计算机研究与发展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202330618
61(7): 1611−1628, 2024

支持访问行为身份追踪的跨域密文共享方案

申　远[1,2]　宋　伟[1,3]　赵常胜[1]　彭智勇[1,3]
[1](武汉大学计算机学院　武汉　430072)
[2](河南省平顶山学院软件学院　河南平顶山　467041)
[3](文化遗产智能计算实验室(武汉大学)　武汉　430072)
(shenyuan@whu.edu.cn)

A Cross-Domain Ciphertext Sharing Scheme Supporting Access Behavior Identity Tracing

Shen Yuan[1,2], Song Wei[1,3], Zhao Changsheng[1], and Peng Zhiyong[1,3]
[1] (*School of Computer Science*, *Wuhan University*, *Wuhan* 430072)
[2] (*School of Software*, *Pingdingshan University*, *Pingdingshan*, *Henan* 467041)
[3] (*Intellectual Computing Laboratory for Cultural Heritage* (*Wuhan University*), *Wuhan* 430072)

Abstract　As a widely used ciphertext authorization access mechanism in cloud environments, ciphertext-policy attribute-based encryption (CP-ABE) has fine-grained, one-to-many and owner-controlled properties. However, the traditional CP-ABE mechanism is difficult to obtain the identities of authorized users who maliciously abuse their decryption privileges since multiple users may have the same attribute set. Although numerous existing studies achieve the identity tracking for some specific decryption privilege abuses (i.e., white-box attacks and black-box attacks), they are challenging to audit authorized users' identities for ciphertext access behaviors, which may lead to potential data security and owners' right-to-be-informed compliance issues. Based on CP-ABE mechanism, to realize identity tracing of ciphertext data access behavior in real application scenarios, this scheme designs a cross-domain ciphertext data sharing method, which generates the access request by binding the traceable decryption key with the authorized user's access behavior. The integrity of access requests is protected by blockchain. Meanwhile, this scheme introduces an encrypted inverted index structure to address the inefficiency of the identity traceability caused by blockchain traversal. The privacy-preserving of index queries is achieved through the BLS signature and privacy set intersection. Theoretical analysis and experimental results demonstrate that the proposed cross-domain ciphertext sharing scheme with authorized users' access behaviors audit trail is efficient and practical.

Key words　CP-ABE; cross-domain ciphertext data sharing; access behavior trace; blockchain; traversing optimization

摘　要　作为在云环境下被广泛应用的密文数据授权访问机制,密文策略属性基加密(ciphertext-policy attribute-based encryption, CP-ABE)具有细粒度、1对多和拥有者可控的特点.由于多个用户可能拥有相同属性集合,传统属性基加密机制难以追溯到滥用解密权限的恶意授权用户身份.虽然现有研究解决了恶意用户的特定解密权限滥用行为(白盒攻击与黑盒攻击)的身份追踪问题,但仍难以实现针对授权用户访问行为的身份追踪,这将导致潜在的安全风险和数据访问知情权合规性问题.为了在现实应用场景中

收稿日期: 2023-07-31; **修回日期**: 2023-11-14
基金项目: 国家自然科学基金项目(62372340, 62072349); 湖北省科技厅技术攻关项目(2023BAA018)
This work was supported by the National Natural Science Foundation of China (62372340, 62072349) and the Major Technical Research Project of Hubei Province (2023BAA018)

实现密文数据访问行为身份追踪,方案基于密文策略属性基加密机制构造跨域密文共享方法,通过数字签名和交互式外包解密流程将可追踪密钥和授权用户访问行为绑定为访问请求,并利用区块链的不可篡改性实现访问请求的完整性保护.为了解决引入区块链所导致的访问行为身份追踪效率低下问题,方案引入加密倒排索引结构以优化区块遍历效率,并通过BLS签名和隐私集合交集思想实现索引查询的隐私保护.理论分析和实验验证表明所提方案是实用与高效的.

关键词 密文策略属性基加密;跨域密文数据共享;访问行为追踪;区块链;遍历优化

中图法分类号 TP391

随着云计算技术在医疗领域的应用与普及,越来越多的用户倾向于将电子健康记录(electronic health record, EHR)外包至云端进行存储,以EHR数据为驱动的各类信息系统与应用为用户提供了各类便利服务.例如:通过基于云共享的电子病历管理系统中保存的患者的各类历史医疗记录,医生在对患者的诊疗过程中能够根据他们的历史医疗数据做出更为精准的医疗服务[1];通过支持跨域数据访问的电子健康数据平台,保险公司能够获知客户近年来的体检信息或医疗诊断病历,为客户提供方便快捷的保险付费与理赔业务[2].由于EHR中包含了诸如患者身份信息和其他医疗敏感信息等私人数据[3],数据安全成为了云环境下EHR驱动的数据共享服务与应用所关注的主要问题.为了避免隐私泄露,数据拥有者通常会在EHR存储于云端之前将其做加密处理.

