问控制。2017 年，Kuchta 等[3] 提出了格上的分布式属性可搜索的 KP-ABE 方案，该方案使用多个云服务器来共同完成复杂的计算任务，还增加了分布式属性可搜索的功能，但这也带来了云存储空间占用过大、加解密慢等问题。2024 年，Singamaneni 等[4] 构造了一个有效的抗量子攻击的 KP-ABE 框架，该框架优化了性能，可减少云存储空间的占用以及缩短密钥生成的时间。

在 ABE 应用中，针对恶意用户攻击或合法用户密钥泄露，需撤销其解密权限。撤销机制分直接撤销和间接撤销。直接撤销在加密阶段动态实现，无需属性权威介入。间接撤销则由属性权威控制撤销列表，追踪并撤销恶意用户权限。2010 年，Sun 等[5] 提出了一种支持属性撤销的属性基加密方案，该方案不仅能够有效抵御合谋攻击，还能通过外包解密计算减轻用户的计算负担。但该方案无法实现对用户行为的全面监控。2022 年，Han 等[6] 提出了一种具有撤销、白盒追踪和策略隐藏的加密方案，该方案追踪到恶意用户身份后迅速撤销权限，同时，其策略隐藏机制也进一步提高了系统的隐私保护能力。以上都是双线性映射下的撤销方案，随着格密码的兴起，2018 年，Wang 等[7] 提出了第一个基于 LWE 问题的可间接撤销 ABE 方案，该方案是由基于 LWE 的可撤销的身份基方案[8] 和基于 LWE 的 ABE 方案[9] 结合形成的。然而，间接撤销涉及复杂的密钥更新过程。为提升撤销操作的效率，Wang 等[7] 提出了一个有效可撤销的 ABE 方案，该方案使用二叉树来处理属性撤销问题。Yang 等[10] 提出了一种基于格可撤销的 ABE 方案，该方案使用完全二叉树来撤销属性。虽然其性能和效率有所提升，但是仍受制于密钥大小和撤销复杂度。同年，Zhao 等[11] 提出了一种可撤销的适用于云存储的基于格的加密方案，该方案能够抵抗量子攻击，并保证对用户权限的细粒度访问控制，实现数据共享。该方案借助云服务器，对数据进行计算和存储，提升了计算效率。2023 年，Luo 等[12] 提出了标准格上可撤销的 KP-ABE 方案，该方案支持策略函数并且具有短密钥，但是其密钥管理和撤销的计算复杂度仍然较大。同年，Huang 等[13] 提出了一种基于格的 ABE 方案，该方案引入了间接撤销和基于二叉树的数据结构，优化了密钥管理和撤销的计算复杂度。考虑到系统中存在恶意用户，需解决追踪和撤销的集成问题。Guo 等[14] 提出了一种新的可追踪可撤销的 ABE 方案，实现了格上追踪和撤销的结合。

2015 年，Green 等[15] 提出了穿刺加密的概念，允许细粒度撤销对特定消息的解密能力。然而，由于该概念是首次提出，并未充分考虑密钥管理的复杂性，特别是在处理大量用户和大规模数据时，密钥的更新可能会变得相对复杂和耗时。2018 年，Phuong 等[16] 设计了一种新的方法，有效地集成了基于属性基加密和穿刺密钥，使得密钥大小与原始基于属性的加密的密钥大小大致相同。这意味着该方案在引入穿刺密钥功能的同时，没有显著增加密钥的存储和管理成本。2022 年，Dutta 等[17] 将基于身份的穿刺与格上的属性基加密结合起来构造了一个方案，该方案虽然在技术上有所创新，但是局限于一对一的共享模式。2024 年，Yang 等[18] 提出了格上可穿刺的 ABE 方案，该方案不仅继承了属性基加密在细粒度访问控制上的优势，还创新性地融入了穿刺加密的灵活撤销特性，突破了先前技术的限制，实现了安全且高效的一对多文件共享功能。

本文提出一种格上具有前向安全、可追踪和可撤销特性的属性基加密方案，该方案基于文献[13]提出的可撤销 KP-ABE 的定义，引入追踪和前向安全机制。本文的具体贡献如下：

1)前向安全

创新性地将穿刺算法应用于格中，有效弥补了格中算法在前向安全方面的不足，确保了密钥在穿刺后能够保护过去的通信记录。

2)高效的用户追踪机制

结合完全二叉树，提出矩阵异或法，实现身份矩阵的快速加解密，为用户身份管理和追踪提供了新的技术手段。

3)灵活且安全的访问权限撤销机制

使用撤销列表动态管理用户访问权限，为系统中的权限管理提供了更灵活、安全的解决方案。

4)职责明确的数据管理机构

数据服务器管理者负责密文更新，避免普通用户随意更改密文，从而确保加密系统的稳定性和安全性。

本文第 2 章介绍了符号说明、格上的定义和引理、UR 二叉树、完全子树法以及矩阵异或法等基础知识；第 3 章介绍了系统结构、方案算法定义和安全模型；第 4 章详细描述了具有多功能的格上的属性基加密方案；第 5 章验证了方案的正确性并进行了参数分析；第 6 章对方案进行了安全性分析；第 7 章对方案的性能了进行评估；最后总结全文。

2 预备知识

2.1 符号说明

本文相关符号含义如表 1 所列。

表 1 符号说明

Table 1 Symbol notations

符号	符号含义
Z	整数集
$\mathbb{R}$	实数集
q	一个大素数
$\boldsymbol{A}$	矩阵
$\|\boldsymbol{A}\|$	矩阵 $\boldsymbol{A}$ 的 l_2 范数
$\boldsymbol{Z}_q^{n\times m}$	模 q 下 $n\times m$ 阶矩阵的集合
$(\boldsymbol{X}\|\boldsymbol{Y})^{n\times(m_1+m_2)}$	矩阵 $\boldsymbol{X}^{n\times m_1}$ 和矩阵 $\boldsymbol{Y}^{n\times m_2}$ 的列连接
$(\boldsymbol{X};\boldsymbol{Y})^{(n+n')\times m}$	矩阵 $\boldsymbol{X}^{n\times m}$ 和矩阵 $\boldsymbol{Y}^{n'\times m}$ 的行连接

2.2 概念

定义 1(格[14]) 设 n 为正整数，$\boldsymbol{B}=\{\boldsymbol{b}_1,\boldsymbol{b}_2,\cdots,\boldsymbol{b}_m\}\subseteq\mathbb{R}^n$ 是一组线性无关的向量，$\boldsymbol{B}$ 中向量线性组合构成的集合记为格 $\Lambda=\mathcal{L}(\boldsymbol{B})=\{\boldsymbol{B}x=\sum x_i\ \boldsymbol{b}_i \mid x_i\in\boldsymbol{Z}, i\in[1,m]\}$，其中 $\boldsymbol{b}_1,\boldsymbol{b}_2,\cdots,\boldsymbol{b}_m$ 为格的基。

定义 2(q-元格[14]) 均匀随机地选取矩阵 $\boldsymbol{A}\in Z^{n\times m}$，向量 $\boldsymbol{u}\in Z_q^n$，$q\geqslant 2$，定义：

$\Lambda_q^{\perp}(\boldsymbol{A}=\{\boldsymbol{x}\in)Z^m:\boldsymbol{A}\boldsymbol{x}=0(\bmod q)\}$

$\Lambda_q^{u}(\boldsymbol{A})=\{\boldsymbol{x}\in Z^m:\boldsymbol{A}\boldsymbol{x}=u(\bmod q)\}$

定义 3(离散高斯函数[12]) 设 r 为大于 0 的实数，以参数 r、向量 $\boldsymbol{c}\in\mathbb{R}^n$ 为中心的高斯分布定义为 $(\forall\boldsymbol{x}\in\mathbb{R}^n,\rho_{r,\boldsymbol{c}}(\boldsymbol{x})=\exp\left(-\pi\frac{\boldsymbol{x}-\boldsymbol{c}}{r^2}\right)$。当满足 $\boldsymbol{c}$ 为原点或 $r=1$ 时，下标 $r,\boldsymbol{c}$ 可省略。对于 $\boldsymbol{c}\in\mathbb{R}^n$，大于 0 的实数 r 和 n 维格 L，定义格 L 上的离散高斯分布为 $\forall\boldsymbol{y}\in\mathbb{R}^n$，$D_{L,r,\boldsymbol{c}}(\boldsymbol{y})=\frac{\rho_{r,\boldsymbol{c}}(\boldsymbol{y})}{\rho_{r,\boldsymbol{c}}(L)}$，其中，对于任意一个有限集合 B 满足 $\rho_{r,B}(B)=\sum_{x\in B}\rho_{r,\boldsymbol{c}}(\boldsymbol{x})$。

定义 4(Decisional Learning with Errors[12]) 令 $n,q\geqslant 1$，$m\geqslant O(n\log q)$，$\chi=\chi(n)$ 是 Z 上的分布，定义 $DLWE_{n,q,m,\chi}$ 问题区分。

$(\boldsymbol{A},\boldsymbol{A}^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e})$ 和 $(\boldsymbol{A},\boldsymbol{u})$ 两种分布，其中 $\boldsymbol{A}\in Z_q^{n\times m}$，$\boldsymbol{s}\in Z_q^n$，$\boldsymbol{e}\leftarrow\chi^m$，$\boldsymbol{u}\leftarrow Z_q^m$。

定义 5(Gadget 矩阵[12]) 对于整数 $q\geqslant 2$，$n\geqslant 1$，定义一种特殊矩阵 $\boldsymbol{G}=\boldsymbol{I}_n\otimes\boldsymbol{g}\in Z_q^{n\times M}$，其中 $M=n\lceil\log q\rceil$，$\boldsymbol{g}=(1,2,\cdots,2^{\lceil\log q\rceil-1})\in Z_q^{\lceil\log q\rceil}$，并定义逆函数 $G^{-1}:\boldsymbol{Z}_q^{n\times m}\rightarrow\{0,1\}^{M\times M}$，对于任意矩阵 $\boldsymbol{A}\in Z_q^{n\times m}$，当 $\|G^{-1}(\boldsymbol{A})\|\leqslant M$ 时，可得到 $\boldsymbol{G}\cdot G^{-1}(\boldsymbol{A})=\boldsymbol{A}$。可通过嵌入 0 或者其他方式将 $\boldsymbol{G}\in Z_q^{n\times M}$ 拓展到 $\widetilde{\boldsymbol{G}}\in Z_q^{n\times M'}(M'>M)$，逆函数 $\widetilde{G}^{-1}$ 的定义与 G^{-1} 类似。

定义 6(KP-ABE) 设属性空间 $\mathcal{X}:\{0,1\}^l$，函数族 $\mathcal{F}=\{f:\{0,1\}^l\rightarrow\{0,1\}\}$。KP-ABE 方案由 $(Setup,keyGen,Enc,Decrypt)$ 组成。

$Setup(1^\lambda,l,N)\rightarrow\{pp,msk\}$：该算法由属性权威运行。输入安全参数 λ、属性最大长度 l 以及用户最大数目 N，输出公共参数 pp 和主私钥 msk。

$KeyGen(pp,msk,f\in\mathcal{F})\rightarrow\{sk\}$：该算法由属性权威运行。输入公共参数 pp、主私钥 msk 以及访问策略函数 f，输出私钥 sk。

$Enc(pp,\mu,att\in\mathcal{X})\rightarrow\{\boldsymbol{ct}\}$：该算法由数据拥有者运行。输入公共参数 pp、明文消息 μ、属性集 $att\in\mathcal{X}$，输出密文 $\boldsymbol{ct}$。

$Decrypt(pp,sk,att\in\mathcal{X},\boldsymbol{ct})\rightarrow\{\mu\}$：此算法由数据用户运行。输入公共参数 pp、私钥 sk、属性集 $att\in\mathcal{X}$ 和密文 $\boldsymbol{ct}$。输出分两种情况：若 $f(att)\neq 0$，则输出 $\perp$；若 $f(att)=0$，则输出明文 μ。

定义 7(Revocable and Traceable Key Policy-Attribute Based Encryption) 设属性空间 $\mathcal{X}:\{0,1\}^l$，函数族 $\mathcal{F}=\{f:\{0,1\}^l\rightarrow\{0,1\}\}$。该方案由 $Setup,keyGen,Enc,Decrypt,KeySanityCheck,Trace$ 组成。

$Setup(1^\lambda,l,N)\rightarrow\{pp,msk,RL\}$：该算法由属性权威运行。输入安全参数 λ、属性个数 l 和用户最大数目 N，初始化 $RL=\{\emptyset\}$，输出公共参数 pp，主私钥 msk 以及撤销列表 RL。

$KeyGen(pp,msk,f\in\mathcal{F},I)\rightarrow\{sk\}$：该算法由属性权威运行。输入公共参数 pp、主私钥 msk、访问策略函数 f 以及用户索引 I。根据完全二叉树和用户索引 I，可查找到身份矩阵 $\boldsymbol{U}_{i_u}$。输出密钥 sk，密钥中包含对身份矩阵 $\boldsymbol{U}_{i_u}$ 的加密信息。

$Enc(pp,\mu,att\in\mathcal{X})\rightarrow\{\boldsymbol{ct}\}$：该算法由数据拥有者运行。输入公共参数 pp、明文消息 μ、属性集 $att\in\mathcal{X}$ 以及标签集 $\{t_1,\cdots,t_d\}$，输出密文 $\boldsymbol{ct}$。

$Decrypt(pp,sk,att\in\mathcal{X},\boldsymbol{ct},RL,I)\rightarrow\{\mu\}$：此算法由数据用户运行。输入公共参数 pp、私钥 sk、属性集 $att\in\mathcal{X}$、密文 $\boldsymbol{ct}$、撤销列表 RL 以及身份索引 $I\in[N]$。输出分为两种情况：1)若 $I\in RL$，则输出 $\perp$；2)若 $I\notin RL$，则判断 $f(att)$ 是否为 0，$f(att)\neq 0$ 则输出 $\perp$，$f(att)=0$ 则输出明文 μ。

$KeySanityCheck(msk,sk,pp)\rightarrow\{1\text{ 或 }\perp\}$：此算法由属性权威运行。输入主密钥 msk、私钥 sk 以及公共参数 pp。如果 sk 通过密钥检查，则输出 1，否则输出 $\perp$。

$Trace(pp,RL,sk)\rightarrow\{\perp\text{ 或 }RL'\}$：此算法由属性权威运行。接收公共参数 pp、撤销列表 RL 以及私钥 sk。若私钥 sk 通过 $KeySanityCheck$ 算法检测，则算法输出用户的身份 u_{i_u}，并更新撤销列表 $RL'=RL\cup\{u_{i_u}\}$，否则输出 $\perp$。

定义 8(Puncturable-Key Policy-Attribute Based Encryption) 属性空间 $\mathcal{X}:\{0,1\}^l$，$\mathcal{X}$ 中的向量为属性向量。设函数族 $\mathcal{F}=\{f:\{0,1\}^l\rightarrow\{0,1\}\}$，$\mathcal{F}$ 中的函数为密钥策略函数。设 T 为标签空间。$S=\{g_t:Z_q^d\rightarrow Z_q\}$，$\boldsymbol{t}=(t_1,\cdots,t_d)$，如果 $\boldsymbol{t}\in(t_1,\cdots,t_d)$，则 $g_t(\boldsymbol{t})\neq 0 \bmod q$，否则 $g_t(\boldsymbol{t})=0 \bmod q$。其由 $(Setup,keyGen,Enc,Puncture,Decrypt)$ 组成。

