面向医疗数据共享的可追踪可撤销注册属性加密方案

田有亮[1,2]，易笃兴[1]，向阿新[1,2]，杨科迪[1,2]，路振华[1]
（1.贵州大学计算机科学与技术学院贵州省大数据实验室，贵州 贵阳 550025；2.贵州大学大数据与信息工程学院，贵州 贵阳 550025）

摘要：医疗数据共享中，内部用户私钥泄露、转让或滥用可能导致敏感数据非法访问，因此需要对泄露私钥进行责任追踪并及时撤销相关权限。传统可追踪属性基加密通常由中心机构生成完整用户私钥并嵌入追踪信息，而注册属性加密中用户自主生成私钥核心分量，密钥管理者不掌握完整私钥，导致追踪信息难以直接嵌入。针对该问题，本文提出一种可追踪可撤销注册属性加密方案TRRABE。该方案在用户自生成私钥分量的基础上构造身份关联追踪组件，使密钥管理者无需掌握完整私钥即可通过泄露私钥定位责任用户；结合密钥更新与密文更新机制实现追踪后撤销，取消泄露私钥用户的访问权限；并通过用户身份哈希处理降低撤销过程中的身份关联风险。安全性分析表明，TRRABE满足选择明文攻击安全性、私钥泄露可追踪性以及撤销机制安全性；性能评估表明，该方案具有可接受的计算与存储开销。

关键词：属性基加密；注册属性加密；私钥追踪；用户撤销；身份保护
中图分类号：TP309
文献标志码：A

A Traceable and Revocable Registered Attribute-Based Encryption Scheme for Medical Data Sharing

TIAN Youliang[1,2], YI Duxing[1], Xiang Axin[1,2], YANG Kedi[1,2], LU Zhenhua[1]
1. Guizhou Provincial Laboratory of Big Data, College of Computer Science and Technology, Guizhou University, Guiyang, Guizhou 550025, China
2. College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China

Abstract: In medical data sharing, the leakage, transfer, or misuse of internal users' private keys may lead to illegal access to sensitive data. Therefore, it is necessary to trace the responsibility for leaked private keys and revoke the relevant permissions in a timely manner. Traditional traceable attribute-based encryption usually generates complete user private keys by a central authority and embeds tracing information into them. However, in registered attribute-based encryption, users independently generate the core components of their private keys, and the key manager does not possess complete private keys, making it difficult to directly embed tracing information. To address this problem, this paper proposes a traceable and revocable registered attribute-based encryption scheme, TRRABE. Based on the user-generated private key components, the scheme constructs identity-associated tracing components, enabling the key manager to locate the responsible user through the leaked private key without possessing the complete private key. It combines key update and ciphertext update mechanisms to realize post-tracing revocation and revoke the access permission of the user whose pri-

收稿日期：XXXX–XX–XX；修回日期：XXXX–XX–XX
通信作者：易笃兴，2961657293@qq.com
基金项目：国家重点研发计划项目(No.2025YFB3109800)；国家自然科学基金项目（62272123）；贵州省科技创新平台科研项目（黔科合平台[2025] 024）；贵州省高层次创新型人才项目（黔科合平台人才[2020]6008）；贵州省基础研究计划（自然科学）面上项目（黔科合基础MS〔2026〕175）。
Foundation Items：The National Key Research and Development Program of China(2025YFB3109800), National Natural Science Foundation of China under Grant(62272123); Scientific and Technological Innovation Platform Research Project of Guizhou Province [2025] 024; Project of High-level Innovative Talents of Guizhou Province under Grant [2020]6008; Guizhou Provincial Basic Research Program (Natural Science) General Project(MS〔2026〕175).

vate key has been leaked. It also reduces the identity association risk during the revocation process through user identity hashing. Security analysis demonstrates that TRRABE is secure against chosen-plaintext attacks and provides private-key leakage traceability and revocation security. Performance evaluation shows that the proposed scheme incurs acceptable computational and storage overheads.

Key words: attribute-based encryption, registered attribute-based encryption, private key tracing, user revocation, identity protection

0 引言

随着智慧医疗、远程诊疗和医疗信息化的快速发展，电子健康记录逐渐由传统本地存储模式转向云端集中存储与跨机构共享模式。医疗数据共享能够缓解医疗机构之间的“信息孤岛”问题，提高诊疗协同效率和医疗数据利用价值。然而，医疗数据通常包含患者身份、疾病信息、诊疗记录和用药情况等敏感内容，具有较强的隐私性。在开放云环境下，数据拥有者不仅需要按照医生、护士、科研人员等不同角色或属性对医疗数据进行细粒度访问控制，还需要保证非授权用户无法获取数据内容，并降低共享过程中患者身份和用户访问行为泄露的风险。因此，如何在保证数据机密性的同时实现多用户、多属性和动态权限变化的受控共享，成为医疗数据安全共享中的热点问题。

属性基加密（attribute-based encryption, ABE）凭借“一次加密，按策略解密”的访问特性，适用于云环境下一对多的细粒度数据共享场景[1]。然而，传统ABE通常由可信权威机构生成用户属性私钥，存在密钥托管风险；多机构ABE虽通过多个属性机构分散密钥生成权限，缓解了单点故障问题，但用户属性私钥仍需由外部机构参与生成，难以从根本上消除密钥托管风险。Garg等[2]提出注册加密思想，允许用户本地生成公私钥对并提交公钥完成注册。Hohenberger等[3]进一步将该思想扩展到属性基加密场景，提出注册属性加密（registered attribute-based encryption, RABE）框架，使用户自主生成私钥并向密钥管理者提交属性相关公钥信息，从根本上解决密钥托管问题。然而，现有RABE方案多是聚焦于性能优化和表达能力，面对高敏感医疗数据场景，仍有以下问题需要解决。

首先是私钥泄露问题，在医疗数据共享过程中，用户私钥可能因存储不当而泄露，也可能被恶意用户主动转让或滥用。若缺乏有效的追踪机制，一旦用户私钥泄露，难以及时识别责任用户，从而威胁数据安全与用户隐私。现有可追踪ABE研究主要分为白盒追踪[4-7]和黑盒追踪[8-10]：白盒追踪通过分析泄露私钥中的追踪组件定位责任用户；黑盒追踪通过与非法解密设备进行交互，并依据其解密行为推断密钥泄露者身份。Xue等[11]提出一种可追踪可撤销的完全隐藏策略的ABE方案，构造解密权限二叉树，实现完全策略隐藏下的白盒追踪；He等[12]提出一种增强型可追踪ABE方案，通过融合白盒追踪与黑盒追踪机制，提升了私钥泄露场景下责任用户定位的适用性与追踪能力。Chen等[13]面向车联网云雾计算环境，设计一种融合多属性机构协同管理、用户追踪与属性撤销的多机构ABE方案，以支持动态网络环境下的细粒度访问控制和权限更新。上述研究从不同角度提升了ABE的私钥泄露追踪能力，但均基于传统ABE框架，追踪信息依赖中心机构在生成用户私钥时嵌入，然而RABE核心是用户自主生成私钥，因此传统追踪信息嵌入方式难以直接适用。李朋祥等[14]提出支持属性验证与用户撤销的IoMT数据共享方案，结合槽位组件与用户辅助解密密钥之间的关联关系，设计了白盒追踪机制；但是其主要针对原始格式密钥，对经过变形、组合或重随机化处理后的泄露密钥追踪能力有限。因此，如何在RABE框架下增强对泄露私钥的责任追踪能力，尤其是在泄露密钥形式可能发生变化的情况下仍能定位责任用户，仍是需要解决的问题。

在追踪责任用户之后，系统还需要通过撤销机制取消泄露私钥的解密能力。若责任用户被追踪后未能及时撤销，则系统难以满足前向安全性和后向安全性。可撤销ABE研究主要通过直接撤销、间接撤销、密钥更新、密文更新或撤销列表等[15-21]方式实现用户权限动态管理。近年来，Peñuelas-Angulo等[22]提出一种面向雾计算物联网的可撤销多机构ABE方案，通过引入基于广播加密的用户撤销组件，避免撤销过程中频繁更新用户密钥；Huang等[23]提出一种支持大属性域和用户属性撤销的多

机构 ABE 方案，实现了用户属性动态撤销及前向安全性和后向安全性。上述方案主要面向多机构 ABE 框架下的用户撤销或属性撤销，密钥生成与撤销过程仍依赖属性机构参与，难以适应 RABE 中用户自主生成私钥核心分量的场景；Li 等[24]提出支持用户注销的可撤销注册属性加密方案，在注册属性加密框架下实现了精确用户撤销和用户注销。然而，该方案侧重于常规用户撤销与注销，缺乏对私钥泄露用户责任追踪及追踪后用户撤销机制；并且，若攻击者与密钥管理者合谋，被注销用户仍可能获得后续辅助信息更新，影响系统的后向安全性。因此，如何在识别责任用户后对其进行撤销，是 RABE 在私钥泄露场景需要解决的问题。

此外，追踪后撤销还可能带来撤销过程中身份暴露风险。与常规权限撤销不同，追踪后撤销以私钥泄露追踪结果为依据；若撤销列表、公共更新信息或密文更新参数中暴露用户身份，攻击者可能关联其角色属性、访问权限和数据访问行为，从而泄露用户隐私。匿名 ABE 通过隐藏用户身份降低访问控制过程中的隐私泄露风险[25-29]。近年来，Deshmukh 等[30]研究云计算环境下具有隐私保护特性的属性基加密机制，通过多属性机构和不经意传输方法降低密钥生成阶段的用户属性泄露风险，但未考虑撤销机制与用户身份之间的关联；Yin 等[31]提出面向医疗数据共享的可追踪动态撤销属性加密方案，利用成员验证机制在不直接暴露用户身份的情况下判断用户是否被撤销，降低公开撤销列表带来的身份泄露风险，但该方案难以直接适用于无密钥托管的注册型访问控制场景；Liu 等[32]提出一种支持问责的隐私增强多机构属性基认证方案，在密钥生成和认证阶段保护用户属性值、全局身份标识和访问行为，但难以直接解决撤销更新信息与责任用户真实身份或权限变化之间的关联风险。上述研究主要关注密钥生成、访问控制、身份认证或撤销列表验证等阶段的隐私保护，仍未解决追踪后撤销场景下撤销更新信息可能引发的真实身份暴露问题。因此，如何在 RABE 中同时实现责任追踪、追踪后撤销与撤销阶段身份保护，是待解决问题。

针对上述问题，本文在注册属性加密框架下引入白盒追踪机制，拓展并改进文献[24]的用户撤销和用户注销机制，提出一种面向医疗数据共享的可追踪可撤销的注册属性加密方案，实现私钥泄露责任追踪、追踪后撤销并降低撤销过程的身份暴露风险。本文主要贡献如下。

1）设计一种适用于注册属性加密的白盒追踪机制，在辅助私钥生成过程中构造与用户身份相关联的追踪组件，建立与用户身份的对应关系，使系统能够依据泄露私钥中的追踪组件匹配责任用户。

2）设计一种追踪后撤销与用户注销机制，将白盒追踪结果引入撤销过程，并结合公共更新信息和密文更新，使泄露私钥用户无法解密更新密文；同时，通过对用户身份标识进行哈希处理，降低撤销阶段真实身份的暴露风险。

3）提出一种面向医疗数据共享的可追踪可撤销注册属性加密方案。在注册属性加密框架下引入追踪组件和撤销更新机制，在保持无密钥托管特性的同时，实现医疗数据细粒度共享中私钥泄露责任追踪、追踪后撤销和撤销过程身份隐私保护。