由于在云环境中能够为用户提供细粒度、1对多和拥有者可控的密文数据安全共享方法[4],密文策略属性基加密(ciphertext-policy attribute-based encryption, CP-ABE)成为了研究者的关注热点并被广泛应用.在传统的CP-ABE机制中,授权用户的解密权限由其拥有的属性集合生成,只有当解密权限对应的属性集合满足嵌入密文的访问策略时,授权用户才能够正确地完成密文解密.但由于不同的用户可能会拥有相同的属性集合[5],CP-ABE机制存在解密权限滥用的安全问题.解密权限滥用问题是由1对多的访问授权特点引起的,授权用户利用此特点可以在执行诸如出售解密密钥[6]或伪造解密设备[7]等恶意行为后来躲避系统问责.例如,Alice和Bob拥有相同的属性集合,当该属性集合对应的解密密钥被恶意泄露后,系统无法从泄露的密钥中推断出泄密者的身份.

为解决由于解密权限恶意滥用而导致的数据安全问题,研究者提出了针对不同攻击行为的解决方案:白盒可追踪性(white-box traceability)[5-6,8-13]和黑盒可问责性(black-box accountability)[7,14-16].前者通过将用户身份信息嵌入到解密密钥来对被恶意用户泄露的解密密钥进行身份追踪,后者通过用户身份信息参与的加密和解密算法来实现对伪造解密设备的恶意用户的身份问责.针对部署环境资源受限的场景,文献[4]基于大属性域(large universe)技术构造公共参数固定的密文数据访问方案,该方案公共参数的大小不会跟随属性数量的增长而改变,降低存储开销的同时还实现了方案的可扩展性.文献[5]将大属性域技术和白盒可追踪技术相结合,提出了一种系统扩展性良好且抵抗用户密钥泄露的密文数据访问方案.为了降低客户端解密过程的计算开销,文献[10]通过构造代理解密密钥的方式将大量的双线性计算外包至云端执行,此外该方案还支持访问权限的撤销.为了解决现实医疗场景中由于患者离线而无法授权的问题,文献[11]在白盒可追踪CP-ABE方案的基础上引入了紧急访问授权(break glass key)技术,满足了不同场景下的医疗密文数据访问授权功能需求,并支持密钥泄露和紧急访问操作的用户身份追踪.由于恶意用户可通过将解密密钥封装在解密设备中而使得上述白盒可追踪方案失效.为了解决此问题,文献[14]为每个解密密钥分配了一个独有的索引参数,并通过对应的加解密算法将解密密钥和索引参数进行绑定,从而实现对构造解密设备的恶意用户的身份问责;文献[15]采用基于身份的集合加密(identity-based set encryption, IBSE)和指纹码(fingerprint code)技术设计了支持黑盒可问责的属性基恶意用户追踪方案,该方案的解密密钥长度与系统用户数量无关,能够有效降低客户端的存储与通信开销;文献[16]采用属性基恶意用户追踪(attribute-based traitor tracing, ABTT)和基于策略的变色龙哈希(policy-based chameleon Hash, PCH)算法构造了属性基可编辑区块链方案,通过引入指纹码和基于身份数字签名使其支持黑盒可问责性,实现了可编辑区块链应用场景的恶意用户身份问责.文献[17]通过将用户对数据的操作记录上传至区块链中存储实现了用户访问行为的审计追踪,但该方案是针对明文数据场

景的，在访问行为存储过程和审计过程中无法保护用户的隐私信息，并且基于区块链的审计追踪过程需要遍历所有区块以提取审计目标的访问记录，增加了额外的时间开销. 为了优化区块遍历过程的时间开销，Ruan 等人[18]针对区块链交易事务提出了有向无环图结构，图中的节点表示交易事务对应的账户在某个版本下的状态值，图中的有向边表示交易，通过有向无环图建立“交易账户状态-交易事务”之间的映射关系，并在此基础上构造跳表索引结构来减少查询账户状态所需遍历的区块数量，优化时间效率. 然而该方案同样是面向明文数据场景的，无法应用于密文数据访问行为审计的应用场景. 文献 [19] 在 CP-ABE 密文访问架构上，利用同步聚合签名（synchronized aggregate signature）技术实现了用户在密文访问过程中对使用云端资源消耗凭据的快速验证.

如上所述，现有研究方案仅能够对特定的解密权限滥用行为实现恶意用户身份追踪，而对于可能会导致潜在安全威胁和隐私泄露的授权用户的正常访问行为还缺少相应的身份追踪方法. 例如，Wang 等人[19]指出分布式拒绝服务（distributed denial-of-service, DDoS）攻击可能由授权用户正常的访问行为触发，虽然文献 [20] 通过在 CP-ABE 解密算法中引入计数器参数来限制授权用户对密文的访问次数，但该方案在响应授权用户的访问请求时，云端仍然需要消耗一定的计算资源，无法有效抵抗 DDoS 攻击模式；此外在现实应用中，患者通常希望加密 EHR 数据仅能够被自己的主管医师所访问，但由于 CP-ABE 的 1 对多特性，患者的这一隐私期望难以得到实现，进而导致潜在的隐私泄露风险；另一方面，《通用数据保护条例》等法律赋予了数据拥有者有关个人数据处理过程的知情权[21]，但由于 CP-ABE 的 1 对多特性带来的匿名性[16]，导致现有基于属性基加密机制的密文数据共享方案在现实应用中存在合规性问题. 如上所述，现有 CP-ABE 数据访问授权方案应用于以数据为驱动的信息共享服务场景时，仍然面临数据安全与合规性等问题. Facebook 与滴滴出行由于恶意滥用用户数据与违反法律法规而遭受处罚的案例表明，若不能有效解决数据安全与合规性问题，则难以在现实中推广以数据驱动的各类信息与数据服务.