$Setup(1^\lambda,l,d)\rightarrow\{pp,msk\}$：该算法由属性权威运行。输入安全参数 λ、属性个数 l 以及标签最大数目 d，输出公共参数 pp 和主私钥 msk。

$KeyGen(pp,msk,f\in\mathcal{F})\rightarrow\{sk_{f,\mathcal{O}}\}$：该算法由属性权威运行。输入公共参数 pp、主私钥 msk 以及访问策略函数 f，输出用户初始密钥 $sk_{f,\mathcal{O}}$。

$Enc(pp,\mu,att\in\mathcal{X},\{t_1,\cdots,t_d\})\rightarrow\{\boldsymbol{ct}\}$：该算法由数据拥有者运行。输入公共参数 pp、明文消息 μ、DO 的属性集 $att\in\mathcal{X}$ 以及标签集 $\{t_1,\cdots,t_d\}$，输出密文 $\boldsymbol{ct}$。

$Puncture(pp,sk_{f,\mathcal{P}_{\eta-1}},\hat{t}_\eta)\rightarrow\{sk_{f,\mathcal{P}_\eta}\}$：该算法由数据用户运行。输入公共参数 pp、当前密钥 $sk_{f,\mathcal{P}_{\eta-1}}$ 以及标签 $\hat{t}_\eta$，输出更新后的密钥 $sk_{f,\mathcal{P}_\eta}$。

$Decrypt(pp,sk_{f,\mathcal{P}_\eta},att,\boldsymbol{ct},\{t_1,\cdots,t_d\})\rightarrow\{\mu\}$：此算法由数据用户运行。输入公共参数 pp、私钥 $sk_{f,\mathcal{P}_\eta}$、属性集 att、密文 $\boldsymbol{ct}$ 以及密文所带的标签集 $\{t_1,\cdots,t_d\}$。输出分为 3 种情况：1)若 $f(att)\neq 0$，则输出 $\perp$；2)若 $f(att)=0$，$g(\hat{t}_i)\neq 0$，则输出 $\perp$；3)若 $f(att)=0$ 且 $g(\hat{t}_i)=0$，则可解密出明文 μ。

引理 1[12] 假设 $n\geqslant 1$，$q\geqslant 2$，$\widetilde{m},m',\hat{m}\geqslant n$，$m=\Theta(n\log q)$，可得到以下性质的多项式时间算法。

1) $TrapGen(1^n,1^m,q)\rightarrow(\boldsymbol{A},\boldsymbol{T}_A)$：输入 n,m,q，输出矩阵

$\boldsymbol{A}\in Z_q^{m\times m}$ 和 陷门基$\boldsymbol{T}_A\in Z_q^{m\times m}$,$\|\boldsymbol{T}_A\|_{GS}\leqslant O(\sqrt{n\log q})$。

2)$SamplePre(\boldsymbol{A},\boldsymbol{T}_A,\boldsymbol{u},\sigma)\rightarrow\boldsymbol{e}$:对于 $q\geqslant 2$,矩阵 $\boldsymbol{A}\in Z_q^{n\times m}$,$\boldsymbol{T}_A\in Z_q^{m\times m}$ 是格 $\Lambda_q^{\perp}(\boldsymbol{A})$的陷门基,$\sigma>\|\widetilde{\boldsymbol{T}}_A\|\omega(\sqrt{\log m})$,$\boldsymbol{u}\in Z_q^n$,存在算法 $SamplePre(\boldsymbol{A},\boldsymbol{T}_A,\boldsymbol{u},\sigma)$,输出从 $D_{\Lambda_q^u(\boldsymbol{A}),\sigma}$ 中采样的 $\boldsymbol{e}\in Z^m$,满足 $\boldsymbol{e}\in\Lambda_q^u(\boldsymbol{A})$。

3)$SampleLeft(\boldsymbol{A},\boldsymbol{B},\boldsymbol{T}_A,\boldsymbol{D},s)\rightarrow\boldsymbol{R}$:假设 $\boldsymbol{A},\boldsymbol{D}\in Z_q^{n\times m}$,$\boldsymbol{B}\in Z_q^{n\times\tilde{m}}$,$\boldsymbol{T}_A$ 是 $\Lambda_q^{\perp}(\boldsymbol{A})$ 的一个短基,$s\geqslant\|\boldsymbol{T}_A\|_{GS}\cdot\omega(\sqrt{\log(m+\tilde{m})})$,输出一个分布接近于 $D_{\Lambda_q^u}(\boldsymbol{A}|\boldsymbol{B}),s$ 的矩阵 $\boldsymbol{R}\in Z_q^{(m+\tilde{m})\times m}$。

4)$SampleRight(\boldsymbol{A},\boldsymbol{B},\boldsymbol{R},\boldsymbol{T}_B,\boldsymbol{u},\sigma)\rightarrow\boldsymbol{e}$:输入矩阵 $\boldsymbol{A}\in Z_q^{n\times m}$,$\boldsymbol{B}\in Z_q^{n\times m}$,低范数矩阵 $\boldsymbol{R}\in Z^{m\times m}$,格 $\Lambda_q^{\perp}(\boldsymbol{B})$的陷门基 $\boldsymbol{T}_B\in Z_q^{m\times m}$,向量 $\boldsymbol{u}\in Z_q^n$,高斯参数 $\sigma>\widetilde{T}_B\cdot\sqrt{m}\cdot\omega(\sqrt{\log m})$,令 $\boldsymbol{F}=\boldsymbol{A}|\boldsymbol{AR}+\boldsymbol{B}$,输出分布满足 $\boldsymbol{F}\cdot\boldsymbol{e}=\boldsymbol{u}\bmod q$ 的向量 $\boldsymbol{e}\in\Lambda_q^u(\boldsymbol{F})$,向量 $\boldsymbol{e}$ 与 $D_{\Lambda_q^u(\boldsymbol{F}),\sigma}$不可区分。

引理 2[17](Leftover Hash Lemma) 假设 $m>(n+1)\log_2 q+\omega(\log n)$,$q$ 为大于 2 的素数。$\boldsymbol{R}\in\{\pm 1\}^{m\times k}$,$k=k(n)$。均匀随机地选择矩阵 $\boldsymbol{A}\in Z_q^{n\times m}$,$\boldsymbol{B}\in Z_q^{n\times k}$。任意选取向量 $\boldsymbol{e}\in Z_q^m$,分布$(\boldsymbol{A},\boldsymbol{AR},\boldsymbol{R}^{\mathrm{T}}\boldsymbol{e})$和$(\boldsymbol{A},\boldsymbol{B},\boldsymbol{R}^{\mathrm{T}}\boldsymbol{e})$不可区分。

引理 3[17] 给定参数(λ,n,m,q,χ),λ 是安全参数,χ 是以 $\chi_{\max}$为上界的分布。$\boldsymbol{B}_1,\cdots,\boldsymbol{B}_l\in Z_q^{n\times m}$,布尔电路 $f:\{0,1\}^l\rightarrow\{0,1\}$,$depth\leqslant d$,$x\in\{0,1\}^l$,若 $i\in[l]$,$c_i=(\boldsymbol{B}_i+x_i\boldsymbol{G})^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e}_i$。对于向量 $\boldsymbol{s}\in Z_q^n$,当 $i\in[l]$,$\boldsymbol{e}_i\in\chi^m$,存在算法$(Eval_{pk},Eval_{ct},Eval_{\mathrm{sim}})$具有以下性质:

1)$Eval_{pk}(f,(\boldsymbol{B}_1,\cdots,\boldsymbol{B}_l))\rightarrow\boldsymbol{B}_f$:输入一个布尔电路 f,l 个矩阵$(\boldsymbol{B}_1,\cdots,\boldsymbol{B}_l)$,输出一个矩阵$\boldsymbol{B}_f$。

2)$Eval_{ct}(f,\{x_i,\boldsymbol{B}_i,c_i\}_{i\in[l]})\rightarrow c_f$:输入布尔电路 f,l 个矩阵$(\boldsymbol{B}_1,\cdots,\boldsymbol{B}_l)$,长度为 l 的字符串 x 和 l 个向量$(\boldsymbol{c}_1,\cdots,\boldsymbol{c}_l)$,输出向量 $\boldsymbol{c}_f=(\boldsymbol{B}_f+f(x)\cdot\boldsymbol{G})^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e}_f$,其中 $\boldsymbol{B}_f=Eval_{pk}(f,(\boldsymbol{B}_1,\cdots,\boldsymbol{B}_l))$,$\|\boldsymbol{e}_f\|\leqslant\chi_{\max}\cdot\sqrt{m}(m+1)^d$。

3)$Eval_{\mathrm{sim}}(f,\{\boldsymbol{S}_i^*,\boldsymbol{x}_i^*\}_{i\in[l]},\boldsymbol{A})\rightarrow\boldsymbol{S}_f^*$:输入布尔电路 f,l 个矩阵$(\boldsymbol{S}_1^*,\cdots,\boldsymbol{S}_l^*)$,矩阵 $\boldsymbol{A}\in Z_q^{n\times m}$,长度为 l 的字符串 x^*,输出矩阵 $\boldsymbol{S}_f^*\in Z_q^{m\times m}$,$\boldsymbol{AS}_f^*-f(x^*)\boldsymbol{G}=\boldsymbol{B}_f$,其中 $\boldsymbol{B}_f=Eval_{pk}(f,(\boldsymbol{AS}_1^*-x_1^*\boldsymbol{G},\cdots,\boldsymbol{AS}_l^*-x_l^*\boldsymbol{G}))$。若$(\boldsymbol{S}_1^*,\cdots,\boldsymbol{S}_l^*)\in\{\pm 1\}^{m\times m}$,可得 $\|\boldsymbol{S}_f^*\|_2\leqslant 20\sqrt{m}\cdot(m+1)^d$。

假设 $S=\{g_t:Z_q^d\rightarrow Z_q\}$,当 $p<q$ 时,函数集 S 可以通过深度为 D 的电路计算,$\alpha_S(n)=O((p^dm)^D\cdot\sqrt{m})$。

引理 4[17] 设 n,m,q 为正整数,其中 q 为素数,多项式时间算法如下。

1)$ExtendRight(\boldsymbol{A},\boldsymbol{B},\boldsymbol{T}_A)\rightarrow\boldsymbol{T}_{(\boldsymbol{A}|\boldsymbol{B})}^{\mathrm{ER}}$:输入满秩矩阵 $\boldsymbol{A},\boldsymbol{B}\in Z_q^{n\times m}$,$\Lambda_q^{\perp}(\boldsymbol{A})$的基$\boldsymbol{T}_A$,输出 $\Lambda_{q(\boldsymbol{A}|\boldsymbol{B})}^{\mathrm{T}}$的基$\boldsymbol{T}_{\boldsymbol{A}|\boldsymbol{B}}$,其中 $\|\boldsymbol{T}_{\boldsymbol{A}}\|_{GS}=\|\boldsymbol{T}_{\boldsymbol{A}|\boldsymbol{B}}\|_{GS}$。

2)$ExtendLeft(\boldsymbol{A},\boldsymbol{G},\boldsymbol{T}_G,\boldsymbol{R})\rightarrow\boldsymbol{T}_M,\boldsymbol{M}=[\boldsymbol{A}|\boldsymbol{G}+\boldsymbol{AR}]$:输入满秩矩阵 $\boldsymbol{A},\boldsymbol{G}\in Z_q^{n\times m}$ 和$\Lambda_q^{\perp}(\boldsymbol{G})$的基$\boldsymbol{T}_G$,令 $\boldsymbol{M}=[\boldsymbol{A}|\boldsymbol{G}+\boldsymbol{AR}]$,输出 $\Lambda_q^{\perp}(\boldsymbol{M})$的基$\boldsymbol{T}_M$,$\|\boldsymbol{T}_M\|_{GS}\leqslant\|\boldsymbol{T}_G\|_{GS}\cdot(1+\|\boldsymbol{R}\|_2)$。

引理 5[17] 假设 $\boldsymbol{A}\in Z_q^{n\times m}$,$\boldsymbol{T}_A\in Z_q^{m\times m}$ 是 $\Lambda_q^{\perp}(\boldsymbol{A})$的基,$\boldsymbol{U}\in Z_q^{n\times k}$,输出 $\boldsymbol{X}\in Z^{m\times k}$。满足 $\boldsymbol{AX}=\boldsymbol{U}$ 的多项式时间算法如下。

1)算法 $SampleD(\boldsymbol{A},\boldsymbol{T}_{\boldsymbol{A}},\boldsymbol{U},\sigma)\rightarrow\boldsymbol{X}$:输入满秩矩阵 $\boldsymbol{A}\in Z_q^{n\times m}$、格 $\Lambda_q^{\perp}(\boldsymbol{A})$的陷门基$\boldsymbol{T}_A\in Z_q^{m\times m}$、$\boldsymbol{U}\in Z_q^{n\times k}$、高斯参数 $\sigma=\widetilde{T}_A\cdot\omega\sqrt{\log m}$,输出分布接近于 $D_{\Lambda_q^U(\boldsymbol{A})}$ 的低范数矩阵 $\boldsymbol{X}\in Z^{m\times k}$。

2)$RandBasis(\boldsymbol{A},\boldsymbol{T}_{\boldsymbol{A}},\delta)\rightarrow\boldsymbol{T}_{\boldsymbol{A}}'$:当 $\delta=\|\boldsymbol{T}_{\boldsymbol{A}}\|_{GS}\cdot\omega(\sqrt{\log m})$ 时,输出 $\Lambda_q^{\perp}(\boldsymbol{A})$的基$\boldsymbol{T}_{\boldsymbol{A}}'$,其分布接近于$(D_{\delta}(\Lambda_q^{\perp}(\boldsymbol{A})))^m$。

假设 $n,q=q(n),m=\Theta(n\log q)$,$\boldsymbol{G}\in Z_q^{n\times m}$ 是 Gadget 矩阵,$x\in Z_q$,$\boldsymbol{B}\in Z_q^{n\times m}$,$\boldsymbol{s}\in Z_q^n$,$\sigma>0$,定义集合:$E_{s,\sigma}(x,\boldsymbol{B})=\{(x\boldsymbol{G}+\boldsymbol{B})^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e}\in Z_q^m,\|\boldsymbol{e}\|<\sigma\}$。