4）安全性与性能分析表明，本文方案在满足选择明文攻击安全性和私钥泄露可追踪性的同时，能够在较低计算开销下追踪责任用户，并通过密钥更新与密文更新实现高效的追踪后撤销，适用于医疗数据共享场景下的可追踪访问控制与动态权限管理需求。

1　预备知识

1.1　复合阶双线性对群

设 $(p_1,p_2,p_3,\mathbb{G},\mathbb{G}_T,e,g)$ 为复合阶双线性群系统，其中 p_1,p_2,p_3 为三个不同素数。$\mathbb{G}$ 和 $\mathbb{G}_T$ 为乘法循环群，群阶为 $N=p_1p_2p_3$，g 为群 $\mathbb{G}$ 的生成元。若映射 $e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$ 满足以下性质，则称为双线性映射。

1）双线性：对任意 $u\in\mathbb{G}$ 以及 $x,y\in\mathbb{Z}_p^*$，满足 $e(u^x,u^y)=e(u,u)^{xy}$；

2）非退化性：当 $u\neq 1_{\mathbb{G}}$、$v\neq 1_{\mathbb{G}}$ 时，有 $e(u,v)\neq 1$；

3）可计算性：对任意 $u,v\in\mathbb{G}$，均存在高效算法能够计算 $e(u,v)$。

1.2　访问结构

当且仅当属性集合满足预设条件时，才可被授予访问权限。严格定义如下：

用符号 U 表示属性全集，访问结构 $\mathbb{A}$ 由属性全集 U 的所有非空子集构成，即 $\mathbb{A}\subseteq 2^U\setminus\{\varnothing\}$。对任意 $B,C\subseteq U$，若满足 $B\subseteq C$ 且 $B\in\mathbb{A}$，可推出

$C \in \mathbb{A}$，则称该访问结构是单调的。

1.3　线性秘密共享方案

设 P 为参与方集合。定义在 $\mathbb{Z}_N$ 上、针对参与方集合 P 的线性秘密共享方案表示为 (M,ρ)，其中 $M \in \mathbb{Z}_N^{l \times n}$ 为共享矩阵，映射 $\rho:[l] \to P$ 为行映射函数。(M,ρ) 需满足以下两个条件：

1）秘密份额生成：设待共享的秘密值为 $s \in \mathbb{Z}_N$，从 $\mathbb{Z}_N$ 中随机选取 $v_2,...,v_n$，构造向量 $v = [s,v_2,...,v_n]^T$。计算秘密份额向量 $u = Mv$，其中分量 $u_i \in \mathbb{Z}_N$ 分配给参与方 $\rho(i)$ 其中 $i \in [l]$。

2）秘密重构：设 $S \subseteq P$ 为参与方集合，定义下标集合 $I = \{i | i \in \{1,...,k\}, \rho(i) \in S\}$ 为矩阵与集合 S 相对应的行下标集。令向量 $e_1 = (1,0,...,0)$。若可对矩阵 M 的行做线性组合，使其结果等于 e_1，则称集合 S 满足访问策略 (M,ρ)。若 S 满足访问策略 (M,ρ)，则必然存在一组系数 $\{\theta_i\}_{i \in I}$，使得

$$\sum_{i \in I} \theta_i M_i = e_1$$

其中 M_i 表示矩阵 M 的第 i 行。

1.4　离散对数问题

离散对数问题（discrete logarithm problem，DLP）是密码学中常用的计算困难问题之一。设 $\mathbb{G}$ 为素数阶 p 的循环群，g 为群 $\mathbb{G}$ 的生成元。对于随机选择的 $x \in \mathbb{Z}_p$，给定 (g,g^x)，计算指数 x 被称为群 $\mathbb{G}$ 上的离散对数问题。

2　系统模型

2.1　系统模型

系统由 6 类实体组成：密钥管理者（key curator, KC）、公共参考串生成中心、身份管理者（Identity Authority, IA）数据拥有者（data owners, DO）、数据用户（data users, DU）和云服务器（cloud server, CS）。本文方案的系统架构如图 1 所示。

1）CRS 生成中心：可信第三方机构，根据安全参数为方案生成公共参考串（common reference string, CRS），为用户注册、数据加密、密文解密、私钥追踪和用户撤销等提供公共参数支持。

2）KC：由医疗机构或医疗数据共享平台中的密钥管理部门承担，负责用户注册、注销、辅助私钥生成、聚合公钥维护以及追踪后撤销等操作。

3）DO：通常为医院、科室或医疗数据管理方，制定访问策略，决定哪些用户有权访问其数据；并按照访问策略对数据加密后上传至 CS。

4）CS：存储数据拥有者 DO 上传的密文，同时执行用户权限撤销操作。

5）DU：包括医生、护士、医学研究人员等授权用户，可使用自身私钥解密密文，恢复得到明文。

6）IA：负责验证用户注册资格，并为合法用户生成和分发身份标识 ID_i。

系统运行过程中，CRS 生成中心首先生成公共参考串，为后续算法提供公共参数支持。身份管理者 IA 为合法数据用户分发身份信息；数据用户 DU（如医生、护士、患者、医学研究人员等）自主生成私钥核心分量，并向密钥管理者 KC 提交注册信息；KC 根据 DU 的医疗角色属性生成辅助私钥并维护聚合公钥，从而完成用户注册。数据拥有者 DO（如医院、科室或医疗数据管理方）根据访问策略对电子健康记录进行加密，并将生成的密文上传至云服务器 CS。授权 DU 从 CS 获取密文后，结合自身私钥和辅助私钥执行解密，获得其权限范围内的医疗数据。当发生私钥泄露时，KC 依据泄露私钥中的追踪组件定位责任用户，并通过密钥更新、密文更新和用户注销等操作撤销其解密权限。

2.2　算法描述

面向医疗数据共享的可追踪可撤销注册属性加密方案 TRRABE 由多个槽式子方案 slot-TRRABE 聚合而成。接下来分两部分介绍算法。

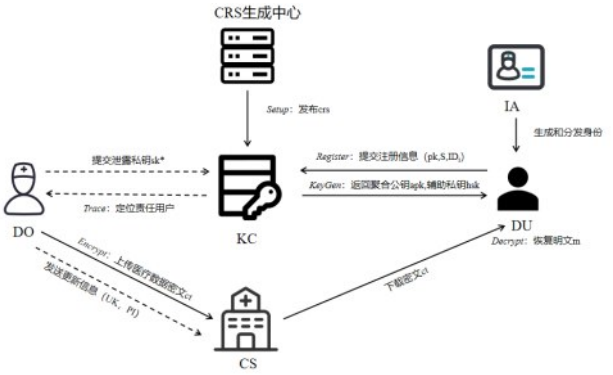

图 1　系统架构

2.2.1　slot-TRRABE

1) $Setup(1^\lambda,1^{|U|},1^L) \to crs$：公共参考串生成中心输入安全参数 λ、属性全集最大规模 $|U|$ 以及可注册用户最大数量 L，生成系统公共参考串 crs。

2) $KeyGen(crs,i,ID_i)\rightarrow(pk_i,sk_i,ID_i')$：系统中第$i$个数据用户DU输入公共参考串$crs$、用户序号$i$与用户身份$ID_i$，本地自主生成自身公私钥对$(sk_i,pk_i)$与身份标识$ID_i'$。

3) $PeckCheck(crs,pk_i,ID_i')\rightarrow True/False$：所有用户均可使用公共参考串$crs$验证用户$i$的公钥$pk_i$以及身份标识$ID_i'$有效性。

4) $Encrypt(apk,\mathbb{A},m)\rightarrow ct$：DO使用聚合公钥$apk$与访问结构$\mathbb{A}$对消息$m$加密，生成密文$ct$。

5) $Decrypt(ct,sk_i,hsk_i)\rightarrow m$：DU输入密文$ct$、本地私钥$sk_i$与辅助私钥$hsk_i$；若解密成功则输出消息$m$，解密失败则输出无效符号$\perp$。

6) $Trace(crs,sk^*,pk_i,ID_i')\rightarrow ID_i'$：KC将公共参考串$crs$，泄露私钥$sk^*$，该私钥的用户身份标识为输入。最后输出泄露私钥的用户身份标识ID_i'。

7) $UpdateKeyGen(crs,x,ct)\rightarrow(UK,PI)$：DO输入公共参考串$crs$、待撤销用户序号$x$及目标密文$ct$，输出公共信息集合$(UK,PI)$。

8) $RevokeCS(ct,UK)\rightarrow ct'$：CS输入原始密文$ct$与更新密钥$UK$，输出更新密文$ct'$。

9) $Deregister(hsk,crs,apk,x)\rightarrow apk'$：KC输入公共参考串$crs$、辅助密钥$hsk$、原始聚合公钥$apk$及待注销用户序号$x$，输出新的聚合公钥$apk'$。

2.2.2 TRRABE

由于TRRABE方案通过组合多个不同规模的slot-TRRABE子方案得到，因此算法描述相似。

1) $Setup(1^\lambda,1^{|U|},1^L)\rightarrow crs$：公共参考串生成中心输入安全参数$\lambda$、属性全集最大规模$|U|$以及可注册用户最大数量$L$，生成聚合系统公共参考串$crs$。

2) $KeyGen(crs,i,ID_i)\rightarrow(pk_i,sk_i,ID_i')$：$KeyGen$算法与子方案设置相同，但输出为聚合后公私钥以及身份信息。

3) $Locate(ver,i)\rightarrow(k,i_k)$：KC在注册阶段执行，输入当前版本号和用户序号，输出用户槽位。

4) $Register(crs,aux,pk_i,S_i,i,ID_i')\rightarrow apk$：KC执行注册算法，用户将公共参考串$crs$、辅助信息$aux$、用户公钥$pk_i$、属性集合$S_i$、用户序号$i$以及身份标识$ID_i'$发送给KC；KC更新本地辅助信息$aux$，并输出聚合公钥$apk$。

5) $Encrypt(apk,\mathbb{A},m)\rightarrow ct$：DO使用聚合公钥$apk$与访问结构$\mathbb{A}$对消息$m$加密，生成密文$ct$。

6) $Decrypt(ct,sk_i,hsk_i)\rightarrow m$：DU输入密文$ct$、本地私钥$sk_i$与辅助私钥$hsk_i$；若解密成功则输出消息$m$，解密失败则输出无效符号$\perp$。

7) $Trace(crs,sk^*,Dic[k,i])\rightarrow ID_i'$：KC输入公共参考串$crs$，泄露私钥$sk^*$，以及该私钥的用户数据字典。最后输出泄露私钥的用户身份ID_i'。

8) $UpdateKeyGen(crs,x,ct)\rightarrow(UK,PI)$：DO输入公共参考串$crs$、待撤销用户序号$x$及目标密文$ct$，输出公共信息集合$(UK,PI)$。

9) $RevokeCS(ct,UK)\rightarrow ct'$：CS输入原始密文$ct$与更新密钥$UK$，输出更新密文$ct'$。

10) $Deregister(aux,crs,apk,x)\rightarrow apk'$：KC输入公共参考串$crs$、辅助信息$aux$、原始聚合公钥$apk$及待注销用户序号$x$，更新本地辅助信息$aux$，并输出新的聚合公钥$apk'$。

2.3 安全模型

2.3.1 IND-CPA安全模型

本文通过挑战者$\mathcal{C}$和敌手$\mathcal{A}$之间的交互式游戏来描述本文提出的方案在选择明文攻击下的安全性。敌手和挑战者之间的游戏描述如下所示：

