计算机研究与发展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202330092
61(6): 1525−1535, 2024

mHealth 中细粒度策略隐藏和可追踪去中心访问控制方案

王静怡[1,2] 阚海斌[1,2,3]
[1](复旦大学计算机科学技术学院 上海 200433)
[2](上海市区块链工程技术研究中心 上海 200433)
[3](复旦大学义乌研究院 浙江义乌 322099)
(jingywang21@m.fudan.edu.cn)

Fine-Grained Policy-Hiding and Traceable Decentralized Access Control Scheme in mHealth

Wang Jingyi[1,2] and Kan Haibin[1,2,3]
[1](*School of Computer Science*, *Fudan University*, *Shanghai* 200433)
[2](*Shanghai Engineering Research Center of Blockchain*, *Shanghai* 200433)
[3](*Yiwu Research Institute of Fudan University*, *Yiwu*, *Zhejiang* 322099)

Abstract With the rapid development of Internet technology, the emergence of mobile health (mHealth) is expected to improve the quality of medical care. However, data security and user privacy issues in the mHealth field have not been fully resolved. The access control protocol based on ciphertext-policy attribute-based encryption (CP-ABE) is a promising technique for the sharing of personal health records (PHRs). However, direct adoption of the traditional CP-ABE in mHealth causes many problems. Firstly, centralized attribute authority has low ability to resist risks. Secondly, the access policies are in cleartext and leak the patient's privacy in the encrypted PHRs. Finally, it is difficult for the traditional CP-ABE scheme to track down the user who intentionally discloses the private key. Therefore, to solve these problems, we propose a fine-grained policy-hiding and traceable decentralized access control in mHealth. This scheme implements a decentralized attribute authority mechanism. Each attribute is expressed by an attribute name and an attribute value. In the encryption phase, the attribute value is hidden in ciphertext and only generic attribute name is exposed. When the private key is maliciously leaked, the regulator can use the identity mapping table to trace the malicious user. Through experimental simulation and comparative analysis, our scheme is suitable for the actual mHealth environment in terms of security and performance.

Key words attribute-based encryption (ABE); blockchain; access control; policy hiding; traceability; decentralization

摘 要 基于属性基加密的访问控制协议在个人健康档案共享中发挥着越来越重要的作用.但传统的基于密文策略属性基加密的访问控制方案存在着些许问题.首先,中心化的属性授权机构的抗风险能力低.其次,随密文发送未隐藏的访问策略可能会泄露患者的隐私.此外,传统方案难以追踪恶意泄露密钥的用户.为解决上述问题,提出一种适用于 mHealth 中细粒度策略隐藏和可追踪去中心访问控制方案.实现了去中

收稿日期: 2023-02-16; **修回日期**: 2023-07-25
基金项目: 国家重点研发计划项目 (2019YFB2101703); 国家自然科学基金项目 (62272107,U19A2066); 上海市科技创新行动计划项目 (21511102200); 广东省重点领域研发计划项目 (2020B0101090001)
This work was supported by the National Key Research and Development Program of China (2019YFB2101703), the National Natural Science Foundation of China (62272107, U19A2066), the Shanghai Science and Technology Innovation Action Plan Project (21511102200), and the Key-Area Research and Development Program of Guangdong Province (2020B0101090001).
通信作者: 阚海斌(hbkan@fudan.edu.cn)

心化的属性授权机构. 属性由属性名称和属性值 2 部分构成，在加密阶段属性值隐藏在密文中，只对外公开通用的属性名称. 当密钥遭到恶意泄露时，监管机构利用身份映射表可以追踪到恶意的用户. 经过实验模拟和对比分析，所提方案在安全性方面和性能上适用于实际的 mHealth 环境.

关键词 属性基加密（ABE）；区块链；访问控制；策略隐藏；可追踪性；去中心化

中图法分类号 TP391

互联网信息技术的飞速发展给移动健康(mobile healthcare, mHealth)领域带来了深刻的变革. 个人健康档案(personal health records, PHR)的大量增长，使 mHealth 进入大数据时代. 越来越多的机构和企业将 PHR 存储在第三方服务器上，这大大降低了本地存储和管理的成本. 但 PHR 涉及到患者的大量敏感信息，如血糖值、脉搏率等，一旦将其存储在云端，传统的安全机制将无法提供对 PHR 的隐私保护和访问控制.

文献 [1−5] 提出了针对云端 PHR 的访问控制方案，一种名为属性基加密(attribute-based encryption, ABE)的密码学原语因其可以同时实现细粒度访问控制和数据秘密共享[6]而受到研究人员的广泛关注. ABE 协议分为基于密钥策略的属性基加密（key-policy attribute-based encryption, KP-ABE)和基于密文策略的属性基加密（ciphertext-policy attribute-based encryption, CP-ABE)[7]. 在 CP-ABE 中，PHR 所有者可以在加密阶段指定访问策略，只有满足访问策略的用户才可以正确地解出密文，在 mHealth 领域被广泛研究.

在传统的 CP-ABE 方案中，属性密钥不包含用户和授权机构的具体信息，同一个属性可以被分发给多个用户. 因此很难对恶意用户和拥有相同属性的普通用户进行区分. 如果有恶意用户合谋共享其密钥，将会对系统的安全构成很大的威胁. 此外，中心化属性授权机构可以生成任意属性集的密钥，授权机构是否腐败会对整个系统的安全性造成严重的影响. 最近热门的区块链技术[8-9]作为分布式存储系统，创新性地将 P2P 网络技术、密码学和分布式共识技术相结合，存储在区块链上的数据不能被篡改，保证了数据的完整性和不可否认性. 因此当区块链上的密钥被滥用时，审计密钥的所有权也更为可信，为解决传统 CP-ABE 方案的密钥可追溯和去中心化问题提供了新的思路，文献 [10−13] 对此做了相关的研究.

当数据存储在分布式网络中，访问结构通常会泄露敏感信息. CP-ABE 中的访问结构是用属性构成的策略表达式，它不加隐藏地跟随密文一起发送. 任何人无论是否获得授权，都可以获得访问策略的具体细节. 对于 mHealth 来说，访问策略会泄露患者的一些隐私信息. 举例来说，如果一个患者将自己的访问策略设置为“(职工号：12345 OR (机构：A 市综合性医院 AND 职位：艾滋病专家))”，那么获得该密文的任意用户都可以推测出该患者很可能是艾滋病感染者. 这对于大部分患者而言是难以接受的，需要对访问策略进行隐藏. 目前的策略隐藏 CP-ABE 主要分为完全策略隐藏 CP-ABE[14]和部分策略隐藏 CP-ABE[15-17]. 在完全策略隐藏 CP-ABE 中，方案不会显示任何关于访问策略的属性信息，只能通过阈值策略和属性隐藏的内积谓词加密(inner-product predicate encryption, IPE)间接构建，但阈值策略在表达性方面远远不如线性秘密共享方案(linear secret sharing scheme, LSSS). 部分策略隐藏 CP-ABE 是将访问策略中涉及的敏感属性值隐藏起来，而将对应的通用属性名称跟随密文一起公开，具有良好的表达性.

基于部分策略隐藏 CP-ABE 的访问控制方案在 mHealth 中被大量研究[15-17]. 文献 [15] 提出了一种部分策略隐藏的 CP-ABE 方案，其中每个属性由一个属性名称和一个属性值表示，访问策略只包含通用属性名称，而对应的敏感属性值被隐藏在密文中. 文献 [16] 提出的 PASH 方案，同时支持部分策略隐藏和大属性集. 文献 [17] 提出了一种大属性集、部分策略隐藏并且可追溯的 mHealth 访问控制方案 HTAC. 然而，上述的 CP-ABE 方案都不能同时解决策略隐藏、可追溯性和去中心化 3 个主要特性. 本文创新性地提出了一种新型的 mHealth 中的细粒度策略隐藏和可追踪去中心化访问控制方案，通过对文献 [16] 的策略隐藏 CP-ABE 方案进行改进，本文方案具有 4 个特点：

1)大属性集. 本文方案对属性集的大小没有任何的限制，与此同时公共参数的大小是恒定的.

