计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210800001

支持访问策略隐藏和密钥追踪的轻量级医疗数据共享方案

王梦宇[1] 殷新春[1,2] 宁建廷[3,4]
1 扬州大学信息工程学院 江苏 扬州 225127
2 扬州大学广陵学院 江苏 扬州 225128
3 福建师范大学计算机与网络空间安全学院 福州 350007
4 中国科学院信息安全国家重点实验室 北京 100093
(MZ120200892@yzu.edu.cn)

摘 要 在传统的密文策略属性基加密(Ciphertext-Policy Attribute-Based,CP-ABE)方案中,访问策略是显式存在的,这可能会泄露数据所有者的隐私,在医疗场景中会给数据所有者带来潜在的安全隐患,因此支持访问策略隐藏的方案被陆续提出。但是多数方案在实现解密测试的过程中需要生成冗余密文或密钥组件,增加了数据所有者的计算开销和数据用户的存储开销。同时,恶意用户可能会受利益驱使,泄露其解密密钥。为了解决以上问题,提出了一个支持访问策略隐藏和密钥追踪的轻量级医疗数据共享方案。首先,采用SGX(Software Guard Extensions)技术,预先将部分主密钥存放在Enclave中,便于准确且快速地计算出测试结果,避免生成冗余密文和密钥组件;然后,为了降低用户的计算开销,同时保证解密结果的正确性和完整性,采用可验证外包技术;最后,通过在数据用户的解密密钥中嵌入身份标识实现了密钥追踪。性能分析表明,该方案在功能和开销上都具备一定的优势,安全性分析证明了该方案在选择明文攻击下是安全的。

关键词:策略隐藏;解密测试;可验证外包;密钥追踪
中图法分类号 TP309

Lightweight Medical Data Sharing Scheme with Access Policy Hiding and Key Tracking

WANG Meng-yu[1], YIN Xin-chun[1,2] and NING Jian-ting[3,4]
1 College of Information Engineering, Yangzhou University, Yangzhou, Jiangsu 225127, China
2 Guangling College of Yangzhou University, Yangzhou, Jiangsu 225128, China
3 College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350007, China
4 State Key Laboratory of Information Security, Chinese Academy of Sciences, Beijing 100093, China

Abstract In the traditional ciphertext-policy attribute-based encryption (CP-ABE) scheme, the access policy exists together with the ciphertext. This may leak the privacy of the data owner and bring potential security risks to the data owner in medical scenarios Therefore, solutions supporting access policy hiding have been proposed. However, most solutions need to generate redundant ciphertexts or key components in the process of implementing the decryption test, which increases the computing overhead of data owners and the storage overhead of data users. At the same time, malicious users may be motivated by its own interest to reveal their decryption keys. In order to solve the problems above, a lightweight medical data sharing scheme with access policy hiding and key tracking is proposed. Firstly, part of the master key is stored in the Enclave in advance by using software guard extensions(SGX) technology, so that the test results can be calculated accurately and quickly, and the generation of redundant ciphertexts and key components are avoided. Then, verifiable outsourcing technology is employed to reduce user's computing overhead, ensuring the correctness and completeness of decryption result. Finally, key tracking is realized by embedding the identity identifier in the decryption key of the data user. Performance analysis shows that the proposed scheme has certain advantages in terms of function and computing. The security analysis proves that the proposed scheme is secure under the selected plaintext attack.

Keywords Strategy hiding, Decryption test, Verifiable outsourcing, Key tracking

1 引言

近年来,随着物联网和云计算技术的快速发展,智能医疗云(Smart Medical Cloud,SMC)系统的建设也随之完善[1-2]。基于云的存储系统比传统存储系统具有更多优势,可使患者更快捷地存储和维护数据,享受云计算带来的高品质数据

到稿日期:2021-07-30 返修日期:2021-09-04
基金项目:国家自然科学基金(61972094)
This work was supported by the National Natural Science Foundation of China(61972094).
通信作者:殷新春(xcyin@yzu.edu.cn)

存储服务[3]。在SMC系统中,病人可以实时上传个人健康数据并将其分享给相应的医生,医生收到数据后通过远程访问对病人进行诊断。相比传统的就医模式,病人不用在各个窗口、科室之间来回奔波,节省了大量的时间和精力。虽然SMC系统的出现给人们带来了极大的便利,但由于健康数据中可能包含病人的个人隐私,一旦泄露会给病人带来极大的安全隐患,因此病人需要先将数据加密,再将加密后的数据上传到SMC系统。在传统的公钥加密中,加密方使用解密方的公钥进行加密,然后将密文发送给解密方,解密方用自己的私钥进行解密。然而,在SMC系统中,一个病人的数据需要共享给多位医生,上述一对一的加密模式会导致病人的存储和计算开销过大,因此并不适合。此外,病人往往还要求医生所属科室与病人患病类型匹配。如果采用传统的公钥加密方案,病人需要先确定医生的具体身份,再根据每个医生的公钥一一加密数据,最后将密文发送给相应的医生。但在现实情况中,病人往往对医生的情况知之甚少,仅仅掌握一些与个人病情相关的信息,很难直接确定合适的医生人选。因此,传统的公钥加密方案并不适用于SMC系统。

属性基加密(Attribute-Based Encryption,ABE)[4]可以实现细粒度的访问控制。ABE分为密钥策略属性基加密(Key-Policy Attribute-Based Encryption,KP-ABE)[5]和密文策略属性基加密(Ciphertext-Policy Attribute-Based Encryption,CP-ABE)[6]。KP-ABE将访问策略嵌入密钥中,将属性集嵌入密文中,只有当密文的属性集合满足密钥的访问策略时才能解密。而CP-ABE则将访问策略嵌入密文中,将属性集嵌入密钥中,只有当密钥中的属性集合满足密文中的访问策略时,用户才能解密该密文。因此,CP-ABE更加适用于SMC。在基于CP-ABE的SMC中,数据所有者可以通过指定访问策略,来限制只有拥有特定属性的数据用户才可以访问其个人数据。考虑到传统的CP-ABE[6-8]方案中访问策略是公开的,如果其中包含一些敏感信息,则可能会给病人带来安全隐患。比如,某个病人设定其数据只有肿瘤专家才能访问,从中可以推断出这个病人可能患有肿瘤疾病。此外,合法数据用户可能会受利益驱使将密钥泄露给他人,从而导致数据所有者的信息被泄露。因此,密钥泄露也是一个需要解决的问题。

针对访问策略可能会泄露数据所有者敏感信息的问题,Nishide等[9]首先提出访问策略隐藏的概念,并设计了一个支持部分访问策略隐藏的属性基加密方案,但该方案只支持"与"门的访问结构,并且计算开销较大。Lai等[10]提出了一个访问策略隐藏的属性基加密方案,并证明了该方案是完全安全的,但该方案是基于合数阶群的,运算效率不高。Cui等[11]提出了基于素数阶群的访问策略隐藏方案,但该方案并不支持解密测试,解密效率较低,而且存储开销较大。Ning等[12]提出了双访问控制的属性基加密方案,他们在方案中利用SGX技术测试数据用户的解密能力,降低了数据所有者的计算开销。但是,该方案不支持访问策略隐藏,而且数据用户需要进行与解密测试等价的计算开销,因此效率不高。Hu等[1]提出了"测试-解密-验证"的智能医疗策略隐藏的方案,该方案基于素数阶群,支持解密测试。同时,Hu等[1]考虑到数据用户计算开销的问题,加入了外包解密技术,并且支持解密验证。但该方案在测试密钥中未包含数据用户身份标识参数,因此无法抵抗共谋攻击。比如,A用户的测试密钥对应的属性集是("市医院""男""主任""肿瘤科"),B用户的测试密钥对应的属性集是("县医院""女""副主任""骨科")。如果A和B共谋,两者就可以共享彼此的任意属性密钥。具体来说,A和B通过共享测试密钥就可以测试出具备访问策略("市医院""女""副主任""肿瘤科")的密文,从而威胁病人隐私安全。此外,以上方案均不支持恶意用户的追踪。基于区块链的数据共享方案[13-14]虽然具备可追溯性,但存储能力较低。Ning等[15]提出了白盒可追踪属性基加密方案,将数据用户的身份标识嵌入密钥中,通过密钥即可追踪恶意用户的身份,但该方案并不具备访问策略隐藏的功能。最近,Zeng等[16]结合医疗云的应用场景,提出了支持大属性以及访问策略隐藏的属性基加密方案。该方案中属性空间不受限制,同时可以追踪泄露密钥的恶意用户,由于该方案基于合数阶群,计算效率不高。而且,Ning等[15]和Zeng等[16]的方案均不支持外包解密功能。

为了解决上述问题,本文在Hu等[1]提出的方案的基础上,提出了一个支持访问策略隐藏和密钥追踪的轻量级医疗数据共享方案。该方案基于素数阶群,采用线性秘密共享(Linear Secret Sharing Scheme,LSSS)技术和SGX技术,同时支持可验证外包计算、用户追踪的功能,在降低存储开销和计算开销的同时,保证了数据的安全性。本文的主要贡献如下:

(1)在解密测试方面,采用Ning等[12]提出的方案使用的SGX技术,创建了一个叫做Enclave[17]的隔离执行环境,将系统主密钥和执行解密计算的代码预储在其中。医疗云服务器只需使用数据用户的测试密钥即可测试解密能力,因此避免了冗余密文和密钥的生成,降低了数据所有者的加密开销和数据用户的存储开销。其中,测试密钥是经过数据用户私钥盲化的,具备解密测试和防共谋的特性,但不能完全解密密文。

(2)在降低计算开销方面,本文方案在完成解密测试后,数据用户不用再次执行与解密测试等价的计算,只需要在解密测试的基础上继续解密即可。因此,本文主要考虑解密测试计算的开销。本文采用外包解密技术,将解密测试计算交给医疗云服务器,通过测试的数据用户只需要进行恒定的常数级运算即可解密,解密计算高效快捷。此外,考虑到医疗云服务器是"隐蔽性"的,由于存储系统损坏或者其他原因,在医疗云服务器上的数据可能丢失或者损坏,医疗云服务器可能会伪造密文,或者直接发送错误的解密结果给用户,以节省计算开销。因此,本文增加了对外包计算结果的验证,将数据所有者的私钥、消息和对称密钥嵌入验证参数中。数据用户可以通过验证算法来验证解密结果的正确性和完整性。

(3)考虑到密钥泄露的问题,本文方案通过白盒可追踪技术来实现密钥追踪,将数据用户的身份嵌入解密密钥中,通过检查数据用户的解密密钥来追溯恶意用户的身份,以保障数据的安全性。

2 预备知识

2.1 符号介绍

本文用到的一些符号及定义如表1所列。

表1 相关符号及定义

Table 1 Related notation and their definition

符号	定义
p	大素数
G,G_T	p 阶循环群
g	群 G 的生成元
e	双线性映射
Z_p	不大于 p 的整数域
$\Lambda=((\boldsymbol{A},\rho),T)$	访问策略
$(\boldsymbol{A},\rho)$	访问结构($\boldsymbol{A}$ 为访问矩阵，ρ 为矩阵映射)
$T=(\pi_{\rho(1)},\pi_{\rho(2)},\cdots,\pi_{\rho(m)})$	Λ 中的属性值集合
$\vec{\boldsymbol{v}}=(s,v_2,v_3,\cdots,v_n)^{\top}$	随机列向量
$U=\{att_1,att_2,\cdots,att_n\}$	属性域(att_x 为属性)
$ATT_x=\{S_{x,1},S_{x,2},\cdots,S_{x,n_x}\}$	att_x 的候选值集合
$L=(I_L,V_L)$	用户属性列表(I_L 为属性名索引集，V_L 为属性值集)
$I=\{x\|\rho(x)\in IL\}$	解密索引集

2.2 访问结构

设$\{P_1,P_2,\cdots,P_n\}$是由 n 个参与者组成的实体集，对于集合 $A\subseteq 2^{\{P_1,P_2,\cdots,P_n\}}$，如果 $\forall B,C,B\in A,B\subseteq C$，有 $C\in A$，那么称 A 是单调的。如果集合 A 是$\{P_1,P_2,\cdots,P_n\}$的非空子集，即 $A\subseteq 2^{\{P_1,P_2,\cdots,P_n\}}\backslash\{\varnothing\}$，那么 A 就是一个访问结构，所有包含在 A 中的集合称为授权集合，不包含在 A 中的集合就称为非授权集合。

2.3 双线性映射

令 G 和 G_T 为两个阶为素数 p 的循环群，g 为群 G 的生成元，$e:G\times G\rightarrow G_T$ 为双线性映射，其中双线性映射 e 满足如下条件：

(1)双线性。对于 $\forall u,v\in G,a,b\in Z_P$，有

$e(u^a,v^b)=e(u,v)^{ab}$。

(2)非退化性。$e(g,g)\neq 1$。

如果群 G 上的计算和双线性映射 $e:G\times G\rightarrow G_T$ 可以有效地执行，那么称群 G 是一个双线性群。可注意到映射 e 是对称的，因为 $e(g^a,g^b)=e(g,g)^{ab}=e(g^b,g^a)$。

2.4 线性秘密共享

秘密共享的提出有效地降低了秘密传输和保存的风险[18]。目前用于秘密共享的主流方案有基于访问控制树和秘密共享矩阵，本文方案采用效率较高的秘密共享矩阵。

令$(\boldsymbol{A},\rho)$为访问结构，其中 $\boldsymbol{A}$ 是一个在 Z_P 中大小为 $m\times n$ 的访问控制矩阵，ρ 是一个映射，将矩阵 $\boldsymbol{A}$ 的每一行映射到一个属性名索引。线性秘密共享方案由以下两种算法组成。

(1)秘密分享。当 $\boldsymbol{A}$ 共享一个秘密值 $s\in Z_P$ 时，首先设置 n 维列向量$\vec{\boldsymbol{v}}=(s,v_2,v_3,\cdots,v_n)^{\top}$，其中 $v_2,v_3,\cdots,v_n$ 是 Z_P 中的随机数。该算法中，$\boldsymbol{A}_y$ 表示 $\boldsymbol{A}$ 的第 y 行，λ_y 表示属性名索引 $\rho(y)$所持有的共享秘密值。

(2)秘密值重构。该算法可以重构出秘密值 s。令 P 是任意授权集合，$I=\{y|\rho(y)\in P\}\subseteq\{1,2,\cdots,m\}$，存在常数集 $\{\omega_y\in Z_P\}_{y\in I}$，使得 $\sum_{y\in I}\omega_y\boldsymbol{A}_y=(1,0,\cdots,0)$，因此有 $\sum_{y\in I}\omega_y\lambda_y=s$。

本文指出，如果 $\boldsymbol{A}'$是未经授权的集合，并且 $I'=\{y|\rho(y)\in\boldsymbol{A}'\}$，则存在系数元组 $\{\omega'_y\}_{y\in I'}$ 且 ω'_1 为非零元素，使得方程 $\sum_{y\in I'}\omega'_y\lambda_y=0$。如果集合 $\{1,2,\cdots,m\}$的一个子集 I 满足$(\boldsymbol{A},\rho)$，并且任意 $I'\subset I$ 都不满足$(\boldsymbol{A},\rho)$，那么称 I 为$(\boldsymbol{A},\rho)$的最小授权集。

令 $U=\{att_1,att_2,\cdots,att_n\}$作为属性空间，其中每个属性 att_x 有 n_x 个候选值。$ATT_x=\{S_{x,1},S_{x,2},\cdots,S_{x,n_x}\}$是第 x 个属性所有候选值的集合。访问策略表示为 $\Lambda=((\boldsymbol{A},\rho),T)$，其中 $T=(\pi_{\rho(1)},\pi_{\rho(2)},\cdots,\pi_{\rho(m)})$是与$(\boldsymbol{A},\rho)$相关的属性值集合且$(\pi_{\rho(x)}\in ATT_{\rho(x)})$。此外，$L=(I_L,V_L)$表示数据用户的属性集，其中 $I_L\subseteq Z_P$ 表示属性名索引集，$V_L=\{v_y\}_{y\in I_L}$ 表示对应的属性值集。如果存在$\{H(v_{\rho(x)})=H(\pi_{\rho(x)})\}_{x\in I}$，其中 $I=\{x|\rho(x)\in IL\}$，则称属性集满足访问策略。

2.5 困难性假设

(1)判定性$(q-1)$BDHE[19]假设

随机选取 $g\in G,a,s,b_1,b_2,\cdots,b_q\in Z_P$，输出如下元组：

g,g^s，

$g^{a^i},g^{b_j},g^{sb_j},g^{a^ib_j},g^{a^i/b_j^2},\forall(i,j)\in[q,q]$，

$g^{a^i/b_j},\forall(i,j)\in[2q,q]$，其中 $i\neq q+1$，

$g^{a^ib_j/b_k^2},\forall(i,j,k)\in[2q,q,q]$，其中 $j\neq k$，

$g^{sa^ib_j/b_k},g^{sa^ib_j/b_k^2},\forall(i,j,k)\in[q,q,q]$，其中 $j\neq k$。

对于任意多项式时间攻击者来说，区分 $e(g,g)^{sa^{q+1}}\in G_T$ 和一个从 G_T 中随机选取的元素是困难的。

(2)判定线性(Decisional Linear，DL)[20]假设

随机选择 $a_1,a_2\in Z_P,g,g_1,g_2$ 是 G 的生成元，$R\in G$。DL 假设指任意多项式时间算法都无法以不可忽略的优势将元组$(g,g_1,g_2,g^{a_1},g_1^{a_2},g_2^{a_1+a_2})$和$(g,g_1,g_2,g^{a_1},g_1^{a_2},R)$区分开来。