系统建立：敌手$\mathcal{A}$首先给出系统最大注册用户数量上限$L=2^l$。挑战者$\mathcal{C}$输入安全参数λ、属性全集U和最大注册用户数L，运行系统初始化算法$Setup$生成公共参考串crs，并将crs发送给敌手$\mathcal{A}$。随后初始化系统辅助信息aux、聚合公钥apk、用户索引计数器以及相关数据字典。

阶段1：敌手$\mathcal{A}$可以自适应地发起腐化密钥查询、诚实密钥查询、腐化诚实密钥查询、注销密钥查询和撤销查询。挑战者根据相应算法返回公钥、辅助密钥、更新后的聚合公钥或撤销后的密文信息。对于诚实用户，挑战者保留其底层私钥；对于腐化用户，敌手可以获得其私钥。

挑战：敌手$\mathcal{A}$选择两个等长消息m_0,m_1和一个挑战访问结构$\mathbb{A}^*$发送给挑战者。但必须满足以下限制：

1）若敌手已经对某用户发起腐化密钥查询或腐化诚实密钥查询，则该用户的属性集合不能满足挑战访问结构$\mathbb{A}^*$。

2）若敌手曾对属性集合满足$\mathbb{A}^*$的用户发起诚实密钥查询，但未获得其私钥，则该用户必须在挑战密文中被撤销。

挑战者根据当前聚合公钥生成挑战密文并返回给敌手。为配合安全性证明，挑战密文可表示为一系列混合游戏$GameH_i$：在相邻游戏中，挑战者逐个改变某一槽式子方案密文所加密的消息，从而将加密m_0的挑战密文逐步转换为加密m_1的挑战密文。若敌手能够区分相邻两个混合游戏，则可归约为攻破底层槽式属性基加密方案的CPA安全性。

阶段2：敌手继续进行与阶段1相同的查询。

猜测：敌手$\mathcal{A}$输出对挑战密文的猜测$b' \in \{0,1\}$。

定义1 如果任意多项式时间敌手$\mathcal{A}$无法以不可忽视的优势攻破上述挑战，则称本文聚合方案满足IND-CPA安全性。其中$\mathcal{A}$的优势定义为$Adv = \left| Pr[b' = b] - \frac{1}{2} \right|$。

2.3.2 可追踪性模型

本文同样通过挑战者$\mathcal{C}$和敌手$\mathcal{A}$之间的交互式游戏来描述可追踪性定义。具体定义如下。

初始化：挑战者运行*Setup*算法生成公共参考串，并初始化用户注册字典、身份信息和追踪信息。

查询阶段：敌手可以自适应地发起用户密钥查询、注册查询和腐化查询。挑战者根据相应算法返回用户公钥、身份标识、辅助信息或私钥，并记录用户身份与追踪组件之间的绑定关系。

伪造阶段：敌手输出一个泄露私钥sk^*及其对应的身份标识，若该私钥能够满足验证方程$e(g_1^{sk^*}, u^{ID_i'}h') = e(g_1, T_i)$，则说明敌手输出的私钥$sk^*$具有合法密钥结构，避免敌手通过输出随机值、残缺密钥或任意篡改后的无效密钥使追踪算法无法识别，最后输出对应用户身份ID_i。若敌手输出一个泄露私钥sk^*，使得$Trace(crs, sk^*, Dic) = \perp$或$Trace(crs, sk^*, Dic) \neq ID_i'$。则认为敌手赢得安全游戏。

定义2 如果任何多项式时间的敌手在上述攻击游戏中的优势是可忽略的，本文的方案是可追踪的。其中优势为$Adv_A^{Trace}(\lambda) \leqslant qAdv_B^{DLP}(\lambda) + negl(\lambda)$。

2.3.3 撤销机制安全模型

本文用时间段t表示系统变化，通过挑战者$\mathcal{C}$和敌手$\mathcal{A}$之间的交互式游戏来描述本文的撤销安全性。具体定义如下。

1）前向安全性

初始化：挑战者$\mathcal{C}$运行*Setup*算法生成公共参考串，并初始化用户注册字典、身份信息和撤销状态。最后将公共参考串发送给敌手$\mathcal{A}$。

查询阶段1：敌手可以自适应地发起用户注册查询、诚实密钥查询、腐化密钥查询、加密查询和撤销查询。挑战者执行相应算法并返回查询结果。敌手可以获得目标用户DU_x在撤销前的完整私钥sk_x，但不能腐化执行挑战密文更新的云服务器CS。

挑战：敌手$\mathcal{A}$选择两个等长消息m_0, m_1和满足访问结构的待撤销用户DU_x发送给挑战者。挑战者随机选择比特$b \in \{0,1\}$，使用当前聚合公钥对apk_t加密得到密文，随后针对DU_x执行撤销更新算法，生成更新后的挑战密文。随后将更新后的挑战密文发送给敌手，公共更新信息中不包含DU_x对应的撤销支持分量，并且敌手不能获得该挑战密文更新前的可解密形式。云服务器获得执行更新所需的*UK*，但按照非合谋假设不向敌手泄露*UK*。

查询阶段2：敌手继续发起与阶段1相同的查询，且不能获得能够直接解密挑战密文的其他未撤销用户完整私钥，也不能发起对挑战密文对应*UK*或腐化云服务器CS的查询。

猜测：敌手输出b'。若$b = b'$，则认为敌手赢得前向安全游戏。

定义3： 如果任意多项式时间敌手□无法以不可忽视的优势攻破上述挑战，则称本文TRRABE方案满足前向安全性。其中敌手$\mathcal{A}$优势定义为$Adv^{FS}(\lambda) = \left| \Pr[b = b'] - \frac{1}{2} \right|$。

2）后向安全性

初始化：挑战者$\mathcal{C}$运行*Setup*算法生成公共参考串和阶段t的聚合公钥apk_t。最后将公共参考串发送给敌手$\mathcal{A}$。

查询阶段1：敌手可以自适应地发起用户注册查询、诚实密钥查询、腐化密钥查询和加密查询。目标用户DU_j在阶段t尚未注册。

挑战：敌手$\mathcal{A}$选择两个等长消息m_0, m_1和一个挑战访问结构$\mathbb{A}^*$发送给挑战者。挑战者随机选择比特$b \in \{0,1\}$，并生成利用阶段t生成历史挑战密文。随后系统进入$t+1$阶段目标用户DU_j完成注册并获得完整私钥。挑战者将历史挑战密文和DU_j注册后的密钥发送给敌手$\mathcal{A}$。

查询阶段2：敌手可以继续发起允许的查询，但不能获得在阶段t已注册、属性集合满足挑战访问结构且能够直接解密挑战密文的用户完整私钥。

猜测：敌手输出b'。若$b=b'$，则认为敌手赢得后向安全游戏。

定义4：如果任意多项式时间敌手$\mathcal{A}$无法以不可忽视的优势攻破上述挑战，则称本文TRRABE方案满足后向安全性。其中敌手$\mathcal{A}$优势定义为$Adv^{BS}(\lambda)=\left|\Pr[b=b']-\frac{1}{2}\right|$。如果一个方案同时满足前向和后向安全性，称方案满足撤销机制安全性。

3　具体方案

3.1　槽式可追踪可撤销注册属性加密子方案

在构造完整的面向医疗数据共享的可追踪可撤销注册属性加密方案TRRABE方案之前，本文首先给出其基础构件，即槽式可追踪可撤销注册属性加密子方案slot-TRRABE。该子方案面向固定数量的用户槽位进行构造，能够支持用户注册、聚合公钥生成、数据加密、用户解密、私钥泄露追踪以及追踪后的密文更新操作。完整TRRABE方案将在3.2节中通过组合多个不同规模的slot-TRRABE子方案得到。

$Setup(1^{\lambda},1^{|U|},1^{L})$:$G=(p_1,p_2,p_3,\mathbb{G},\mathbb{G}_T,e,g)$是由安全参数$\lambda$生成的复合阶双线性群，其中$p_1,p_2,p_3$为三个不同的素数。群$\mathbb{G},\mathbb{G}_T$均为乘法循环群，阶为$N=p_1p_2p_3$；$e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$是双线性映射，$g$为$\mathbb{G}$的生成元。对于槽位$i$，$\mathbb{G}_i$为$\mathbb{G}$中阶为$p_i$的子群，$g_i$为子群$\mathbb{G}_i$的生成元。CRS生成中心随机选取$\alpha,\beta\in\mathbb{Z}_N$，计算$h=g_1^{\beta}$，随机选取$u,h'\in\mathbb{G}_{p_1}$对于方案中含$L$个槽位的每个槽位$i$，$CRS$生成中心随机选取$t_i,\delta_i,\tau_i\in\mathbb{Z}_N$，计算$a_i=(g_1g_3)^{t_i}$、$b_i=g_1^{\alpha}a_i^{\beta}g_3^{\tau_i}$以及$P_i=(g_1g_3)^{\delta_i}$。对于每个槽位$i$以及大型属性集合$U$中每个属性$w$，CRS生成中心随机选取$u_{i,w}\in\mathbb{Z}_N$，计算$U_{i,w}=g_1^{u_{i,w}}$；对于每个满足$j\neq i$的槽位$j$，CRS生成中心随机选取$\gamma_{i,j,w}\in\mathbb{Z}_N$，计算$W_{i,j,w}=a_i^{u_{j,w}}g_3^{\gamma_{i,j,w}}$。最后CRS生成中心选择$H_2:\{0,1\}^*\to\{0,1\}^{256}$，$H_1:\{0,1\}^*\to\mathbb{Z}_N$，计算$Z=e(g,g)^{\alpha}$，并输出公共参考串

$$crs=(N,g_1,g_3,h,Z,u,h',H_1,H_2,\{a_i,b_i,P_i\}_{i\in[L]},\{U_{i,w},W_{i,j,w}\}_{i\neq j,w\in U}) \tag{1}$$

$KeyGen(crs,i,ID_i)$：密钥生成算法由数据用户DU本地执行。算法输入公共参考串crs、用户所在槽位索引i以及用户身份ID_i。用户i随机选取$r_i\in\mathbb{Z}_N$作为私钥，即$sk_i=r_i$，选取随机值$salt\in\{0,1\}^{128}$，计算$Tag_i=H_2(ID_i')$，计算$ID_i'=H_1(ID_i\,//\,salt_i)$，计算追踪组件$T_i=(u^{ID_i'}\cdot h')^{r_i}$，并利用公共参考串$crs$计算公钥分量$E_i=g_1^{r_i}$，$F_i=P_i^{r_i}$，$Q_i=g_3^{r_i}$，对于每个$j\neq i$，用户$i$计算$V_{j,i}=a_j^{r_i}$。输出公钥

$$pk_i=(E_i,F_i,Q_i,\{V_{j,i}\}_{j\neq i},T_i,Tag_i)$$

和身份标识ID_i'。

$PeckCheck(crs,pk_i,ID_i')$：公钥合法性验证算法用于验证用户提交的公钥和身份标识是否满足注册要求。所有用户均可使用公共参考串crs验证用户i的公钥pk_i、身份标识ID_i'以及身份标签Tag_i有效性。首先验证以下三个方程

$$\begin{gathered}e(g_3,E_i)=1_{\mathbb{G}_T}=e(g_1,Q_i)\\ e(E_i,P_i)=e(g_1,F_i),e(Q_i,P_i)=e(g_3,F_i)\end{gathered} \tag{2}$$