2)策略隐藏. 每个属性由属性名称和属性值表示. 在加密阶段，属性值隐藏在密文中，只有通用属性名称在访问策略中是可显示的.

3)可追踪性. 监管机构可以准确追溯非法泄露密钥的恶意用户，并且未经授权的访问是不被允许的.

4)去中心化. 引入门限秘密共享算法实现了去

中心化的属性授权机构，机构的创建、属性的创建和分发由用户合作完成.

1 预备知识

1.1 合数阶双线性映射

定义 1. 合数阶双线性映射[18]. 运行群生成器 $\mathcal{G}\left(1^{\lambda}\right)$，输出元组$(p_1,p_2,p_3,p_4,\mathbb{G},\mathbb{G}_T,e)$. 其中$p_1,p_2,p_3,p_4$是不同的素数，$\mathbb{G}$和$\mathbb{G}_T$是阶为$N=p_1\times p_2\times p_3\times p_4$的循环群. 双线性映射$e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$满足：

1）对于$\forall g,h\in\mathbb{G},a,b\in\mathbb{Z}_N,e\left(g^a,h^b\right)=e(g,h)^{ab}$；

2）$\exists g\in\mathbb{G}$ 使得$e(g,g)$在$\mathbb{G}_T$中阶为N；

3）对于$\forall g,h\in\mathbb{G}$, $e(g,h)$在λ的多项式时间内是可计算的.

$\mathbb{G}_{p_1},\mathbb{G}_{p_2},\mathbb{G}_{p_3},\mathbb{G}_{p_4}$分别是$\mathbb{G}$中阶为$p_1,p_2,p_3,p_4$的子群，注意$\mathbb{G}=\mathbb{G}_{p_1}\times\mathbb{G}_{p_2}\times\mathbb{G}_{p_3}\times\mathbb{G}_{p_4}$. 对于$\forall g_i\in\mathbb{G}_{p_i},g_j\in\mathbb{G}_{p_j}$, $i\neq j$，则$e\left(g_i,g_j\right)=1$.

1.2 访问结构

定义 2. 访问结构[19]. 假设$\mathcal{U}$为一系列参与者的集合. 一个集合$A\subseteq 2^{\mathcal{U}}$是单调的，当且仅当对于$\forall B\in A, C\in 2^{\mathcal{U}}$，如果$B\subseteq C$，则$C\in A$. 若$A\subseteq 2^{\mathcal{U}}\backslash\varnothing$，则集合$A$为一个访问结构.称在$A$中的集合为授权集合，不在$A$中的集合为非授权集合.

1.3 线性秘密共享方案

线性秘密共享方案最早在文献 [19] 中被提及. 假设$\mathcal{U}$为一系列参与者集合，Π为访问结构A的一个秘密共享方案. 如果Π为$\mathcal{U}$在$\mathbb{Z}_p$上的线性秘密共享方案，则其包括一个l行和n列的矩阵$\boldsymbol{M}$和将$\boldsymbol{M}$的每一行映射到$\mathcal{U}$上对应参与者的函数ρ. 那么Π包括 2 个算法：

1）考虑一个列向量$\boldsymbol{v}=(s,r_2,\cdots,r_n)$，其中$s\in\mathbb{Z}_p$是要共享的秘密，而$r_2,\cdots,r_n$为$\mathbb{Z}_p$上的随机值. 则$\boldsymbol{Mv}$是将秘密$s$共享给$\mathcal{U}$的$l$个秘密共享值，其中$\lambda_i=(\boldsymbol{Mv})_i$是属于$\rho(i)$方的秘密共享值.

2）设$S\in\mathcal{U}$为任意的授权集合，令$I\subseteq\{1,2,\cdots,l\}$且$I=\{i|\rho(i)\in S\}$. 则根据高斯消元法，可以计算出常数集$\{\omega_i\in\mathbb{Z}_p\}_{i\in I}$使得$\sum_{i\in I}\omega_i\boldsymbol{M}_i=(1,0,\ldots,0)$. 由此可以重构出共享的秘密$s=\sum_{i\in I}\omega_i\lambda_i$.

本文方案将在$\mathbb{Z}_N$上使用 LSSS 矩阵，其中N是合数. 假设用户的属性集$\mathcal{S}=\{s_1,s_2,\cdots,s_n\}$，其中$s_i\in\mathbb{Z}_N$表示属性$i$. 在部分策略隐藏 CP-ABE 中，每个属性包括属性名称和属性值 2 部分，每个属性名称有多个值，将其表示为$\mathcal{S}=(I_S,S)$，其中$I_S\subseteq\mathbb{Z}_N$表示属性名称索引，对应的属性值集合是$S=\{s_i\}_{i\in I_S}$. 用$A=(\boldsymbol{M},\rho,\mathcal{T})$表示一个特定的访问控制结构，其中$\boldsymbol{M}$和$\rho$的定义不变，$\mathcal{T}$可以被解析为$\left(t_{\rho(1)},\cdots,t_{\rho(l)}\right)$，其中$t_{\rho(i)}$指由$(\boldsymbol{M},\rho)$表示的属性$\rho(i)$的对应属性值. 如果说用户的属性集$\mathcal{S}$满足$A$，当且仅当存在$I\subseteq\{1,2,\cdots,l\}$和常数集$\{\omega_i\}_{i\in I}$，使得$\sum_{i\in I}\omega_i\boldsymbol{M}_i=(1,0,\ldots,0)$且$s_{\rho(i)}=t_{\rho(i)}$, $\forall i\in I$.

1.4 (t,n)门限秘密共享方案

(t,n)门限秘密共享方案由 Shamir[20] 和 Blakley[21] 同时提出，将需共享的秘密分为n份并分配给不同的用户.n个用户各自持有共享秘密的一部分，当且仅当t个及以上的用户联合起来才可以重构出共享秘密. 为了满足属性密码学和区块链的去中心特性，本文将使用 Pedersen[22] 提出的无需可信第三方的门限秘密共享算法.

$\mathcal{U}$为一系列参与者集合$(U_1,U_2,\cdots,U_n)$，每个用户U_i分别生成一个唯一标识符GID_i，用户间的标识符互不冲突. 用户分别选取一个随机值x_i，共享秘密值为$x=\sum_{i=1}^{n}x_i$. 再随机生成一个阶为$t-1$的多项式$f_i(z)=f_{i,0}+f_{i,1}z+\cdots+f_{i,t-1}z^{t-1}$，令$f_i(0)=x_i$，计算$share_{ij}=f_i\left(GID_j\right)$$(j\in[1,n])$. 用户$U_i$将$share_{ij}$共享给用户$U_j$同时保存自己的秘密共享值$share_{ii}$. 用户$U_i$收到剩余$n-1$个$share_{ji}$后计算自己的秘密值$share_i=\sum_{j=1}^{n}share_{ji}=\sum_{j=1}^{n}f_j(GID_i)$. 共享秘密值$x$可以被$n$个参与者中的任意$t$个重构，假设存在一个函数$F(x)=\sum_{j=1}^{n}f_j(x)$，每个用户的秘密值$share_i=\sum_{j=1}^{n}share_{ji}=\sum_{j=1}^{n}f_j(GID_i)=F(GID_i)$. 最后可以通过拉格朗日插值定理计算出$F(0)=x$.

1.5 部分策略隐藏 CP-ABE

在部分策略隐藏 CP-ABE 中，不完全的访问策略使得用户不能直接判断出自己是否具有满足访问策略的属性集. 因此在完全解密之前需要测试属性集是否满足密文的访问策略. 在现有的策略隐藏 CP-ABE 方案中，用户通常通过多次重复解密来决定其属性是否满足密文中的访问策略，直到找到满足访问策略的属性集或执行完所有可能的测试. 将 Zhang 等人[16] 提出的 CP-ABE 方案描述为：

1）$Setup\left(1^{\lambda}\right)$. 输入安全参数$\lambda$，运行群生成器$\mathcal{G}(\lambda)$得到$(N,p_1,p_2,p_3,p_4,\mathbb{G},\mathbb{G}_T,e)$. 设置属性集$U=\mathbb{Z}_N$，然后随机选择$\alpha,a\in\mathbb{Z}_N,g,h\in\mathbb{G}_{p_1},X_3\in\mathbb{G}_{p_3},Z,X_4\in\mathbb{G}_{p_4}$，计算$Y=e(g,g)^{\alpha},H=hZ$. 系统公共参数$PK=(N,g,g^a,Y,H,X_4)$，主私钥为$MSK=(\alpha,h,X_3)$.