2.6 Intel SGX

Intel SGX(Intel Software Guard Extension)是在原有Intel架构上扩展的一组新的指令集和内存访问机制[21]，允许应用程序创建一个叫作 Enclave 的隔离执行环境。Enclave 作为一个可信和安全的实体，用于存储数据和执行代码。Enclave 具有 3 个安全特性：隔离、密封和认证。隔离限制了对硬件保护的内存区域的访问，只有特定的 Enclave 可以访问它。同一处理器上的任何其他进程，甚至是操作系统、管理程序等其他外部实体，都不能访问该内存。密封提供了一个将 Enclave 秘密加密到磁盘上的持久存储的方法，以实现即使 Enclave 被拆毁也能检索秘密。加密是使用特定 Enclave 私有的密封密钥执行的，除了完全相同的 Enclave 之外，没有任何进程可以对其进行解密或修改。认证可以使验证者验证代码在 Enclave 内安全运行且不被修改的真伪。SGX 提供两种身份认证方式，分别是本地认证和远程认证[22]。本地认证用于同一平台上的两个 Enclave 之间的认证，同一台机器上的两个 Enclave 可以使用它们之间共享的根密封密钥派生一个共享密钥。远程认证使 Enclave 能够生成任何远程实体都可以验证的报告。

3 系统定义

3.1 系统模型

本文方案的系统模型如图 1 所示，该方案由 5 个实体组成：授权中心(Authorization Center，AC)、智能医疗云服务器(Smart Medical Cloud，SMC)、数据所有者(Data Owner，DO)、数据使用者(Data User，DU)、飞地(Enclave)。每个实体在这个系统中负责不同的任务，具体如下。

(1)授权中心(AC)。授权中心是完全可信的，负责初始化系统公共参数、数据所有者的注册以及追踪恶意用户。

(2)智能医疗云服务器(SMC)。智能医疗云服务器是

"隐蔽性"的,负责处理数据用户的下载请求和测试数据用户的解密能力。但是,它可能会任意偏离协议规范以试图作弊。由于存储系统损坏或者其他原因,医疗云服务器上的数据可能丢失或者损坏,医疗云服务器可能会伪造密文。此外,也可能直接发送错误的解密结果给用户,以减小计算开销。

(3)数据所有者(DO)。数据所有者是诚实的,负责加密将要外包的数据,并将加密后的数据上传到医疗云服务器。

(4)数据用户(DU)。数据用户是不可信的,被授权的数据用户可以申请下载并解密共享数据,以验证外包解密的正确性;未授权的数据用户可能与其他数据用户串通,以获取自身能力之外的数据,同时可能发动 DDoS 攻击,引起资源消耗。

(5)飞地(Enclave)。Enclave 是完全可信的,是通过英特尔软件保护扩展(Intel SGX)技术在计算机内存中创建的一个隔离的环境,即所谓的"飞地"。即使在不可信的主机上,它也可以安全地执行程序和保存数据。在本方案中,Enclave 被用来存储系统主密钥和执行测试程序。

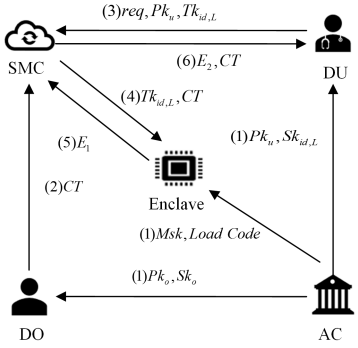

图 1 系统模型图

Fig. 1 System model diagram

3.2 算法定义

一个支持访问策略隐藏和密钥追踪的轻量级医疗数据共享方案由以下 9 种算法组成。

(1)系统初始化算法

$Setup(\lambda)\rightarrow\{Pk, Msk, Ul\}$:该算法由 AC 执行,输入安全参数 λ,输出公共参数 Pk 和系统主密钥 Msk,初始化一个空的数据用户列表 Ul。

(2)密钥生成算法

$KeyGen(Pk, Msk, L, id)\rightarrow\{Pk_o, Sk_o, Pk_u, SK_{id,L}\}$:该算法由 AC 执行,输入公共参数 Pk、主密钥 Msk、DU 的属性集合 L 和身份 id,输出 DO 的公私钥对(Pk_o,Sk_o)、DU 的公钥 Pk_u 和解密密钥 $Sk_{id,L}$。

(3)加密算法

$Encrypt(Pk, M, ((\boldsymbol{A}, \rho), T))\rightarrow CT$:该算法由 DO 执行,输入公共参数 Pk、消息 M 和访问策略$((\boldsymbol{A}, \rho), T)$,输出密文 CT。

(4)请求算法

$Request(Pk, Sk_u)\rightarrow req$:该算法由 DU 执行,输入公共参数 Pk 和 DU 的私钥 Sk_u,输出下载请求 req。

(5)测试算法

$Test(Pk, req, CT, Tk_{id,L})\rightarrow E_2$ or $\perp$:该算法由 SMC 执行,输入公共参数 Pk、下载请求 req、密文 CT 和测试密钥 $Tk_{id,L}$。如果 DU 请求通过,则输出测试计算结果 E_2;否则输出 $\perp$。

(6)解密算法

$Decrypt(Pk, E_2, CT, Sk_{id,L})\rightarrow M \| \overset{\vee}{c}$:该算法由 DU 执行,输入公共参数 Pk、测试计算结果 E_2、DU 的解密密钥 $Sk_{id,L}$ 和密文 CT,输出消息和解密验证参数 $M \| \overset{\vee}{c}$。

(7)验证算法

$Verify(M \| \overset{\vee}{c}, EK, CT, Pk_o)\rightarrow 1$ or 0:该算法由 DU 执行,输入消息和解密验证参数 $M \| \overset{\vee}{c}$、对称密钥 EK、密文 CT 和 DO 的公钥 Pk_o。如果解密结果通过验证,则输出 1;否则输出 0。

(8)密钥检查算法

$KeySanityCheck(Pk, Sk_{id,L})\rightarrow 1\ or\ 0$:该算法由 AC 执行,输入公共参数 Pk 和 DU 的解密密钥 $Sk_{id,L}$。如果用户的解密秘钥 $Sk_{id,L}$ 通过密钥完整性检查,则算法输出 1;否则输出 0。

(9)密钥追踪算法

$Trace(Sk_{id,L}, Ul)\rightarrow id\ or\ \perp$:该算法由 AC 执行,输入 DU 的解密密钥 $Sk_{id,L}$ 和 DU 的身份列表 Ul,如果 $KeySanityCheck(Pk, Sk_{id,L})\rightarrow 1$,则输出 DU 的身份 id;否则输出 $\perp$。

3.3 安全模型

本文方案基于选择明文安全模型,通过攻击者 A 和挑战者 B 之间的安全游戏来定义安全模型,具体步骤如下:

初始化阶段:攻击者 A 将挑战访问策略 $\Lambda^*=((\boldsymbol{A}^*, \rho^*), \mathcal{T}^*)$ 发送给挑战者 B。

系统建立阶段:挑战者 B 先运行 $Setup$ 算法,输入安全参数 λ,然后将公共参数 Pk 发送给攻击者 A 并保存主密钥 Msk。

查询阶段 1:攻击者 A 自适应地发起关于(id_1, L_1),(id_2, L_2),…,(id_q, L_q)的密钥请求。挑战者 B 运行 $KeyGen$ 算法,将生成的密钥$\{Sk_{id,L_i}, id_i\}_{i\in[1,q]}$发送给攻击者 A。其中,规定 Sk_{id,L_i} 不满足访问策略 Λ^*。

挑战阶段:攻击者 A 向挑战者 B 提交两个等长的明文消息 M_0 和 M_1。挑战者 B 随机投掷一枚硬币 $\sigma\in\{0,1\}$,选取 M_σ,运行 $Encrypt(Pk, M_\sigma, \Lambda^*)$算法生成挑战密文 CT,将 CT 发送给攻击者 A。

查询阶段 2:与查询阶段 1 相同。攻击者 A 请求属性集 $L_{q+1}, L_{q+2}, \cdots, L_Q$ 的解密密钥。但是,它们都不能满足初始化阶段定义的挑战访问策略。

猜测阶段:攻击者 A 输出猜测 $\sigma'\in\{0,1\}$。如果 $\sigma'=\sigma$,则攻击者 A 将赢得游戏。

攻击者 A 赢得这个游戏的优势定义为 $\varepsilon=|\Pr[\sigma=\sigma']-\frac{1}{2}|$。

定义 1 若没有 PPT 攻击者能够以不可忽略的优势 ε 打破上述安全性游戏,则说明本文方案在选择明文攻击下是安全的。

4 方案设计

本节提出了一个支持访问策略隐藏和密钥追踪的轻量级医疗数据共享方案。该方案采用 SGX 技术,云服务器可以与 Enclave 交互执行解密测试,采用可验证外包技术,在减小