对于所有满足$j\neq i$的索引j，进一步验证

$$e(g_1,V_{j,i})=e(E_i,a_j)$$

$$e(g_3,V_{j,i})=e(Q_i,a_j)$$

最后验证

$$Tag_i=H_2(ID_i')$$

$$e(E_i,u^{ID_i'}\cdot h')=e(g_1,T_i)$$

若用户i的公钥pk_i满足上述所有方程，算法输出$True$，否则输出$False$。

$APKGen(crs,\{(pk_i,S_i,ID_i')\}_{i\in[n]})$：聚合公钥算法由KC执行。首先对每个时隙$i$，运行$PeckCheck$算法验证公钥$pk_i$和身份标识$ID_i'$的合法性，验证失败则拒绝聚合。KC计算属性无关公钥

$$E'=\prod\nolimits_{i\in[n]}E_i \tag{3}$$

并对每个用户i，计算噪声消除参数$\xi_i=H_1(Tag_i)$，并计算辅助密钥

$$V_i'=\Big(\prod_{j\in[n],j\neq i}V_{i,j}\Big)\cdot g_3^{\xi_i}$$

对每个属性$w\in U$，每个用户i计算属性相关公钥U_w'和辅助密钥$W_{i,w}'$

$$U_w' = \prod_{i\in[n],w\notin S_i} U_{i,w}$$
$$W_{i,w}' = \prod_{j\in[n],j\neq i,w\notin S_j} W_{i,j,w} \tag{4}$$

由于聚合公钥生成阶段先于任何加密和撤销阶段执行，因此为每个用户 i 设置撤销支持信息 $RSI_i = 1_{\mathbb{G}}$。最后返回聚合公钥

$$apk = (\mathbb{G},g_1,h,Z,E',\{U_w'\}_{w\in U})$$

每个用户的辅助密钥

$$hsk_i = (apk,i,S_i,a_i,b_i,V_i',\{W_{i,w}'\}_{W\in U},Tag_i,RSI_i)$$

Encrypt$(apk,\mathbb{A},m)$：加密算法由数据所有者 DO 执行。数据所有者 DO 根据访问结构 $\mathbb{A}=(M,\rho)$，利用聚合公钥 apk 对明文 m 进行加密。其中，M 为 K 行 n 列的矩阵，ρ 为将矩阵行号映射至属性的单射函数，且 $m_k^{\top}$ 为矩阵 M 的第 K 行。DO 选取并存储随机数 $s\in\mathbb{Z}_N$，以便后续撤销，从群 $\mathbb{G}_1$ 中选取 h_1，计算 $h_2 = h/h_1$，随机选取向量 $v_2,...,v_n\in\mathbb{Z}_N$，令向量 $v=(s,v_2,...,v_n)^{\top}$。然后计算密文：

$$C = m\cdot Z^s, C_0 = g_1^s, C_2 = h_1^s\cdot E'^{-s} \tag{5}$$

对于 M 的每行 k，计算 $\lambda_k = M_k\cdot v$，然后计算密文

$$C_{1,k} = h_2^{\lambda_k}\cdot U'^{-s}_{\rho(k)} \tag{6}$$

最后输出密文

$$ct = (ver,C,C_0,\{C_{1,k}\}_{k\in[K]},C_2)$$

Decrypt(ct,sk_i,hsk_i)：数据用户 i 首先验证辅助密钥 hsk_i 中的数据集 S_i 是否满足密文 ct 中的访问结构 $\mathbb{A}=(M,\rho)$。若不满足，解密算法终止；若满足，计算满足策略的系数 ω_k，然后用户检查密文是否附带撤销公共信息 PI，若 PI 中存在 $P_i' = a_i^{r'}$，用户将其取出并令 $RSI_i = P_i'$，若 PI 中不存在 P_i'，则说明该用户 DU 已被 DO 针对该密文撤销，解密中止。若密文未被撤销，则 $RSI_i = 1_{\mathbb{G}}$。随后记集合 $I=\{k_1,k_2,...,k_{|I|}\}$，解密属性相关部分

$$D_{pol} = \prod_{1<k<|I|}(e(C_{1,k},a_i)\cdot e(C_0,W_{i,\rho(k)}'))^{\omega_k} \tag{7}$$

利用密钥 $sk_i = r_i$ 解密属性无关部分

$$Y = e(C_0,V_i'\cdot a_i^{sk_i}\cdot RSI_i) \tag{8}$$

输出明文为

$$m = C\cdot\frac{e(C_2,a_i)\cdot D_{pol}\cdot Y}{e(C_0,b_i)} \tag{9}$$

Trace(crs,sk^*,pk_i,ID_i')：当私钥 sk^* 被泄露时，KC 执行追踪算法，若等式

$$e(g_1^{sk^*},u^{ID_i'}\cdot h') = e(g_1,T_i) \tag{10}$$

成立，输出泄露者身份标识 ID_i'。

UpdateKeyGen(crs,x,ct)：DO 以需撤销用户 x 的索引为输入，生成随机数 $r'\in\mathbb{Z}_N$。随后计算标量形式的代理更新密钥 $UK = -r'(\mathrm{mod}\,N)$，同时为所有未被撤销的合法用户 $j\neq x$ 计算公共信息更新集合 $PI=\{P_j' = a_j^{r'}\}$，最后 DO 将 (UK,PI) 发送给 CS。

RevokeCS(ct,UK)：云服务器 CS 接收到更新请求后，提取待更新密文 ct 中的组件 C_0。CS 利用 DO 下发的标量 UK，计算密文更新因子 $C_{update} = (C_0)^{UK}$，随后更新密文组件 $C_2' = C_2\cdot C_{update}$。最后 CS 将公共信息 PI 附加在密文，输出更新密文

$$ct' = (ver,C,C_0,\{C_{1,k}\}_{k\in[K]},C_2')$$

Deregister(hsk,crs,apk,x)：密钥管理者 KC 以需注销用户 x 的索引为输入，选择随机数 $r_x\in\mathbb{Z}_N$。首先计算更新后属性无关公钥 $E^* = E'\cdot g_1^{r_x}$，并计算更新后辅助密钥分量当 $i\neq x$ 时 $V_i^* = V_i'\cdot a_i^{r_x}$，当 $i=x$ 时 $V_i^* = V_i'$。随后，KC 输出更新后的聚合公钥

$$apk' = (G,g_1,h,Z,E^*,\{U_w'\}_{w\in U})$$

并为每个用户输出更新后的辅助密钥

$$hsk_i' = (apk,i,S_i,a_i,b_i,V_i^*,\{W_{i,w}'\}_{W\in U},Tag_i,RSI_i) \tag{11}$$

3.2 可追踪可撤销注册属性加密方案

完整的 TRRABE 方案需要由多种不同规模的 slot-TRRABE 方案组合构建。

密钥管理者 KC 需要同时管控多组 slot-TRRABE 子方案，因此需在本地维护多类管理信息。字典 $Dic1[k,i]=(pk_{i,k},S_i,ID_i')$ 用以记录第 k 组子方案下第 i 个时隙对应的用户公钥与属性集合；字典 $Dic2[i,k]=(hsk_{i,k})$ 用以记录第 k 组子方案下第 i 名用户对应的辅助私钥；字典 $DeR[x]=k$ 用以记录已注销用户 x 所属的子方案编号。三个字典共同构成系统整体辅助信息

$$aux = (Dic1,Dic2,DeR)$$

由于本文 TRRABE 方案需要由多种不同规模的 slot-TRRABE 方案组合构建，为确定用户 i 所属的子方案，给出定位算法。

$Locate(ver,i)$：若 $i-1\geqslant ver$，则算法输出空。若 $i-1<ver$，算法从二进制最低位开始逐位对比 $i-1$ 和版本号 ver，将二者最高不同位的位置记为 k，随后计算 $i_k=(i-1 \bmod 2^k)+1$，最终输出 (k,i_k)。

$Setup(1^\lambda,1^{|U|},1^L)$: 输入属性全集 U，安全参数 λ 以及最大注册用户数 L(记为 $L=2^l$)。CRS 生成中心需为每个槽式子方案生成 $l+1$ 份公共参考串。该生成中心重复调用槽式初始化算法共 $l+1$ 次。为每个子方案生成公共参考串

$$sTRRABE.Setup(1^\lambda,1^{|U|},1^{2^k})\rightarrow crs_k,$$

并定义哈希函数

$H_1:\{0,1\}^*\rightarrow\mathbb{Z}_N, H_2:\{0,1\}^*\rightarrow\{0,1\}^{256}$。

最终输出参考串集合

$$crs=(crs_0,crs_1,...,crs_l)$$

$KeyGen(crs,i,ID_i)$：数据用户 i 需加入 $l+1$ 个槽式可撤销注册属性加密子方案。根据其在子方案中的索引 i，计算出该用户在第 k 个槽式子方案中的索引为 $i_k=(i-1 \bmod 2^k)+1$。数据用户重复调用子方案的密钥生成算法共 $l+1$ 次，每个子方案生成公私钥对和身份标识

$$sTRRABE.KeyGen(crs,i,ID_i)$$

最终，数据用户输出自己的公私钥和身份标识

$$pk_i=(pk_{i,0},pk_{i,1},...,pk_{i,l})$$

$$sk_i=(sk_{i,0},sk_{i,1},...,sk_{i,l})$$

$$ID_i'=(ID'_{i,0},ID'_{i,1},...,ID'_{i,l})$$

$Register(crs,aux,pk_i,S_i,i,ID_i')$：密钥管理者令版本号 $ver=i$,根据该用户在子方案中的注册索引 i，密钥管理者计算出该用户在 $l+1$ 个槽位子方案中的槽位索引 $i_k=(i-1 \bmod 2^k)+1$。对于每个 $k\in[0,l]$，确定用户 i 在第 k 个槽式子方案中的局部槽位索引，并验证该用户在该子方案中的公钥分量和身份标识分量是否满足若子方案的公钥验证算法满足

$$sTRRABE.PeckCheck(crs_k,pk_{i,k},ID_{i,k}')=1$$

则密钥管理者更新字典 $Dic1[k,i_k]=(pk_{i_k},S_i,ID_{i_k}')$。对于所有 $k\in[0,l]$，若槽位索引 $i_k\neq 2^k$，则令 $apk_k'=apk_k$。否则，密钥管理者执行子方案的聚合公钥生成算法，生成对应的聚合公钥 apk_k'

$$APKGen(crs_k,\{Dic1[k,1],...,Dic1[k,2^k]\})$$

然后密钥管理者对所有 $k\in[0,l]$，为每个 $j\in[i-2^k+1,i]$ 更新字典 $Dic2[j,k]=hsk_{j,k}$，并将新的聚合公钥计算为

$$apk'=(ver,apk_0',apk_1',...,apk_l')$$

用户完成注册后，历史注销用户若需重新加入系统，可作为新实体重新注册。密钥管理者遍历注销字典 DeR 所有值，对于每个注销用户 x，调用定位算法得到该用户当前所属子方案索引，以及其在激活子方案的槽索引，即 $Locate(i,x)\rightarrow(k,x_k)$。若 $k\neq DeR[x]$，则密钥管理者调用注销算法，更新第 k 个子方案的聚合公钥，执行

$$Deregister(aux,crs,apk,x)\rightarrow apk'_k$$

若 $k=DeR[x]$，则密钥管理者不执行任何处理。最终，密钥管理者输出聚合公钥

$$apk'=(ver,apk_0',apk_1',...,apk_l')$$

$Encrypt(apk,\mathbb{A},m)$：数据拥有者根据聚合公钥 $apk=(ver,apk_0,apk_1,...,apk_l)$，重复调用子方案中的加密算法共 $l+1$ 次，为每个 $k\in[0,l]$ 生成密文组件

$$sTRRABE.Encrypt(apk,\mathbb{A},m)\rightarrow ct_k$$

若 $apk_k=\perp$，则数据拥有者把 ct_k 的值置为空。最终输出完整密文 $ct=(ver,ct_0,ct_1,...,ct_l)$。

$Trace(crs,sk^*,Dic1[k,i])$：KC 将公共参考串 crs，泄露私钥 sk^*，以及疑似恶意用户数据字典为输入，KC 读取数据字典 $Dic1[k,i]=(pk_{i,k},S_i,ID_i')$ 并提取出身份标识 ID_i'，重复调用子方案中

$$sTRRABE.Trace(crs,sk^*,pk_i,ID_i')\rightarrow ID_i'$$

算法，最后输出恶意用户身份标识 ID_i'。

$Decrypt(ct,sk_i,hsk_i)$：用户 i 接收密文 ct。若 $i-1\geqslant ver$，即用户 i 索引大于加密时聚合的公钥版本，解密失败。若 $i-1<ver$，数据用户从二进制最低位开始对比 $i-1$ 和版本号 ver 的每一位，将最高不同位记为 k，该 k 为包含此数据用户的激活子方案索引。用户重复调用子方案的解密算法，执行

$$sTRRABE.Decrypt(ct_k,sk_k,hsk_{i,k})\rightarrow m_k。$$

恢复明文 m_k 并输出。

$UpdateKeyGen(crs,x,ct)$：数据拥有者将待撤销用户索引 x 作为输入。根据密文 ct 中携带的版本号信息，若 $x-1\geqslant ver$，则算法运行中止，输出空值。若 $x-1<ver$，数据拥有者先提取版本号 ver