2) $KeyGen(PK,MSK,\mathcal{S})$. 令属性集为 $\mathcal{S}=(\mathcal{J}_S,S)$，其中 $\mathcal{J}_S\subseteq\mathbb{Z}_N$ 且 $S=\{s_i\}_{i\in\mathcal{J}_S}$. 对于 $i\in\mathcal{J}_S$，随机选择 $t\in\mathbb{Z}_N$，$R,R',R_i\in\mathbb{G}_{p_3}$，输出密钥 $SK_S=\left(\mathcal{S},K,K',\{K_i\}_{i\in\mathcal{J}_S}\right)$，其中

$$K=g^{\alpha}g^{at}R,K'=g^tR',K_i=(g^{s_i}h)^tR_i.$$

3) $Encrypt(PK,m,A)$. $m\in\mathbb{G}_T$ 且 $A=(\boldsymbol{M},\rho,\mathcal{T})$，其中 $\boldsymbol{M}$ 是一个 $l\times n$ 的矩阵，ρ 是一个将 M 的每一行 $\boldsymbol{M}_x$ 映射到对应属性名的函数且 $\mathcal{T}=\left(t_{\rho(1)},t_{\rho(2)},\cdots,t_{\rho(l)}\right)\in\mathbb{Z}_N^l$. 加密算法随机选择 2 个向量 $\boldsymbol{v},\boldsymbol{v}'\in\mathbb{Z}_N^n$，其中 $\boldsymbol{v}=(s,v_2,\cdots,v_n),\boldsymbol{v}'=\left(s',v_2',\cdots,v_n'\right)$. 对于 $x\in[1,l]$，随机选择 $r_x\in\mathbb{Z}_N$，$Z_1,Z_{1,x},Z_{2,x},Z_{2,x}'\in\mathbb{G}_{p_4}$，计算密文

$$CT_A=\left((\boldsymbol{M},\rho),\tilde{C}_1,\hat{C}_1,\tilde{C}_2,\hat{C}_2,\{C_{1,x},C_{2,x},C_{2,x}'\}_{x\in[1,l]}\right),$$

其中

$$\tilde{C}_1=Y^{s'},\hat{C}_1=g^{s'}Z_1,$$

$$\tilde{C}_2=mY^s,\hat{C}_2=g^s,$$

$$C_{1,x}=g^{a\boldsymbol{M}_x\boldsymbol{v}'}\left(g^{t_{\rho(x)}}H\right)^{-s'}Z_{1,x},$$

$$C_{2,x}=g^{a\boldsymbol{M}_x\boldsymbol{v}}\left(g^{t_{\rho(x)}}H\right)^{-r_x}Z_{2,x},C_{2,x}'=g^{r_x}Z_{2,x}'.$$

4) $Decrypt(PK,CT_A,SK_S)$. 给定 CT_A 和 SK_S，首先计算满足 $(\boldsymbol{M},\rho)$ 的最小属性集 $I_{\boldsymbol{M},\rho}$，然后判断是否存在子集 $\mathcal{J}\in I_{\boldsymbol{M},\rho}$ 满足 $\{\rho(i)|i\in\mathcal{J}\}\subseteq\mathcal{J}_S$ 且

$$\tilde{C}_1=\frac{e(\hat{C}_1,K)}{\prod_{i\in\mathcal{J}}\left(e(C_{1,i},K')e(\hat{C}_1,K_{\rho(i)})\right)^{\omega_i}},$$

其中向量 $\boldsymbol{\omega}=(\omega_1,\omega_2,\cdots,\omega_l)$ 使得 $\sum_{i\in\mathcal{J}}\omega_i\boldsymbol{M}_i=(1,0,\ldots,0)$. 如果这样的 $\mathcal{J}$ 不存在，表明 $\mathcal{S}$ 不满足隐藏的访问结构 A，输出 $\perp$；否则，计算 $m=\tilde{C}_2/Y^s$，其中

$$Y^s=\frac{e\left(\hat{C}_2,K\right)}{\prod_{i\in\mathcal{J}}\left(e\left(C_{2,i},K'\right)e\left(C_{2,i}',K_{\rho(i)}\right)\right)^{\omega_i}}.$$

2 系统模型

2.1 系统架构

图 1 描述了本文的系统架构，它由 6 个通用实体组成：属性授权机构(attribute authority, AA)、云服务提供商(cloud service provider, CSP)、个人健康档案所有者(PHR owner, PO)、个人健康档案使用者(PHR user, PU)、监管机构(regulator, RA)和去中心化应用(decentralized application, DAPP).

1)DAPP 通过发布公共参数来初始化系统，并参与 RA 和 AA 的初始化.

2)AA 由多个用户组成，为 PU 生成其属性集的

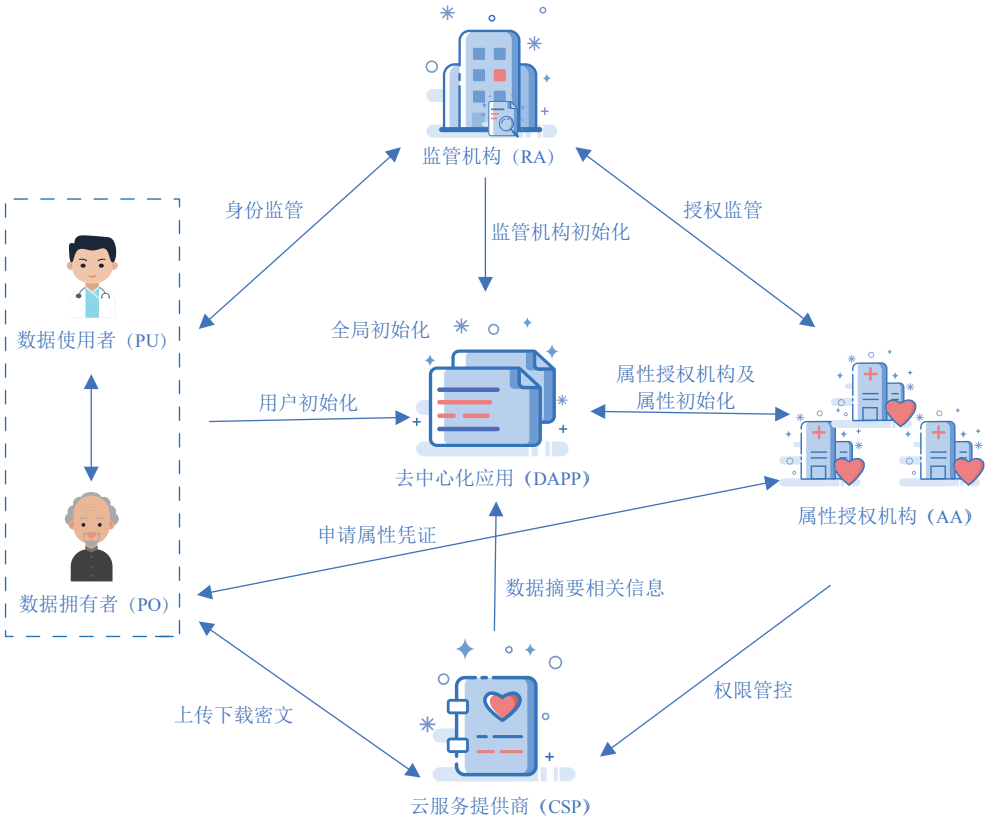

Fig. 1 Our system architecture diagram

图 1 本文的系统架构图

属性凭证 (attributes certificate, AC).

3）CSP 为 PO 提供 PHR 存储服务，如有必要，也可以删除 PHR.

4）PO 通过各种智能设备收集和整合自己的 PHR，为加密 PHR 定义任意访问策略并将加密后 PHR 上传至 CSP.

5）PU 访问加密的 PHR 以提供相应的医疗保健服务，只有在其属性满足访问策略时才能解密出 PHR.