用户计算开销的同时保证了数据的正确性和完整性。通过密钥追踪机制，追踪泄露密钥的恶意用户。

(1) $Setup(\lambda)$：系统的初始化算法由 AC 执行，该算法随机选择 $\theta,\beta,a,\alpha\in Z_P$，$g,g_1,g_2,u,h,f\in G$，选择一个散列函数 $H:\{0,1\}^*\rightarrow Z_P$。$G$ 和 G_T 是阶为素数 p 的循环群，e 是一个双线性映射。然后，算法初始化一个空的数据用户列表 Ul。该算法输出公共参数 Pk 和主密钥 Msk，具体如下：

$$Pk=\{g,g^a,g^a,g_2,g_2^\beta,u,h,f,e(g,g_1)^\theta,H\}$$

$$Msk=\{\theta,\beta,a,g_1,\alpha\}$$

(2) $KeyGen(Pk,Msk,L,id)$：密钥生成算法由 AC 执行，该算法随机选取 $c,r,\gamma\in Z_P^*$ 和 $\{\delta,\delta_1,\delta_2,\cdots,\delta_{|I_L|}\}\in Z_P$，$L=(I_L,V_L)$ 是属性集。该算法输出 DO 的公私钥对 (Pk_o,Sk_o)、DU 的公钥 Pk_u 和解密密钥 $Sk_{id,L}$，具体如下：

$$Pk_o=g^\gamma,Sk_o=\gamma,Pk_u=g^r,Sk_u=r$$

$$K=g_1^{\frac{\theta r}{a+c}}g^{\alpha\delta r},K_0=g^{\delta r},K_0'=g^{a\delta r},K_1=c$$

$$\{K_{1,x}=g^{\frac{\delta_x r}{\beta}},K_{2,x}=(u^{H(V_x)}h)^{\frac{\delta_x r}{\beta}}f^{(a+c)\delta r},K_{3,x}=(u^{H(V_x)}h)^{\delta_x r}f^{(a+c)\beta\delta r}\}_{x\in[|I_L|]}$$

$$Tk_{id,L}=(K,K_0,K_0',K_1,\{K_{1,x},K_{2,x},K_{3,x}\}_{x\in[|I_L|]})$$

$$Sk_{id,L}=(Sk_u,Tk_{id,L})$$

最后，该算法将元组 (K_1,id) 加入数据用户列表 Ul 中。

(3) $Encrypt(Pk,M,((\boldsymbol{A},\rho),T))$：加密算法由 DO 执行，输入系统公开参数 Pk、一个明文消息 M 和一个访问策略 $((\boldsymbol{A},\rho),T)$。其中，$\boldsymbol{A}$ 是一个 $m\times n$ 大小的矩阵，ρ 是一个把 $\boldsymbol{A}$ 的每一行 $\boldsymbol{A}_y$ 映射到一个属性 $\rho_{(y)}$ 的映射，T 是一个属性值列表，$T=(\pi_{\rho(1)},\pi_{\rho(2)},\cdots,\pi_{\rho(m)})\in Z_P^m$，其中 $(\pi_{\rho(x)}\in ATT_{\rho(x)})$。该算法随机选取 $\vec{v}=(s,v_2,\cdots,v_n)^\top\in Z_P^n$，其中 s 是用于分享的随机秘密值，计算 $\lambda_y=\boldsymbol{A}_y\cdot\vec{v}$。该算法随机选取一个对称密钥 EK，利用对称加密算法 SE 加密明文 M，并计算 DO 的一对验证参数 $(\hat{c},\overset{\vee}{c})$，计算式如下：

$$\hat{c}=e(g,g)^{\frac{1}{H(M\|EK)+\gamma}},\overset{\vee}{c}=g^{\frac{1}{H(M\|EK)+\gamma}}$$

$$C=SE.Enc(EK,M\|\overset{\vee}{c}),C_0=EK\cdot e(g,g_1)^{\theta s}$$

$$C_1=g^s,C_1'=g^{\alpha s}$$

随机选取 $z_y,k_y\in Z_P$，$1\leqslant y\leqslant m$，计算属性密文组件：

$$C_{0,y}=g_2^{z_y},C_{1,y}=g_2^{\beta z_y}g^{k_y}$$

$$C_{2,y}=g^{a\lambda_y}f^{k_y},C_{3,y}=(u^{H(\pi_{\rho(y)})}h)^{k_y}$$

该算法的输出密文为 $CT=\{C,C_0,C_1,C_{0,i},C_{1,i},C_{2,i},C_{3,i},\hat{c}\}$。

(4) $Request(Pk,Sk_u)$：请求算法由 DU 执行，该算法随机选择 $t\in Z_p$，计算 g^t 和身份验证参数 y 和 π，计算式如下：

$$y=e(g,g)^{\frac{1}{t+r}},\pi=g^{\frac{1}{t+r}}$$

算法输出请求 $req=(Pk_u,y,\pi,g^t)$。

(5) $Test(Pk,req,CT,Tk_{id,L})$：外包测试算法由 SMC 执行。该算法首先验证 DU 的身份，然后验证其解密能力。

第一步　验证 DU 的身份，防止系统外的用户访问，造成了大量的资源消耗。验证过程如下：

$$y=e(g,\pi),e(g^t\cdot Pk_u,\pi)=e(g,g)$$

如果等式成立，则证明 DU 是系统用户，进入解密能力测试。

第二步　解密测试。SMC 首先将 $Tk_{id,L}$ 中的 K_0,K_0'，K_1 和 CT 中的 C_1 发送给 Enclave，Enclave 调用系统主密钥中的 a 并执行计算：$E_1=e((K_0^{K_1}\cdot K_0')^a,C_1)=e(g,g)^{a(a+c)\delta sr}$，然后发送 E_1 给 SMC。SMC 的计算式如下：

$$E_2=\sum_{y\in I}\left(\frac{e(C_{0,y},K_{3,\rho(y)})e(C_{2,y},K_0^{K_1}\cdot K_0')e(C_{3,y},K_{1,\rho(y)})}{e(C_{1,y},K_{2,\rho(y)})}\right)^{\omega_y}=\sum_{y\in I}\frac{e(g_2^{z_y},(u^{H(V_{\rho(y)})}h)^{\delta_x r}f^{(a+c)\beta\delta r})^{\omega_y}}{e(g_2^{\beta z_y}g^{k_y},(u^{H(V_{\rho(y)})}h)^{\frac{\delta_x r}{\beta}}f^{(a+c)\delta r})^{\omega_y}}$$

$$=\sum_{y\in I}(e(g^{a\lambda_y},g^{(a+c)\delta r}))^{\omega_y}=e(g,g)^{a(a+c)\delta sr}$$

SMC 验证 $E_1=E_2$，如果等式成立，则说明 DU 具备解密能力，该算法输出 E_2；反之，输出 $\perp$。

(6) $Decrypt(Pk,E_2,CT,Sk_{id,L})$：解密算法由 DU 执行。如果 DU 通过解密测试，则 SMC 将测试结果 E_2 发送给数据用户，DU 执行最后的解密，计算式如下：

$$F=\frac{e((C_1^{K_1}\cdot C_1'),K)}{E_2}=\frac{e(g^{cs}g^{\alpha s},g_1^{\frac{\theta r}{a+c}}g^{a\delta r})}{e(g,g)^{a(a+c)\delta r}}=e(g,g_1)^{\theta sr}$$

$$EK=\frac{C_0}{F^{\frac{1}{r}}}=\frac{EK\cdot e(g,g_1)^{\theta s}}{(e(g,g_1)^{\theta sr})^{\frac{1}{r}}}=\frac{EK\cdot e(g,g_1)^{\theta s}}{e(g,g_1)^{\theta s}}$$

$$M\|\overset{\vee}{c}=SE.Dec(C,EK)$$

该算法输出 $M\|\overset{\vee}{c}$，然后提取消息 M。

(7) $Verify(M\|\overset{\vee}{c},EK,CT,Pk_o)$：验证算法由 DU 执行，提取验证密钥 $(\hat{c},\overset{\vee}{c})$ 对，验证解密结果的正确性。验证过程如下：

$$\hat{c}=e(g,\overset{\vee}{c}),e(g^{H(M\|EK)}\cdot Pk_o,\overset{\vee}{c})=e(g,g)$$

如果上述等式成立，则说明解密结果正确，该算法输出 1；反之，输出 0。

(8) $KeySanityCheck(Pk,Sk_{id,L})$：密钥完整性算法由 AC 执行，该算法输入系统公共参数 Pk 和 DU 的解密密钥 $Sk_{id,L}$，首先检查 $Sk_{id,L}$ 是否具有 $\{Sk_u,K,K_0,K_0',K_1,K_{1,x},K_{2,x},K_{3,x}\}$ 这种形式，$K_{1,x},K_{2,x},K_{3,x}\in G$，$Sk_u\in Z_P$ 是否成立，验证以下等式：

$$e(K_0,g^a)=e(K_0',g)$$

$$e(g^ag^{K_1},K)=e(K_0^{K_1}\cdot K_0',g^a)(e(g,g_1)^\theta)^{Sk_u}$$

$$e(K_{2,x},g_2^\beta)=e(K_{3,x},g_2)$$

如果 $Sk_{id,L}$ 通过密钥完整性检验，则该算法输出 1；否则输出 0。

(9) $Trace(Sk_{id,L},Ul)$：追踪算法由 AC 执行，如果算法 $KeySnaityCheck(Pk,Sk_{id,L})\rightarrow 0$，则算法输出 $\perp$；否则 $Sk_{id,L}$ 是一个完整的解密密钥，算法在用户列表 Ul 中搜索 K_1。如果 K_1 存在，则算法输出 DU 的身份 id；否则输出 $\perp$。