的二进制表示中所有数字1的位置，将这些位置按从大到小顺序排列为有序集合$P=\{k_1,k_2,...,k_p\}$，再标记出$x-1$和ver的二进制表示中最高不同位k^*。

DO选择$r'\in\mathbb{Z}_N$。随后计算$UK=-r' \bmod N$，并将其发送给云服务器CS。对于集合P中所有满足$k\neq k^*$的k，DO计算$PI_k=(a_{k,1}^{r'},a_{k,2}^{r'},...,a_{k,2^k}^{r'})$，对于满足$k=k^*$的情况，计算

$$PI_{k^*}=(a_{k,1}^{r'},a_{k,2}^{r'},...,a_{k,(x-1)\bmod 2^k}^{r'},1_{\mathbb{G}_1},a_{k,((x-1)\bmod 2^k)+2}^{r'},...,a_{k,2^k}^{r'}) \tag{12}$$

最后，DO将(UK,PI_{k^*})发送给CS。

RevokeCS(ct,UK)：对于每个$k\in[0,l]$，CS重复调用子方案撤销算法生成新的密文$RevokeCS(ct,UK)\rightarrow ct'_k$。最终输出完整密文

$$ct'=(ver,ct'_0,ct'_1,...,ct'_l)$$

Deregister(aux,crs,apk,x)：KC将待注销用户的索引x作为输入，根据聚合公钥apk中的版本号，调用定位算法$Locate(i,x)\rightarrow(k,x_k)$。随后选择随机数$r_{x_k}\in\mathbb{Z}_N$。首先计算更新后属性无关公钥$E^*_k=E'_k\cdot g_1^{r_{x_k}}$，并计算更新后辅助密钥分量当$i\in[1,2^k],i\neq x_k$时$V^*_{k,i}=V'_{k,i}\cdot a_{k,i}^{r_{x_k}}$，当$i=x_k$时$V^*_{k,i}=V'_{k,i}$。随后，KC对每个$i\in[1,2^{k_1}]$，计算$Dic2[x-x_k+i,k][V]=V'_{k,i}$。并将注销字典的条目更新为$DeR[x]=k$。最后输出更新后的第$k$个子方案的聚合公钥$apk'_k$。

4 安全性分析

4.1 选择明文攻击

定理1： 若Hohenberger等人[3]的槽式属性基加密方案在子群判定假设(SD_1,SD_2,SD_3)下是CPA安全的，则本文的槽式子方案也是CPA安全的。

证明：本文槽式子方案是在Hohenberger [3]等人方案的基础上扩展得到的。与原方案相比，本文额外引入了追踪组件T_i、撤销支持信息RSI_i。因此，只需说明这些新增组件能够在安全性归约中被有效模拟，且不会影响原方案的CPA安全性。

1）T_i由模拟器用已知的μ,v和公开的ID_i'自行计算，不依赖挑战者。

2）RSI_i初始为$1_{\mathbb{G}_1}$，只在撤销时更新，模拟器可正常处理(SD_1,SD_2,SD_3)下是CPA安全的，则本文的槽式子方案也是CPA安全的。

我们定义序列$GameH_{-1},GameH_0,...,GameH_l$。

在$Game_{H_{k^*}}$中，挑战密文构造规则为敌手$\mathcal{A}$选取一个访问策略$\mathbb{A}^*$，并从群$\mathbb{G}_T$中选取两个随机消息m_0、m_1。随后将其发送给挑战者。令挑战者本地当前聚合公钥为

$$apk=(ver,apk_0,apk_1,...,apk_l),$$

挑战者对所有$k\in[0,k^*-1]$，执行$sTRRABE.Encrypt(apk_k,\mathbb{A}^*,m_0)$算法，对所有$k\in[k^*,l]$，执行

$$sTRRABE.Encrypt(apk_k,\mathbb{A}^*,m_1),$$

生成密文$ct=(ver,ct_0,ct_1,...,ct_l)$并返回给敌手$\mathcal{A}$。

引理1： 若槽式属性基加密方案是安全的，则对于所有高效对手，均存在一个可忽略概率ε，使得对任意$k^*\in[0,l-1]$，都满足

$$\left|Pr[GameH_{k^*}(A)=1]-Pr[GameH_{k^*+1}(A)=1]\right|\leqslant\varepsilon \tag{13}$$

证明：假设存在一个高效敌手$\mathcal{A}$，使得

$$\left|Pr[GameH_{k^*}(A)=1]-Pr[GameH_{k^*+1}(A)=1]\right|$$

是不可忽略的，那么我们可以构造一个模拟器$\mathcal{B}$来攻破槽式属性基加密方案。敌手$\mathcal{A}$首先选取一个值$L=2^l$，作为系统可注册用户数量的上限。

初始化：对于所有$k\in[0,l]$，模拟器运行$sTRRABE.Setup(1^\lambda,1^{|U|},1^{2^k})$算法生成$crs_k$。当$k=k^*$时，模拟器$\mathcal{B}$将$2^{k^*}$发送给挑战者，挑战者返回$crs_k$。随后，模拟器$\mathcal{B}$将公共参考串

$$crs=(crs_0,crs_1,...,crs_l)$$

发送给敌手$\mathcal{A}$。模拟器$\mathcal{B}$初始化本地基础信息，令辅助信息$aux=\perp$，聚合公钥$apk=\perp$，用户索引$i=1$，计数器$ctr=1$，用于统计敌手发起的诚实密钥查询次数；维护字典D，用于记录密钥生成查询对应的私钥。此外，模拟器$\mathcal{B}$需维护两个有序列表S_{cur}与S_{new}，用于跟踪聚合到apk_{k^*}中的公钥与属性集合。这两个列表初始化为$(\perp,\perp,\cdots,\perp)$，长度均为$2^{k^*}$。

阶段1：$\mathcal{B}$初始化字典D和列表S_{cur}与S_{new}。敌手$\mathcal{A}$可发起以下查询：

1）腐化密钥查询；敌手$\mathcal{A}$将其选定的公钥pk与属性集S发送给模拟器$\mathcal{B}$。模拟器$\mathcal{B}$执行$Register(crs,aux,pk_i,S_i,i,ID_{i'})$算法，得到新的聚合

公钥 apk' 与辅助信息 aux'，并 (apk',aux') 返回给敌手 $\mathcal{A}$。模拟器 $\mathcal{B}$ 更新本地状态，令 $apk = apk'$，$aux = aux'$，$i = i + 1$。此外，$\mathcal{B}$ 计算

$$i_{k^*} = \left(i - 1 \bmod 2^{k^*}\right) + 1,$$

并更新 $S_{new}\left[i_{k^*}\right] = (\perp, S, pk)$。若 $i_{k^*} = 2^{k^*}$，则模拟器 $\mathcal{B}$ 令 $S_{cur} = S_{new}$。

2）诚实密钥查询：模拟器 $\mathcal{B}$ 对所有 $k \in [0,l]$ 计算 $i_k = \left(i - 1 \bmod 2^k\right) + 1$。对于 $k \in [0,l]$，模拟器 $\mathcal{B}$ 自行运行 $sTRRABE.KeyGen(crs,i,ID_i)$ 算法，生成 $\left(pk_k,sk_k\right)$。当 $k = k^*$ 时，模拟器 $\mathcal{B}$ 针对 i^{k^*} 向挑战者发起密钥生成查询，挑战者返回 $\left(t,pk^{k^*}\right)$，其中 t 是挑战者记录的密钥生成查询次数。$\mathcal{B}$ 设置公钥 $pk = \left(pk_0,pk_1,...,pk_l\right)$，并在字典 D 中记录 $\mathrm{D}[t] = \left(t,sk_0,sk_1,...,sk_{k^*-1},sk_{k^*+1},...,sk_l\right)$。随后，$\mathcal{B}$ 运行 $Register(crs,aux,pk_i,S_i,i,ID_i')$ 算法得到 apk' 与 aux'，并将 $\left(apk',aux',t,pk\right)$ 返回给敌手 $\mathcal{A}$。模拟器 $\mathcal{B}$ 更新本地状态 $apk = apk'$，$aux = aux'$，$i = i + 1$，$ctr = ctr + 1$。此外，模拟器 $\mathcal{B}$ 更新

$S_{new}\left[i^{k^*}\right] = (t,S,\perp)$；

若 $i_{k^*} = 2^{k^*}$，则令 $S_{cur} = S_{new}$。

3）腐化诚实密钥查询：敌手 $\mathcal{A}$ 选取一个索引 $1 \leqslant i_{ctr} \leqslant ctr$，模拟器 $\mathcal{B}$ 遍历字典 $D[i_{ctr}]$，获取

$$(i_{ctr},sk_0,sk_1,...,sk_{k^*-1},sk_{k^*+1},...,sk_l),$$

向挑战者发起针对索引 i 的腐化查询以获取 sk^{k^*}。最终，$\mathcal{B}$ 将私钥 $sk = (sk_0,sk_1,...,sk_l)$ 返回给 $\mathcal{A}$。

4）注销密钥查询：敌手 $\mathcal{A}$ 选取一个索引 $1 \leqslant x \leqslant ver$，$\mathcal{B}$ 运行 $Deregister(aux,crs,apk,x)$，

生成 $\left(apk_k',aux'\right)$。模拟器 $\mathcal{B}$ 将 $\left(apk_k',aux'\right)$ 返回给敌手，并更新本地状态 $apk_k = apk_k'$，$aux = aux'$。

5）撤销查询：敌手 $\mathcal{A}$ 选取一个密文 ct 以及待撤销用户的索引 x，将其发送给模拟器 $\mathcal{B}$。模拟器 $\mathcal{B}$ 运行 $UpdateKeyGen(crs, x, ct)$ 算法，生成代理更新密钥 UK 与公共更新信息集合 PI，并将 (UK, PI) 返回给敌手 $\mathcal{A}$。

挑战：敌手 $\mathcal{A}$ 选取一个访问策略 $\mathbb{A}^*$，并从群 $\mathbb{G}_T$ 中随机选取两个消息 m_0^*、m_1^*，随后将它们发送给模拟器 $\mathcal{B}$，但需满足以下两个条件，若任一不满足则游戏中止：