6）RA 负责验证系统中所有用户的身份，为其生成相应的属性私钥，并追踪非法泄露其密钥的恶意 PU.

在此系统中，AA 由多个用户合作组建，是完全去中心化的，它和 RA 是值得信赖的. 但是 CSP 是诚实且好奇的，它诚实地执行指定的程序，但试图从加密的 PHR 中获取隐私信息. 攻击者可能是 1 个或 1 组恶意 PU，他们不仅试图获得 PHR 中的隐私信息，还想知道隐藏的访问策略的属性值.

2.2 架构定义

1）$GlobalSetup\left(1^{\lambda}\right)\rightarrow PP$.全局系统初始化. 输入安全参数$\lambda$，然后输出系统公共参数$PP$.

2）$RASetup(PP)\rightarrow RPK,RSK$.RA 初始化. 生成监管机构的公钥$RPK$和私钥$RSK$，并将监管机构公钥$RPK$公开上链.

3）$UserSetup(PP,baseinfo)\rightarrow UPK,USK$.用户初始化. 生成用户的公开密钥$UPK$和签名私钥$USK$，并将用户公钥$UPK$公开上链.

4）$AASetup(PP,n,t)\rightarrow OPK$. AA 初始化，多个机构成员合作生成属性授权机构的公钥OPK并公开上链.

5）$AttrGen(PP,attrname,n,t)\rightarrow APK$. AA 机构属性初始化. 多个机构成员合作生成授权机构属性的公钥APK并公开上链.

6）$UACertGen(PP,GID,\mathcal{S})\rightarrow AC$. 属性凭证申请. 用户需要向属性授权机构提出申请，请求其为属性集$\mathcal{S}$颁发属性凭证AC，在这一过程中，为确保安全性和可靠性至少需要达到设定的阈值数量的机构成员同意方可进行后续操作.

7）$KeyGen(PP,GID,AC,\{RSK,APK\},\mathcal{S})\rightarrow SK_{\{GID,\mathcal{S}\}}$. 属性密钥生成. PU 首先计算身份签名$\sigma$并提交给 RA，验证成功后 RA 为其计算属性密钥$SK_{\{GID,\mathcal{S}\}}$，并将签名σ和GID记录到身份映射表（identity table, IT）中.

8 ）$Encrypt(PP,m,\{RPK,UPK,APK\},A)\rightarrow CT_A$. 属性加密. PO 定义特定的访问控制策略，将明文m加密后生成密文CT_A并上传到云端服务器 CSP.

9）$Decrypt\left(SK_{\{GID,\mathcal{S}\}},PP,CT_A\right)\rightarrow m$ or $\perp$.属性解密. PU 从云端服务器 CSP 上下载密文CT_A，然后测试其属性密钥集$SK_{\{GID,\mathcal{S}\}}$是否满足访问策略. 若满足，则可以成功解密获得明文m.

10）$Trace\left(PP,SK^{*}_{\{GID,\mathcal{S}\}},\mathcal{S}\right)\rightarrow GID$ or $\perp$.恶性用户身份跟踪. RA 首先验证密钥$SK^{*}_{\{GID,\mathcal{S}\}}$的格式是否合法；然后根据密钥中的$\sigma$和身份映射表 IT 恢复出恶意泄露密钥用户的真实身份.

3 方案设计

3.1 符号说明

本文涉及到的符号及其描述如表 1 所示.

Table 1 Symbols and Their Description

表 1 符号及其描述

符号	描述	符号	描述
$\mathbb{G},\mathbb{G}_T$	乘法循环群	$\mathbb{G}_{p_i}$	阶为p_i的子群
GID	用户唯一标识符	PP	系统公共参数
IC	中间密文	RPK	监管机构公钥
RSK	监管机构私钥	UPK	用户公钥
USK	用户私钥	OPK	属性授权机构公钥
APK	机构属性公钥	AC	属性凭证
σ	用户签名	$SK_{\{GID,\mathcal{S}\}}$	属性集$\mathcal{S}$的密钥
$\mathcal{S}$	用户属性集	$\mathcal{J}_{\mathcal{S}}$	用户属性名索引集
S	用户属性值集	$\boldsymbol{M},\rho$	策略矩阵及映射
$\mathcal{T}$	$\boldsymbol{M}$每一行属性值集	m	待加密 PHR 数据
CT_A	PHR 数据密文	$\mathcal{J}$	最小授权集

3.2 方案设计

本文提出的 mHealth 中的细粒度策略隐藏和可追踪去中心化访问控制方案是在 PASH 方案[16] 的基础上，结合文献 [10] 实现了属性授权机构的去中心化，具体的方案设计如下文所示.

3.2.1 初始化

系统的初始化主要包括 5 个步骤：

1）$GlobalSetup\left(1^{\lambda}\right)\rightarrow PP$. 此算法由智能合约完成系统的初始化. 输入安全参数λ，选择阶为$N=p_1\times p_2\times p_3\times p_4$的循环群$\mathbb{G}$和一个双线性映射$e:\mathbb{G}\times\mathbb{G}\rightarrow\mathbb{G}_T$，其中$\mathbb{G}=\mathbb{G}_{p_1}\times\mathbb{G}_{p_2}\times\mathbb{G}_{p_3}\times\mathbb{G}_{p_4}$.$g,g_1$为$\mathbb{G}$的生成元. 选择 2 个安全的哈希值 $f_0:\{0,1\}^{*}\rightarrow\mathbb{G}$, $f_1:\{0,1\}^{*}\rightarrow\mathbb{Z}_N$.

最终系统公共参数为

$$PP=\langle\mathbb{G},\mathbb{G}_T,e,N,p_1,p_2,p_3,p_4,g,g_1,f_0,f_1\rangle.$$

2）$RASetup(PP)\rightarrow RPK,RSK$. 此算法负责 RA 的初始化. 输入系统公共参数PP，随机选择$a\in\mathbb{Z}_N,h\in\mathbb{G}_{p_1}$，$X_3\in\mathbb{G}_{p_3},Z,X_4\in\mathbb{G}_{p_4}$，计算$H=hZ$. 监管机构生成公钥

$RPK=\langle g^a,X_4,H\rangle$和私钥$RSK=\langle a,X_3,h\rangle$，并将公钥$RPK$通过智能合约公开上链. 此外，RA还初始化一个空的身份映射表IT，记录用户签名σ和GID的对应关系，用来对用户密钥进行追踪.

3) $UserSetup(PP,baseinfo)\rightarrow UPK,USK$. 此算法负责用户初始化. 输入系统公共参数$PP$和用户的基本信息$baseinfo$. 用户生成随机值$x_I\in\mathbb{Z}_N$，计算$P_I=g^{x_I}$，其中$P_I$用作用户公开密钥$UPK$，$x_I$用作用户签名私钥$USK$. 用户计算全局唯一身份标识$GID=f_1(baseinfo||P_I)$，将密钥$UPK$和身份标识$GID$通过智能合约公开上链.

4) $AASetup(PP,n,t)\rightarrow OPK$. 此算法负责AA初始化. 假设属性授权机构由具备唯一身份标识GID_i的n个机构成员$User_i(i\in[1,n])$构成，并且门限秘密共享算法的t值已被设定好. 初始化主要包括2个阶段：机构初始化阶段和机构公钥生成阶段.

在机构初始化阶段,机构中的每个成员$User_i(i\in[1,n])$随机选择$\alpha_i\in\mathbb{Z}_N$作为部分机构秘密. 各个成员部分机构秘密之和，即$\alpha=\sum_{i=1}^{n}\alpha_i$为方案中的最终机构秘密. 随后，每个成员$User_i(i\in[1,n])$随机生成阶为$t-1$的多项式$f_i(x)=a_0+a_1x+a_2x^2+\cdots+a_{t-1}x^{t-1}$，令$f_i(0)=a_0=\alpha_i$. 计算秘密共享值$share_{ij}=f_i(GID_j)$，并将其发送给$User_j(j\in[1,n],j\neq i)$，自己储存好$share_{ii}$. 当机构成员$User_i$拿到其他$n-1$个机构成员发送的秘密共享值后，计算部分机构私钥$osk_i=\sum_{j=1}^{n}share_{ji}$,部分机构公钥为$opk_i=e(g,g_1)^{osk_i}$，并将$opk_i$公开上链.