导致上述问题的根本原因是现有基于属性基加密机制的密文共享方案无法提供密文数据访问行为的身份追踪. 为了解决这一问题，本文在白盒可追踪的密文策略属性基加密机制基础上提出了一个可证明安全的支持授权用户访问行为身份追踪的跨域密文数据共享方案. 在该方案中，授权用户的所有访问请求将被记录在区块链中用于访问行为的身份追踪. 本文的主要创新点包括 3 点：1）针对来自于不同机构授权用户的密文访问需求，通过多授权机构管理各自机构的访问属性，并利用内嵌密文的双重访问策略设计支持白盒可追踪的跨域密文数据共享方案，该方案基于大属性域技术构造，具有较好的可扩展性；2）通过轻量级 Gamma 签名和交互式外包解密算法实现解密密钥和访问行为绑定，并将用户访问行为以密文形式存储在区块链中，实现访问行为完整性保护；3）为了减轻数据拥有者针对访问行为的身份审计过程需要遍历区块链的时间开销，本方案设计了用于区块链遍历优化的倒排索引结构，并基于 BLS 签名和隐私集合求交（private set intersection, PSI）运算构造陷门查询算法，在优化区块链上对密文内容遍历效率的基础上，对用户的敏感信息和访问模式进行了有效保护.

1 相关知识

1.1 双线性映射

假设G和G_T为 2 个阶为素数p的乘法循环群，令g为群G的生成元，群G和G_T之间的双线性映射$e:G\times G\to G_T$具有 3 个性质：

1）双线性. $\forall f, h\in_R G$ 和 $\forall a, b\in_R \mathbb{Z}_p$，有 $e(f^a,h^b)=e(f,h)^{ab}$.

2）可计算性. $\forall f,h\in_R G$, $e(f,h)$是可计算的.

3）非退化性. $e(g,g)\neq 1$.

1.2 访问结构

定义 1. 访问结构[22]. 令$\{P_1,P_2,\cdots,P_n\}$为参与实体的集合，如果集合$\mathbb{A}\subseteq 2^{\{P_1,P_2,\cdots,P_n\}}$对于$\forall B,C$：若$B\in\mathbb{A}$且$B\subseteq C$，有$C\in\mathbb{A}$，则称$\mathbb{A}$是单调的. 若访问结构$\mathbb{A}$是参与实体$\{P_1,P_2,\cdots,P_n\}$构成幂集的非空子集合，即$\mathbb{A}\in 2^{\{P_1,P_2,\cdots,P_n\}}\backslash\{\varnothing\}$，则包含在$\mathbb{A}$中的集合被称为授权集合，不包含在$\mathbb{A}$内的集合为非授权集合.

1.3 线性秘密分享方案

定义 2. 线性秘密分享方案（linear secret sharing scheme, LSSS）[22]. 令p为素数，构建在参与者集合$\mathbb{A}$上的秘密共享方案Π满足 2 个条件时，则称Π在$\mathbb{Z}_p$上是线性的：

1）每个实体分享的子秘密是$\mathbb{Z}_p$上的向量；

2）对于秘密分享方案Π存在一个分享生成矩阵$\boldsymbol{M}\in\mathbb{Z}_p$，其中$\boldsymbol{M}$是$l\times n$矩阵. 对于矩阵$\boldsymbol{M}$中的每一行$i=1,2,\cdots,l$，映射函数$\rho:\{1,2,\cdots,l\}\to\mathbb{A}$把每一个矩阵

行向量映射为对应的参与者$\rho(i)$. 构造列向量$\boldsymbol{v}=(s,v_2,v_3,\cdots,v_n)$, 其中$s\in\mathbb{Z}_p$是主秘密, 随机数$v_2,\cdots,v_n\in\mathbb{Z}_p$用来掩饰$s$. 由$\boldsymbol{Mv}$得到主秘密$s$的$l$个子秘密, 其中$(\boldsymbol{Mv})_i$表示参与者$\rho(i)$所分配到的子秘密.

令$S\in\mathbb{A}$为任意授权集合, 且$I\subset\{1,2,\cdots,l\}$被定义为$I=\{i:\rho(i)\in S\}$, 则存在$\{\omega_i\in\mathbb{Z}_p\}_{i\in I}$对于分享生成矩阵$\boldsymbol{M}$满足

$$\sum_{i\in I}\omega_i\boldsymbol{M}_i=(1,0,\cdots,0).$$

1.4 安全性假设

定义 3. q-type 假设[23]. 令g为双线性群G中的随机元素, $d,s,b_1,b_2,\cdots,b_q\in\mathbb{Z}_p$为$q+2$个随机数. 若对给定的双线性映射$(p,G,G_T,e)$和多元组$(g,g^s,\ g^{d^i},g^{b_j},\ g^{sb_j},g^{d^ib_j},g^{d^i/b_j^2},\ \forall(i,j)\in[q,q],\ g^{d^ib_j/b_{j'}^2},\ \forall(i,j,j')\in[2q,q,q]$且$j\neq j'$, $g^{d^i/b_j},\ \forall(i,j)\in[2q,q]$且$i\neq q+1$, $g^{sd^ib_j/b_{j'}},g^{sd^ib_j/b_{j'}^2},\ \forall(i,j,j')\in[q,q,q]$且$j\neq j'$), 若攻击者无法在多项式时间内以不可忽略的优势区分$e(g,g)^{sd^{q+1}}$和随机元素$R\in G_T$, 则称q-type 假设成立.