1）腐化限制：若敌手 $\mathcal{A}$ 曾针对某用户 i 发起过腐化密钥查询或腐化诚实密钥查询，则该用户的属性集 S_i 绝对不能满足挑战访问策略。

2）撤销限制：若敌手 $\mathcal{A}$ 曾利用属性集合 S（S 满足挑战访问策略）发起过“诚实密钥查询”获取了用户的公钥与辅助密钥，但未获取底层私钥 r_i，则敌手 $\mathcal{A}$ 必须针对该用户发起关于挑战密文的撤销查询。

两个条件满足后，令当前聚合公钥为

$$apk = (i - 1,apk_0,apk_1,...,apk_l)。$$

对于任意 $k \in [0,l]$，若 $apk_k = \perp$，则模拟器 $\mathcal{B}$ 令 $ct_k^* = \perp$；若 $apk_k \neq \perp$ 且 $k < k^*$，则 $\mathcal{B}$ 运行

$$sTRRABE.Encrypt(apk_k,\mathbb{A}^*,m_0^*) \rightarrow ct_k^*$$

算法得到 ct_k^*；若 $apk_k \neq \perp$ 且 $k > k^*$，则 $\mathcal{B}$ 运行 $sTRRABE.Encrypt(apk_k,\mathbb{A}^*,m_1^*)$ 算法得到 ct_k^*。当 $k = k^*$ 时，模拟器 $\mathcal{B}$ 向挑战者发起关于 $(\mathbb{A}^*,m_0^*,m_1^*,S_{cur})$ 的挑战查询，挑战者将 $ct_{k^*}^*$ 返回给 $\mathcal{B}$。最终，$\mathcal{B}$ 输出挑战密文 $ct^* = (i - 1,ct_0^*,ct_1^*,...,ct_l^*)$ 发送给敌手 $\mathcal{A}$。

阶段 2：重复阶段 1。

猜测：敌手 $\mathcal{A}$ 输出一个比特 $b' \in \{0,1\}$，模拟器 $\mathcal{B}$ 同样输出 b'。模拟器 $\mathcal{B}$ 能够对敌手 $\mathcal{A}$ 完美模拟混合游戏。

模拟器 $\mathcal{B}$ 能够完美地向敌手 $\mathcal{A}$ 模拟混合游戏。$ct_{k^*}^*$ 为查询响应结果，

$$sTRRABE.Encrypt(apk_k,\mathbb{A}^*,m_0^*) \rightarrow ct_{k^*}^*,$$

则 $\mathcal{B}$ 模拟游戏 $GameH_{k^*}$；若

$$sTRRABE.Encrypt(apk_k,\mathbb{A}^*,m_1^*) \rightarrow ct_{k^*}^*,$$

则 $\mathcal{B}$ 模拟游戏 $GameH_{k^*+1}$。

若敌手 $\mathcal{A}$ 能以不可忽略的概率区分 $GameH_{k^*}$ 与 $GameH_{k^*+1}$，则 $\mathcal{B}$ 能以不可忽略的概率攻破该槽式属性基加密 slot ABE 方案。引理 1 证毕。

根据引理 1，对任意 $k \in [0,l-1]$，均存在一个可忽略概率 ε，使得

$$\left|Pr\left[GameH_{k^*}(A) = 1\right] - Pr\left[GameH_{k^*+1}(A) = 1\right]\right| = \varepsilon \tag{14}$$

由此可推得，存在可忽略概率 ε'，满足

$$\left|Pr\left[GameH_{k^*}(A) = 1\right] - Pr\left[GameH_{k^*+1}(A) = 1\right]\right| = \varepsilon' \tag{15}$$

由于混合游戏 $Game_{H_0}$ 对应挑战阶段中 $b = 0$ 的

原始游戏 $Game_R$，而混合游戏 $Game_{H_l}$ 对应挑战阶段中 $b=1$ 的原始游戏 $Game_R$，因此有

$$|Pr[Game_{R|b=0}(A)=1]-Pr[Game_{R|b=1}(A)=1]|=\varepsilon' \tag{16}$$

若存在多项式时间敌手 $\mathcal{A}$ 使式（15）中的概率差为不可忽略量 ε'，则根据混合论证，必存在某个 $k\in[0,l-1]$，使得混合游戏 $GameH_{k^*}$ 与 $GameH_{k^*+1}$ 可被以不可忽略优势区分。由此可构造模拟器 $\mathcal{B}$ 破坏底层困难假设，这与假设成立相矛盾。因此敌手无法以非忽略优势区分挑战比特 b，本文方案满足选择明文攻击安全性。证毕。

4.2 可追踪性

定理2： 若DLP假设成立且哈希函数 H_1 抗碰撞，则本文提出的TRRABE方案具备可追踪性。

证明： 假设存在多项式敌手 $\mathcal{A}$，其能够以不可忽略优势 ε 赢得可追踪性游戏，即能够生成一个合法的泄露私钥分量 sk^*，使得追踪算法不能正确输出对应的泄露用户身份标识。我们可构造一个多项式时间算法 $\mathcal{B}$，以相同优势 ε 求解DLP挑战问题。

设 $\mathbb{G}_1$ 为素数阶 p_1 的双线性群，$e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$ 为双线性映射。为求解DLP问题，算法 $\mathcal{B}$ 获得实例 $(g_1,X=g_1^x)$，其中 $x\in\mathbb{Z}_{p_1}$。要求算法 $\mathcal{B}$ 算出 x。

初始化：算法 $\mathcal{B}$ 接收实例 $(g_1,X=g_1^x)$，设置 crs 为：随机选择 $\mu,v\in\mathbb{Z}_N$，设置 $u=g_1^{\mu}$，$h'=g_1^{v}$，然后算法 $\mathcal{B}$ 根据真实 $Setup$ 算法生成公共参考串 crs，并将 crs 发送给敌手 $\mathcal{A}$。算法 $\mathcal{B}$ 随机猜测敌手最终伪造所对应的目标用户索引 i^*。若敌手最终输出的伪造私钥不对应 i^*，则算法 $\mathcal{B}$ 终止。由于敌手最多进行 q 次相关查询，因此该猜测成功的概率至少为 $1/q$。

密钥查询：当敌手 $\mathcal{A}$ 对身份 $ID_i'\neq ID_{i^*}'$ 发起查询时，算法 $\mathcal{B}$ 按照正常密钥生成算法为该身份生成密钥 sk_i，并计算追踪组件 $T_i=(u^{ID_i'}h')^{sk_i}$。对于目标身份 ID_{i^*}'，算法 $\mathcal{B}$ 将 $X=g_1^x$ 嵌入公钥分量中。并计算出追踪组件 $T_{i^*}=(u^{ID_{i^*}'}h')^{x}$。算法 $\mathcal{B}$ 将生成的密钥、公钥以及身份标识返回给敌手 $\mathcal{A}$，同时记录查询过的身份及其对应的追踪信息。若敌手 $\mathcal{A}$ 要求获取目标身份 ID_{i^*}' 的私钥，算法 $\mathcal{B}$ 中止。

密钥伪造：查询结束后，敌手 $\mathcal{A}$ 输出一个伪造密钥 sk^* 以及对应身份标识 ID'。算法 $\mathcal{B}$ 首先验证

$$e(g_1^{sk^*},u^{ID'}\cdot h')=e(g_1,T^*),$$

若上式不成立，则说明敌手输出的伪造密钥 sk^* 不具有合法密钥构造，属于无效密钥，不能构成对方案可追踪性的有效攻击，$\mathcal{B}$ 中止本次游戏。若成立，则说明该伪造密钥具有合法的密钥结构，可作为有效伪造进入后续归约分析。将

$$u=g_1^{\mu},\quad h'=g_1^{v},\quad T_{i^*}=X^{\mu ID'+v}$$

带入得

$$e(g_1,g_1)^{sk^*(\mu ID'+v)}=e(g_1,g_1)^{x(\mu ID'+v)},$$

由于双线性映射非退化，且除可忽略概率外有 $\mu ID'+v\neq 0$，因此可推出 $sk^*=x$。于是，$\mathcal{B}$ 输出 sk^* 作为DLP挑战实例的解。若敌手试图构造不同身份 $(ID_i\,//\,salt_i)\neq(ID_j\,//\,salt_j)$ 产生相同身份标识 $ID_i'=ID_j'$，即构造 $H_1(ID_i\,//\,salt_i)=H_1(ID_j\,//\,salt_j)$，这与 H_1 的抗碰撞性矛盾。因此，敌手无法通过身份碰撞破坏身份与追踪组件之间的绑定关系。设敌手最多进行 q 次相关查询，则 $\mathcal{B}$ 猜中目标用户的概率至少为 $1/q$。因此有

$$Adv_B^{DLP}(\lambda)\geqslant\frac{1}{q}Adv_A^{Trace}(\lambda)-negl(\lambda),$$

即

$$Adv_A^{Trace}(\lambda)\leqslant qAdv_B^{DLP}(\lambda)+negl(\lambda)。$$

由于DLP假设成立，任意多项式时间算法求解DLP的优势 $Adv_B^{DLP}(\lambda)$ 均为可忽略函数，因此 $Adv_A^{Trace}(\lambda)$ 也是可忽略函数。综上，本文方案满足可追踪性。证毕。

4.3 撤销机制安全性

定理3： 若本文槽式子方案是IND-CPA安全的，且云服务器CS诚实执行更新并不与被撤销用户合谋，则TRRABE方案满足前向和后向安全性。

证明： 撤销机制的前向和后向安全性由敌手 $\mathcal{A}$ 与模拟器 $\mathcal{B}$ 之间的安全游戏证明。假设存在概率多项式时间敌手 $\mathcal{A}$ 能够以不可忽略的优势破坏TRRABE撤销机制的前向安全性或后向安全性，则构造模拟器 $\mathcal{B}$ 能以不可忽略的优势攻破TRRABE的IND-CPA安全性。

初始化阶段、查询阶段：$\mathcal{B}$ 按照本文完整TRRABE方案执行 $Setup$ 算法，初始化子方案，得到 crs、apk 以及辅助信息 aux。$\mathcal{B}$ 按照真实算法回答 $\mathcal{A}$ 发起的查询。对于非挑战密文，$\mathcal{B}$ 依次执行 $UpdateKeyGen(crs,x,ct)$ 与 $RevokeCS(ct,UK)$ 并返

回更新结果；对于挑战密文，$\mathcal{A}$不能获得与其对应的更新密钥UK、密文更新因子C_{update}，也不能腐化云服务器CS。

挑战阶段：$\mathcal{A}$提交挑战访问策略$\mathbb{A}^*$、两个等长消息m_0^*、m_1^*，当前版本号ver'以及待撤销用户DU_x。用户DU_x在撤销前已完成注册，其属性集合满足挑战访问策略；进入挑战阶段后，DU_x必须处于撤销状态，并且$\mathcal{A}$不得通过其他腐化用户获得一组能够满足$\mathbb{A}^*$的有效完整私钥。$\mathcal{B}$随机选择$b \in \{0,1\}$，依据ver'的二进制表示确定完整方案中参与本次加密的slot-TRRABE子方案。对每个参与加密的第k个子方案，$\mathcal{B}$生成初始挑战子密文ct_k^*。