智能合约完成机构公钥生成阶段的主要工作，随机选取t个机构成员生成的部分机构公钥opk_i，由秘密共享原理可生成AA的公钥OPK并将其公开上链，计算

$$OPK=\prod_{i=1}^{t}opk_i^{\prod_{j=1,j\neq i}^{t}\frac{GID_j}{GID_j-GID_i}}=e(g,g_1)^{\sum_{i=1}^{t}\left(osk_i\prod_{j=1,j\neq i}^{t}\frac{GID_j}{GID_j-GID_i}\right)}=e(g,g_1)^{\sum_{i=1}^{n}\alpha_i}=e(g,g_1)^{\alpha}.$$

5) $AttrGen(PP,attrname,n,t)\rightarrow APK$. 此算法负责AA的初始化，与机构初始化类似，其包括机构属性初始化阶段和属性公钥生成阶段. 对属性$attrname$,n个用户需要协作进行5个步骤.

①在机构属性初始化阶段，机构中的每个成员$User_i(i\in[1,n])$随机选择$\beta_i\in\mathbb{Z}_N$作为其部分机构属性秘密，各个成员部分机构属性秘密之和，即$\beta=\sum_{i=1}^{n}\beta_i$为方案中的最终的机构属性秘密. ②每个成员$User_i$随机生成阶为$t-1$的多项式$f_i(x)=a_0+a_1x+\cdots+a_{t-1}x^{t-1}$，令$f_i(0)=a_0=\beta_i$. ③计算秘密共享值$share_{ij}=f_i(GID_j)$ $(j\in[1,n],j\neq i)$，并将其秘密共享值通过智能合约传输给其他用户. ④当机构成员$User_i$拿到其他$n-1$个秘密共享值后，计算$ask_i=\sum_{j=1}^{n}share_{ji}$和$apk_i=g^{ask_i}$，并将$apk_i$通过智能合约公开上链.⑤在属性公钥生成阶段，智能合约从中随机选取t个机构成员生成的apk_i，由秘密共享原理可生成授权机构属性公钥APK并公开上链，计算

$$APK=\prod_{i=1}^{t}apk_i^{\prod_{j=1,j\neq i}^{t}\frac{GID_j}{GID_j-GID_i}}=g^{\sum_{i=1}^{t}\left(ask_i\prod_{j=1,j\neq i}^{t}\frac{GID_j}{GID_j-GID_i}\right)}=g^{\sum_{i=1}^{n}\beta_i}=g^{\beta}.$$

3.2.2 PU授权

新加入的PU首先需要向AA申请其属性集$\mathcal{S}=(\mathcal{J}_{\mathcal{S}},S)$的属性凭证$AC$，再计算其签名$\sigma$并向RA提交进行身份验证. 最后RA为其分发$\mathcal{S}$的属性密钥. 具体细节为：

1) $UACertGen(PP,GID,\mathcal{S})\rightarrow AC$. 此算法负责$AC$的申请过程. PU向AA申请其$AC$，需要经过AA下至少$t$个机构成员的同意. 每个成员$User_j(j\in[1,t])$计算$auth=(g_1)^{osk_j}$，并返回给PU. 在收到$t$个$auth$后，PU计算$AC$：

$$AC=\prod_{i=1}^{t}auth^{\prod_{j=1,j\neq i}^{t}\frac{GID_j}{GID_j-GID_i}}=g_1^{\sum_{i=1}^{t}\left(osk_i\prod_{j=1,j\neq i}^{t}\frac{GID_j}{GID_j-GID_i}\right)}=g_1^{\sum_{i=1}^{n}\alpha_i}=g_1^{\alpha}.$$

2) $KeyGen(PP,GID,AC,\{RSK,APK\},\mathcal{S})\rightarrow SK_{\{GID,\mathcal{S}\}}$. 此算法主要由RA负责属性集密钥生成过程，包括验证身份和生成密钥2个部分. 具体实现细节有：

PU首先利用AC和其私钥USK计算其身份签名$\sigma=f_0(AC,P_I)^{x_I}$，连同AC一起提交到RA进行验证.

RA首先验证PU的身份签名是否有效，即检查

$e(\sigma,g)=e(f_0(AC,P_I),P_I)$是否成立. 如果验证通过, RA 计算$u=f_0(GID)$, 随机选择$R,R',R_i\in\mathbb{G}_{p_3},t\in\mathbb{Z}_N$, 计算出$SK_{\{GID,S\}}=(\sigma,K,K',\{K_i\}_{i\in S})$并将$\sigma$和对应的$GID$记录在 IT 中. 其中

$$K=g_1^{\alpha u}g^{at}R,K'=g^tR',K_i=(APK_ih)^tR_i,$$

否则 RA 将拒绝为其分发密钥.

3.2.3 PHR 加密

要加密 PHR 数据, PO 定义一个访问结构$A=(\boldsymbol{M},\rho,\mathcal{T})$, 通过运行 Encrypt 算法对 PHR 进行加密生成密文CT_A, 然后将CT_A发送到 CSP.

$Encrypt(PP,m,\{RPK,UPK,APK\},A)\rightarrow CT_A$: *Encrypt* 算法分为线下(offline)阶段和线上(online)阶段, 假设 LSSS 访问控制结构中的矩阵最多有P行, 设计的具体细节为:

1) offline 阶段. PO 随机选择 2 个秘密值$s,s'\in\mathbb{Z}_N$, 生成随机向量$\boldsymbol{v}=(s,v_2,\cdots,v_n)$和$\boldsymbol{v}'=(s',v_2',\cdots,v_n')$. 再随机选择$Z_{2,x}'\in\mathbb{G}_{p_4}$, 计算

$$C_{\text{base}}=e(g,g_1)^{\alpha s},C_1'=g^s,$$

$$C_2=e(g,g_1)^{\alpha s'},C_2'=g^{s'}Z_{2,x}'.$$

对于$\forall x\in[1,P]$, 随机选择$Z_{1,x},Z_{1,x}',Z_{2,x}\in\mathbb{G}_{p_4},r_x,\gamma_x,\gamma_x'\in\mathbb{Z}_N$, 计算

$$D_{1,x}=g^{r_x}Z_{1,x}',C_{2,x}=g^{a\gamma_x'}(APK_xH)^{-s'}Z_{2,x},$$

生成中间密文$IC=\left(C_{\text{base}},C_1',C_2,C_2'\right)$.

2) online 阶段. 当 PHR 数据m已知时, PO 根据指定的访问控制结构对秘密值s, s'进行线性秘密共享. 对于$\forall x\in[1,l]$, 计算$\lambda_x=\boldsymbol{M}_xv_x,\lambda_x'=\boldsymbol{M}_xv_x'$. 利用中间密文$IC$和共享矩阵计算

$$C_1=mC_{\text{base}},C_{1,x}=g^{a\lambda_x}(APK_xH)^{-r_x}Z_{1,x},$$

$$C_{2,x}'=\lambda_x'-\gamma_x'.$$

最后, 生成最终密文为

$$CT_A=\left((\boldsymbol{M},\rho),C_1,C_1',C_2,C_2',\{C_{1,x},D_{1,x},C_{2,x},C_{2,x}'\}_{x\in[1,l]}\right)$$

3.2.4 PHR 访问

如果 PU 的特定属性集$\mathcal{S}$匹配CT_A中的访问策略, 则可以通过 *Decrypt* 算法恢复出 PHR.