2.3 UR 二叉树[14]

本文构造如图 1 所示的完全二叉树,以实现用户的追踪和撤销。二叉树共有 $2N-1$ 个节点,其中有 N 个叶子节点,这 N 个叶子节点分别与 N 个用户相连。每个节点具有 *key* 和 *value* 两个属性,其中 *key* 代表每个节点在二叉树中的编号,*value* 代表每个节点所存储的矩阵。根节点的 *value* 值是具有陷门基的矩阵$\boldsymbol{U}_0$。非叶子节点的 *value* 值是 $\boldsymbol{U}_i=\boldsymbol{U}_0\cdot\boldsymbol{M}_i$,$i\in[1,N-2]$,$\boldsymbol{M}_i\in\{0,1\}^{m\times m}$,其中$\boldsymbol{M}_i$ 是可逆矩阵,每行每列只有一个 1,其余为 0。叶子节点的 *value* 值是$\boldsymbol{U}_j=\boldsymbol{U}_0\cdot\boldsymbol{N}_j$,$j\in[N-1,2N-2]$,$\boldsymbol{N}_j\in\{-1,1\}^{m\times m}$。

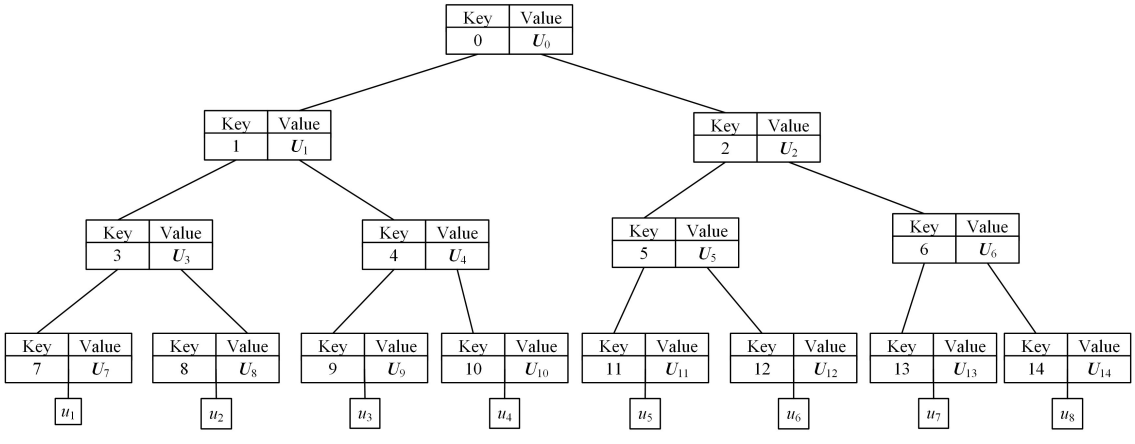

图 1 UR 二叉树

Fig. 1 User revocation binary tree

2.4 完全子树法[12]

算法 1 采用文献[12]提出的 KUNodes 算法。若 I 为图 1 中叶子节点，则 $Path(I)$ 指图 1 中从完全二叉树的根结点到与用户 I 相关联的身份矩阵这一段路径上 key 的集合。对于 $u_2 \in RL$，可得 $Path(u_2)=\{0,1,3,8\}$，假设 $RL=\{u_2,u_5\}$，与撤销列表中的用户相连的结点分别是{8,11}，除了与用户相连的{8,11}这两个结点之外，BT 中能够覆盖的二叉树中叶子结点的最小集合是：$KUNodes(BT,RL)\rightarrow\{4,6,7,12\}$。可得到 $Path(8)\cap KUNodes(BT,RL)=\{\emptyset\}$。

算法 1 KUNodes algorithm

输入：BT，RL

输出：Y

1. X，Y∈∅，
2. ∀ r∈RL，SET X←X∪Path(r)
3. ∀ r∈X：
   IF $r_L \notin$ X，SET Y←Y∪{r_L}
   IF $r_R \notin$ X，SET Y←Y∪{r_R}
4. IF Y=∅，SET Y=Y∪{root}
5. RETURN Y.

2.5 矩阵异或算法

算法 2 将两个行数和列数均相同的矩阵相应位置的元素进行异或，生成一个新的矩阵。

算法 2 Matrix_XOR algorithm

输入：$\mathbf{A},\mathbf{B}\in Z_q^{n\times m}$

输出：$\mathbf{C}\in Z^{n\times m}$

1. For i IN range(n)：
2. For j IN range(m)：
3. $\mathbf{C}[i][j]=\mathbf{A}[i][j]\oplus\mathbf{B}[i][j]$
4. Return **C**.

3 系统和安全模型

3.1 系统结构

系统中有 5 个实体，即云服务提供商(Cloud Service Provider，CSP)、属性权威(Attribute Authority，AA)、数据拥有者(Data Owner，DO)、数据服务器管理者(Data Server Manager，DSM)和用户(Data User，DU)。如图 2 所示。

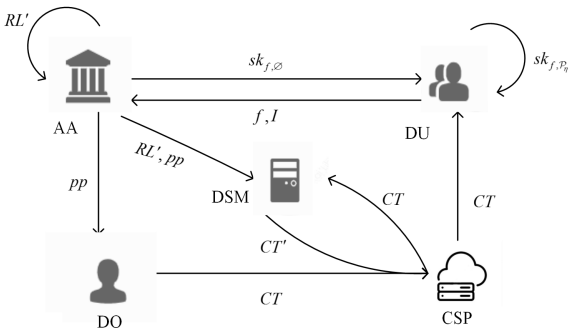

图 2 系统模型

Fig. 2 System model

1)CSP：是诚实且好奇的，负责存储密文。DO 将密文上传至 CSP 中，当用户需要密文时，从 CSP 获取密文。

2)AA：是完全可信的，负责在系统中生成公共参数，根据访问策略函数及身份为每个用户生成密钥，定义和维护撤销列表。

3)DO：拥有数据，使用属性集和标签集加密数据。

4)DU：定义访问策略函数，将策略函数和身份发送给 AA，获得密钥。当密文属性满足密钥的访问策略，密文标签不满足标签函数且用户不属于撤销列表时，用户可以对密文进行解密。用户可以重复地更新解密密钥，以撤销对过去密文的解密权限。

5)DSM：撤销列表发生变化后，由 DSM 更新密文。

3.2 算法定义

具有多功能的格上属性加密方案由 8 个算法组成。

$Setup(1^\lambda,l,N,d)\rightarrow\{pp,msk\}$：该算法由 AA 运行。输入安全参数 λ、属性个数 l、用户最大数目 N、标签最大数目 d，输出公共参数 pp 和主私钥 msk。

$KeyGen(pp,msk,f,I)\rightarrow\{sk_{f,\emptyset}\}$：该算法由 AA 运行。输入公共参数 pp、主私钥 msk、访问策略函数 f 和用户索引 I，输出用户私钥 $sk_{f,\emptyset}$。

$Enc(pp,\mu,att,\{t_1,\cdots,t_d\},RL)\rightarrow\{\boldsymbol{ct}\}$：该算法由 DO 运行。输入公共参数 pp、明文消息 μ、属性集 att、标签集 $\{t_1,\cdots,t_d\}$ 和撤销列表 RL，输出密文 $\boldsymbol{ct}$。

$Puncture(pp,sk_{f,\mathcal{P}_{\eta-1}},\hat{t}_\eta)\rightarrow\{sk_{f,\mathcal{P}_\eta}\}$：该算法由 DU 运行。输入公共参数 pp、当前密钥 $sk_{f,\mathcal{P}_{\eta-1}}$ 以及标签 $\hat{t}_\eta$(此标签为想要撤销访问权限的密文标签集中的任意一个标签)，输出更新后的密钥 $sk_{f,\mathcal{P}_\eta}$。

$Decrypt(pp,sk_{f,\mathcal{P}_\eta},att,\boldsymbol{ct},\{t_1,\cdots,t_d\})\rightarrow\{\mu=(\mu_1,\cdots,\mu_m)\}$：此算法由 DU 运行。输入公共参数 pp、私钥 $sk_{f,\mathcal{P}_\eta}$、属性集 att、密文 $\boldsymbol{ct}$ 以及密文所带的标签集 $\{t_1,\cdots,t_d\}$。输出分为 4 种情况：1)若 $f(att)\neq 0$，则输出 $\perp$；2)若 $f(att)=0$，$g(\hat{t}_i)\neq 0$，则输出 $\perp$；3)若 $f(att)=0$，$g(\hat{t}_i)=0$ 且 $I\in RL$，则输出 $\perp$；4)若 $f(att)=0$，$g(\hat{t}_i)=0$ 且 $I\notin RL$，则输出 μ。

$KeySanityCheck(sk,pp)\rightarrow\{1\text{ 或 }\perp\}$：此算法由 AA 运行。输入 sk 和公共参数 pp，算法检查 sk 是否需要被追踪，如果 sk 通过密钥检查，则输出 1，否则输出 $\perp$。

$Trace(pp,RL,sk)\rightarrow\{\perp\text{ 或 }RL'\}$：此算法由 AA 运行。输入 sk、公共参数 pp 以及撤销列表 RL，如果 sk 通过了 $KeySanityCheck$，则算法可得到用户的身份 u_{i_u}，并更新撤销列表 $RL'=RL\cup\{u_{i_u}\}$，否则输出 $\perp$。

$CTUpdate(RL',pp,\boldsymbol{ct})\rightarrow\{\boldsymbol{ct}'\}$：该算法由 DSM 执行。输入最新的撤销列表 RL'、公共参数 pp 和密文 $\boldsymbol{ct}$，输出更新后的密文 $\boldsymbol{ct}'$。

3.3 安全模型

$\mathcal{A}$ 为概率多项式时间的敌手，$\Pi=(Setup,KeyGen,Enc,Puncture,Decrypt,KeySanityCheck,Trace,CTUpdate)$ 是具有多功能的 KP-ABE 方案，$\mathcal{M}$ 为明文空间，f 为属性策略函数，I 为用户索引。

安全模型由挑战者 $\mathcal{B}$ 和敌手 $\mathcal{A}$ 之间的博弈来描述。

系统初始化：敌手 $\mathcal{A}$ 声明目标属性向量 $\boldsymbol{x}^*=(x_1^*,x_2^*,\cdots,x_l^*)\in\mathcal{X}^l$、目标标签集 $\{t_1^*,t_2^*,\cdots,t_d^*\}$ 和撤销列表 $RL^*\subseteq[I]$，并将目标属性向量、目标标签集以及撤销列表发送给挑战者 $\mathcal{B}$。

参数设置阶段：挑战者$\mathcal{B}$运行 Setup 算法，将公钥 pp 发送给敌手$\mathcal{A}$，主私钥 msk 保密，挑战者维持一个元组$(f, I, sk_{f,\mathcal{P}_i}, P_f, C_f)$，$P_f$代表当前密钥被穿刺的标签集合，$C_f$代表当前密钥之前已经完成穿刺的标签集合。

密钥查询阶段 1：$\mathcal{A}$执行以下查询。

1）$Q_{\text{Puncture}}(f\in\mathcal{F}, \hat{t})$：给定一个函数 f，目标属性向量 $\boldsymbol{x}^*$，一个标签 $\hat{t}$。

情形 1　对于 f，如果 $f(x^*)\neq 0$：

(1)若存在元组$(f, I, sk_{f,\mathcal{P}_i}, P_f, C_f)$，则执行 $Puncture(pp, sk_{f,\mathcal{P}_i}, \hat{t})\rightarrow\{sk_{f,\mathcal{P}_{i+1}}\}$，将 $\hat{t}$ 加入P_f，$P_f=P_f\cup\{\hat{t}\}$，使用新元组$(f, I, sk_{f,\mathcal{P}_{i+1}}, P_f, C_f)$替代旧元组。

(2)若不存在元组，则运行 $KeyGen(pp, msk, f, I)\rightarrow\{sk_{f,\emptyset}\}$，$Puncture(pp, sk_{f,\emptyset}, \hat{t})\rightarrow\{sk_{f,\mathcal{P}_1}\}$，$P_f=\{\hat{t}\}$，创建新元组$(f, I, sk_{f,\mathcal{P}_1}, P_f, C_f)$。

情形 2　对于 f，如果 $f(x^*)=0$，$I\in RL^*$：

(1)若存在元组$(f, I, sk_{f,\mathcal{P}_i}, P_f, C_f)$，则执行 $Puncture(pp, sk_{f,\mathcal{P}_i}, \hat{t})\rightarrow\{sk_{f,\mathcal{P}_{i+1}}\}$，将 $\hat{t}$ 加入P_f，$P_f=P_f\cup\{\hat{t}\}$，用新元组$(f, I, sk_{f,\mathcal{P}_{i+1}}, P_f, C_f)$替代旧元组。

(2)若不存在元组，密钥构造和元组构造与上述情形 1 中(2)的情况相同。

情形 3　对于 f，如果 $f(x^*)=0$，$I\notin RL^*$：

(1)若存在元组$(f, I, -, P_f, C_f)$，则挑战者添加 $\hat{t}$ 到P_f中，用新元组替代旧元组。

(2)否则，添加 $\hat{t}$ 到P_f中，构造一个新元组$(f, I, -, P_f, C_f)$，$P_f=\{\hat{t}\}$，$C_f=\emptyset$。

2）$Q_{\text{Corrupt}}(f)$：当敌手首次对函数 f 进行查询时，挑战者考虑以下两种情况。

情形 1　对于 f，如果 $f(x^*)\neq 0$ 或者 $f(x^*)=0$ 且 $I\in RL^*$：

(1)若存在元组$(f, I, sk_{f,\mathcal{P}_i}, P_f, C_f)$，则挑战者返回 $sk_{f,\mathcal{P}_i}$ 给敌手$\mathcal{A}$，设置$C_f\leftarrow P_f$。

(2)否则，计算初始密钥(如上所述)，将 $sk_{f,\emptyset}$返回给敌手$\mathcal{A}$，设置$C_f\leftarrow P_f=\emptyset$。

情形 2　对于 f，如果 $f(x^*)=0$，$I\notin RL^*$：

(1)若存在元组$(f, I, -, P_f, C_f)$，检查 $P_f\bigcap\{t_1^*, t_2^*, \cdots, t_d^*\}$是否为空集。

①若$P_f\bigcap\{t_1^*, t_2^*, \cdots, t_d^*\}=\emptyset$，则挑战者输出$\perp$；

②若$P_f\bigcap\{t_1^*, t_2^*, \cdots, t_d^*\}\neq\emptyset$，则计算 $sk_{f,\mathcal{P}_k}$并返回给$\mathcal{A}$，设置$C_f\leftarrow P_f$。

(2)不存在元组，则输出$\perp$。

(3)其他情况下的查询均输出$\perp$。

挑战阶段：$\mathcal{A}$发送两个等长的消息 $\mu_0\neq\mu_1$ 给挑战者$\mathcal{B}$，挑战者$\mathcal{B}$随机选择 $b\in\{0,1\}$，计算对应密文并发送给敌手$\mathcal{A}$。

密钥查询阶段 2：与密钥查询阶段 1 相似，敌手$\mathcal{A}$继续向挑战者$\mathcal{B}$询问私钥。

猜测阶段：$\mathcal{A}$输出 b 的猜测 $b'\in\{0,1\}$，若 $b=b'$，则输出 1，否则输出 0。

以上 IND-ST-SA-SR-CPA（Indistinguish-ability under Selective Tag and Selective Attribute and Selective Revocation list against Chosen Plaintext Attack）游戏，敌手$\mathcal{A}$的优势定义如下：

$$Adv_{\mathcal{A}}^{\text{IND-ST-SA-SR-CPA}}(\lambda)=\left|\Pr[b=b']-\frac{1}{2}\right|$$