5　安全性分析

定理 1　假设 $(q-1)$BDHE 假设和 DL 假设都为真，则本文方案可以实现选择明文不可区分和策略隐藏。

本文方案将借鉴 Hu 等[1]的方案对 $(q-1)$BDHE 假设和 DL 假设给出证明。

证明：本文方案的证明是通过一系列游戏给出的，这些游戏在计算上是不可区分的。设置 $\{C^*,C_0^*,C_1^*,C_{0,y}^*,C_{1,y}^*,C_{2,y}^*,C_{3,y}^*,\hat{c}^*\}$ 作为挑战密文发送给游戏中的攻击者，随机

选择 $R\in G_T$，$\{R_{i,0},R_{i,1},R_{i,2}\}_{i\in[1,m]}\in G$，定义一个游戏集 $Game_r$，$Game_0$，$Game_1$，…，$Game_m$，$Game_{m+1}$，…，$Game_{2m}$，具体描述如下。

$Game_r$：这是一个原始游戏，也就是说这个游戏中的所有密文都正常。

$Game_0$：这是一个经过修改的游戏。挑战密文中 C_0^* 被 R 取代，并且 CT^* 中其余项仍是原始的。

$Game_{j'}$：为了方便，本文方案用 $Game_{j'}$ 代替 $Game_1$，…，$Game_m$，其中 $j'\in[1,m]$。在第 j' 次游戏 $Game_{j'}$ 中，C_0^* 被 R 代替，$C_{0,y}^*$ 和 $C_{1,y}^*$ 分别被替换为 $R_{j',0}$ 和 $R_{j',1}$，其中 $j'\in[1,m]$。总之，$Game_1$ 到 $Game_m$ 中的 $C_{0,y}^*$ 和 $C_{1,y}^*$ 分别被 $R_{j',0}$ 和 $R_{j',m}$ 代替。

$Game_{k'}$：$Game_{j'}$ 和 $Game_{k'}$ 的表达方式相同，即 $Game_{m+1}$，…，$Game_{2m}$ 被 $Game_{k'+m}$ 代替，其中 $k'\in[1,m]$。在第 k' 次游戏 $Game_{k'}$ 中，C_0^* 被 R 替换，然后 $C_{0,y}^*$ 和 $C_{1,y}^*$ 分别被 $R_{k',0}$ 和 $R_{k',1}$ 代替。

下文将详细展示这些游戏 $Game_r$ 到 $Game_{2m}$ 在计算上无法区分。

引理 1 假设 $(q-1)$BDHE 假设为真，对于任何 PPT 算法，都没有区分游戏 $Game_r$ 和 $Game_0$ 的优势。

证明：假设存在一个攻击者 A，他可以区分 $Game_r$ 和 $Game_0$。那么挑战者 B 可以通过攻击者 A 来解决具有不可忽略优势的 $(q-1)$BDHE 问题。

初始化阶段：攻击者 A 将访问策略 $\Lambda^*=((\boldsymbol{A}^*,\rho^*),\mathcal{T}^*)$ 发送给挑战者 B。

系统建立阶段：挑战者 B 随机选择 $a,d_1,d_2,\tilde{\beta},\tilde{f},\tilde{u},\tilde{h},\tilde{\theta}\in Z_P$，计算 $g_1=g^{d_1}$，$g_2=g^{d_2}$。然后，选择一个将属性值或群 G_T 映射到 Z_P 的哈希函数 H，初始化一个数据用户列表 $Ul=\emptyset$。此外，隐式设置 $\theta=a^{q+1}+\tilde{\theta}$，其余公共参数构造为：

$$g_2^{\beta}=g^{d_2\tilde{\beta}},u=g^{\tilde{u}}\cdot\prod_{(j,k)\in[m,n]}(g^{a^k/b_j^2})^{A_{j,k}^*}$$

$$h=g^{\tilde{h}}\cdot\prod_{(j,k)\in[m,n]}(g^{a^k/b_j^2})^{H(\pi_{\rho^*(j)})A_{j,k}^*}$$

$$f=g^{\tilde{f}}\cdot\prod_{(j,k)\in[m,n]}(g^{a^k/b_j})^{A_{j,k}^*}$$

$$e(g,g_1)^{\theta}=e(g,g_1)^{a^{q+1}}\cdot e(g,g_1)^{\tilde{\theta}}$$

查询阶段 1：攻击者 A 自适应地发起关于 (id,L) 的密钥请求。挑战者 B 运行 $KeyGen$ 算法，将生成的密钥 $\{Sk_{id,L},id\}$ 发送给攻击者 A。具体过程如下。

挑战者 B 随机选择 $r,c,\tilde{\delta}\in Z_P$ 并隐式设置以下值：

$$\delta=\tilde{\delta}+\omega_1a^q+\omega_2a^{q-1}+\cdots+\omega_na^{q+1-n}$$

$$=\tilde{\delta}+\sum_{i\in[n]}\omega_ia^{q+1-i}$$

挑战者 B 计算与秘密值 δ 相关联的密钥，具体如下：

$$\tilde{K}=g_1^{\theta/a+c}g^{a\delta}=(g^{d_1})^{(a^{q+1}+\tilde{\theta})/a+c}\cdot(g^a)^{\tilde{\delta}}\prod_{i\in[n]}(g^{a^{q+1-i}})^{\omega_i}$$

$$=g_1^{\tilde{\theta}/a+c}(g^a)^{\tilde{\delta}}\prod_{i=2}^{n}(g_1^{a^{q+2-i}})^{\omega_i/a+c}$$

$$\tilde{K}_0=g^{\delta}=g^{\tilde{\delta}}\prod_{i\in[n]}(g^{a^{q+1-i}})^{\omega_i}$$

$$\tilde{K}_0'=\tilde{K}_0^{\alpha},K_1=c,Sk_u=r$$

计算密钥中的公共部分：

$$f^{\delta}=f^{\tilde{\delta}}(g^{\tilde{f}}\prod_{(j,k)\in[m,n]}(g^{a^k/b_j})^{A_{j,k}^*})^{\sum_{i\in[n]}\omega_ia^{q+1-i}}$$

$$\Phi=f^{\tilde{\delta}}\prod_{i\in[n]}(g^{a^{q+1-i}})^{\tilde{f}\omega_i}\cdot\prod_{(i,j,k)\in[n,m,n],i\neq k}(g^{\frac{a^{q+1+k-i}}{b_j}})^{\omega_iA_{j,k}^*}$$

$$=f^{\tilde{\delta}}\prod_{i\in[n]}(g^{a^{q+1-i}})^{\tilde{f}\omega_i}\cdot\prod_{(i,j,k)\in[n,m,n]}g^{\omega_iA_{j,k}^*a^{q+1+k-i}/b_j}$$

$$=f^{\tilde{\delta}}\prod_{i\in[n]}(g^{a^{q+1-i}})^{\tilde{f}\omega_i}\cdot\prod_{(i,j,k)\in[n,m,n],i\neq k}(g^{\frac{a^{q+1+k-i}}{b_j}})^{\omega_iA_{j,k}^*}\cdot\prod_{j\in[m],\rho(j)^*\notin I_L}(g^{a^{q+1}})^{\langle\vec{\omega},A_j^*\rangle}$$

即 $f^{\delta}=\Phi\cdot\prod_{j\in[m],\rho_{(j)}^*\notin I_L}(g^{a^{q+1}})^{\langle\vec{\omega},A_j^*\rangle}$

对于每个属性索引 $x(x\in I_L)$，挑战者 B 选择 $\tilde{\delta}\in Z_P$ 并且隐式地设置如下参数：

$$\delta_x=\tilde{\delta}_x+\delta\cdot\sum_{i'\in[m],\rho^*(i')\notin I_L}\frac{b_i'}{H(V_x)-H(V_{\rho^*(i')})}$$

$$=\tilde{\delta}_x+\tilde{\delta}\sum_{i'\in[m],\rho^*(i')\notin I_L}\frac{b_i'}{H(V_x)-H(V_{\rho^*(i')})}+\sum_{i,i'\in[m,n],\rho^*(i')\notin I_L}\frac{\omega_ib_i'a^{q+1-i}}{H(V_x)-H(V_{\rho^*(i')})}$$

剩下的密钥可以通过秘密值 δ_x 来计算：

$$\tilde{K}_{1,x}=g^{\delta_x/\beta}$$

$$=g^{\tilde{\delta}_x/\beta}\cdot\prod_{i'\in[m],\rho^*(i')\in I_L}(g^{b_{i'}/\beta})^{\frac{\tilde{\delta}}{H(V_x)-H(V_{\rho^*(i')})}}\cdot\prod_{(i,i')\in[n,m],\rho^*(i')\in I_L}(g^{b_{i'}a^{q+1+k-i}/\beta})^{\frac{\omega_i}{H(V_x)-H(V_{\rho^*(i')})}}$$