$Decrypt\left(SK_{\{GID,\mathcal{S}\}},PP,CT_A\right)\rightarrow m$ or $\perp$. *Decrypt* 算法分为匹配(*matching*)和解密(*decryption*)2 个算法. PU 首先计算满足$(\boldsymbol{M},\rho)$的最小子集$I_{\boldsymbol{M},\rho}$. 设计的具体细节为:

1) *matching* 算法. PU 首先检查是否存在$\mathcal{J}\in I_{\boldsymbol{M},\rho}$满足$\{\rho(i)|i\in\mathcal{J}\}\subseteq\mathcal{J}_S$和等式

$$C_2=\left(\frac{e\left(C_2',K\right)}{\prod_{i\in\mathcal{J}}\left(e\left(C_{2,i}g^{aC_{2,i}'},K'\right)e\left(C_2',K_{\rho(i)}\right)\right)^{\omega_i}}\right)^{\frac{1}{u}},$$

其中$\{\omega_i\}_{i\in\mathcal{J}}$使得$\sum_{i\in\mathcal{J}}\omega_i\boldsymbol{M}_i=(1,0,\ldots,0)$. 如果存在这样的$\mathcal{J}$,则 PU 使用$\mathcal{J}$和$\{\omega_i\}_{i\in\mathcal{J}}$执行 *decryption* 算法; 否则意味着 PU 的属性集$\mathcal{S}$不满足$\boldsymbol{M}$, 算法返回$\perp$.

2) *decryption* 算法. PU 计算$m=C_1/B$, 其中

$$B=\left(\frac{e\left(C_1',K\right)}{\prod_{i\in\mathcal{J}}\left(e\left(C_{1,i},K'\right)e\left(D_{1,i},K_{\rho(i)}\right)\right)^{\omega_i}}\right)^{\frac{1}{u}}.$$

3.2.5 追 溯

如果恶意的 PU 泄露了他的密钥$SK_{\{GID,\mathcal{S}\}}^*$, RA 可以通过运行 Trace 算法来识别其真实身份.

$Trace\left(PP,SK_{\{GID,\mathcal{S}\}}^*,\mathcal{S}\right)\rightarrow GID$ or $\perp$.RA 首先检查被泄露的密钥$SK_{\{GID,\mathcal{S}\}}^*=(\sigma,K,K',\{K_i:\forall i\in\mathcal{S}\})$的格式是否正确. 如果$SK_{\{GID,\mathcal{S}\}}^*$能够通过 3 个检查, 则称为格式良好.

① $\sigma,K,K',K_i\in G$;

② $e(g,K)=e(g,g_1)^{\alpha u}e(g^a,K')\neq1$;

③ $\exists i\in\mathcal{S}$, 使得$e(K_i,g)=e\left(K',g^{\beta_i}\right)$.

然后 RA 通过在 IT 表中寻找σ来找出用户的真实身份. 如果找不到, RA 返回$\perp$.

4 方案分析

4.1 稳健性分析

本文方案中的稳健性指 PU 可以访问到 PHR, 当且仅当其拥有的特定属性集$\mathcal{S}$匹配底层的访问结构A. 一方面, *matching* 算法是正确的. 当且仅当$t_{\rho(i)}=s_{\rho(i)}$, 对于$i\in\mathcal{J}$, 计算出

$$C_2=\left(\frac{e\left(C_2',K\right)}{\prod_{i\in\mathcal{J}}\left(e\left(C_{2,i}g^{aC_{2,i}'},K'\right)e\left(C_2',K_{\rho(i)}\right)\right)^{\omega_i}}\right)^{\frac{1}{u}}=\frac{e(g,g_1)^{\alpha s'}e(g,g)^{ats'}}{\prod_{i\in\mathcal{J}}\left(e\left(g^{a\lambda_i'},g^t\right)e\left((APK_iH)^{-s'},g^t\right)e\left(g^{s'},(APK_ih)^t\right)\right)^{\omega_i}}=$$

$$\frac{e(g,g_1)^{\alpha s'}e(g,g)^{ats'}}{e(g,g)^{at\sum_{i\in\mathcal{J}}\lambda_i'\omega_i'}}=\frac{e(g,g_1)^{\alpha s'}e(g,g)^{ats'}}{e(g,g)^{at\left(\sum_{i\in\mathcal{J}}A_i\omega_i'\right)v'}}=e(g,g_1)^{\alpha s'}.$$

另一方面, 在 *matching* 算法找到属性集$\mathcal{S}$满足访问结构A时, 则存在一个属性集合$\mathcal{J}$和常数集$\{\omega_i\}_{i\in\mathcal{J}}$使得

$\forall i \in J, \sum_{i\in\mathcal{J}} \boldsymbol{M}_i\omega_i' = (1,0,\ldots,0), t_{\rho(i)} = s_{\rho(i)}$. 计算出

$$B = \left(\frac{e\left(C_1', K\right)}{\prod_{i\in\mathcal{J}}\left(e\left(C_{1,i}, K'\right)e\left(D_{1,i}, K_{\rho(i)}\right)\right)^{\omega_i}}\right)^{\frac{1}{u}} = \frac{e(g,g_1)^{\alpha s}e(g,g)^{ats}}{\prod_{i\in\mathcal{J}}\left(e\left(g^{a\lambda_i}, g^t\right)e\left((APK_iH)^{-r_i}, g^t\right)e\left(g^{r_i}, (APK_ih)^t\right)\right)^{\omega_i}} =$$

$$\frac{e(g,g_1)^{\alpha s}e(g,g)^{ats}}{e(g,g)^{at\sum_{i\in\mathcal{J}}\lambda_i\omega_i}} = \frac{e(g,g_1)^{\alpha s}e(g,g)^{ats}}{e(g,g)^{at\left(\sum_{i\in\mathcal{J}}M_i\omega_i\right)v}} = e(g,g_1)^{\alpha s}.$$

PU 计算 $m = C_1/B = me(g,g_1)^{\alpha s}/e(g,g_1)^{\alpha s}$ 可以恢复出 PHR, 解密阶段也是正确的. 因此, 本文方案具有稳健性.

4.2 安全性分析

在数据安全方面, 攻击者可以窃听公共通道上传输的加密 PHR, 并试图访问未经授权的 PHR.在属性隐私保护方面, 攻击者的目标是从加密的 PHR 中提取敏感属性的属性值. 具体来说, 本文将考虑的安全性需求有 3 点:

1)PHR 机密性. 外包的 PHR 对于 POs 来说是私有的和敏感的, 因此应该防止未经授权的访问.

2)抗合谋攻击. 合谋攻击指不同的 PUs 可以通过合并他们的密钥来读取他们无权访问的 PHR.

3)隐私性. 在本文方案中, 与访问策略关联的属性值是敏感的. 为了保护患者的隐私, 应该将其隐藏在加密的 PHR 中.

本文方案能够有效满足上述 3 个安全需求: 首先, PHR 在 CSP 上都是密文, 基于属性的加密算法保证其安全性, 攻击者无法得到敏感信息, 保证了 PHR 的机密性. 密钥集合 $SK_{\{GID,\mathcal{S}\}}$ 中的密钥 K 生成合法的 PU 的 GID 的哈希值. 多个不合法的 PUs 合谋是没有办法共享其属性, 也无法使用具备不同 GID 的哈希值的属性密钥进行解密, 可以抵抗合谋攻击. 在隐私性方面, 本文方案中的每个属性包括 2 部分: 属性名称及属性值. 如果 PU 的属性集合 $\mathcal{S}$ 不满足与密文关联的访问结构, 则隐藏访问结构中的属性值, 而其他访问结构的相关信息如属性名称是公开的. 如上文提到的例子: 如果 PO 使用本文方案来加密他的 PHR, 任何获得密文的人都只能获得访问结构的信息, 如“(职工号: * OR(机构: * AND 职位: *))”; 而相对敏感的属性值, 如“12345”“A 市综合性医院”“艾滋病专家”则是不可见的, 实现了对用户更高层次的隐私保护.