4　方案构造

属性空间$\mathcal{X}$：$\{0,1\}^l$，定义函数族如下：$\mathcal{F}=\{f:\{0,1\}^l\rightarrow\{0,1\}\}$，输入 l 维属性向量，若 f 输出为 0，则说明属性向量满足访问策略函数，否则，不满足。函数族：$S=\{g_t: Z_q^d\rightarrow Z_q\}$，标签集 $t=(t_1,\cdots,t_d)$，若标签 $t\in(t_1,\cdots,t_d)$，则 $g_t(\boldsymbol{t})\neq 0 \bmod q$；若标签 $t\notin(t_1,\cdots,t_d)$，则 $g_t(t)=0 \bmod q$。

$Setup(1^\lambda, l, N, d)\rightarrow\{pp, msk\}$：此算法由 AA 运行。输入安全参数 λ、属性最大长度 l、用户最大数目 N 以及标签最大数目 d，执行以下操作。

1）运行 $TrapGen(1^n, 1^m, q)$算法，生成均匀随机的矩阵 $\boldsymbol{A}\in Z_q^{n\times m}$ 以及 $\Lambda_q^{\perp}(\boldsymbol{A})$的基$\boldsymbol{T}_A\in Z_q^{m\times m}$。

2）采样$(l+d+3)$个均匀随机的矩阵$\boldsymbol{B}_1,\cdots,\boldsymbol{B}_l,\boldsymbol{A}_1,\cdots,\boldsymbol{A}_d,\boldsymbol{U},\boldsymbol{D},\boldsymbol{F}\in\boldsymbol{Z}_q^{n\times m}$。Gadget 矩阵 $\boldsymbol{G}\in Z_q^{n\times m}$，初始化 $RL=\{\emptyset\}$。

3）构建具有 N 个叶子节点的完全二叉树 BT。在二叉树 BT 中，除了引入根节点的 $value$ 值$\boldsymbol{U}_0=\boldsymbol{A}$ 之外，叶子节点的 $value$ 值、非叶子节点的 $value$ 值、矩阵$\boldsymbol{M}_i$，$i\in[1,N-2]$以及矩阵$\boldsymbol{N}_j$，$j\in[N-1,2N-2]$的描述均与 2.3 节相同。

输出：$pp=\{\boldsymbol{A},\boldsymbol{A}_1,\cdots,\boldsymbol{A}_d,\boldsymbol{B}_1,\cdots,\boldsymbol{B}_l,\boldsymbol{U},\boldsymbol{D},\boldsymbol{G},BT,\{\boldsymbol{M}_i\}_{i\in[1,N-2]},\{\boldsymbol{N}_j\}_{j\in[N-1,2N-2]},RL\}$，$msk=\{\boldsymbol{T}_A,\boldsymbol{F}\}$。

$KeyGen(pp, msk, f, I)\rightarrow\{sk_{f,\emptyset}\}$：此算法由 AA 运行。输入 msk、策略函数 $f\in\mathcal{F}$、索引 $I\in[N]$，执行以下操作。

1）选择均匀随机的矩阵$\boldsymbol{D}_I\in Z_q^{n\times m}$。

2）计算$\boldsymbol{B}_f=Eval_{pk}^{\mathcal{T}}(f,(\boldsymbol{B}_1,\cdots,\boldsymbol{B}_l))$，运行 $SampleLeft(\boldsymbol{A},\boldsymbol{B}_f,\boldsymbol{T}_A,\boldsymbol{D}_I,s)\rightarrow\boldsymbol{R}_f$，产生满足$(\boldsymbol{A}|\boldsymbol{B}_f)\cdot\boldsymbol{R}_f=\boldsymbol{D}_I$ 的低范数矩阵 $\boldsymbol{R}_f\in Z_q^{2m\times m}$。对于 $\gamma\in Path(I)$，运行 $SampleLeft(\boldsymbol{A},\boldsymbol{U}_\gamma,\boldsymbol{T}_A,\boldsymbol{D}-\boldsymbol{D}_I,s)\rightarrow\boldsymbol{R}_\gamma$，产生满足$(\boldsymbol{A}|\boldsymbol{U}_\gamma)\boldsymbol{R}_\gamma=\boldsymbol{D}-\boldsymbol{D}_I$ 的低范数矩阵 $\boldsymbol{R}_\gamma\in Z_q^{2m\times m}$。运行 $ExtendRight(\boldsymbol{A},\boldsymbol{B}_f,\boldsymbol{T}_A)$ 算法生成$\boldsymbol{T}_{(\boldsymbol{A}|\boldsymbol{B}_f)}^{\text{ER}}$，运行 $RandBasis([\boldsymbol{A}\mid\boldsymbol{B}_f],T_{(\boldsymbol{A}|\boldsymbol{B}_f)}^{\text{ER}},\delta_0)$算法生成 $T_{(\boldsymbol{A}|\boldsymbol{B}_f)}\in Z_q^{2m\times 2m}$，$\delta_0=\omega(\alpha_{\mathcal{F}}\cdot\sqrt{\log m})$。

3）计算 $T'_{(\boldsymbol{A}|\boldsymbol{B}_f)}=T_{(\boldsymbol{A}|\boldsymbol{B}_f)}\cdot(D|U)^{\text{T}}$。根据 I，找到用户对应叶子节点的 $value$ 值，记为$\boldsymbol{U}_{i_u}$，计算 $\boldsymbol{K}=\text{Matrix\_XOR}(\boldsymbol{F},\boldsymbol{U}_{i_u})$。

输出：$sk_{f,\emptyset}=\{T_{(\boldsymbol{A}|\boldsymbol{B}_f)},T'_{(\boldsymbol{A}|\boldsymbol{B}_f)},\boldsymbol{R}_f,\boldsymbol{K},\{\boldsymbol{R}_\gamma\}_{\gamma\in Path(I)}\}$。

$Enc(pp,\mu,att,\{t_1,\cdots,t_d\})\rightarrow\{\boldsymbol{ct}\}$：此算法由 DO 运行。输入公开参数 pp、明文消息 $\mu\in\{0,1\}^m$、属性 $att\in\{0,1\}^l$、标签集$\{t_1,\cdots,t_d\}$，执行以下操作。

1）选择均匀随机的向量 $\boldsymbol{s}\in Z_q^n$，噪声向量$\boldsymbol{e}_0$，$\boldsymbol{e}_0{}'$，$\boldsymbol{e}_1$，$\boldsymbol{e}_{\text{out}}\in\chi^m$，分布 χ 的上限为 $\chi_{\max}$，矩阵$\boldsymbol{R}_1,\cdots,\boldsymbol{R}_d\in\{\pm1\}^{m\times m}$，矩阵$\boldsymbol{S}_\gamma^i,\boldsymbol{S}_i\in\{\pm1\}^{m\times m}$，其中 $i\in[l]$，$\overset{\wedge}{\gamma}\in KUNodes(BT,RL)$。

2）令$\boldsymbol{H}_1=(\boldsymbol{B}_1+att_1\boldsymbol{G}\mid\cdots\mid\boldsymbol{B}_l+att_l\boldsymbol{G})\in Z_q^{n\times lm}$，$\boldsymbol{H}_2=(\boldsymbol{A}_1+t_1\boldsymbol{G}\mid\cdots\mid\boldsymbol{A}_d+t_d\boldsymbol{G})\in Z_q^{n\times dm}$，$\boldsymbol{H}=(\boldsymbol{A}\mid\boldsymbol{H}_1\mid\boldsymbol{H}_2)\in Z_q^{n\times(l+d+1)m}$，$\boldsymbol{e}=(\boldsymbol{S}_1\mid\cdots\mid\boldsymbol{S}_l)^{\text{T}}$，$\boldsymbol{e}_0\in Z_q^{lm}$，$\boldsymbol{e}'=(\boldsymbol{I}_m\mid\boldsymbol{S}_1\mid\cdots\mid\boldsymbol{S}_l\mid\boldsymbol{R}_1\mid\cdots\mid\boldsymbol{R}_d)^{\text{T}}\cdot\boldsymbol{e}_0{}'=(\boldsymbol{e}_{\text{in}}^{\text{T}};\overline{\boldsymbol{e}}_1^{\text{T}};\cdots;\overline{\boldsymbol{e}}_l^{\text{T}};\boldsymbol{e}_1^{\text{T}};\cdots;\boldsymbol{e}_d^{\text{T}})^{\text{T}}\in Z_q^{(l+d+1)m}$。

3) 计算 $\boldsymbol{c}=\boldsymbol{H}^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e}'\in Z_q^{(l+d+1)m}$，令 $\boldsymbol{c}=[\boldsymbol{c}_{\mathrm{in}};\bar{\boldsymbol{c}}_1;\cdots;\bar{\boldsymbol{c}}_l;\boldsymbol{c}_1;\cdots;\boldsymbol{c}_d]\in Z_q^{(l+d+1)m}$，$\boldsymbol{c}_{\mathrm{in}}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e}_{\mathrm{in}}$，$\bar{\boldsymbol{c}}_i=(\boldsymbol{B}_i+att_i\boldsymbol{G})^{\mathrm{T}}\boldsymbol{s}+\bar{\boldsymbol{e}}_i$，$i\in\{1,\cdots,l\}$，$\boldsymbol{c}_j=(\boldsymbol{A}_j+t_j\boldsymbol{G})^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e}_j$，$j\in\{1,\cdots,d\}$。计算 $\boldsymbol{c}_0=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e}_0\in Z_q^m$，$\boldsymbol{c}_1=\boldsymbol{H}_1^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e}\in Z_q^{lm}$，对于 $\hat{\gamma}\in KUNodes(BT,RL)$，$\hat{\boldsymbol{c}}_{\hat{\gamma}}=\boldsymbol{U}_{\hat{\gamma}}^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{S}_{\hat{\gamma}}\boldsymbol{e}_0\in Z_q^m$，令 $\boldsymbol{c}_2=\{\hat{\boldsymbol{c}}_{\hat{\gamma}}\}$，$\boldsymbol{c}_{\mathrm{out}}=\boldsymbol{D}^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{U}^{\mathrm{T}}s+\boldsymbol{e}_{\mathrm{out}}+\mu\left\lfloor\frac{q}{2}\right\rfloor$。

输出：密文 $\boldsymbol{ct}=(\boldsymbol{c},\boldsymbol{c}_0,\boldsymbol{c}_1,\boldsymbol{c}_2,\boldsymbol{c}_{\mathrm{out}})$。

$Puncture(pp,sk_{f,\mathcal{P}_{\eta-1}},\hat{t}_\eta)\to\{sk_{f,\mathcal{P}_\eta}\}$：此算法由 DU 运行。输入公共参数 pp、密钥 $sk_{f,\mathcal{P}_{\eta-1}}$，以及标签 $\hat{t}_\eta$，执行以下操作。

计算 $A_{g_{\hat{t}_\eta}}\leftarrow Eval_{pk}^{S}(\{\boldsymbol{A}_i\}_{i=1}^{d},g_{\hat{t}_\eta})$，$\boldsymbol{T}_{f,\mathcal{P}_\eta}^{\mathrm{ER}}\leftarrow ExtendRight([\boldsymbol{A}|\boldsymbol{B}_f|\boldsymbol{A}_{g_{\hat{t}_{\eta-1}}}|\cdots|\boldsymbol{A}_{g_{\hat{t}_{\eta-1}}}],\boldsymbol{A}_{g_{\hat{t}_\eta}},\boldsymbol{T}_{id,\eta-1})$，$\boldsymbol{T}_{f,\mathcal{P}_\eta}\leftarrow ExtendRight([\boldsymbol{A}|\boldsymbol{B}_f|\boldsymbol{A}_{g_{\hat{t}_1}}|\cdots|\boldsymbol{A}_{g_{\hat{t}_\eta}}],\boldsymbol{T}_{id,\eta}^{\mathrm{ER}},\sigma_\eta)$，其中 $\sigma_\eta=\sigma_0\cdot(\sqrt{m\log m})^\eta$，$\sigma_0=\omega(\alpha_S\cdot\sqrt{\log m})$，$p_{\eta-1}=\{\hat{t}_1,\cdots,\hat{t}_{\eta-1}\}$，$p_\eta=\{\hat{t}_1,\cdots,\hat{t}_\eta\}$。

输出：$sk_{f,\mathcal{P}_\eta}=\{\boldsymbol{T}_{f,\mathcal{P}_\eta},\boldsymbol{R}_f,\boldsymbol{K},\{\boldsymbol{R}_\gamma\}_{\gamma\in Path(I)}\}$。

$Decrypt(pp,sk_{f,\mathcal{P}_\eta},att,\boldsymbol{ct},\{t_1,\cdots,t_d\})\to\{\mu=(\mu_1,\cdots,\mu_m)\}$：此算法由 DU 运行。输入公共参数 pp、密钥 $sk_{f,\mathcal{P}_\eta}$、属性集 att、密文 $\boldsymbol{ct}$，以及标签集 $\{t_1,\cdots,t_d\}$，执行以下操作。

1) 若 $f(att)\neq 0$，输出 $\perp$；

2) $\boldsymbol{t}=\{t_1,\cdots,t_d\}$，若 $f(att)=0$ 且 $g_{\hat{t}_j}(\boldsymbol{t})\neq 0$，$j\in\{1,\cdots,\eta\}$，则输出 $\perp$；

3) 若 $f(att)=0$，$g_{\hat{t}_j}(\boldsymbol{t})=0$ 且 $I\in RL$，则输出 $\perp$；

4) 若 $f(att)=0$，$g(\hat{t}_i)=0$ 且 $I\in RL$，则执行以下操作。

计算 $\bar{\boldsymbol{c}}_f=Eval_{ct}^{\mathcal{F}}(f,\{\boldsymbol{B}_i,att_i,\bar{\boldsymbol{c}}_i\})$，$i\in[l]$，$\boldsymbol{R}\in Z_q^{(\eta+2)m\times m}\leftarrow SampleD([\boldsymbol{A}|\boldsymbol{B}_f|\boldsymbol{A}_{g_{\hat{t}_1}}|\cdots|\boldsymbol{A}_{g_{\hat{t}_\eta}}],\boldsymbol{T}_{f,\mathcal{P}_\eta},\boldsymbol{U},\sigma_\eta)$。当 $\gamma\in Path(I)$ 和 $\hat{\gamma}\in KUNodes(BT,RL)$ 存在相同的节点时，计算 $\boldsymbol{w}_{(\gamma,\hat{\gamma})}=\boldsymbol{R}_f^{\mathrm{T}}(\boldsymbol{c}_0;\bar{\boldsymbol{c}}_f)+\boldsymbol{R}_\gamma^{\mathrm{T}}(\boldsymbol{c}_0;\hat{\boldsymbol{c}}_{\hat{\gamma}})$，$\boldsymbol{c}_{g_{i_j}}=Eval_{ct}^{g}(g_{t_j},\{t_i,\boldsymbol{A}_i,\boldsymbol{c}_i\}_{i=1}^{d})$，$j\in\{1,\cdots,\eta\}$。计算 $(\mu_1,\cdots,\mu_m)=\boldsymbol{c}_{\mathrm{out}}-\boldsymbol{R}^{\mathrm{T}}\boldsymbol{c}-\boldsymbol{w}_{(\gamma,\hat{\gamma})}$，如果 $|\mu_i|<\frac{q}{4}$，则 $\mu_i=0$，否则 $\mu_i=1$。