对于密钥 $K_{2,x}$ 和 $K_{3,x}$ 的模拟，挑战者 B 只需要考虑公共部分 h^{δ_x} 和 u^{δ_x} 的有效性，具体构造如下：

$$(u^{H(V_x)}h)^{\delta_x}=(u^{H(V_x)}h)^{\tilde{\delta}_x}\cdot(\tilde{K}_{1,x}/g^{\tilde{\delta}_x})^{\tilde{u}H(V_x)+\tilde{h}}\cdot\prod_{\substack{(i',j,k)\in[m,m,n]\\ \rho^*(i')\notin I_L}}(g^{\frac{b_{i'}'a^k}{b_j^2}})^{\frac{\tilde{\delta}(H(V_x)-H(V_{\rho^*(j)}))A_{j,k}^*}{(H(V_x)-H(V_{\rho^*(i')}))}}\cdot\prod_{\substack{(i,i',j,k)\in[n,m,m,n]\\ \rho^*(i')\in I_L,(j\neq i,i\neq k)}}(g^{\frac{b_{i'}'a^{q+1+k-i}}{b_j^2}})^{\frac{(H(V_x)-H(V_{\rho^*(j)}))A_{j,k}^*\omega_i}{(H(V_x)-H(V_{\rho^*(i')}))}}$$

令 $\Psi=(u^{H(V_x)}h)^{\delta_x}$

$$\tilde{K}_{2,x}=\Psi^{\frac{1}{\beta}}\cdot\Phi^{a+c}\cdot\prod_{j\in[m],\rho_{(j)}^*\notin I_L}(g^{a^{q+1}})^{\langle\vec{\omega},A_j^*\rangle(a+c)}$$

$$\tilde{K}_{3,x}=\tilde{K}_{2,x}^{\beta}$$

随后，挑战者 B 计算：

$$SK_{id,L}=(Sk_u,K=\tilde{K}^r,K_0=\tilde{K}_0{}^r,K_0'=\tilde{K}_0'^r,K_1,\{K_{1,x}=\tilde{K}_{1,x}^r,K_{2,x}=\tilde{K}_{2,x}^r,K_{3,x}=\tilde{K}_{3,x}^r\}_{x\in[|I_L|]})$$

挑战者 B 将查询密钥 $(SK_{id,L},id)$ 发送给攻击者 A。随后，将元组 (K_1,id) 放入数据用户列表 Ul 中。

挑战阶段：攻击者 A 向挑战者 B 提交两个等长度的挑战明文消息 M_0 和 M_1。挑战者 B 随机投掷一枚硬币 $\sigma\in\{0,1\}$，选取 M_σ，运行 $Encrypt(Pk,M_\sigma,\Lambda^*)$ 算法生成密文 CT，将 CT 发送给攻击者 A。具体过程如下：

挑战者 B 选择 $\tilde{\gamma},d,s,\tilde{s}\in Z_p$，隐式设置 $s'=d\tilde{s}$，然后进行如下构造：

$$\hat{c}=(g,g)^{\frac{1}{H(M\|Ek)+\tilde{\gamma}}},\overset{\vee}{c}=g^{\frac{1}{H(M\|EK)+\tilde{\gamma}}}$$

$$C=SE.Enc(EK,M_\sigma\|\overset{\vee}{c}),$$

$$C_0=EK\cdot R\cdot e(g,g_1)^{\tilde{s}s},C_1=g^s,C_1'=g^{as}$$

挑战者 B 隐式设置列向量 $\vec{v}=(s,sa+\tilde{v}_2,\cdots,sa^{n-1}+$

$\tilde{v}_n)^{\top}$，其中 $\tilde{v}_i \in Z_p$，$i \in [2,n]$。因为 $\lambda_y = \boldsymbol{A}_y^* \cdot \vec{v}$，在不失一般性的情况下进行如下构造：

$$\lambda_y = \sum_{i\in[n]} \boldsymbol{A}_{y,i}^* s a^{i-1} + \sum_{i=2}^{n} \boldsymbol{A}_{y,i}^* \tilde{v}_i = \sum_{i\in[n]} \boldsymbol{A}_{y,i}^* s a^{i-1} + \tilde{\lambda}$$

此外，挑战者 B 还对访问矩阵的第 y 行隐式设置 $k_y = sb_y$，$z_y = s' b_y{}'$，其中 $\{b_y, b_y{}'\}_{y\in[1,m]} \in Z_P$。然后构造了挑战密文 CT^*。

$$C_{0,y} = g_2^{z_y} = (g^{s' d_2})^{b_y{}'}$$

$$C_{1,y} = (g_2^{\beta})^{z_y} g^{k_y} = (g^{s' d_2})^{\tilde{\beta} b_y{}'} (g^s)^{b_y}$$

$$C_{2,y} = (g^a)^{(\sum_{i\in[n]} A_{y,i}^* s a^{i-1} + \tilde{\lambda}_y)} (g^{\tilde{f}} \cdot \prod_{(j,k)\in[m,n]} g^{\frac{a^k A_{j,k}^*}{b_j}})^{sb_y}$$

$$= (g^a)^{\tilde{\lambda}_y} (g^{s'\tilde{f}})^{b_y{}'} \cdot \prod_{i\in[n]} g^{A_{y,i}^* s a^i} \cdot \prod_{k\in[n]} ((g^{sa^k/b_y})^{A_{y,k}^*}) b_y \cdot \prod_{\substack{(j,k)\in[m,n]\\ j\neq k}} (g^{sa^k A_{j,k}^*/b_j})^{b_y}$$

$$= (g^a)^{\tilde{\lambda}_y} (g^{s'\tilde{f}})^{b_y{}'} \cdot \prod_{i\in[n]} (g^{sa^i A_{y,i}^*})^2 \cdot \prod_{\substack{(j,k)\in[m,n]\\ j\neq k}} (g^{\frac{sa^k b_y A_{j,k}^*}{b_j}})$$

$$C_{3,y} = g^{(\tilde{u}+\tilde{h})sb_y} \cdot (\prod_{(j,k)\in[m,n]} (g^{H(\pi_{\rho^*(y)}) + H(\pi_{\rho^*(j)})})^{\frac{a^k A_{j,k}^*}{b_j^2}})^{sb_y}$$

$$= ((g^{s\tilde{u}})^{H(\pi_{\rho^*(y)})} g^{s\tilde{h}})^{b_y} \cdot D_7 \cdot D_8$$

这里：

$$D_7 = (\prod_{(j,k)\in[m,n]} (g^{a^k/b_j^2})^{A_{j,k}^*})^{H(\pi_{\rho^*(y)})k_y} = \prod_{\substack{(j,k)\in[m,n]\\ j\neq k}} (g^{\frac{sa^k b_y A_{j,k}^* H(\pi_{\rho^*(y)})}{b_j^2}} \cdot \prod_{k\in[n]} (g^{\frac{sa^k A_{y,k}^*}{b_y}}))^{H(\pi_{\rho^*(y)})^{b_y}}$$

$$D_8 = (\prod_{(j,k)\in[m,n]} (g^{a^k/b_j^2})^{H(\pi_{\rho^*(j)})A_{j,k}^*})^{k_y} = \prod_{\substack{(j,k)\in[m,n]\\ j\neq k}} (g^{\frac{sa^k b_y H(\pi_{\rho^*(j)}) A_{j,k}^*}{b_j^2}}) \cdot \prod_{k\in[n]} (g^{\frac{sa^k H(\pi_{\rho^*(y)})}{b_y}})^{A_{y,k}^* b_y}$$

查询阶段 2：重复查询阶段 1。需要强调的是，属性集 $\{L_{\zeta_i}\}_{\zeta_i\in[q+1,Q]}$ 不满足访问策略。

猜测阶段：在查询阶段 1 和查询阶段 2，挑战密钥由挑战者 B 生成，如上文所述。完成上述互动游戏后，攻击者 A 输出一个猜测 σ'。如果 $\sigma' = \sigma$，则表示挑战分量为 $R = e(g,g_1)^{s\cdot a^{q+1}}$，挑战者 B 将输出 0；否则输出 1，表示 R 是从群 G_T 中随机选择的元素；则攻击者 A 的优势为 0。若攻击者 A 的优势 $Adv_A = \varepsilon$ 不可忽视；那么挑战者 B 可以以不可忽略的优势打破 $(q-l)$-BDHE 假设。

引理 2 假设 DL 假设为真，那么对于任何 PPT[23] 算法就不能区分 $Game_{j'}$ 和 $Game_{j'+1}$，其中 $j' \in [0,m]$。

证明：假设存在一个攻击者 A，它能够以不可忽略的优势区分 $Game_{j'}$ 和 $Game_{j'+1}$ 这两个游戏；那么可以使用攻击者 A 的输出构造挑战者 B，以相同优势来解决 DL 问题。

初始化阶段：攻击者 A 将访问策略 $\Lambda^* = ((\boldsymbol{A}^*, \rho^*), \mathcal{T}^*)$ 发送给挑战者 B。