4.3 可追踪性分析

当密钥 $SK_{\{GID,\mathcal{S}\}}^* = (\sigma, K, K', \{K_i\}_{i\in\mathcal{S}})$ 被滥用, 需要 RA 来追踪密钥的真实用户身份信息. RA 首先对密钥的格式进行检查, 检查 $e(g,K) = e(g,g_1)^{\alpha u}e(g^a,K')$ 是否成立. 若 $SK_{\{GID,\mathcal{S}\}}^*$ 格式正确, 则

$$e(g,g_1)^{\alpha u}e(g^a,K') = e(g,g_1)^{\alpha u}e\left(g^a,g^tR'\right) =$$
$$e\left(g,g_1^{\alpha u}\right)e\left(g,g^{at}\right) = e\left(g,g_1^{\alpha u}g^{at}\right) = e(g,K).$$

若不成立, RA 输出 $\perp$; 若成立, 则寻找是否存在非空集 $\mathcal{S}' \in \mathcal{S}$ 满足方程 $e\left(K', g^{\beta_i}\right) = e(K_i, g)(i \in \mathcal{S}')$. 若 $SK_{\{GID,\mathcal{S}\}}^*$ 格式标准, 则

$$e(K_i,g) = e\left((APK_ih)^tR_i, g\right) =$$
$$e\left(\left(g^{\beta_i}h\right)^t, g\right) = e(g,g)^{\beta_i t} = e\left(K', g^{\beta_i}\right).$$

如果 $\mathcal{S}'$ 非空, RA 可根据密钥中的 σ 到 IT 中查找用户对应的 GID; 否则 RA 输出 $\perp$.

4.4 方案对比分析

4.4.1 功能特性对比

表 2 比较了文献 [10,15−17,23−24] 和本文方案的功能特性, 包括可表达性、属性集的规模、可追溯性等. 可以看出, 只有本文方案才能同时支持大属性集、可追溯性和策略隐藏, 并实现了属性授权机构完全的去中心化.

Table 2 Function Feature Comparison of Related Work

表 2 相关工作的功能特性对比

方案	可表达性	群	大属性集	可追踪性	策略隐藏	分布式
文献 [10]	LSSS	Prime	是	否	否	是
文献 [15]	LSSS	Composite	否	否	是	否
文献 [16]	LSSS	Composite	是	否	是	否
文献 [17]	LSSS	Composite	是	是	是	否
文献 [23]	AND	Prime	否	否	是	否
文献 [24]	LSSS	Prime	是	否	否	否
本文	LSSS	Composite	是	是	是	是

4.4.2 效率对比

表 3 和表 4 显示了文献 [16−17] 和本文方案在效率方面的对比结果. 注意表 3 中本文方案的公共参数包括系统公共参数 PP、监管机构公共参数 RPK 等本文方案中涉及到的所有公开的参数.

从表 3 中可以看出本文方案相比文献 [16−17], 公共参数长度略大, 这是因为增加了用于实现可追溯性和去中心化的附加参数, 并且一部分的线下加

Table 3 Comparison of the Parameter Length

表 3 参数长度对比

方案	公共参数长度	私钥长度	密文长度
文献 [16]	$4\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	$(\|\mathcal{S}\|+2)\|\mathbb{G}\|+2$	$(3l+2)\|\mathbb{G}\|+2\|\mathbb{G}_T\|$
文献 [17]	$5\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	$(\|\mathcal{S}\|+3)\|\mathbb{G}\|+1\|\mathbb{Z}_N\|$	$(3l+4)\|\mathbb{G}\|+2\|\mathbb{G}_T\|$
本文	$6\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	$(\|\mathcal{S}\|+2)\|\mathbb{G}\|+2$	$(4l+2)\|\mathbb{G}\|+2\|\mathbb{G}_T\|$

密导致生成的密文长度略大些. 在密钥大小方面, 本文方案未造成过多的开销.

从表 4 中可以看出相较于文献 [16-17], 本文方案在*KeyGen*阶段和*Encryption*阶段所需的计算开销较低, 主要原因是一些参数已经在初始化阶段和线下阶段完成计算.*Matching*阶段需要比文献 [16] 更多的计算开销, 这主要是由于其支持可追溯性而产生的. 在*Decryption*阶段本文方案没有增加更多的开销.

4.4.3 实验仿真

本节在 Ubuntu 18.04 64 位, 4 核 16 GB 的 PC 机上基于 JPBC 库(Java pairing-based cryptography library)的 Hyperledger Fabric 网络环境下对本文方案、文献 [16] 方案以及文献 [17] 方案进行仿真实现. 本文方案在对比分析相关算法的计算开销后, 给出了 3 种方案在密钥生成、加密、匹配和解密阶段的时间效率方面的实验结果, 如图 2 所示.

由图 2 可知, 在 *KeyGen* 和 *Encryption* 阶段, 本文方案的效率明显优于其他 2 个方案. 这是因为采用了预加密技术, 在知道待加密 PHR 数据之前进行了大

Table 4 Comparison of the Computation Cost

表 4 计算开销对比

方案	*KeyGen*开销	*Encryption*开销	*Matching*开销	*Decryption*开销
文献 [16]	$(2\|\mathcal{S}\|+3)E$	$(6l+2)E+2E_T$	$2\|\mathcal{J}\|E+2P$	$\|\mathcal{J}\|E_T+(2\|\mathcal{J}\|+1)P$
文献 [17]	$(2\|\mathcal{S}\|+4)E$	$(6l+4)E+2E_T$	$(2\|\mathcal{J}\|+2)E+3P$	$2E+\|\mathcal{J}\|E_T+(2\|\mathcal{J}\|+1)P$
本文	$(1\|\mathcal{S}\|+3)E$	$3lE+1E_T$	$3\|\mathcal{J}\|E+2P$	$\|\mathcal{J}\|E_T+(2\|\mathcal{J}\|+1)P$

注: E和E_T分别表示在群$\mathbb{G}$和$\mathbb{G}_T$上的指数运算, P指双线性配对操作. $|\mathcal{S}|$表示用户属性集$\mathcal{S}$的大小, l表示访问控制矩阵$\boldsymbol{M}$的行数. $|\mathcal{J}|$表示解密阶段使用的最小授权属性集$\mathcal{J}$的大小.

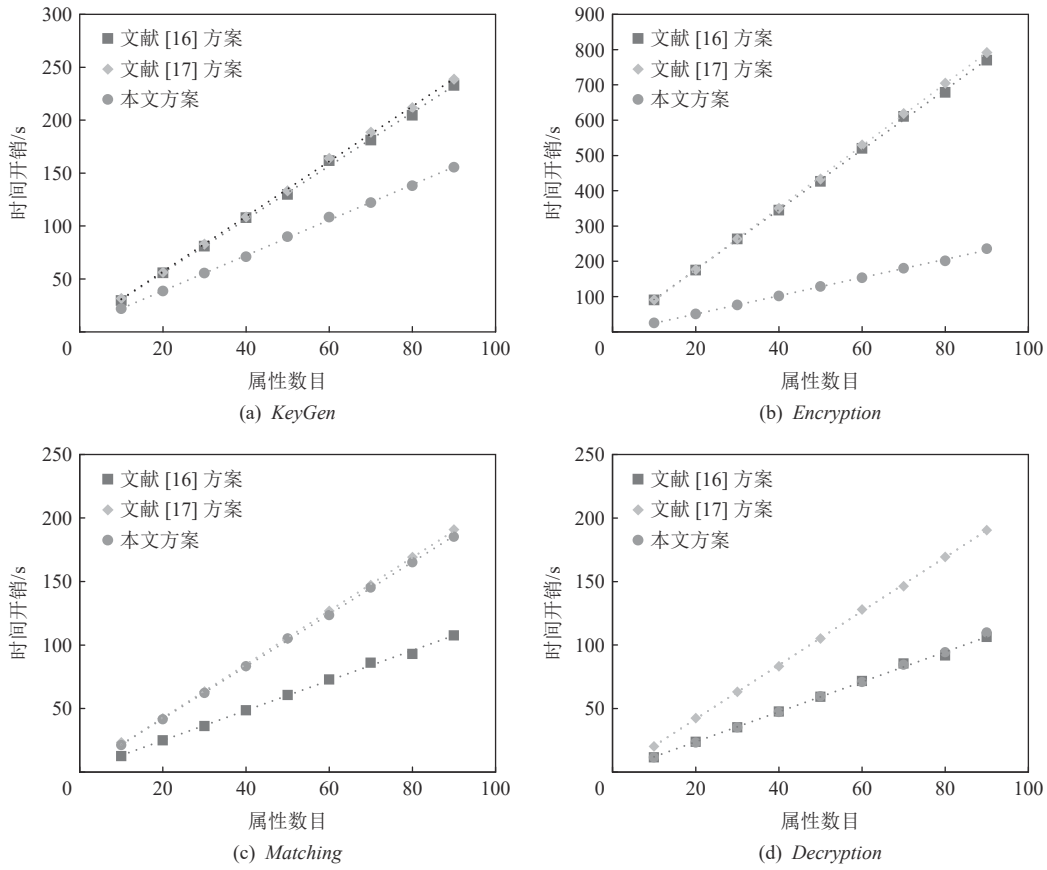

Fig. 2 Time cost of computing process in *KeyGen*, *Encryption*, *Matching* and *Decryption*

图 2 *KeyGen*, *Encryption*, *Matching* 和 *Decryption* 计算过程的时间开销

量的预计算. 当知道要加密的数据时, 可以快速生成密文. 在 *Matching* 阶段, 本文方案在时间效率上略低于文献 [16] 方案, 但在 *Decryption* 阶段没有增加更多的开销且优于文献 [17] 方案, 符合表 4 的理论结果. 总之, 本文方案在 *KeyGen* 和 *Encryption* 阶段效率得到了很大的提高, 此外还增加了属性隐藏和可追踪的功能特性.