输出：$(\mu_1,\cdots,\mu_m)$。

$KeySanityCheck(msk,sk,pp)\to\{1\text{ 或 }\perp\}$：此算法由 AA 运行。输入用户私钥 sk 以及公开参数 pp。$KeySanityCheck$ 评估解密密钥是否需要被追踪。若解密密钥被怀疑，则算法需要检查解密密钥是否满足以下条件：

$$(\boldsymbol{A}|\boldsymbol{B}_f)\boldsymbol{R}_f+(\boldsymbol{A}|\boldsymbol{U}_\gamma)\boldsymbol{R}_\gamma=\boldsymbol{D},\ \gamma\in Path(I) \tag{1}$$

$$Matrix\_XOR(\boldsymbol{F},\boldsymbol{K})=\boldsymbol{U}_{i_u} \tag{2}$$

$$\boldsymbol{T}'_{(\boldsymbol{A}|\boldsymbol{B}_f)}=\boldsymbol{T}_{(\boldsymbol{A}|\boldsymbol{B}_f)}\cdot(\boldsymbol{D}|\boldsymbol{U})^{\mathrm{T}} \tag{3}$$

输出：如果解密密钥 sk 满足条件(1)－(3)，说明解密密钥能通过密钥完整性检测，则算法输出 1，否则算法输出 $\perp$。

$Trace(pp,RL,sk)\to\{\perp\text{ 或 }RL'\}$：此算法由 AA 运行。输入公共参数 pp、撤销列表 RL，以及恶意用户的私钥 sk。若 sk 未通过 $KeySanityCheck$，则输出 $\perp$；若通过，则计算 $\boldsymbol{U}_{i_u}=Matrix\_XOR(\boldsymbol{F},\boldsymbol{K})$，遍历二叉树，找到相应的用户 u_{i_u}，若用户 $u_{i_u}\notin RL$，则更新撤销列表 $RL'=RL\cup\{u_{i_u}\}$。

输出：$\perp$ 或者 RL'。

$CTUpdate(RL',pp,\boldsymbol{ct})\to\{\boldsymbol{ct}'\}$：此算法由 DSM 运行。输入公共参数 pp、密文 $\boldsymbol{ct}$，以及撤销列表 RL'。

只需更新与撤销列表相关的部分密文即可，只有 $\boldsymbol{c}_2=\{\hat{\boldsymbol{c}}_{\hat{\gamma}}\}$，$\hat{\gamma}\in KUNodes(BT,RL)$，这一部分密文与撤销列表相关。对于 $\hat{\gamma}\in KUNodes(BT,RL)$，$\hat{\boldsymbol{c}}_{\hat{\gamma}}=\boldsymbol{U}_{\hat{\gamma}}^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{S}_{\hat{\gamma}}\boldsymbol{e}_0\in Z_q^m$。对于 $j'\in KUNodes(BT,RL')$：

1) 若存在 $j\in KUNodes(BT,RL)$，$j=j'$，那么 $\hat{\boldsymbol{c}}_j'=\hat{\boldsymbol{c}}_j$。

2) 若存在 $j\in KUNodes(BT,RL)$，j 是 j' 的祖先节点，分为以下两种情况：

(1) 假如 j' 是非叶子节点。

$$\begin{aligned}\hat{\boldsymbol{c}}_j'&=(\boldsymbol{M}_{j'}^{\mathrm{T}})(\boldsymbol{M}_j^{\mathrm{T}})^{-1}\cdot\hat{\boldsymbol{c}}_j\\&=(\boldsymbol{M}_{j'}^{\mathrm{T}})(\boldsymbol{M}_j^{\mathrm{T}})^{-1}(\boldsymbol{U}_j^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{S}_j\boldsymbol{e}_0)\\&=\boldsymbol{U}_{j'}^{\mathrm{T}}\boldsymbol{s}+(\boldsymbol{M}_{j'}^{\mathrm{T}})(\boldsymbol{M}_j^{\mathrm{T}})^{-1}\boldsymbol{S}_j\boldsymbol{e}_0\end{aligned}$$

(2) 假如 j' 是叶子节点。

$$\begin{aligned}\hat{\boldsymbol{c}}_{j'}&=(\boldsymbol{N}_{j'}^{\mathrm{T}})(\boldsymbol{M}_j^{\mathrm{T}})^{-1}\hat{\boldsymbol{c}}_j\\&=(\boldsymbol{N}_{j'}^{\mathrm{T}})(\boldsymbol{M}_j^{\mathrm{T}})^{-1}(\boldsymbol{U}_j^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{S}_j\boldsymbol{e}_0)\\&=\boldsymbol{U}_{j'}^{\mathrm{T}}\boldsymbol{s}+(\boldsymbol{N}_{j'}^{\mathrm{T}})(\boldsymbol{M}_j^{\mathrm{T}})^{-1}\boldsymbol{S}_j\boldsymbol{e}_0\end{aligned}$$

$\boldsymbol{c}_2'=\{\hat{\boldsymbol{c}}_{j'}\}$，$j'\in KUNodes(BT,RL')$

输出：密文 $\boldsymbol{ct}'=(\boldsymbol{c},\boldsymbol{c}_0,\boldsymbol{c}_1,\boldsymbol{c}_2',\boldsymbol{c}_{\mathrm{out}})$。

5 正确性

5.1 密文解密

使用以下等式来实现解密：

$$\boldsymbol{c}=[\boldsymbol{c}_{\mathrm{in}};\bar{\boldsymbol{c}}_1;\cdots;\bar{\boldsymbol{c}}_l;\boldsymbol{c}_1;\cdots;\boldsymbol{c}_d]\in Z_q^{(l+d+1)m}$$

其中，$\bar{\boldsymbol{c}}_i=(\boldsymbol{B}_i+att_i\boldsymbol{G})^{\mathrm{T}}\boldsymbol{s}+\bar{\boldsymbol{e}}_i$，$i\in\{1,\cdots,l\}$，当 $f(att)=0$ 时，$\bar{\boldsymbol{c}}_f=\boldsymbol{B}_f^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e}_f$，$\boldsymbol{c}=[\boldsymbol{c}_{\mathrm{in}};\bar{\boldsymbol{c}}_f;\boldsymbol{c}_{g_{\hat{t}_1}};\cdots;\boldsymbol{c}_{g_{\hat{t}_\eta}}]$。

$$\begin{aligned}(\mu_1,\cdots,\mu_m)&=\boldsymbol{c}_{\mathrm{out}}-\boldsymbol{R}^{\mathrm{T}}\boldsymbol{c}-\boldsymbol{w}_{(\gamma,\hat{\gamma})}\\&=\boldsymbol{D}^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{U}^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e}_{\mathrm{out}}+\mu\left\lfloor\frac{q}{2}\right\rfloor-\boldsymbol{R}^{\mathrm{T}}[\boldsymbol{c}_{\mathrm{in}};\bar{\boldsymbol{c}}_f;\\&\quad\boldsymbol{c}_{g_{\hat{t}_1}};\cdots;\boldsymbol{c}_{g_{\hat{t}_\eta}}]-R_f^{\mathrm{T}}(\boldsymbol{c}_0;\boldsymbol{c}_f)-\boldsymbol{R}_\gamma^{\mathrm{T}}(\boldsymbol{c}_0;\hat{\boldsymbol{c}}_\gamma)\\&=\boldsymbol{D}^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{U}^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e}_{\mathrm{out}}+\mu\left\lfloor\frac{q}{2}\right\rfloor-\boldsymbol{R}^{\mathrm{T}}[\boldsymbol{A}|\boldsymbol{B}_f|\\&\quad\boldsymbol{A}_{g_{\hat{t}_1}}|\cdots|\boldsymbol{A}_{g_{\hat{t}_\eta}}]^{\mathrm{T}}s-\boldsymbol{R}_f^{\mathrm{T}}(\boldsymbol{A}^{\mathrm{T}};\boldsymbol{B}_f^{\mathrm{T}})\boldsymbol{s}-\boldsymbol{R}_\gamma^{\mathrm{T}}\cdot\\&\quad(\boldsymbol{A}^{\mathrm{T}};\boldsymbol{U}_\gamma^{\mathrm{T}})\boldsymbol{s}-\boldsymbol{R}^{\mathrm{T}}[\boldsymbol{e}_{\mathrm{in}};\boldsymbol{e}_f;\boldsymbol{e}_{g_{\hat{t}_1}};\cdots;\boldsymbol{e}_{g_{\hat{t}_\eta}}]-\\&\quad\boldsymbol{R}_f^{\mathrm{T}}\cdot(\boldsymbol{e}_0;\boldsymbol{e}_f)-R_\gamma^{\mathrm{T}}\cdot(\boldsymbol{e}_0;S_{\hat{\gamma}}\boldsymbol{e}_0)\\&=\mu\left\lfloor\frac{q}{2}\right\rfloor+\boldsymbol{e}_{\mathrm{out}}-\boldsymbol{R}^{\mathrm{T}}[\boldsymbol{e}_{\mathrm{in}};\boldsymbol{e}_f;\boldsymbol{e}_{g_{\hat{t}_1}};\cdots;\boldsymbol{e}_{g_{\hat{t}_\eta}}]-\\&\quad\boldsymbol{R}_f^{\mathrm{T}}\cdot(\boldsymbol{e}_0;\boldsymbol{e}_f)-\boldsymbol{R}_\gamma^{\mathrm{T}}(\boldsymbol{e}_0;S_{\hat{\gamma}}\boldsymbol{e}_0)\end{aligned}$$

令 $x=\boldsymbol{e}_{\mathrm{out}}-\boldsymbol{R}^{\mathrm{T}}[\boldsymbol{e}_{\mathrm{in}};\boldsymbol{e}_f;\boldsymbol{e}_{g_{\hat{t}_1}};\cdots;\boldsymbol{e}_{g_{\hat{t}_\eta}}]-\boldsymbol{R}_\gamma^{\mathrm{T}}(\boldsymbol{e}_0;S_{\hat{\gamma}}\boldsymbol{e}_0)-\boldsymbol{R}_f^{\mathrm{T}}\cdot(\boldsymbol{e}_0;\boldsymbol{e}_f)$，因为 $20\sqrt{2}\chi_{\max}s(m+1)^{d+2}+3(\eta+2)^2m^{1+\frac{\eta}{2}}\cdot s\chi_{\max}\alpha^2<\frac{q}{4}$，所以 $\|x\|<\frac{q}{4}$。因此当 $|\mu_i|<\frac{q}{4}$ 时 $\mu_i=0$，否则 $\mu_i=1$。

5.2 方案的参数估计

由引理 2 可知，更新后的密文与原始密文不可区分，因此参数估计不考虑更新密文后改变的噪声，只考虑初始密文的噪声即可。本方案的安全参数是 λ，属性数量上界为 l，用户最大数目是 N。由文献[18]可得，$\bar{\boldsymbol{c}}_f\in E_{s,\Delta_1}(0,\boldsymbol{B}_f)$，其中 $\Delta_1<\alpha_{\mathcal{F}}$，$\Delta_2<\alpha_{\Xi}$，$\|\boldsymbol{e}_f\|<20\chi_{\max}m\cdot(m+1)^d$。由计算可得：

$\|\boldsymbol{e}_{\text{out}}\| \leqslant \sqrt{m}\chi_{\max}$，$\|\boldsymbol{e}_0\| \leqslant \sqrt{m}\chi_{\max}$，$\|\boldsymbol{R}_f^{\mathrm{T}}\| \leqslant \sqrt{2}ms$，$\|\boldsymbol{R}_r^{\mathrm{T}}\| < \sqrt{2}ms$，$\|[\boldsymbol{e}_{\text{in}};\boldsymbol{e}_f;\boldsymbol{e}_{g_{i_1}};\cdots;\boldsymbol{e}_{g_{i_\eta}}]\| \leqslant \chi_{\max}+\Delta_1+\eta\Delta_2 < (\eta\alpha_G+\alpha_F+1)\chi_{\max}$，$\|\boldsymbol{R}\| < (\eta+2)\cdot m\delta_\eta$，$\|\boldsymbol{R}^{\mathrm{T}}[\boldsymbol{e}_{\text{in}};\boldsymbol{e}_f;\boldsymbol{e}_{g_{i_1}};\cdots;\boldsymbol{e}_{g_{i_\eta}}]\| < (\eta+2)\cdot m\delta_\eta\cdot(\eta\alpha_G+\alpha_F+1)\chi_{\max}$。令 $\boldsymbol{z}=\boldsymbol{e}_{\text{out}}-\boldsymbol{R}^{\mathrm{T}}[\boldsymbol{e}_{\text{in}};\boldsymbol{e}_f;\boldsymbol{e}_{g_{i_1}};\cdots;\boldsymbol{e}_{g_{i_\eta}}]-\boldsymbol{R}_f^{\mathrm{T}}(\boldsymbol{e}_0;\boldsymbol{e}_f)-\boldsymbol{R}_\gamma^{\mathrm{T}}(\boldsymbol{e}_0;S_\gamma^{\downarrow}\boldsymbol{e}_0)$，为保证方案的正确性，需使 $\|\boldsymbol{z}\| < \frac{q}{4}$ 成立。

$$\begin{aligned}\|\boldsymbol{z}\| <{}& \sqrt{m}\chi_{\max}+(\eta+2)m\delta_\eta(\eta\alpha_S+\alpha_{\mathcal{F}}+1)\chi_{\max}+\\ & \sqrt{2}ms(\sqrt{m}\chi_{\max}+20\chi_{\max}m\cdot(m+1)^d)+\\ & \sqrt{2}ms(\sqrt{m}\chi_{\max}+m\cdot\sqrt{m}\chi_{\max})\end{aligned}$$

其中，$\delta_\eta=\delta_0\cdot(\sqrt{m\log m})^\eta$，$\delta_0=w(\alpha_{\mathcal{F}}\cdot\sqrt{\log m})$。因为 $\alpha=\max\{\alpha_S,\alpha_{\mathcal{F}}\}$，令 $y_1=\sqrt{m}\chi_{\max}+\sqrt{2}ms(\sqrt{m}\cdot\chi_{\max}+20\cdot\chi_{\max}m\cdot(m+1)^d)+\sqrt{2}ms(\sqrt{m}\chi_{\max}+m\sqrt{m}\cdot\chi_{\max})$，可得 $y_1 \leqslant 20\sqrt{2}\chi_{\max}s(m+1)^{d+2}$；令 $y_2=(\eta+2)\cdot m\delta_\eta\cdot(\eta\alpha_S+\alpha_{\mathcal{F}}+1)\chi_{\max}$，可得 $y_2 \leqslant 3(\eta+2)^2\cdot m^{1+\frac{\eta}{2}}\cdot s\chi_{\max}\alpha^2$。所以 $\|\boldsymbol{z}\| \leqslant 20\sqrt{2}\chi_{\max}s(m+1)^{d+2}+3(\eta+2)^2m^{1+\frac{\eta}{2}}s\cdot\chi_{\max}\alpha^2$。