定义 4. l-SDH 假设[24]. 假设G是阶为素数p的双线性群且g为生成元, 在群G上的l-SDH(l-strong Diffie-Hellman)问题被定义为: 对给定元组$(g,g^x,g^{x^2},\cdots,g^{x^l})$和$(c,g^{1/(c+x)})\in\mathbb{Z}_p\times G$, 若攻击者无法在多项式时间内以不可忽略的优势区分上述参数, 则称l-SDH 假设成立.

2 系统和安全模型

2.1 系统模型

如图 1 所示, 本文方案包含 6 个实体, 其中密钥生成中心和授权代理被假设为完全可信.

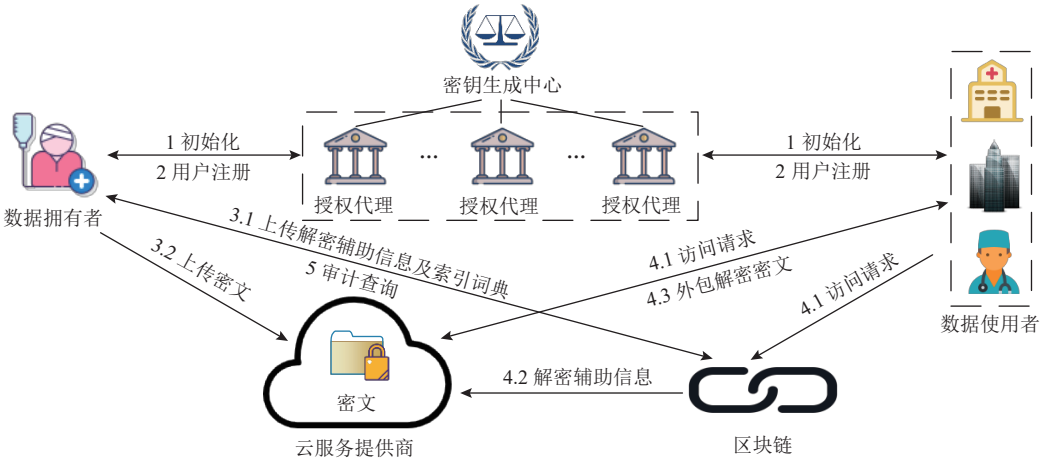

Fig. 1 Illustration of system architecture

图 1 系统架构示意图

1)密钥生成中心(key generation center, KGC). 该实体负责生成系统公开参数和系统主私钥.

2)授权代理(authority agent, AA). 为了有效管理不同机构用户的访问属性, 方案包含若干个 AA, 每个 AA 管理对应的属性域. AA 为系统用户生成公私钥及访问密钥; 同时全体 AA 参与区块链运行维护, 并负责管理用于区块链遍历优化的倒排索引结构.

3)云服务提供商(cloud service provider, CSP). CSP 为系统用户提供加密数据存储服务和外包解密服务.

4)数据拥有者(data owner, DO). DO 生成加密数据并上传到 CSP 端, 同时将外包解密算法所需的辅助信息发送至区块链存储, 此外 DO 还能够针对外包数据发起访问行为的审计查询.

5)数据使用者(data user, DU). DU 向所属的 AA 注册以获得解密密钥. 访问数据时, DU 向 CSP 发送访问请求以获得外包解密密文, 并在本地完成最终的解密操作.

6)区块链(blockchain, BC). 区块链作为方案的分布式账本, DU 的访问请求以交易事务的形式存储在区块链中. 链中各类节点由 AA 担任, 包括主节点(peer node)、排序节点(ordering node)和查询节点(query node, QN)[25], 其中主节点负责交易事务的存储, 排序节点负责共识协议, 查询节点负责响应用户的数据访问请求和审计请求.

2.2 算法描述

本文方案包含 11 个多项式算法, 各算法描述为:

1)系统初始化$Setup(1^\kappa)\rightarrow(pp,msk)$. KGC 执行该算法. 该算法以安全参数$1^\kappa$为输入, 输出系统公共参

数pp和主私钥msk，下列算法描述中将省略系统公共参数.

2）授权代理初始化$AASetup(msk)\rightarrow(aPK,aSK)$. KGC 执行该算法. 该算法以主私钥为输入，输出授权代理AA_i的公私钥对(aPK_i,aSK_i).

3）实体初始化$EntSetup(msk)\rightarrow(PK,SK)$. KGC 执行该算法. 该算法以主私钥$msk$为输入，输出 CSP 和 BC 的公私钥对$(PK,SK)$.

4）授权用户密钥生成$KeyGen(aSK_i,uid_j,S)\rightarrow(uPK_{i,j},uSK_{i,j},uDK_{i,j})$. AA_i执行该算法. 该算法以私钥aSK_i和数据使用者DU_j身份uid_j和访问属性集合S为输入，输出DU_j的公/私/解密密钥三元组$(uPK_{i,j},uSK_{i,j},uDK_{i,j})$.

5）加密$Enc(m,(\boldsymbol{M},\delta),(\boldsymbol{A},\rho))\rightarrow\{CT,aux\}$. DO 执行该算法. 该算法以访问策略$\{(\boldsymbol{M},\delta),(\boldsymbol{A},\rho)\}$、明文$m$为输入，输出密文$CT$和辅助解密参数$aux$.