系统建立阶段：挑战者 B 随机选择 $\tilde{a}, \tilde{\alpha}, \theta, d_1{}', d_2{}', \tilde{f}, \tilde{h}, \tilde{\beta} \in \mathbb{Z}_p, g, g_1 \in G$，隐式设置 $g_2 = g^{d_2{}'}$，$a_2 = \tilde{\beta}/\theta$，设置公共参数 $Pk = (g, g^{\tilde{a}}, g^{\tilde{\alpha}}, g_2, g_2^{\tilde{\beta}}, u, h, f, e(g,g_1)^{\theta}, H)$。然后，初始化一个数据用户列表 $Ul = \emptyset$。常用参数的详细构造如下：

$$u = g^{d_1{}'\theta}, f = g^{\tilde{f}}, h = g^{-d_1{}'\theta H(V_{\rho^*(l')})} g^{\tilde{h}}$$

查询阶段 1：密钥查询阶段，攻击者 A 发送属性集 $S = \{L_1, L_2, \cdots, L_{|I_L|}\}$ 到挑战者 B 并查询他们的密钥。挑战者 B 随机选择 $\tilde{c}, \delta, \delta_1, \delta_2, \cdots, \delta_{|I_L|} \in Z_P$ 并隐式设置：

$$\tilde{\delta} = (\delta\theta)/(d_1{}' + \tilde{h})$$

$$\delta_x = \frac{\tilde{\delta}_x (\theta(H(V_{\rho^*(x)}) - H(V_{\rho^*(l')}))d_1{}' + \tilde{h}) + \tilde{f}\tilde{\delta}\tilde{\beta}}{\theta(H(V_{\rho^*(x)}) - H(V_{\rho^*(l')}))\tilde{\beta}}$$

挑战者 B 通过如下方式构造密钥：

$$K = g_1^{\theta r/\alpha + \tilde{c}} (g^{\tilde{a}\delta r})^{\theta/(d_1{}' + \tilde{h})}, K_0 = (g^{\delta r})^{\theta/(d_1{}' + \tilde{h})}$$

$$K_1 = \tilde{c}, K_0{}' = (g^{\tilde{a}\delta r})^{\theta/(d_1{}' + \tilde{h})}$$

$$K_{1,x} = (g^{a_2 r}) \frac{\delta_x \theta(H(V_{\rho^*(x)}) - H(V_{\rho^*(y')}) - \tilde{f}\tilde{\delta})}{(H(V_{\rho^*(x)}) - H(V_{\rho^*(y')})d_1{}'\tilde{h}} = g^{\tilde{\delta}_x r/\tilde{\beta}}$$

$$K_{2,x} = (u^{H(V_x)} h)^{\tilde{\delta}_x r/\tilde{\beta}} f^{(\tilde{a}+\tilde{c})\tilde{\delta}}$$

$$K_{3,x} = (u^{H(V_x)} h)^{r\tilde{\delta}_x} (f^{(\tilde{a}+\tilde{c})\tilde{\delta}})^{\tilde{\beta}}$$

随后，将元组 (K_1, id) 放入数据用户列表 Ul 中。

挑战阶段：攻击者 A 向挑战者 B 提交两个等长度的挑战明文消息 M_0 和 M_1。挑战者 B 随机投掷一枚硬币 $\sigma \in \{0,1\}$，选取 M_σ，运行 $Encrypt(Pk, M_\sigma, \Lambda^*)$ 算法生成密文 CT，将 CT 发送给攻击者 A。挑战者 B 随机设置一个列向量 $\vec{v} = (s, v_2, \cdots, v_n)^{\top}$ 其中，$v_{j_1} \in Z_P$，$j_1 \in [2,n]$，$\lambda_y = \boldsymbol{A}_y \cdot \vec{v}$。此外，它还随机选择 $\{k_y, z_y\}_{y\in[1,m]}, \tilde{\gamma}, d' \in Z_P$。

如果 $y = m$，则挑战者 B 隐式设置 $a_1 = s, z_m = 1/\theta, k_m = d_2 a_1$，计算密文 CT^*：

$$\hat{c} = e(g,g)^{\frac{1}{H(M_\sigma \| Ek) + \tilde{\gamma}}}, \overset{\vee}{c} = g^{\frac{1}{H(M_\sigma \| Ek) + \tilde{\gamma}}}$$

$$C = SE.Enc(EK, M_\sigma \| \overset{\vee}{c}), C_0^* = EK \cdot e(g,g_1)^{\theta s}$$

$$C_1^* = g^{a_1} = g^s, C_1' = g^{\tilde{\alpha} a_1} = g^{\tilde{\alpha} s}$$

$$C_{0,y}^* = (g^{a_2})^{d_2/\tilde{\beta}} = g_2^{z_m}$$

$$C_{1,y}^* = R^{z_1 m} g^{d_2 a_1} = g_2^{a_1 + a_2} = (g_2^{\tilde{\beta}})^{z_m} g^{k_m}$$

$$C_{2,y}^* = g^{\tilde{a}\lambda_m} R^{\tilde{f}} = g^{a\lambda_m} f^{k_m}, C_{3,y}^* = (u^{H(\pi_{\rho(m)})} h)^{k_m}$$

如果 $y \neq m$，挑战者 B 计算密文 CT^*：

$$\hat{c} = e(g,g)^{\frac{1}{H(M_\sigma \| Ek) + \tilde{\gamma}}}, \overset{\vee}{c} = g^{\frac{1}{H(M_\sigma \| Ek) + \tilde{\gamma}}}$$

$$C = SE.Enc(EK, M_\sigma \| \overset{\vee}{c}), C_0^* = EK \cdot e(g,g_1)^{\theta s}$$

$$C_1^* = g^s, C_1' = g^{\tilde{\alpha} s}, C_{0,y}^* = g_2^{z_y}, C_{1,y}^* = (g_2^{\beta})^{z_y} g^{k_y}$$

$$C_{2,y}^* = g^{a\lambda_y} f^{k_y}, C_{3,y}^* = (u^{H(\pi_{\rho(y)})} h)^{k_y}$$

最后，挑战者 B 将 CT^* 发送给攻击者 A。

查询阶段 2：它是查询阶段 1 的重复。

猜测阶段：攻击者 A 输出一个猜测 σ'。如果 $\sigma' = \sigma$，则表示挑战分量为 $R = g_2^{a_1 + a_2}$，挑战者 B 将输出 0；否则输出 1，表示 R 是从群 G 中选择的随机元素，那么攻击者 A 的优势为 0。总之，如果攻击者 A 的优势 $Adv_A = \varepsilon$ 不可忽视，那么挑战者 B 能够以不可忽视的优势打破 DL 假设。

引理 3 假设 DL 假设为真，那么对于任何 PPT 算法都没有区分 $Game_{k'}$ 和 $Game_{k'+1}$ 游戏的优势，其中 $k' \in [m, 2m-1]$。

证明：引理 3 的证明与引理 2 相同。

可追踪性证明：本文的追踪算法类似于 Ning 等[15] 提出

的算法。本文在数据用户的密钥中嵌入其身份 id，对应的密钥 $K_1=c$。通过检测用户密钥的真实性，根据密钥 K_1 查询数据用户列表 Ul，即可获得数据用户的身份 id。因此，在可追踪性证明上本文方案与 Ning 等[15]的方案相似。

6 性能分析

本节描述本文方案与其他方案的性能对比，分别从方案的基本功能特征、存储开销和计算开销这 3 个方面进行对比。

6.1 功能性比较

本节中，本文方案与其他采用 LSSS 和访问策略隐藏的方案进行功能特征比较，如表 2 所列。

表 2 方案功能特征比较

Table 2 Comparison of program features

方案	素数阶	解密测试	外包计算	可验证	可追踪
文献[10]的方案	◇	√	◇	◇	◇
文献[24]的方案	◇	√	◇	◇	◇
文献[11]的方案	√	◇	◇	◇	◇
文献[1]的方案	√	√	√	√	◇
本文方案	√	√	√	√	√

从表中可以看出，文献[10]和文献[24]的方案仅支持解密测试，文献[11]的方案仅支持素数阶。文献[1]的方案虽然具备其他功能，但不支持用户追踪；本文方案同时支持素数阶、解密测试、外包计算、可验证、可追踪特征，因此实用性更好。

6.2 开销比较

本节将本文方案与其他基于素数阶群的访问策略隐藏方案分别从存储开销和计算开销这两方面进行对比。本文方案在 Intel(R) Core(TM) i5-7300HQ CPU 2.5GHz、16RAM、Win10 操作系统、基于 Java 配对加密库(JPBC)[25]进行实验。

表 3 列出了存储开销的对比统计。其中，p，m，ω 分别表示配对操作、矩阵行数以及数据用户的属性个数，γ 是一个授权集，即它可以满足算法中的访问策略。$|G|$ 和 $|G_T|$ 表示群 G 和 G_T 中元素的大小，E 和 E_T 分别表示群 G 和 G_T 中的指数运算。实验模拟了表 4 中的计算开销，实验数据展示在图 2 和图 3 中。