因此只需要不等式 $20\sqrt{2}\chi_{\max}s(m+1)^{d+2}+3(\eta+2)^2\cdot m^{1+\frac{\eta}{2}}s\chi_{\max}\alpha^2 < \frac{q}{4}$ 成立，则本方案正确。

6 安全性分析

定理 1 本方案基于 *DLWE* 困难问题证明 IND-ST-SA-SR-CPA 是安全的。

证明：假设 $\boldsymbol{x}^*=(x_1^*,x_2^*,\cdots,x_l^*)\in\mathcal{X}^l$ 是目标属性向量，$\{t_1^*,t_2^*,\cdots,t_d^*\}$是目标标签集合，$RL^*\subseteq[N]$是撤销列表，$t^*=\{t_1^*,t_2^*,\cdots,t_d^*\}\in T^d$。对于函数 $f\in\mathcal{F}$，挑战者会维持一个元组$(f,I,sk_{f,\varnothing},P_f,C_f)$。其中，$f$ 指的是每个用户生成的关于属性的策略函数，I 指的是用户的身份，$sk_{f,\varnothing}$指的是 AA 为用户 I 生成的初始密钥，P_f代表当前密钥被穿刺的标签集合，C_f代表当前密钥之前已经完成穿刺的标签集合。初始时，P_f 和 C_f 是两个空集合。

证明过程是一个博弈序列。第一个游戏和 IND-SA-CPA (Indistinguishability under Selective Attribute against Chosen Plaintext Attack)游戏不可区分。后两个游戏和 DLWE 问题不可区分。

Game0 Game0 是初始的 IND-ST-SA-SR-CPA 游戏，敌手$\mathcal{A}$借助 IND-ST-SA-SR-CPA 的挑战者攻击方案。在设置阶段，挑战者选择 $l+d+3$ 个均匀随机的矩阵$\boldsymbol{B}_1,\cdots,\boldsymbol{B}_l,\boldsymbol{A}_1,\cdots,\boldsymbol{A}_d,\boldsymbol{U},\boldsymbol{D},\boldsymbol{F}\in Z_q^{n\times m}$，运行 $TrapGen(1^n,1^m,q)\to(\boldsymbol{A},\boldsymbol{T}_{\boldsymbol{A}})$，2.2 节提到的完全二叉树 BT，其中$\boldsymbol{U}_0=\boldsymbol{A}$，挑战者将 $pp=\{\boldsymbol{A},\boldsymbol{A}_1,\cdots,\boldsymbol{A}_d,\boldsymbol{B}_1,\cdots,\boldsymbol{B}_l,\boldsymbol{U},\boldsymbol{D},\boldsymbol{G},BT,\{\boldsymbol{M}_i\}_{i\in[1,N-2]},\{\boldsymbol{N}_j\}_{j\in[N-1,2N-2]}\}$发送给敌手$\mathcal{A}$，主密钥 $msk=\{\boldsymbol{T}_{\boldsymbol{A}},\boldsymbol{F}\}$。挑战者选择 $l+d$ 个均匀随机的矩阵$\boldsymbol{S}_1^*,\cdots,\boldsymbol{S}_l^*,\boldsymbol{R}_1^*,\cdots,\boldsymbol{R}_d^*\in\{\pm1\}^{m\times m}$。

Game1 挑战者以与 Game0 不同的方式产生$\boldsymbol{B}_1,\cdots,\boldsymbol{B}_l,\boldsymbol{A}_1,\cdots,\boldsymbol{A}_d$，挑战者选择 $l+d$ 个均匀随机的矩阵$\boldsymbol{S}_1^*,\cdots,\boldsymbol{S}_l^*,\boldsymbol{R}_1^*,\cdots,\boldsymbol{R}_d^*\in\{\pm1\}^{m\times m}$。设置阶段，在 Game0 中产生矩阵 $\boldsymbol{A}$，设置$\boldsymbol{B}_1,\cdots,\boldsymbol{B}_l,\boldsymbol{A}_1,\cdots,\boldsymbol{A}_d$ 如下：

$\boldsymbol{B}_i=\boldsymbol{A}\boldsymbol{S}_i^*-x_i^*\boldsymbol{G}, i\in\{1,\cdots,l\}$

$\boldsymbol{A}_j=\boldsymbol{A}\boldsymbol{R}_j^*-t_j^*\boldsymbol{G}, j\in\{1,\cdots,d\}$

构建完全二叉树 BT：对于每一个 $r\in BT$，选择$\boldsymbol{S}_r^*\in\{\pm1\}^{m\times m}$，计算：

$\boldsymbol{U}_r^*=\boldsymbol{A}\boldsymbol{S}_r^*, r\in KUNodes(BT,RL^*)$

$\boldsymbol{U}_r^*=\boldsymbol{A}\boldsymbol{S}_r^*+\boldsymbol{G}, r\notin KUNodes(BT,RL^*)$

在挑战查询阶段，挑战者计算$\boldsymbol{S}_i^{*\mathrm{T}}\boldsymbol{e}_0, i\in[l]$用于产生密文$\boldsymbol{c}_1$，$\boldsymbol{S}_r^*\boldsymbol{e}_0, r\in BT$ 用于产生密文$\boldsymbol{c}_2$ 以及$\boldsymbol{e}_0,\boldsymbol{e}_1,\boldsymbol{e}_0',\boldsymbol{e}_{\text{out}}\in\chi^m$。

之后的游戏和 Game0 一样，在$\mathcal{A}$的视角，Game0 和 Game1 是不可区分的。在 Game0 中，公开参数中的$\boldsymbol{B}_i,\boldsymbol{A}_j,\boldsymbol{U}_r, i\in[l], j\in[d], r\in BT$，可以通过下式计算：$\boldsymbol{B}_i=\boldsymbol{A}\boldsymbol{S}_i^*-x_i^*\boldsymbol{G}, i\in\{1,\cdots,l\}$，$\boldsymbol{A}_j=\boldsymbol{A}\boldsymbol{R}_j^*-t_j^*\boldsymbol{G}, j\in\{1,\cdots,d\}$，$\boldsymbol{U}_r^*=\boldsymbol{A}\boldsymbol{S}_r^*+\rho_r\boldsymbol{G},\rho_r\in\{0,1\}$，$\boldsymbol{S}_i^*,\boldsymbol{R}_j^*,\boldsymbol{S}_r^*\in\{\pm1\}^{m\times m}$，根据引理 2，在 $Z_q^{n\times m}$ 中选取均匀随机矩阵$\boldsymbol{B}_i,\boldsymbol{A}_j,\boldsymbol{U}_r, i\in[l], j\in[d], r\in BT$。分布$(\boldsymbol{A},\boldsymbol{A}\boldsymbol{S}_i^*-x_i^*\boldsymbol{G},\boldsymbol{S}_i^{*\mathrm{T}}\boldsymbol{e}_0)$和$(\boldsymbol{A},\boldsymbol{B}_i,\boldsymbol{S}_i^{*\mathrm{T}}\boldsymbol{e}_0)$不可区分，分布$(\boldsymbol{A},\boldsymbol{A}\boldsymbol{R}_j^*-t_j^*\boldsymbol{G},\boldsymbol{R}_j^{*\mathrm{T}}\boldsymbol{e}_0)$和$(\boldsymbol{A},\boldsymbol{A}_j,\boldsymbol{R}_j^{*\mathrm{T}}\boldsymbol{e}_0)$不可区分。分布$(\boldsymbol{A},\boldsymbol{A}\boldsymbol{S}_r^*+\rho_r\boldsymbol{G},\boldsymbol{S}_r^{*\mathrm{T}}\boldsymbol{e}_0)$和$(\boldsymbol{A},\boldsymbol{U}_r^*,\boldsymbol{S}_r^{*\mathrm{T}}\boldsymbol{e}_0)$不可区分。因此 Game0 中的公共参数$(\boldsymbol{B}_1,\cdots,\boldsymbol{B}_l,\boldsymbol{A}_1,\cdots,\boldsymbol{A}_d,BT)$和 Game1 中的公共参数不可区分，即 Game0 和 Game1 不可区分。

Game2 挑战者选择一个随机矩阵 $\boldsymbol{A}\in Z_q^{n\times m}$ 替代 $TrapGen(1^n,1^m,q)$算法产生的 $\boldsymbol{A}$。均匀随机地选择一个矩阵 $\boldsymbol{F}\in Z_q^{n\times m}$。根据引理 2 可知，Game1 和 Game2 中的 $\boldsymbol{A},\boldsymbol{F}$ 不可区分。$\boldsymbol{B}_1,\cdots,\boldsymbol{B}_l,\boldsymbol{A}_1,\cdots,\boldsymbol{A}_d$，的构造和 Game1 一样。在 Game2 中挑战者没有 $\Lambda_q^{\perp}(\boldsymbol{A})$的陷门，但是可以回答所有的私钥提取查询。在查询阶段 1，敌手进行自适应查询，挑战者执行如下操作。

注意，只有函数-索引$(f,I)\in\mathcal{F}\times[N]$，满足 $f(x^*)\neq0$ 或者 $f(x^*)=0$ 且 $P_f\cap\{t_1^*,t_2^*,\cdots,t_d^*\}\neq\varnothing$ 或者 $f(x^*)=0$，$P_f\cap\{t_1^*,t_2^*,\cdots,t_d^*\}=\varnothing$ 且 $I\in RL^*$ 时允许所有的私钥提取查询。为了产生函数索引对$(f,I)\in\mathcal{F}\times[N]$的私钥，挑战者需要输入 $f\in\mathcal{F}$、用户身份 $I\in[N]$、挑战属性 $x^*\in\{0,1\}^l$、撤销列表 $RL^*\subseteq[N]$，输出 $sk_{f,\mathcal{P}_\eta}=\{\boldsymbol{T}_{f,\mathcal{P}_\eta},\boldsymbol{T}'_{(\boldsymbol{A}|\boldsymbol{B}_f)},\boldsymbol{R}_f,\boldsymbol{K},\{\boldsymbol{R}_\gamma\}_{\gamma\in Path(I)}\}$，其中$\boldsymbol{T}'_{(\boldsymbol{A}|\boldsymbol{B}_f)}=\boldsymbol{T}_{(\boldsymbol{A}|\boldsymbol{B}_f)}\cdot(\boldsymbol{D}|\boldsymbol{U})^{\mathrm{T}}$。

1) $Q_{\text{Puncture}}(f\in\mathcal{F},\hat{t})$：给定一个函数 f、目标属性向量 $\boldsymbol{x}^*$，以及一个标签 $\hat{t}$。

情形 1 对于 f，如果 $f(x^*)\neq0$：

(1)若存在元组$(f,I,sk_{f,\mathcal{P}_i},P_f,C_f)$，执行算法 $Puncture(pp,sk_{f,\mathcal{P}_i},\hat{t}_\eta)\to sk_{f,\mathcal{P}_{i+1}}$，挑战者添加 $\hat{t}$ 到 P_f 中，使用新的$(f,I,sk_{f,\mathcal{P}_{i+1}},P_f,C_f)$替代旧的元组$(f,I,sk_{f,\mathcal{P}_i},P_f,C_f)$。

(2)若不存在元组$(f,I,sk_{f,\mathcal{P}_i},P_f,C_f)$，则计算$\boldsymbol{B}_f\leftarrow Eval_{pk}^{\mathcal{T}}(f,(\boldsymbol{B}_1,\cdots,\boldsymbol{B}_l))$，并计算$\boldsymbol{S}_f^*\leftarrow Eval_{\text{sim}}(f,\{x_j^*,\boldsymbol{S}_j^*\}_{j=1}^l,\boldsymbol{A})$，以及 $\boldsymbol{K}=Matrix\_XOR(\boldsymbol{F},\boldsymbol{U}_I)$。令$\boldsymbol{B}_f=\boldsymbol{A}\boldsymbol{S}_f^*+f(x^*)\boldsymbol{G}$，$\|\boldsymbol{S}_f^*\|_2\leqslant\alpha_{\mathcal{F}}$，假设$\boldsymbol{T}_{\boldsymbol{G}}$是 $\boldsymbol{G}$ 的陷门，当 $f(x^*)=1$ 时，$\boldsymbol{T}_{\boldsymbol{G}}$也是 $f(x^*)\boldsymbol{G}$ 的陷门。执行$\boldsymbol{T}_{\boldsymbol{A}|\boldsymbol{B}_f}^{\text{EL}}\leftarrow ExtendLeft(\boldsymbol{A},f(x^*)\boldsymbol{G},\boldsymbol{T}_{\boldsymbol{G}},\boldsymbol{S}_f^*)$，最后计算$\boldsymbol{T}_{(\boldsymbol{A}|\boldsymbol{B}_f)}\leftarrow RandBasis([\boldsymbol{A}|\boldsymbol{B}_f],\boldsymbol{T}_{(\boldsymbol{A}|\boldsymbol{B}_f)}^{\text{EL}},\delta_0)$。

①如果 $I\in RL^*$ 选择均匀随机的矩阵$\boldsymbol{D}_I\in Z_q^{n\times m}$，则对于每一个 $\gamma\notin KUNodes(BT,RL^*)$，$\boldsymbol{U}_r=\boldsymbol{A}\boldsymbol{S}_r^*+\boldsymbol{G}$。所以对于每一个 $\gamma\in Path(I)$，采样$\boldsymbol{R}_\gamma\leftarrow SampleRight(\boldsymbol{A},\boldsymbol{G},\boldsymbol{S}_\gamma^*,\boldsymbol{T}_{\boldsymbol{G}},\boldsymbol{D}-$

$\boldsymbol{D}_I, s)$和$\boldsymbol{R}_f \leftarrow SampleRight(\boldsymbol{A}, \boldsymbol{G}, \boldsymbol{B}_f, \boldsymbol{T}_G, \boldsymbol{D}_I, I)$。

②若$I \notin RL^*$，存在一个节点$\gamma^* \in Path(I) \cap KUNodes(BT, RL^*)$，采样$\boldsymbol{R}_{\gamma^*} \leftarrow D_{\mathbb{Z}^{2m\times m},s}$，设置$\boldsymbol{D}_I = \boldsymbol{D} - (\boldsymbol{A}|\boldsymbol{U}_{\gamma^*})\boldsymbol{R}_{\gamma^*}$，则对于$\gamma \notin KUNodes(BT, RL^*)$，$\boldsymbol{U}_r = \boldsymbol{A}\boldsymbol{S}_r^* + \boldsymbol{G}$。所以对于$\gamma \in Path(I)\backslash\{\gamma^*\}$，采样$\boldsymbol{R}_\gamma \leftarrow SampleRight(\boldsymbol{A}, \boldsymbol{G}, \boldsymbol{S}_\gamma^*, \boldsymbol{T}_G, \boldsymbol{D} - \boldsymbol{D}_I, s)$，采样$\boldsymbol{R}_f \leftarrow SampleRight(\boldsymbol{A}, f(x^*)\boldsymbol{G}, \boldsymbol{S}_f^*, \boldsymbol{T}_G, \boldsymbol{D}_I, I)$。