6）索引生成$IdxGen(fid,oSK,PK_{\mathrm{BC}})\rightarrow idx$. DO 执行该算法. 该算法以外包文件名$fid$、私钥$oSK$和区块链公钥$PK_{\mathrm{BC}}$为输入，输出索引结构$idx$.

7）外包解密$OutDec(CT,OK,aux)\rightarrow CT_{\mathrm{out}}$. CSP 执行该算法. 该算法以 DU 的外包解密密钥OK和辅助解密参数aux为输入，输出密文CT的外包解密密文CT_{out}.

8）解密$Dec(CT_{\mathrm{out}})\rightarrow m/\perp$. 数据使用者$DU_j$执行该算法. 该算法以外包解密密文$CT_{\mathrm{out}}$为输入，输出明文$m$或$\perp$.

9）审计查询生成$QryGen(fid_{\mathrm{t}},oSK)\rightarrow T_{\mathrm{Q}}$. DO 执行该算法. 该算法以 DO 私钥$oSK$和目标文件名$fid_{\mathrm{t}}$为输入，输出审计查询$T_{\mathrm{Q}}$.

10）查询匹配$QryInt(T_{\mathrm{Q}},idx)\rightarrow\{bn\}$. BC 执行该算法. 该算法以审计查询$T_{\mathrm{Q}}$和索引结构$idx$为输入，输出区块编号集合$\{bn\}$.

11）身份追踪$Trace(OK,aSK_i)\rightarrow uid$. AA_i执行该算法. 该算法以外包解密密钥OK和AA_i私钥aSK_i为输入，输出内嵌于OK中的 DU 的身份信息uid.

2.3 安全模型

在该方案中，KGC 和 AA 被假定为完全可信的；CSP 被假定为“诚实且好奇”的，即 CSP 会诚实地存储加密数据和执行预定义的算法，但会从整个存储和解密流程中试图获取用户的隐私信息；系统的攻击者被假定为非授权用户、恶意授权用户和其他攻击者，其中非授权用户会尝试利用已掌握的解密权限来解密无权限访问的密文数据，恶意授权用户会试图对自己的访问行为进行抵赖或出售解密权限，其他攻击者会试图获取系统内用户的各项隐私信息（例如从区块链中存储的访问请求、倒排索引内容和审计查询中获取的访问模式等隐私信息）.

选择性安全（selective security）[6]. 本文方案的数据机密性（data confidentiality）建立在选择性安全基础之上. 选择性安全通过敌手和挑战者之间的游戏来定义，假设X为敌手发起密钥查询的次数上限且$X_1\in\{0,1,\cdots,X\}$，游戏具体流程为：

1）初始化. 敌手选择要挑战的访问结构ST^*，并将其发送给挑战者.

2）参数建立. 挑战者执行$Setup()$和$AASetup()$算法，生成公共参数pp和授权代理 AA 的公私钥(aPk,aSK)，并将pp和aPK发送给敌手.

3）查询阶段 1. 敌手为其身份信息uid_{X_1}和属性集合S_{X_1}向挑战者发出密钥请求，其中$S_{X_1}\notin ST^*$. 挑战者执行$KeyGen()$算法为其生成对应的解密私钥$uDK_{uid_{X_1},S_{X_1}}$.

4）挑战. 敌手将 2 个等长的消息(m_0,m_1)发送给挑战者，挑战者随机选取$\beta\in\{0,1\}$，并基于访问结构ST^*执行$Enc()$算法对m_β进行加密，并将密文CT发送给敌手.

5）查询阶段 2. 敌手在$S_{X_1}\notin ST^*$和$X_1\leqslant X$的约束下延续查询阶段 1 的行为以获取不同属性集合的解密密钥.

6）猜测. 敌手输出对β的猜测结果$\beta'\in\{0,1\}$. 在该游戏中，敌手的优势被定义为$|Pr[\beta'=\beta]-1/2|$.

定义 5. 如果对于所有的概率多项式时间（probabilistic polynomial time, PPT）的敌手无法以不可忽略的优势赢得上述游戏，则称该方案是选择性安全的.

追踪正确性（tracing correctness）. 本文方案访问行为的可追踪性通过白盒可追踪性（white-box traceability）[9]实现，即授权用户无法伪造其他用户的解密权限，进而无法抵赖自己的访问行为. 审计正确性通过敌手和挑战者之间的游戏来定义，假设O为敌手发起的密钥查询的次数上限，游戏具体流程为：

1）参数建立. 挑战者执行$Setup()$和$AASetup()$算法，生成公共参数pp和授权代理 AA 的公私钥(aPk,aSK)，并将pp和aPK发送给敌手.

2）密钥查询. 敌手向挑战者提交用户身份和属性集合$(uid_1,S_1),(uid_2,S_2),\cdots,(uid_O,S_O)$以获取对应的解密密钥.

3）访问请求伪造. 敌手通过输出解密密钥uDK^*来伪造一个访问请求req^*，若$Trace()$算法的输出结果不属于$\{uid_1,uid_2,\cdots,uid_O\}$，则敌手赢得该游戏. 敌手

的优势被定义为 $Pr[Trace() \notin \{uid_1, uid_2, \cdots, uid_O\}]$.

定义 6. 如果对于所有的概率多项式时间敌手无法以不可忽略的优势赢得上述游戏，则称该方案能够实现追踪正确性.