对于包含DU_x注册位置且需要执行撤销的第k个子方案，$\mathcal{B}$按照本文$UpdateKeyGen(crs, x, ct)$算法独立选择$r_k' \in \mathbb{Z}_N$，生成代理更新密钥$UK_k = -r_k'$，并仅为该子方案中的未撤销用户生成公共更新信息$PI_k = \{P_{j,k} = a_{j,k}^{r_k'}\}_{j \neq x}$。随后CS更新$ct_k^*$。更新完成后，按照TRRABE方案聚合得到聚合密文ct^*。并将撤销后的挑战密文ct^*及各子方案对应的公共更新信息PI_k返回给$\mathcal{A}$。被撤销用户DU_x对应的更新分量$P_{x,k}$不包含在PI_k中，更新前的挑战密文ct^*也不向$\mathcal{A}$公开。

猜测阶段：$\mathcal{A}$输出对b猜测$b' \in \{0,1\}$，$\mathcal{B}$将$\square'$作为对挑战密文的猜测。

1）前向安全性

用户DU_x被撤销后仍可能保留撤销前获得的sk_x。然而，在每个执行撤销的第k个子方案中，PI_k均不包含$\{P_{x,k} = a_{x,k}^{r_k'}\}$，故$\mathrm{DU}_x$无法按照本文$Decrypt$算法取得$RSI_{x,k}$并继续解密；即使其绕过该判定并直接组合原私钥与更新后的密文，$C_{2,k}^*$中仍会留下由更新随机数引入的残余双线性因子$e(C_{0,k}^*, a_{x,k})^{-r_k'}$，该因子不能由撤销前私钥消去。若$\mathcal{A}$能利用撤销前私钥以不可忽略优势区分$m_0^*$与$m_1^*$，则$\mathcal{B}$输出同一猜测即可获得不可忽略的IND-CPA优势，与TRRABE的IND-CPA安全性矛盾。因此，本文撤销机制满足前向安全性。

2）后向安全性

设历史挑战密文的版本号为ver'，用户DU_j在该密文生成后才注册，则其注册序号满足$j > ver'$。按照本文完整TRRABE的$Decrypt$算法，用户首先比较自身序号与密文版本号；当$j > ver'$时算法直接输出$\perp$。从子方案结构看，DU_j的公钥并未参与版本ver'下聚合公钥的生成，而且其当前密钥所对应的活动子方案由当前版本号的二进制表示确定，不能替代历史密文中通过j与ver'的最高不同位所定位的子方案密钥。若$\mathcal{A}$能利用新注册用户的密钥以不可忽略优势区分历史密文中的m_0^*与m_1^*，则$\mathcal{B}$同样输出$\mathcal{A}$的猜测，从而以不可忽略优势攻破TRRABE的IND-CPA安全性。因此，本文撤销机制满足后向安全性。综上，定理成立。证毕。

5　分析与实验

5.1　功能分析

表1对比了本文方案与相关注册属性基加密方案的功能特性。功能比较表明，文献[3]仅支持注册属性加密，未考虑撤销、注销、密钥追踪和身份保护；文献[24]支持撤销和用户注销，但不具备密钥泄露追踪和身份保护能力；文献[7]支持密钥追踪，但不属于注册框架，且不支持注销与身份保护；文献[14]支持注册、撤销和密钥追踪，但缺乏用户注销和身份保护。相比之下，本文TRRABE方案同时支持注册框架、用户撤销、用户注销、密钥追踪和身份保护，更适用于医疗数据共享的需求。

表1　本文方案与相关方案的功能对比

	注册框架	撤销	用户注销	密钥追踪	身份保护
文献[3]	√	×	×	×	×
文献[24]	√	√	√	×	×
文献[7]	×	√	×	√	×
文献[14]	√	√	×	√	×
TRRABE	√	√	√	√	√

5.2　计算开销分析

为评估TRRABE的性能表现，选取了Hohenberger等[3]提出的基础RABE方案（RABE），Li等[24]提出的支持撤销和用户注销的RABE方案（RRABE），李朋祥等[14]提出的支持属性验证和用户撤销的RABE方案（VTR-RABE）进行对比分析。分析过程中涉及的符号定义如表2所示。

由表3可知，4种方案在初始化阶段的计算开销均为$O(L^2|U|E)$，说明本文方案在系统公共参数生成阶段未改变注册属性加密的主要复杂度。在密钥生成阶段，RABE、RRABE和VTR-RABE的开

销均为LE，本文方案为实现私钥泄露追踪，在用户自生成私钥分量基础上增加追踪组件和哈希计算，开销为$(L+4)E+2H$，仅增加常数级指数运算和哈希运算。

表3　本文方案与相关方案的计算开销

步骤	RABE	RRABE	VTR-RABE	TRRABE
初始化	$O(L^2\|U\|E)$	$O(L^2\|U\|E)$	$O(L^2\|U\|E)$	$O(L^2\|U\|E)$
密钥生成	LE	LE	LE	$(L+4)E+2H$
加密	$2ME$	$2ME$	$2ME+3RE$	$2(M+3)E+E_T$
解密	$2IP+IE_T$	$2IP+IE_T$	E_T	$(2I+2)P+IE_T$
追踪	-	-	$3P$	$2E+2P+H$
撤销更新	-	LE	$3E+E_{\mathrm{T}}$	$(N_u+1)E$
用户注销	-	LE	-	LE

表4　本文方案与相关方案的存储开销

方案	公共参考串	聚合公钥	辅助私钥	密文
RABE	$L^2\|U\|G$	$\|U\|G$	$\|U\|G$	MG
RRABE	$L^2\|U\|G$	$\|U\|G$	$\|U\|G$	MG
VTR-RABE	$L^2\|U\|G$	$\|U\|G$	$\|U\|G$	$MG+2N_uG$
TRRABE	$L^2\|U\|G$	$\|U\|G$	$\|U\|G$	$(M+N_u)G$

在加密阶段，RABE和RRABE的开销为$2ME$，VTR-RABE因引入撤销列表相关密文分量，开销为 $2ME+3RE$。本文方案的开销为$2(M+3)E+E_T$，相比基础RABE增加常数级运算，主要用于后续撤销处理，整体仍随加密属性数量M线性增长。

在解密阶段，本文方案开销为$(2I+2)P+IE_T$，相比RABE和RRABE增加常数次双线性对运算；VTR-RABE由于采用外包解密，用户端解密开销仅为E_T。

在追踪阶段，RABE和RRABE不支持私钥泄露追踪，VTR-RABE的单次追踪开销为$3P$。本文方案单次追踪开销为$2E+2P+H$。完整追踪算法需要依次读取用户注册字典中的候选用户公钥和身份标识，并逐一执行追踪验证。因此在最坏的情况下追踪开销为$N_u(2E+2P+H)$。为统一比较不同方案完成一次追踪验证所需的密码学计算量，本文表3及相关实验中的追踪开销均指针对单个候选用户的单次追踪验证开销。在撤销更新阶段，RRABE的开销为LE，VTR-RABE的开销为$3E+E_T$，本文方案开销为$(N_u+1)E$。用户注销阶段，本文方案与RRABE的开销均为LE。综上，本文方案在支持追踪、撤销和用户注销功能的同时，仅在密钥生成、加密、解密和追踪阶段引入有限额外开销，整体计算开销保持可接受。

表2　符号定义

符号	含义
$\|U\|$	系统属性个数
L	系统设定槽位个数
M	加密所需属性个数
I	解密所需属性个数
R	撤销列表元素个数
N_u	合法用户数量
E	群$\mathbb{G}$中的指数运算
E_T	群$\mathbb{G}_T$中的指数运算
P	双线性对运算
H	哈希运算
G	群$\mathbb{G}$中一个元素大小
G_T	群$\mathbb{G}_T$中一个元素大小

5.3　存储开销分析

各方案的存储开销对比如表4所示。TRRABE在公共参考串、聚合公钥和辅助私钥方面的存储开销与RABE方案保持同一量级，并未因拓展功能导致存储开销增高。但在密文尺寸方面，VTR-RABE由于将撤销列表相关分量嵌入密文，密文开销增加为$MG+2N_uG$，TRRABE不采用撤销列表，而是在密文更新阶段附加面向合法用户的公共更新信息，因此密文开销为$(M+N_u)G$。总体来看，本文方案的额外存储开销主要来自追踪后撤销所需的公共更新信息，这是可接受的代价。

5.4　实验对比

为评估TRRABE方案的性能，本文对比了RABE，RRABE以及VTR-RABE的算法计算开销。本文使用Java编程语言，并基于JPBC库实现双线性配对运算。实验环境设置为Windows 11操作系统，处理器为Intel Core i5-13400 2.50 GHz，内存为16 GB。

以槽位数量L为变量，各方案的初始化计算开销如图2所示。随着槽位数量L增加，4种方案的初始化开销均呈明显增长趋势；由图2可以看出，

TRRABE 在初始化阶段开销与 RABE，RRABE 以及 VTR-RABE 三个方案相差不大，说明本文方案在引入追踪、撤销和用户注销功能后，并未改变系统初始化阶段的主要计算量级。

密阶段计算开销如图 3 所示。随着加密所需属性个数的增加，各方案的加密开销相应增大，但由于 TRRABE 方案引入追踪与撤销支持所需的固定辅助分量，因此开销高于 RABE。从图中可以看出，当加密属性个数由 5 增加至 25 时，VTR-RABE 相较于 TRRABE 的运行时间约高 3.0～4.0s。相比之下，TRRABE 虽然引入了追踪与撤销支持所需的辅助分量，但加密开销仍低于 VTR-RABE，说明本文方案在支持私钥追踪和追踪后撤销的同时，保持了较好的加密效率。

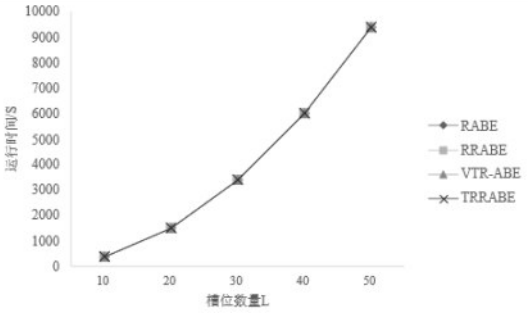

图 2　初始化阶段计算开销

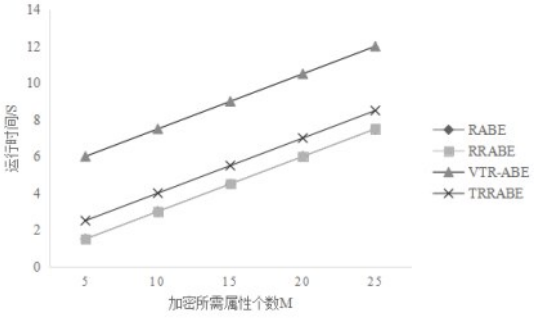

图 3　加密阶段计算开销

加解密阶段计算开销如图 4 所示。由于本文 TRRABE 方案在解密时需要增加撤销有效性判断和更新密文解密支持，因此相比 RABE 和 RRABE 方案增加了常数级的双线性对运算，当解密属性数量为 25 时，TRRABE 运行时间约为 13s，与 RABE 和 RRABE 相比仅小幅增加，说明本文方案在支持私钥泄露追踪和追踪后撤销的同时，仍保持了可接受的解密计算开销；而 VTR-RABE 方案引入外包解密，因此开销最低。