5 结 论

本文提出了一种 mHealth 中细粒度策略隐藏和可追踪去中心访问控制方案, 有效地解决了 mHealth 中的数据安全和用户隐私问题. 本文方案支持大属性集和 LSSS 策略, 访问策略具有良好的表达性. 与此同时, 访问策略中涉及到的敏感属性值均被隐藏, 从而进一步提高了对用户的隐私保护. 在正式加密之前增加了一个线下加密过程, 很大程度上提高了用户加密的效率. 经过理论分析和实验结果表明, 本文方案比现有方案更高效, 具有实际的应用可行性. 未来研究的方向是找到适合本文方案的属性撤销机制.

作者贡献声明: 王静怡提出了论文思路和技术方案、优化方法技术, 完成部分实验并撰写论文; 阚海斌提出指导意见并修改论文.

参 考 文 献

[1] Chen T S, Liu C H. Secure dynamic access control scheme of PHR in cloud computing[J]. Journal of Medical Systems, 2012, 36((2):): 4005−4020

[2] Li Ming, Yu Shucheng, Ren Kui, et al. Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings[C]//Proc of the 6th Int ICST Conf on Security and Privacy in Communication Networks. Berlin: Springer, 2010: 89−106

[3] Zhang Leyou, Hu Gongcheng, Mu Yi, et al. Hidden ciphertext policy attribute-based encryption with fast decryption for personal health record system[J]. IEEE Access, 2019, 7: 33202−33213

[4] Wang Changji, Liu Xuan, Li Wentao. Implementing a personal health record cloud platform using ciphertext-policy attribute-based encryption[C]//Proc of the 4th Int Conf on Intelligent Networking and Collaborative Systems. Piscataway, NJ : IEEE, 2012: 8−14

[5] Sun Jianfei, Xiong Hu, Liu Ximeng, et al. Lightweight and privacy-aware fine-grained access control for IoT-oriented smart health[J]. IEEE Internet of Things Journal, 2020, 7(7): 6566−6575

[6] Sahai A, Waters B. Fuzzy identity-based encryption[C]//Proc of the 24th Annual Int Conf on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2005: 457−473

[7] Goyal V, Pandey O, Sahai A, et al. Attribute-based encryption for fine-grained access control of encrypted data[C]//Proc of the 13th ACM Conf on Computer and Communications Security. New York: ACM, 2006: 89−98

[8] Wei Pengcheng, Wang Dahu, Zhao Yu, et al. Blockchain data-based cloud data integrity protection mechanism[J]. Future Generation Computer Systems, 2020, 102: 902−911

[9] Zikratov I, Kuzmin A, Akimenko V, et al. Ensuring data integrity using blockchain technology[C]//Proc of the 20th Conf of Open Innovations Association (FRUCT). Piscataway, NJ : IEEE, 2017: 534−539

[10] Chen Zening, Zhang Liang, Zhang Shuangjun, et al. Access control scheme on blockchain and decentralized attributed-based algorithm with identity[J]. SCIENTIA SINICA Informationis, 2021, 51(8): 1345−1359(in Chinese)
(陈泽宁, 张亮, 张双俊, 等. 基于区块链和去中心属性密码的访问控制身份方案[J]. 中国科学: 信息科学, 2021, 51(8): 1345−1359)

[11] Fang Ning, Liu Baixiang, Kan Haibin. Controllable anonymous authentication scheme based on blockchain and decentralized traceable attribute-based signature[J]. SCIENTIA SINICA Informationis, 2021, 51(10): 1706−1720(in Chinese)
(方宁, 刘百祥, 阚海斌. 基于区块链和去中心可追踪属性签名的可控匿名认证方案[J]. 中国科学: 信息科学, 2021, 51(10): 1706−1720)

[12] Yuan Hexin, Liu Baixiang, Kan Haibin, et al. Distributed public key infrastructure scheme based on blockchain and decentralized undeniable attribute-based signature[J]. SCIENTIA SINICA Informationis, 2022, 52((6):): 1135−1148(in Chinese)
(袁和昕, 刘百祥, 阚海斌, 等. 基于区块链和去中心不可否认属性签名的分布式公钥基础设施方案[J]. 中国科学: 信息科学, 2022, 52(6): 1135−1148)

[13] Banerjee S, Bera B, Das A K, et al. Private blockchain-envisioned multi-authority CP-ABE-based user access control scheme in IIoT[J]. Computer Communications, 2021, 169: 99−113

[14] Katz J, Sahai A, Waters B. Predicate encryption supporting disjunctions, polynomial equations, and inner products[C]//Proc of the 27th Annual Int Conf on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2008: 146−162

[15] Lai Junzuo, Deng R H, Li Yingjun. Expressive CP-ABE with partially hidden access structures[C]//Proc of the 7th ACM Symp on Information, Computer and Communications Security. New York: ACM, 2012: 18−19

[16] Zhang Yinghui, Zheng Dong, Deng R H. Security and privacy in smart health: Efficient policy-hiding attribute-based access control[J]. IEEE Internet of Things Journal, 2018, 5(3): 2130−2145

[17] Li Qi, Zhang Yinghui, Zhang Tao, et al. HTAC: Fine-grained policy-hiding and traceable access control in mHealth[J]. IEEE Access, 2020, 8: 123430−123439

[18] Boneh D, Goh E J, Nissim K. Evaluating 2-DNF formulas on ciphertexts[C]//Proc of the 2nd Theory of Cryptography Conf. Berlin:

Springer, 2005: 325−341

[19] Beimel A. Secure schemes for secret sharing and key distribution[D/OL].1996[2022-11-06].https://www.cs.bgu.ac.il/~beimel/Papers/thesis.pdf

[20] Shamir A. How to share a secret[J]. Communications of the ACM, 1979, 22(11): 612−613

[21] Blakley G R. Safeguarding cryptographic keys[C]//Proc of 1979 Int Workshop on Managing Requirements Knowledge. Los Alamitos, CA: IEEE Computer Society, 1979: 313−313

[22] Pedersen T P. A threshold cryptosystem without a trusted party[C]//Proc of the 10th Workshop on the Theory and Application of Cryptographic Techniques Brighton. Berlin: Springer, 1991: 522−526

[23] Phuong T V X, Yang G, Susilo W. Hidden ciphertext policy attribute-based encryption under standard assumptions[J]. IEEE Transactions on Information Forensics and Security, 2015, 11(1): 35−45

[24] Rouselakis Y, Waters B. Practical constructions and new proof methods for large universe attribute-based encryption[C]//Proc of the 2013 ACM SIGSAC Conf on Computer & Communications Security. New York: ACM, 2013: 463−474

Wang Jingyi, born in 1999. Master candidate. Her research interests include attribute-based cryptography, blockchain, and privacy protection.

王静怡, 1999 年生. 硕士研究生. 主要研究方向为属性基密码学、区块链、隐私保护.

Kan Haibin, born in 1971. PhD, professor, PhD supervisor. His main research interests include cryptography and information security, coding and information theory, algorithms and computational complexity, and blockchain.

阚海斌, 1971 年生. 博士, 教授, 博士生导师. 主要研究方向为密码学与信息安全、编码与信息论、算法与计算复杂性、区块链.