表 3 存储开销

Table 3 Storage overhead

方案	公钥	密钥	密文
文献[11]的方案	$9\|G\|+\|G_T\|$	$(5\omega+2)\|G\|$	$(6m+1)\|G\|+\|G_T\|$
文献[1]的方案	$7\|G\|+\|G_T\|$	$(4\omega+2)\|G\|$	$(5m+1)\|G\|+2\|G_T\|$
本文方案	$8\|G\|+\|G_T\|$	$(3\omega+5)\|G\|$	$(4m+2)\|G\|+2\|G_T\|$

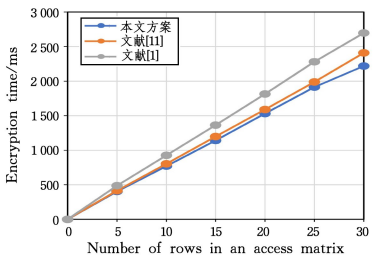

图 2 加密时间对比

Fig. 2 Encryption time comparison

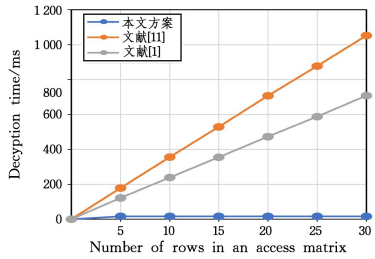

图 3 解密时间对比

Fig. 3 Decryption time comparison

从表 3 可以看出，本文方案在公钥存储开销上比文献[11]的方案少 1 个 $|G|$。本文方案相比文献[1]的方案增加了用户追踪功能，因此公钥的存储开销比它多一个 $|G|$。在密钥和密文存储开销上，本文方案明显低于文献[11]的方案。相比文献[1]的方案，本文方案采用 SGX 技术进行解密测试，避免了计算冗余的密文和密钥组件。因此，在密钥和密文的存储开销上，本文方案都比文献[1]的方案少一个 $\omega|G|$，随着属性数量的增加，本文方案的优势更加明显。

表 4 计算开销

Table 4 Computational overhead

方案	加密	解密
文献[11]的方案	$(8m+1)E+E_T$	$(6\|\gamma\|+1)p+\|\gamma\|E_T$
文献[1]的方案	$(8m+3)E+2E_T$	$(4\|\gamma\|+1)p+\|\gamma\|E_T$
本文方案	$(7m+3)E+2E_T$	$p+E+E_T$

从表 4 可以看出，本文方案在加密开销上比文献[1]和文献[11]的方案小一个 mE 左右的开销。从图 2 也可以看到，本文方案随着矩阵行数的增加，加密开销的增速低于文献[1]和文献[11]的方案。在解密开销上，由于本文方案在解密测试中进行了大部分的运算，且考虑到用户计算开销的问题，本文方案将解密测试的计算外包给服务器。因此，本文方案只需要进行常数级的计算。从图 3 可以看出，本文方案明显优于文献[1]和文献[11]的方案。

结束语 本文提出了一个支持访问策略隐藏和密钥追踪的轻量级医疗数据共享方案，有效地解决了医疗系统中数据所有者的隐私和数据安全性问题，在降低用户的存储开销和计算开销的基础上增加了系统的安全性。本文方案采用 SGX 技术实现了在策略隐藏方案中的解密测试，避免了冗余密钥和密文的生成，降低了数据所有者的计算开销和数据用户的存储开销。此外，本文方案采用外包可验证技术，将大部分计算外包给医疗云服务器，降低了数据用户的计算开销，并且可以验证解密结果的正确性和完整性。此外，还采用白盒可追踪技术，追踪泄露密钥的恶意用户，进一步增强系统的安全性。最后，对本文方案进行了安全性证明，证明所提方案能够抵抗选择明文攻击。在未来的工作中，期望通过降低整个解密过程的存储和计算开销来完善本文方案。

参 考 文 献

[1] HU G, ZHANG L, MU Y, et al. An expressive "test-decrypt-verify" attribute-based encryption scheme with hidden policy for smart medical cloud[J]. IEEE Systems Journal, 2021, 15(1): 365-376.

[2] TANG H F. Research on security access and privacy protection

mechanism in medical cloud[D]. Xi'an:Xidian University,2020.
[3] NIUS U,LIU W K,CHEN L X,et al. Data Sharing Scheme of Electronic Medical Record Based on Proxy Re-Encryption[J]. Computer Engineering,2021,47(6):164-171.
[4] SAHAI A,WATERS B. Fuzzy identity-based encryption [C]// Proceedings of the 2005 Annual International Conference on the Theory and Applications of Cryptographic Techniques, LNCS 3494. Berlin:Springer,2005:457-473.
[5] GOYAL V,PANDEY O,SAHAI A,et al. Attribute-based encryption for fine-grained access control of encrypted data[C]// Proceedings of the 13th ACM Conference on Computer and Communications Security. New York:ACM,2006:89-98.
[6] BETHENCOURT J,SAHAI A,WATERS B. Ciphertext-policy attribute-based encryption[C]// Proceedings of the 2007 IEEE Symposium on Security and Privacy. Piscataway: IEEE, 2007: 321-334.
[7] LEWKO A,OKAMOTO T,SAHAI A,et al. Fully secure functional encryption:attribute-based encryption and (hierarchical) inner product encryption[C]// Proceedings of the 29th Annual International Conference on Theory and Applications of Cryptographic Techniques. 2010:62-91.
[8] CHEUNG L,NEWPORT C. Provably secure ciphertext policy ABE[C]// Proceedings of the 14th ACM Conference on Computer and Communications Security. ACM,2007:456-465.
[9] NISHIDE T,YONEYAMA K,OHTA K. Attribute-based encryption with partially hidden encryptor-specified access structures[C]// Proceedings of the 2008 International Conference on Applied Cryptography and Net-work Security. Springer-Verlag, 2008:111-129.
[10] LAI J,DENG R H,LI Y. Expressive CP-ABE with partially hidden access structures [C]// Proceedings of the 7th ACM Symposium on Information,Computer and Communications Security. New York:ACM,2012:18-19.
[11] CUI H,DENG R,LAI J,et al. An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures, revisited[J]. Computer Networks, 2018,133:157-165.
[12] NING J,HUANG X,SUSILO W,et al. Dual access control for cloud-based data storage and sharing[J/OL]. IEEE Transactions on Dependable and Secure Computing. https://doi.org/10.1109/TDSC.2020.3011525.
[13] ZHU X D,ZHANG Y Y,YAO R K,et al. Research on Government Information Opening and Sharing Model and Application Based on Blockchain[J]. Journal of Chongqing Technology and Business University (Natural Science Edition), 2020, 37(5):122-128.
[14] HUANG Z Z,ZHANG X D,ZHAO J H,et al. Design of knowledge sharing mechanism based on blockchain[J]. Journal of Chongqing University of Technology(Natural Science), 2021, 35(9):143-151.
[15] 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 & Security,2015,10(6):1274-1288.
[16] ZENG P,ZHANG Z,LU R,et al. Efficient policy-hiding and large universe attribute-based encryption with public traceability for internet of medical things[J]. IEEE Internet of Things Journal,2021,8(13):10963-10972.
[17] MCKEEN F,ALEXANDROVICH I,BERENZON A,et al. Innovative instructions and software model for isolated execution [J/OL]. Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy. https://doi.org/10.1145/2487726.2488368.
[18] XIE Y,MIAO F Y,BAI J F. Secret sharing scheme with general access structure based on integer programming[J]. Computer Engineering,2019,45(6):165-170.
[19] ROUSELAKIS Y,WATERS B. Practical constructions and new proof methods for large universe attribute-based encryption [C]// Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security. 2013:463-474.
[20] BONEH D,BOYEN X,SHACHAM H. Short group signatures [C]// Proceeding of the 24th Annual International Cryptology Conference. 2004:41-55.
[21] SHINDE S,CHUA Z L,NARAYANAN V,et al. Preventing your faults from telling your secrets:defenses against pigeonhole attacks[C]// Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security. 2016:317-328.
[22] FISCH B,VINAYAGAMURTHY D,BONEH D,et al. Iron: functional encryption using Intel SGX[C]// Proceedings of the 2017 ACM SIGSAC Conference. 2017:765-782.
[23] ROUSELAKIS Y,WATERS B. New constructions and proof methods for large universe attribute-based encryption[C]// Proceedings of the ACM Conference on Computer and Communications Security. 2013:463-474.
[24] LIU L,LAI J,DENG R,et al. Ciphertext-policy attribute-based encryption with partially hidden access structure and its application to privacy-preserving electronic medical record system in cloud environment[C]// Proceedings of the Security and Communication Networks. 2016:4897-4913.
[25] CARO D A,LOVINO V. Java pairing based cryptography[C]// Proceedings of the 2011 IEEE Symposium on Computers and Communications. 2011:850-855.

WANG Meng-yu, born in 1997, postgraduate, is a member of China Computer Federation. His main research interests include attribute based encryption and information safety.

YIN Xin-chun, born in 1962, Ph.D, professor, Ph. D supervisor, is a senior member of China Computer Federation. His main research interests include cryptology, software quality assurance and high performance computing.

(责任编辑:喻藜)