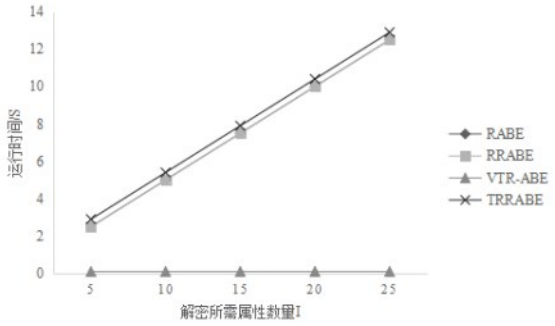

图 4　解密阶段计算开销

追踪阶段计算开销如图 5 所示。VTR-RABE 的追踪开销约为 600ms，TRRABE 的追踪开销约为 700ms，相比之下仅增加约 100ms。TRRABE 由于在追踪过程中需要结合泄露私钥中的追踪组件进行身份验证，因此引入了少量额外计算开销。但该开销仍为常数级，整体运行时间低于 1s，说明本文方案在支持注册属性加密下责任用户追踪的同时，具有较好的追踪效率。

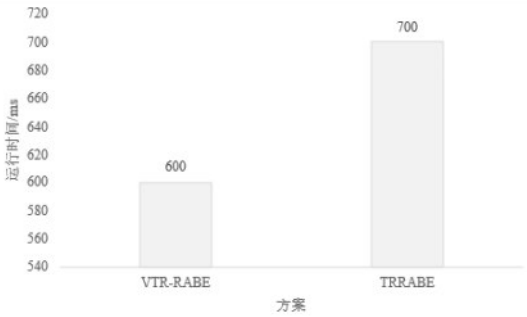

图 5　追踪阶段计算开销

以槽位数量、撤销密文数量和撤销用户数量为变量时，各方案撤销计算开销分别如图 6-图 7 和图 8 所示。由图 6 可知，当槽位数量为 100 时，TRRABE 运行时间约为 15s，相比 RRABE 的约 75s 降低约 80%，但高于 VTR-RABE。由图 7 可知，当撤销密文数量为 50 时，TRRABE 运行时间约为 11s，相比 RRABE 的约 75s 降低约 85%，说明本文方案在多密文撤销场景下具有较好的扩展性。由图 8 可知，当撤销用户数量为 10 时，TRRABE 运行时间约为 30s，相比 RRABE 的 75s 降低约 60%，仅略高于 VTR-RABE 的约 27s。综上，TRRABE 在支持追踪后定向撤销的同时，相比 RRABE 显著降低了撤销计算开销，并在多密文撤销场景下表现出较好的性能优势。

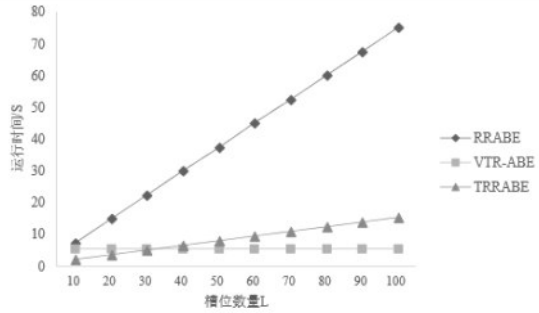

图6　撤销计算开销（变量为L）

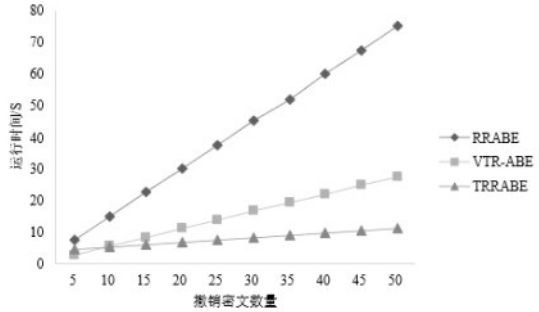

图7　撤销计算开销（变量为撤销密文数量）

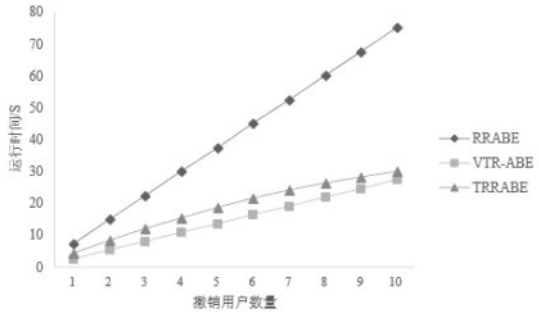

图8　撤销计算开销（变量为撤销用户数量）

6　结束语

本文面向医疗数据共享中的私钥泄露追责与权限撤销问题，提出一种可追踪可撤销注册属性加密方案。该方案利用用户自主生成私钥核心分量的特点缓解密钥托管风险，并通过构造与用户身份相关联的追踪组件，实现泄露私钥责任用户定位；在此基础上，结合密钥更新、密文更新和用户注销机制，阻止被追踪用户继续解密更新后的医疗数据，并在撤销过程中避免直接暴露用户真实身份。安全性分析表明，本文方案满足选择明文攻击安全性和私钥泄露可追踪性；性能评估表明，方案在引入追踪、撤销和注销功能的同时，计算开销仍处于可接受范围。后续工作将进一步研究多医疗机构协同场景下的高效追踪与撤销机制，并降低大规模用户环境下的计算开销。

参考文献：

[1] Sahai A, Waters B. Fuzzy identity-based encryption[C]//Annual international conference on the theory and applications of cryptographic techniques. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005: 457-473.

[2] Garg S, Hajiabadi M, Mahmoody M, et al. Registration-based encryption: removing private-key generator from IBE[C]//Theory of Cryptography Conference. Cham: Springer International Publishing, 2018: 689-718.

[3] Hohenberger S, Lu G, Waters B, et al. Registered attribute-based encryption[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. Cham: Springer Nature Switzerland, 2023: 511-542.

[4] Liu Z, Cao Z, Wong D S. White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures [J]. IEEE Transactions on Information Forensics and Security, 2012, 8 (1): 76-88.

[5] Ning J, Cao Z, Dong X, et al. White-box traceable CP-ABE for cloud storage service: How to catch people leaking their access credentials effectively[J]. IEEE Transactions on Dependable and Secure Computing, 2016, 15(5): 883-897.

[6] Ning J, Dong X, Cao Z, et al. White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes[J]. IEEE Transactions on Information Forensics and Security, 2015, 10(6): 1274-1288.

[7] Meng F, Cheng L. TSR-ABE: Traceable and server-aided revocable ciphertext-policy attribute-based encryption under static assumptions[J]. IEEE Transactions on Information Forensics and Security, 2024, 20: 955-967.

[8] Luo F, Al-Kuwari S. Generic construction of black-box traceable attribute-based encryption[J]. IEEE Transactions on Cloud Computing, 2021, 11(1): 942-955.

[9] Liu Z, Cao Z, Wong D S. Blackbox traceable CP-ABE: how to catch people leaking their keys by selling decryption devices on ebay[C]//Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. 2013: 475-486.

[10] Zhang R, Hui L, Yiu S, et al. A traceable outsourcing cp-abe scheme with attribute revocation[C]//2017 IEEE Trustcom/BigDataSE/ICESS. IEEE, 2017: 363-370.

[11] Xue J, Shi L, Zhang W, et al. Poly-ABE: A traceable and revocable fully hidden policy CP-ABE scheme for integrated demand response in multi-energy systems[J]. Journal of Systems Architecture, 2023, 143: 102982.

[12] He X, Li L, Peng H. An enhanced traceable CP-ABE scheme against various types of privilege leakage in cloud storage[J]. Journal of Systems Architecture, 2023, 136: 102833.

[13] 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.

[14] 李朋祥, 王玥欢, 贾春福, 等. 支持属性验证与用户撤销的 IoMT 数据共享方案[J]. 通信学报, 2025, 46(12).

[15] Cui H, Deng R H, Li Y, et al. Server-aided revocable attribute-based encryption[C]//European symposium on research in computer security. Cham: Springer International Publishing, 2016: 570-587.

[16] Ge C, Susilo W, Baek J, et al. Revocable attribute-based encryption with data integrity in clouds[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 19(5): 2864-2872.

[17] Hur J, Noh D K. Attribute-based access control with efficient revocation in data outsourcing systems[J]. IEEE Transactions on Parallel and Distributed Systems, 2010, 22(7): 1214-1221.

[18] Chen S, Li J, Zhang Y, et al. Efficient revocable attribute-based encryption with verifiable data integrity[J]. IEEE Internet of Things Journal, 2023, 11(6): 10441-10451.

[19] Yanfeng S, Qingji Z, Jiqiang L, Zhen H, et al. Directly revocable key-policy attribute-based encryption with verifiable ciphertext delegation [J]. Information Sciences, 2015, 295: 221-231.

[20] Han D, Pan N, Li K C. 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.

[21] Prantl T, Lauer M, Horn L, et al. Security analysis of a decentralized, revocable and verifiable attribute-based encryption scheme[C]//Proceedings of the 19th International Conference on Availability, Reliability and Security. 2024: 1-11.

[22] Peñuelas-Angulo A, Feregrino-Uribe C, Morales-Sandoval M. A revocable multi-authority attribute-based encryption scheme for fog-enabled IoT[J]. Journal of Systems Architecture, 2024, 155: 103265.

[23] Huang K. Traceable and Revocable Large Universe Multi-Authority Attribute-Based Access Control with Resisting Key Abuse[J]. Computer Networks, 2025: 111694.

[24] Li J, Chen S, Lu Y, et al. Revocable registered attribute-based encryption with user deregistration[J]. IEEE Internet of Things Journal, 2025.

[25] Jung T, Li X Y, Wan Z, et al. Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption[J]. IEEE Transactions on Information Forensics and Security, 2014, 10(1): 190-199.

[26] Zhang Y, Li J, Chen X, et al. Anonymous attribute-based proxy re-encryption for access control in cloud computing[J]. Security and Communication Networks, 2016, 9(14): 2397-2411.

[27] Zhang Y, Chen X, Li J, et al. Anonymous attribute-based encryption supporting efficient decryption test[C]//Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security. 2013: 511-516.

[28] Zhang K, Zhang Y, Li Y, et al. A blockchain-based anonymous attribute-based searchable encryption scheme for data sharing[J]. IEEE Internet of Things Journal, 2023, 11(1): 1685-1697.

[29] Meng L, Chen L, Tian Y, et al. Fabesa: Fast (and anonymous) attribute-based encryption under standard assumption[C]//Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security. 2024: 4688-4702.

[30] Deshmukh J Y, Yadav S K, Bhandari G M. Attribute-Based encryption mechanism with Privacy-Preserving approach in cloud computing[J]. Materials Today: Proceedings, 2023, 80: 1786-1791.

[31] Yin H, Li J, Zhu Y, et al. A Traceable CP-ABE Scheme Supporting Dynamic Revocation and Efficient Decryption for Medical Data Sharing [J]. IEEE Internet of Things Journal, 2025.

[32] Liu X, Wang H, Zhang B, et al. Accountable privacy-enhanced multi-authority attribute-based authentication scheme for cloud services[J]. Computer Communications, 2025: 108205.

[33] Boneh D, Goh E J, Nissim K. Evaluating 2-DNF formulas on ciphertexts[C]//Theory of cryptography conference. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005: 325-341.

[34] Beimel A. Secure schemes for secret sharing and key distribution[J]. PhD thesis, Israel Institute of Technology, Technion, 1996.

田有亮（1982- ），男，贵州盘州人，博士，贵州大学教授、博士生导师，主要研究方向为博弈论、密码学与安全协议、大数据隐私保护。

易笃兴（2002− ），男，贵州铜仁人，贵州大学硕士生，主要研究方向为属性基加密，基于身份加密。

向阿新(1996-)，男，贵州遵义人，博士，贵州大学副教授、硕士生导师，主要研究方向为区块链、密码学、密钥管理等。