设置初始密钥$sk_{f,\emptyset} = \{\boldsymbol{T}_{(\boldsymbol{A}|\boldsymbol{B}_f)}, \boldsymbol{T}'_{(\boldsymbol{A}|\boldsymbol{B}_f)}, \boldsymbol{R}_f, \boldsymbol{K}, \{\boldsymbol{R}_\gamma\}_{\gamma\in Path(I)}\}$，之后执行算法$Puncture(pp, sk_{f,\emptyset}, \hat{t}) \rightarrow \{sk_{f,\mathcal{P}_1}\}$，其中$\mathcal{P}_1 = \{\hat{t}\}$。将$\{\hat{t}\}$添加到$P_f$，构造一个新的元组$(f, I, sk_{f,\mathcal{P}_1}, P_f, C_f)$，其中$P_f = \{\hat{t}\}$，$C_f = \emptyset$。

情形 2　对于f，如果$f(x^*) = 0, I \in RL^*$。

(1)若存在元组$(f, I, sk_{f,\mathcal{P}_i}, P_f, C_f)$，则直接执行算法$Puncture(pp, sk_{f,\mathcal{P}_i}, \hat{t}_\eta) \rightarrow sk_{f,\mathcal{P}_{i+1}}$，将挑战者添加$\hat{t}$到$P_f$中，使用新的$(f, I, sk_{f,\mathcal{P}_{i+1}}, P_f, C_f)$替代旧的元组$(f, I, sk_{f,\mathcal{P}_i}, P_f, C_f)$。

(2)若不存在元组，则密钥构造和元组构造与上述情形 1 中①的情况相同。

情形 3　对于f，如果$f(x^*) = 0, I \notin RL^*$：

(1)若存在元组$(f, I, -, P_f, C_f)$，则挑战者只添加$\hat{t}$到P_f中，使用新元组替代旧元组。

(2)若不存在元组$(f, I, -, P_f, C_f)$，则添加$\hat{t}$到P_f中，构造一个新元组$(f, I, -, P_f, C_f)$，$P_f = \{\hat{t}\}$，$C_f = \emptyset$。

Game1 和 Game2 中的矩阵$\boldsymbol{R}_f, \{\boldsymbol{R}_r\}, r \in Path(I), \boldsymbol{F} \in \mathbb{Z}_q^{n\times m}$不可区分，证明如下。

(1)Game1 中，$\boldsymbol{R}_f$通过左采样算法$SampleLeft$产生，满足$(\boldsymbol{A}|\boldsymbol{B}_f)\boldsymbol{R}_f = \boldsymbol{D}_I$；Game2 的情形 1 和情形 2 中，$\boldsymbol{R}_f$通过右采样算法$SampleRight$产生，满足$(\boldsymbol{A}|\boldsymbol{B}_f)\boldsymbol{R}_f = \boldsymbol{D}_I$。根据引理 2，通过$SampleLeft$产生的$\boldsymbol{R}_f$和通过$SampleRight$产生的$\boldsymbol{R}_f$分布接近$D_{\mathbb{Z}^{2m\times m},s}$，满足$\Lambda_q^{\boldsymbol{D}_I}(\boldsymbol{A}|\boldsymbol{B}_f)$。因此，Game1 和 Game2 中矩阵$\boldsymbol{R}_f$的分布不可区分。

(2)根据引理 2，*Game*1 中的$SampleLeft$算法产生的$\{\boldsymbol{R}_\gamma\}$和$\gamma \in Path(I)$与通过$SampleRight$算法从$D_{\mathbb{Z}^{2m\times m},s}$采样的$\boldsymbol{R}_\gamma$是不可区分的。

(3)Game1 中矩阵$\boldsymbol{F}$是均匀随机选取的，Game2 中矩阵$\boldsymbol{F}$也是均匀随机选取的，因此 Game1 和 Game2 中矩阵$\boldsymbol{F}$也是不可区分的。

因此，Game1 和 Game2 中的公共参数和查询阶段的回复是不可区分的。

2)$Q_{\text{Corrupt}}(f)$：当敌手第一次对函数f进行查询时，挑战者考虑以下两种情况。

(1)对于f，如果$f(x^*) \neq 0$或者$f(x^*) = 0$且$I \in RL^*$：

①若存在元组$(f, I, sk_{f,\mathcal{P}_i}, P_f, C_f)$，则挑战者返回$sk_{f,\mathcal{P}_i}$给敌手$\mathcal{A}$，设置$C_f \leftarrow P_f$。

②否则，计算初始密钥(如上所述)，将$sk_{f,\emptyset}$返回给敌手$\mathcal{A}$，设置$C_f \leftarrow P_f = \emptyset$。

注意，对于f，$f(x^*) \neq 0$或者$f(x^*) = 0$且$I \in RL^*$，并不要求$P_f \cap \{t_1^*, t_2^*, \cdots, t_d^*\}$是否为空集。在任何情况下，$f(x^*) \neq 0$时挑战者都可以响应穿刺查询，所以挑战者可以计算初始密钥。

(2)对于f，如果$f(x^*) = 0, I \notin RL^*$：

①若存在元组$(f, I, -, P_f, C_f)$，检查$P_f \cap \{t_1^*, t_2^*, \cdots, t_d^*\}$是否为空集。

a)若$P_f \cap \{t_1^*, t_2^*, \cdots, t_d^*\} = \emptyset$，挑战者输出$\perp$。

b)若$P_f \cap \{t_1^*, t_2^*, \cdots, t_d^*\} \neq \emptyset$，至少存在一个$t_i^* \in P_f$，$g_{\hat{t}_i}(t^*) \neq 0$。假设$P_f = \{\hat{t}_1, \cdots, \hat{t}_k\}$，$g_{\hat{t}_i}(t^*) \neq 0, j \in \{1, \cdots, k\}$，计算$\boldsymbol{A}_{g_{\hat{t}_j}} \leftarrow Eval_{pk}^S(\{\boldsymbol{A}_i\}_{i=1}^d, g_{\hat{t}_j})$，计算$\boldsymbol{R}_{g_{\hat{t}_k}}^* \leftarrow Eval_{\text{sim}}(g_{\hat{t}_k}, \{t_j^*, R_j^*\}_{j=1}^d, \boldsymbol{A})$，$\boldsymbol{A}_{g_{\hat{t}_k}}^* = \boldsymbol{A}\boldsymbol{R}_{g_{\hat{t}_k}}^* - g_{\hat{t}_k}(t^*)\boldsymbol{G}$，$\|\boldsymbol{R}_{g_{\hat{t}_k}}^*\| \leqslant \alpha_{\mathfrak{S}}$，计算$\boldsymbol{T}_{(\boldsymbol{A}|\boldsymbol{A}_{g_{\hat{t}_k}}^*)}^{\text{EL}} \leftarrow ExtendLeft(\boldsymbol{A}, -g_{\hat{t}_k}(t^*)\boldsymbol{G}, \boldsymbol{T}_G, \boldsymbol{R}_{g_{\hat{t}_k}}^*)$，计算$\boldsymbol{T}_{(\boldsymbol{A}|\boldsymbol{A}_{g_{\hat{t}_k}}^*|\boldsymbol{B}_f|\boldsymbol{A}_{g_{\hat{t}_1}}|\cdots|\boldsymbol{A}_{g_{\hat{t}_{k-1}}})}^{\text{ER}} \leftarrow ExtendRight((\boldsymbol{A}|\boldsymbol{A}_{g_{\hat{t}_k}}), \boldsymbol{T}_{(\boldsymbol{A}|\boldsymbol{A}_{g_{\hat{t}_k}}^*)}^{\text{EL}}, (\boldsymbol{B}_f|\boldsymbol{A}_{g_{\hat{t}_1}}|\cdots|\boldsymbol{A}_{g_{\hat{t}_{k-1}}}))$。通过选择矩阵$\boldsymbol{T}_{(\boldsymbol{A}|\boldsymbol{A}_{g_{\hat{t}_k}}^*|\boldsymbol{B}_f|\boldsymbol{A}_{g_{\hat{t}_1}}|\cdots|\boldsymbol{A}_{g_{\hat{t}_{k-1}}})}^{\text{ER}}$的行，得到矩阵$\boldsymbol{T}_{(\boldsymbol{A}|\boldsymbol{B}_f|\boldsymbol{A}_{g_{\hat{t}_1}}|\cdots|\boldsymbol{A}_{g_{\hat{t}_{k-1}}})}^{\text{ER}}$是矩阵$(\boldsymbol{A}|\boldsymbol{B}_f|\boldsymbol{A}_{g_{\hat{t}_1}}|\cdots|\boldsymbol{A}_{g_{\hat{t}_{k-1}}})$的陷门。$\boldsymbol{T}_{f,\mathcal{P}_k}^{\text{ER}} = \boldsymbol{T}_{(\boldsymbol{A}|\boldsymbol{B}_f|\boldsymbol{A}_{g_{\hat{t}_k}}^*|\cdots|\boldsymbol{A}_{g_{\hat{t}_{k-1}}})}^{\text{ER}}$，$\|\boldsymbol{T}_{f,\mathcal{P}_k}^{\text{ER}}\|_{\text{GS}} \leqslant \|\boldsymbol{T}_G\|_{\text{GS}} \cdot \|\boldsymbol{R}_{g_{\hat{t}_k}}^*\| \leqslant \sqrt{5}\alpha_S$。计算陷门$\boldsymbol{T}_{f,\mathcal{P}_k} \leftarrow RandBasis([\boldsymbol{A}|\boldsymbol{B}_f|\boldsymbol{A}_{g_{\hat{t}_1}}|\cdots|\boldsymbol{A}_{g_{\hat{t}_k}}], \boldsymbol{T}_{f,\mathcal{P}_k}^{\text{ER}}, \delta_0)$，$k = P_f$。$\boldsymbol{R}_f, \boldsymbol{K}, \{\boldsymbol{R}_\gamma\}_{\gamma\in Path(I)}$的计算同情形 2 中$I \in RL^*$。输出函数的穿刺密钥$sk_{f,\mathcal{P}_k} = \{\boldsymbol{T}_{f,\mathcal{P}_k}, \boldsymbol{T}'_{f,\mathcal{P}_k}, R_f, K, \{R_\gamma\}_{\gamma\in Path(I)}\}$，其中$\boldsymbol{T}'_{f,\mathcal{P}_k} = \boldsymbol{T}_{f,\mathcal{P}_k} \cdot (\boldsymbol{D}|\boldsymbol{U})^{\text{T}}$。计算$sk_{f,\mathcal{P}_k}$，返回给$\mathcal{A}$，设置$C_f \leftarrow P_f$。

②若不存在元组，则输出$\perp$。

③其他情况下的查询均输出$\perp$。

Game2 和 Game1 相同，由于 Game2 和 Game1 的公共参数和查询阶段不可区分，因此敌手$\mathcal{A}$在 Game2 和 Game1 中的优势可忽略不计。

Game3　Game3 与 Game2 中，挑战密文$\boldsymbol{ct} = (\boldsymbol{c}, \boldsymbol{c}_0, \boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_{\text{out}})$，随机选取$\boldsymbol{c} \in \mathbb{Z}_q^{(l+d+1)m}$，$\boldsymbol{c}_0 \in \mathbb{Z}_q^m$，$\boldsymbol{c}_1 \in \mathbb{Z}_q^{lm}$，$\boldsymbol{c}_2 = \{\overset{\wedge}{\boldsymbol{c}}_{\overset{\wedge}{\gamma}}\} \in \mathbb{Z}_q^m$，$\overset{\wedge}{\gamma} \in KUNodes(BT, RL)$，$\boldsymbol{c}_{\text{out}} \in \mathbb{Z}_q^m$，其他的参数与 Game2 相同，因此敌手$\mathcal{A}$在 Game3 的优势为 0。

通过规约到 DLWE 问题，表明对于 PPT 敌手，Game3 与 Game2 是不可区分的。

从 DLWE 问题规约：假设敌手$\mathcal{A}$以不可忽略的优势区分 Game3 与 Game2。使用$\mathcal{A}$，构造一个 DLWE 构造器$\mathcal{B}$。

DLWE 实例：$\mathcal{B}$获得 DLWE 挑战，由 3 个随机向量$\boldsymbol{A}, \boldsymbol{D}, \boldsymbol{U} \in \mathbb{Z}_q^{n\times m}$和$\boldsymbol{c}_0 \in \mathbb{Z}_q^m, \boldsymbol{c}_{\text{out}} \in \mathbb{Z}_q^m$组成，$\boldsymbol{c}_0, \boldsymbol{c}_{\text{out}}$要么随机在$\mathbb{Z}_q^m$中选取，要么$\boldsymbol{c}_0 = \boldsymbol{A}^{\text{T}}\boldsymbol{s} + \boldsymbol{e}_0 \in \mathbb{Z}_q^m$，$\boldsymbol{c}_{\text{out}} = \boldsymbol{D}^{\text{T}}\boldsymbol{s} + \boldsymbol{U}^{\text{T}}\boldsymbol{s} + \boldsymbol{e}_{\text{out}} = (\boldsymbol{D}+\boldsymbol{U})^{\text{T}}\boldsymbol{s} + \boldsymbol{e}_{\text{out}} \in \mathbb{Z}_q^m$，$\boldsymbol{e}_0, \boldsymbol{e}_{\text{out}} \in \chi^m$，$\boldsymbol{s} \in \mathbb{Z}_q^n$。$\mathcal{B}$借助$\mathcal{A}$的优势区分这两种分布。

系统初始化：$\mathcal{A}$声明目标属性向量$\boldsymbol{x}^* = (x_1^*, x_2^*, \cdots, x_l^*) \in \mathcal{X}^l$和目标标签集$\{t_1^*, t_2^*, \cdots, t_d^*\}$，以及撤销列表$RL^* \subseteq [N]$。

参数设置阶段：$\mathcal{B}$构造公共参数同 Game2，选择随机矩阵$\boldsymbol{A}, \boldsymbol{D}, \boldsymbol{F} \in \mathbb{Z}_q^{n\times m}$。挑战者选择$(l+d)$个均匀随机的矩阵$\boldsymbol{S}_1^*, \cdots, \boldsymbol{S}_l^*, \boldsymbol{R}_1^*, \cdots, \boldsymbol{R}_d^* \in \{\pm 1\}^{m\times m}$。设置$\boldsymbol{B}_1, \cdots, \boldsymbol{B}_l, \boldsymbol{A}_1, \cdots, \boldsymbol{A}_d$如下：

$$\boldsymbol{B}_i = \boldsymbol{A}\boldsymbol{S}_i^* - x_i^*\boldsymbol{G}, i \in \{1, \cdots, l\}$$

$$\boldsymbol{A}_j = \boldsymbol{A}\boldsymbol{R}_j^* - t_j^*\boldsymbol{G}, j \in \{1, \cdots, d\}$$

构建一棵完全二叉树 BT，对于每一个 $r\in BT$，选择$\boldsymbol{S}_r^*\in\{\pm1\}^{m\times m}$，设置：

$\boldsymbol{U}_r^*=\boldsymbol{AS}_r^*$，$r\in KUNodes(BT,RL^*)$

$\boldsymbol{U}_r^*=\boldsymbol{AS}_r^*+\boldsymbol{G}$，$r\notin KUNodes(BT,RL^*)$