此外，系统还需要保证数据拥有者在审计过程中的索引隐私、查询隐私和访问模式隐私：

1）索引隐私（index privacy）. 索引隐私保证了在区块链查询节点中存储的索引结构的机密性，使得攻击者无法根据索引内容推测出 DO 的敏感信息.

2）查询隐私（query privacy）. 查询隐私保证了其他攻击者无法从 DO 在发起审计查询中获取审计目标的相关信息. 在区块链查询节点处存储的倒排索引以密文形式存储，审计查询以陷门形式存在.

3）访问模式隐私（access pattern privacy）. 访问模式隐私保证了审计查询结果的隐私性，以防止攻击者从查询结果中推测出索引结构的隐私信息和陷门内容[26]，为了实现访问模式隐私，需要确保不同查询的返回结果是不可区分的.

3 方案设计

如图 1 所示，支持用户访问行为审计的跨域密文共享方案包含 5 个阶段，各阶段主要流程为：

1）KGC 初始化生成系统参数，并为 AA、BC 和 CSP 生成对应的公私钥.

2）AA 为各自管理域中的 DU 生成公私钥和解密密钥，并为 DO 生成全局公私钥对.

3）DO 对明文加密，将密文上传至 CSP 存储；同时基于明文数据文件名构造倒排索引的索引词典，连同外包解密过程所需的辅助信息上传至 BC 存储.

4）当 DU 访问云端密文数据时，DU 生成外包解密密钥，并构造密文访问请求，在此基础上生成访问请求数字签名，将访问请求和数字签名一并发送给 BC 和 CSP；BC 查询节点对数字签名验证后，对访问请求进行加密后发送至 BC 存储，完成对访问目标对应的倒排索引的更新，并将对应的辅助解密参数发送给 CSP；CSP 利用辅助解密参数和访问请求完成外包解密，将外包解密密文发送给 DU，最终由 DU 在本地完成外包解密密文的解密操作.

5）DO 对外包密文数据进行访问者身份审计时，首先针对审计目标文件构造陷门查询，并将其发送给 BC；BC 查询节点收到查询后在倒排索引结构上进行密文求交运算，根据匹配结果定位审计目标文件的访问请求所在区块编号，并依据区块编号找到待审计文件对应的加密访问请求，对其解密后从中抽取出 DU 的身份信息，并将其加密后发送给 DO.

方案中常用的符号和语义描述如表 1 所示.

Table 1 Commonly Used Symbols and Their Descriptions

表 1 常用符号和描述

符号	描述
pp, msk	系统公共参数，系统主私钥
aPK, aSK	授权代理公、私钥
PK_{BC}, SK_{BC}	区块链节点公、私钥
PK_{CSP}, SK_{CSP}	云服务提供商公、私钥
uid, S	DU 身份信息及其属性集合
uPK, uSK, uDK, OK	DU 公、私钥，解密密钥和外包解密密钥
oid, oPK, oSK	DO 身份信息及公、私钥
m, fid, ftg	明文数据，对应文件名及文件名标签
$SEnc$, $SDec$	对称加、解密函数
H, H_1, H_2, H_3, H_4, F	密码学哈希函数，伪随机函数
CT, CT_{out}	密文，外包解密密文
bn, ntg	区块编号，区块编号标签
req, C_{req}	访问请求，访问请求密文
idx, T_{Q}	倒排索引，审计查询

3.1 方案初始化

该阶段由 KGC 分别为系统、授权代理和其他实体生成对应的参数和公私钥.

1）KGC 调用算法 $Setup(1^{\kappa}) \rightarrow \{pp, msk\}$ 生成系统公共参数 pp 与系统主私钥 msk，该算法具体流程为：

KGC 选择 1^{κ} 为系统安全参数，令 g 是阶为素数 p 的双线性群 G 生成元，双线性映射 $e: G \times G \rightarrow G_T$. 令 $SEnc, SDec$ 为安全对称加、解密函数，F 为伪随机函数（pseudorandom function）[24]，KGC 随机选择 $g_1, g_2, g_3, g_4 \in G$ 和 $\alpha, a \in \mathbb{Z}_p^*$，计算 $h = g_1^a, f = g^a, v = g^{\alpha}$，选择密码学哈希函数 $H_1: \{0,1\}^* \rightarrow \mathbb{Z}_p^*, H_2: G \rightarrow \mathbb{Z}_p^*, H_3: \{0,1\}* \rightarrow G, H_4: \{0,1\}^* \rightarrow \mathbb{Z}_p, H: \{0,1\}^* \rightarrow \{0,1\}^C$，$C$ 为常数. 系统公共参数 $pp = \{G, G_T, e, g, g_1, g_2, g_3, g_4, h, f, v, v^{-1}\}$，系统主私钥 $msk = \alpha$.

2）KGC 调用算法 $AASetup() \rightarrow \{aPK, aSK\}$ 为授权代理 AA_i 生成公私钥. KGC 随机选择 $\alpha_i \in \mathbb{Z}_p^*$，计算 AA_i 公钥 $aPK_i = g_1^{\alpha_i}$，私钥 $aSK_i = \{aSK_{i,1} = \alpha_i, aSK_{i,2} = a\}$，其中 $aSK_{i,2}$ 为所有 AA 的全局私钥.

3）KGC 调用算法 $EntSetup() \rightarrow \{PK, SK\}$ 为 BC 和 CSP 分别生成对应的公私钥对. KGC 随机选择 $b, c \in \mathbb{Z}_p^*$，则 BC 的公私钥对为 $PK_{\mathrm{BC}} = h^b, SK_{\mathrm{BC}} = b$，由于 BC 由全体 AA 维护，因此 AA 拥有 BC 公私钥对 $(PK_{\mathrm{BC}}, SK_{\mathrm{BC}})$；CSP 的公私钥对为 $PK_{\mathrm{CSP}} = h^c, SK_{\mathrm{CSP}} = c$.

3.2 用户注册

在该阶段，DO 和 DU 向 AA_i 注册获得密钥.

对于 DO，AA_i 基于身份加密机制[27]为其生成公私钥对，令 oid 为 DO 的身份信息，DO 公钥 $oPK=H_3(oid)$，私钥 $oSK=oPK^{aSK_{i,2}}$.

对于 DU_j，AA_i 调用算法 $KeyGen(aSK_i,uid_j,S)\rightarrow\{uPK_{i,j},uSK_{i,j},uDK_{i,j}\}$ 为其生成公钥、私钥与解密密钥三元组，该算法具体流程为：

假设用户 DU_j 身份信息为 uid_j，AA_i 计算 $o_{j,1}=SEnc_{aSK_{i,2}}(uid_j)$，$z_j=SEnc_{aSK_{i,2}}(o_{j,1}\|o_{j,2})$，其中 $o_{j,2}=g_1^{\alpha_i+o'_{j,2}}$，且 $o'_{j,2}\in\mathbb{Z}_p^*$ 为随机数，使 z_j 满足 $gcd(a+z_j,p)=1$. 假设 DU_j 访问属性集合为 $S=\{A_1,A_2,\cdots,A_k\}\subset\mathbb{Z}_p^*$，$AA_i$ 令 $r=o'_{j,2}$ 并随机选择 $r_1,r_2,\cdots,r_\tau\in\mathbb{Z}_p^*$，其中 $\tau\in[k]$. DU_j 公钥 $uPK_{i,j}$、私钥 $uSK_{i,j}$ 和解密密钥 $DK_{i,j}$ 构造为：

$$uPK_{i,j}=\{(g_1v^{-1})^{o'_{j,2}},g_4^{a+z_j}\},\ uSK_{i,j}=o'_{j,2},\ dk_{i,j,1}=v^{1/(a+z_j)}g_1^r,$$
$$dk_{i,j,2}=z_j,\ dk_{i,j,3}=f^r,\ dk_{i,j,4}=g^r,\ dk_{i,j,5}=\{g^{r_\tau}\}_{\tau\in[k]},$$
$$dk_{i,j,6}=\{(g_2^{A_\tau}g_3)^{r_\tau}g_4^{-(a+z_j)r}\}_{\tau\in[k]},\ dk_{i,j,7}=\{aPK_\mu^{a+z_j}g_2^{z_j}\}_{\mu\in[L]},$$

其中 L 代表方案中 AA 的数量.

3.3 数据加密

在该阶段，DO 生成辅助解密参数并构造倒排索引词典，将其上传至 BC 存储. DO 采用密钥封装机制[28]并调用 $Enc()$ 算法对外包文件 fid 的明文数据 m 进行加密，下面介绍具体过程.

1）DO 随机选择 $\gamma_1,\gamma_2,\xi\in\mathbb{Z}_p^*$，计算 $\xi_1=H_2(PK_{\mathrm{BC}}^{\gamma_1+\gamma_2}\times PK_{\mathrm{CSP}}^{\gamma_1+\gamma_2})$ 和 $\xi_2=\xi+\xi_1$，DO 调用算法 $IdxGen(fid,oSK,PK_{\mathrm{BC}})\rightarrow idx$ 生成倒排索引 idx. 该算法具体流程为：

DO 计算 $ftg=F(e(oSK,PK_{\mathrm{BC}})\|H(fid))$，构造倒排索引文件名标签词典 $dict$，其中索引词典元素 $dict[i]=ftg_i$，DO 将审计词典 $dict$ 和辅助解密参数 $aux=\{h^{\gamma_1+\gamma_2},g^{\xi_2}\}$ 发送给 BC，其中 $dict$ 存储于查询节点 QN 中，QN 为该 DO 创建倒排索引结构，解密辅助参数 aux 存储在 BC 中. 在密文数据访问阶段，QN 将外包文件 fid_i 的访问请求 req_{fid_i} 所在区块编号 bn 转化为编号标签 ntg_{bn}，将其插入到 $dict[i]$ 对应的索引列表当中. 用于 BC 遍历优化的倒排索引构造如图 2 所示.