$\mathcal{B}$发送 $pp=\{\boldsymbol{A},\boldsymbol{A}_1,\cdots,\boldsymbol{A}_d,\boldsymbol{B}_1,\cdots,\boldsymbol{B}_l,\boldsymbol{U},\boldsymbol{D},\boldsymbol{G},BT,\{\boldsymbol{M}_i\}_{i\in[1,N-2]},\{\boldsymbol{N}_j\}_{j\in[N-1,2N-2]}\}$给$\mathcal{A}$。

密钥查询阶段 1：同 Game2。

挑战阶段：$\mathcal{A}$发送 $\mu_0,\mu_1\in\{0,1\}^m$ 给$\mathcal{B}$，$\mathcal{B}$选择一个随机比特 $\beta\in\{0,1\}$，计算 $\boldsymbol{c}^*=[\boldsymbol{c}_{\text{in}};\bar{\boldsymbol{c}}_1;\cdots;\bar{\boldsymbol{c}}_l;\boldsymbol{c}_1;\cdots;\boldsymbol{c}_d]\in Z_q^{(l+d+1)m}$，$\boldsymbol{c}_0^*=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e}_0\in Z_q^m$，$\boldsymbol{c}_1^*=\boldsymbol{H}_1^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e}\in\boldsymbol{Z}_q^{lm}$，$\boldsymbol{c}_2^*=\{\hat{\boldsymbol{c}}_{\hat{\gamma}}\}$，$\hat{\gamma}\in KUNodes(BT,RL)$，$\hat{\boldsymbol{c}}_\gamma=\boldsymbol{U}_\gamma^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{S}_{\hat{\gamma}}\boldsymbol{e}_0\in Z_q^m$，$\boldsymbol{c}_{\text{out}}^*=(\boldsymbol{D}+\boldsymbol{U})^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e}_{\text{out}}+\mu_\beta\cdot\left\lfloor\frac{q}{2}\right\rfloor\in Z_q^m$。$\mathcal{B}$将 $\boldsymbol{ct}_\beta^*=(\boldsymbol{c}^*,\boldsymbol{c}_0^*,\boldsymbol{c}_1^*,\boldsymbol{c}_2^*,\boldsymbol{c}_{\text{out}}^*)$作为挑战密文发送给$\mathcal{A}$。

假设$\boldsymbol{c}_0^*,\boldsymbol{c}_{\text{out}}^*$通过 DLWE 问题产生$\boldsymbol{c}_0^*=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e}_0\in Z_q^m$，$\boldsymbol{c}_{\text{out}}^*=\boldsymbol{D}^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{U}^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e}_{\text{out}}=(\boldsymbol{D}+\boldsymbol{U})^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e}_{\text{out}}\in Z_q^m$，根据加密算法可得：

(1) $\boldsymbol{H}=(\boldsymbol{A}\mid\boldsymbol{H}_1\mid\boldsymbol{H}_2)\in Z_q^{n\times(l+d+1)m}$，其中 $\boldsymbol{H}_1=(\boldsymbol{B}_1+att_1^*\boldsymbol{G}\mid\cdots\mid\boldsymbol{B}_l+att_l^*\boldsymbol{G})=(\boldsymbol{AS}_1^*\mid\cdots\mid\boldsymbol{AS}_l^*)\in Z_q^{n\times lm}$，$\boldsymbol{H}_2=(\boldsymbol{A}_1+t_1^*\boldsymbol{G}\mid\cdots\mid\boldsymbol{A}_l+t_d^*\boldsymbol{G})=(\boldsymbol{AR}_1^*\mid\cdots\mid\boldsymbol{AR}_d^*)\in Z_q^{n\times dm}$。

(2) $\boldsymbol{e}=(\boldsymbol{S}_1\mid\cdots\mid\boldsymbol{S}_l)^{\mathrm{T}}\cdot\boldsymbol{e}_0\in Z_q^{lm}$，$\boldsymbol{e}'=(\boldsymbol{I}_m\mid\boldsymbol{S}_1\mid\cdots\mid\boldsymbol{S}_l\mid\boldsymbol{R}_1\mid\cdots\mid\boldsymbol{R}_d)^{\mathrm{T}}\cdot\boldsymbol{e}_0'=(\boldsymbol{e}_{\text{in}}^{\mathrm{T}},\bar{\boldsymbol{e}}_1^{\mathrm{T}},\cdots,\bar{\boldsymbol{e}}_l^{\mathrm{T}},\boldsymbol{e}_1^{\mathrm{T}},\cdots,\boldsymbol{e}_d^{\mathrm{T}})\in Z_q^{(l+d+1)m}$。

(3) $\boldsymbol{c}_{\text{in}}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e}_{\text{in}}$，$\bar{\boldsymbol{c}}_i=(\boldsymbol{B}_i+att_i^*\boldsymbol{G})^{\mathrm{T}}\boldsymbol{s}+\bar{\boldsymbol{e}}_i$，$i\in\{1,\cdots,l\}$，$\boldsymbol{c}_j=(\boldsymbol{A}_j+t_j^*\boldsymbol{G})^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{e}_j$，$j\in\{1,\cdots,d\}$。

显然很容易在 Game2 中计算其他密文，$\boldsymbol{c}_{\text{out}}^*=(\boldsymbol{D}+\boldsymbol{U})^{\mathrm{T}}\cdot\boldsymbol{s}+\boldsymbol{e}_{\text{out}}+\mu_\beta\left\lfloor\frac{q}{2}\right\rfloor\in Z_q^m$，因此 $\boldsymbol{ct}_\beta^*=(\boldsymbol{c}^*,\boldsymbol{c}_0^*,\boldsymbol{c}_1^*,\boldsymbol{c}_2^*,\boldsymbol{c}_{\text{out}}^*)$是具有标签集$\{t_1^*,t_2^*,\cdots,t_d^*\}$的明文 μ_β 的有效密文。由引理 2 可知，$\boldsymbol{c}_0^*,\boldsymbol{c}_1^*,\boldsymbol{c}_2^*,\boldsymbol{c}_{\text{out}}^*$是随机的，$\boldsymbol{c}^*\in Z_q^{(l+d+1)m}$ 是均匀随机的。故 Game2 与 *Game*3 中的$\boldsymbol{ct}^*$不可区分。

密钥查询阶段 2：同 Game2。

猜测阶段：$\mathcal{A}$猜测与 Game3 或 Game2 交互，$\mathcal{B}$输出$\mathcal{A}$的回复作为 DLWE 问题的答案。

因此，$\mathcal{B}$解决 DLWE 问题的优势与$\mathcal{A}$在区分 Game3 或 Game2 方面的优势不可区分。

定理 2 本方案可抵抗非法用户合谋攻击。

证明：在 KP-ABE 中，若两个未授权的用户共享密钥，首先，由于密钥依赖于不同的访问策略生成，无法由密文属性集不满足的两个访问策略组合出密文属性集可以满足的一个访问策略，在解密操作的第一步就会中止；其次 AA 为不同用户生成的密钥 $\boldsymbol{D}_l$ 的不同，使得即使他们的密钥合并也无法解密未经授权的密文。

7 性能评估

本章对比了不同方案在安全模型、前向安全、是否可撤销、是否可追踪和量子安全方面的表现，如表 2 所列。本方案在安全性上展现出显著优势：它不仅实现了前向安全，有效降低了密钥泄露后的潜在风险；还具备撤销恶意用户权限的功能，增强了系统的灵活性和防御能力；同时，通过用户身份矩阵的追踪机制，实现了对用户身份的有效追踪；尤为重要的是，作为格上的属性基加密方案，它能够抵御量子计算的攻击威胁，确保在量子计算时代的数据安全。与其他方案相比，本方案在功能方面具有多样性，能够满足多种安全需求，更适用于需要保证通信安全的场景。

表 2 相关方案功能对比

Table 2 Comparison of related scheme functions

方案	安全假设	前向安全	可撤销	可追踪	抗量子安全
Luo 等[12]	DLWE	×	√	×	√
Huang 等[13]	RLWE	×	√	×	√
Guo 等[14]	LWE	×	√	√	√
Dutta 等[17]	DLWE	√	×	×	√
Ours	DLWE	√	√	√	√

表 3 对不同方案在空间存储开销方面的表现进行了对比，其中 l 为属性最大个数，q 为大素数，N 为 UR 二叉树中叶子节点的个数，w 为访问结构中的属性个数，d 为标签集的标签个数，r 为撤销属性的个数，p 为权威机构个数，n 为矩阵行数，m 为矩阵列数，同时设 $k=q+w-s+1$。本方案除了提供较完善的功能之外，还尽可能地减少了存储空间开销。而其他方案仅实现可追踪、可撤销和前向安全中的一到两个功能，所以空间存储开销较小。相比目前功能较全的文献[15]提出的方案，本方案公共参数和密文储存开销仅增加 $O(d)$，解密密钥仅增加 $O(N)$。因此，综合而言，本方案优于其他方案。

表 3 空间存储开销对比

Table 3 Comparison of space storage costs

方案	公共参数	密文	解密密钥
Luo 等[12]	$(l+2N)mn\lceil\log q\rceil+nk\lceil\log q\rceil$	$(l+N+1)m\lceil\log q\rceil+k\lceil\log q\rceil$	$(1+\lceil\log 2^{N+1}\rceil)\cdot 2mk\cdot\lceil\log q\rceil+\lceil\log N\rceil$
Huang 等[13]	$(l+3p+1)mn\lceil\log q\rceil$	$\leqslant(2l+2p)mn\lceil\log q\rceil$	$(l+3p)mn\lceil\log q\rceil$
Guo 等[14]	$(lm+2m+1)m\lceil\log q\rceil+N(m+1)m$	$[(lw+r+1)m+l+3]\lceil\log q\rceil$	$(lm+2m+3)m\lceil\log q\rceil$
Dutta 等[17]	$(l+d+2)mn\lceil\log q\rceil+nm(\lceil\log q\rceil-1)$	$(l+d+2)m\lceil\log q\rceil+d$	$2m^2\lceil\log q\rceil$
Ours	$[(l+d+4)m]n\lceil\log q\rceil+Nm^2+(N-2)m$	$(2l+d+N+3)m\lceil\log q\rceil+d$	$(6m^2+3mn+\lceil\log 2^{N+1}\rceil\cdot 2m^2)\lceil\log q\rceil$

结束语 本文基于格上的 DLWE 困难问题构造了一种具备可追踪、可撤销以及前向安全的多功能属性基加密方案，该方案的核心亮点在于大幅优化了身份矩阵的加/解密效率，显著提升了大规模数据处理场景下的加密操作性能。该方案采用模块化设计，允许用户按需选择功能模块，既能保持核心优势，又可减少计算负担，实现资源高效配置与成本控制。本文通过理论分析证明了该方案的安全性和正确性，与现有文献相比，该方案在撤销效率、安全性等方面均展现出显著优势。此外，该方案还填补了格密码上多功能融合的空白，为未来抗量子属性基加密技术的研究提供了参考。

参 考 文 献

[1] REGEV O. On Lattices, Learning with Errors, Random Linear

Codes, and Cryptography[J]. Journal of the ACM, 2009, 56(6): 1-40.

[2] BOYEN X. Attribute-based Functional Encryption on Lattices [C] // Theory of cryptography conference. Berlin: Springer, 2013: 122-142.

[3] KUCHTA V, MARKOWITCH O. Multi-authority Distributed Attribute-based Encryption with Application to Searchable Encryption on Lattices [C] // Paradigms in Cryptology-Mycrypt 2016. Springer, 2017: 409-435.

[4] SINGAMANENI K K, BUDATI A K, BIKKU T. An Efficient Q-KPABE Framework to Enhance Cloud-Based IoT Security and Privacy [J/OL]. Wireless Personal Communications. (2024). https://doi.org/10.1007/s11277-024-10908-8.

[5] SUN L, ZHAO Z, WANG J, et al. Attribute-based Encryption Scheme Supporting Attribute Revocation in Cloud Storage Environment[J]. Journal on Communication/Tongxin Xuebao, 2019, 40(5): 47-56.

[6] HAN D Z, PAN N N, KUAN C L. A Traceable and Revocable Ciphertext-Policy Attribute-based Encryption Scheme Based on Privacy Protection[J]. IEEE Transactions on Dependable and Secure Computing, 2020, 19(1): 316-327.

[7] WANG S, ZHANG X, ZHANG Y. Efficient Revocable and Grantable Attribute-based Encryption from Lattices with Fine-Grained Access Control [J]. IET Information Security, 2018, 12(2): 141-149.

[8] CHEN J, LIM H W, LING S, et al. Revocable Identity-based Encryption from Lattices[C] // Information Security and Privacy: 17th Australasian Conference. Springer, 2012: 390-403.

[9] WANG Y. Lattice Ciphertext Policy Attribute-based Encryption in the Standard Model[J]. Interational Journal of Network Security, 2014, 16(6): 444-451.

[10] YANG K, WU G, DONG C, et al. Attribute Based Encryption with Efficient Revocation from Lattices[J]. Interational Journal of Network Security, 2020, 22(1): 161-170.

[11] ZHAO S, JIANG R, BHARGAVA B. RL-ABE: A Revocable Lattice Attribute-based Encryption Scheme based on R-LWE Problem in Cloud Storage[J]. IEEE Transactions on Services Computing, 2020, 15(2): 1026-1035.

[12] LUO F, AL-KUWARI S, WANG H, et al. Revocable Attribute-based Encryption from Standard Lattices [J]. Computer Standards & Interfaces, 2023, 84: 103698.

[13] HUANG B, GAO J, LI X. Efficient Lattice-based revocable attribute-based Encryption Against Decryption Key Exposure for Cloud File Sharing [J]. Journal of Cloud Computing, 2023, 12(1): 37.

[14] GUO L, WANG L, MA X, et al. New Traceable and Revocable Attribute Based Encryption on Lattices[C] // 2023 International Conference on Networking and Network Applications (NaNA). IEEE, 2023: 359-364.

[15] GREE N, MATTHEW D, IAN M. Forward Secure Asynchronous Messaging from Puncturable Encryption [C] // 2015 IEEE Symposium on Security and Privacy. IEEE, 2015: 305-320.

[16] PHUONG T V X, NING R, XIN C, et al. Puncturable Attribute-based Encryption for Secure Data Delivery in Internet of Things [C] // IEEE INFOCOM 2018 — IEEE Conference on Computer Communications. IEEE, 2018: 1511-1519.

[17] DUTTA P, SUSILO W, DUONG D H, et al. Puncturable Identity-based and Attribute-based Encryption from Lattices[J]. Theoretical Computer Science, 2022, 929(11): 18-38.

[18] YANG M, WANG H, HE D. Puncturable Attribute-based Encryption from Lattices for Classified Document Sharing [J]. IEEE Transactions on Information Forensics and Security, 2024, 929(11): 4028-4042.

GUO Lifeng, born in 1975, Ph.D, professor, postgraduate supervisor, is a member of CCF (No. Q22710M). Her main research interests include privacy protection technologies for digital encryption, signature and blockchain.

（责任编辑：何杨）