2）DO 调用算法 $Enc(m,(\boldsymbol{M},\delta),(\boldsymbol{A},\rho))\rightarrow CT$ 将文件 fid 所对应的明文数据 m 加密为密文 CT. 该算法具体流程为：

令 $(\boldsymbol{M},\delta)$ 和 $(\boldsymbol{A},\rho)$ 为 2 个基于线性秘密分享方案构造的访问结构，其中 $\boldsymbol{M}$ 和 $\boldsymbol{A}$ 分别为 $l\times n$ 和 $L\times n$ 的矩阵，函数 $\delta()$ 将 M 的每一行映射为一个访问属性，函数 $\rho()$ 将 $\boldsymbol{A}$ 的每一行映射为一个 AA. DO 选择随机数并构造列向量 $\boldsymbol{s}=(s,s_2,\cdots,s_n)^{\mathrm{T}}$ 和 $\boldsymbol{\varepsilon}=(0,\varepsilon_2,\cdots,\varepsilon_n)^{\mathrm{T}}$，通过线性秘密分享方案的秘密分享过程，计算 $\lambda_x=\boldsymbol{M}_x\boldsymbol{s}$ 和 $\theta_y=\boldsymbol{A}_y\boldsymbol{\varepsilon}$，其中 $x\in[l]$，$y\in[L]$. DO 随机选择 $t_x,t_y\in\mathbb{Z}_p^*$，$key\in G_T$，密文 CT 构造为：

$$C_m=SEnc_{key}(m\|H(m)),\ C=key\times e(g,v)^s,$$
$$C_0=g^s,\ C_0'=f^sg^{\xi},$$
$$\left\{C_{1,x}=g_1^{\lambda_x}g_4^{t_x},C_{2,x}=(g_2^{\delta(x)}g_3)^{-t_x},C_{3,x}=g^{t_x}\right\}_{x\in[l]},$$
$$\left\{C_{4,y}=g^{\alpha_{\rho(y)}t_y},C_{5,y}=g^{-t_y},C_{6,y}=g^{t_y}g_2^{\theta_y}\right\}_{y\in[L]},$$

DO 将 $(\boldsymbol{M},\delta)$、$(\boldsymbol{A},\rho)$ 和密文 CT 发送给 CSP.

3.4 密文数据访问

在该阶段，DU 向 BC 和 CSP 发送密文数据访问请求；BC 对访问请求验证后将该访问请求写入到 BC，并向 CSP 发送辅助解密参数；CSP 将访问请求对应的密文进行外包解密后发送给 DU，DU 在本地完成最终的解密操作，实现密文数据访问，下面介绍具体流程.

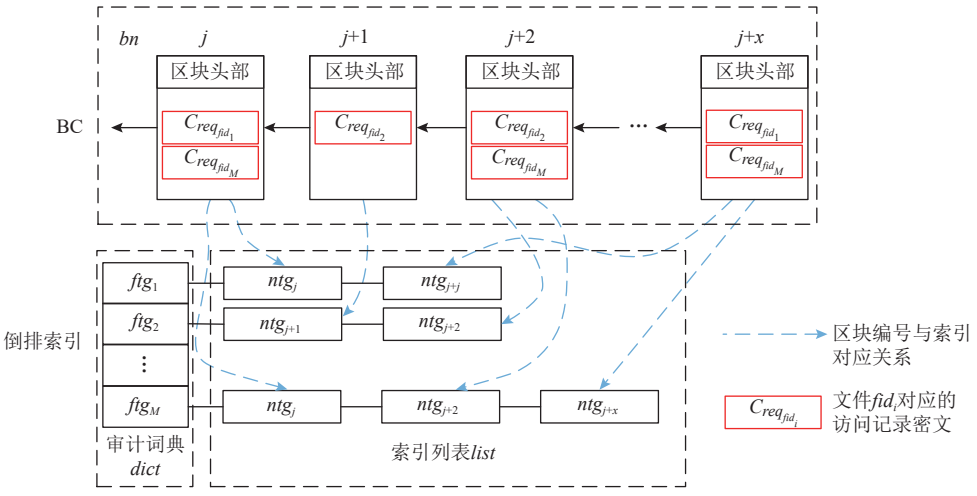

Fig. 2 Structural diagram of inverted index

图 2 倒排索引构造图

1）DU_j以线下协商方式[29]从DO处获取访问目标元数据oid和$H(fid)$，例如患者向医生提供其姓名或身份证号码作为oid、EHR文件名称作为fid，并使用$H(fid)$替换fid以保护EHR文件名中包含的敏感信息. DU_j选择随机盲因子$\eta\in\mathbb{Z}_p^*$，生成外包解密密钥OK：

$$ok_1=dk_{i,j,1}^{\eta},\ ok_2=dk_{i,j,2},\ ok_3=dk_{i,j,3}^{\eta},\ ok_4=dk_{i,j,4}^{\eta},$$
$$ok_5=dk_{i,j,5}^{\eta},\ ok_6=dk_{i,j,6}^{\eta},\ ok_7=dk_{i,j,7},\ ok_8=g^{\eta}.$$

DU_j构造访问请求$req=\{oid,H(fid),OK\}$及其Gamma数字签名[30]sig：随机选择$u\in\mathbb{Z}_p$，令$w=H_1(req)$，$U=(g_1v^{-1})^u$，$u_1=H_4(uPK_{i,j}\|U)$，并计算$w_1=(u\times u_1-w\times uSK_{i,j})\bmod p$，$DU_j$访问请求$req$的Gamma签名为$sig=\{u_1,w_1\}$；$DU_j$将访问信息$\{req,sig\}$发送给BC和CSP.

2）BC收到数据访问者DU_j的访问信息$\{req,sig\}$后，基于DU_j的公钥$uPK_{i,j}$对req进行验证：假设待验证签名为$sig'=\{u_1',w_1'\}$，计算$w=H_1(req)$，$U'=((g_1v^{-1})^{w_1'}uPK_{i,j,1}^{w})^{(u_1')^{-1}}$，验证等式$H_4(uPK_{i,j}\|U')=u_1'$是否成立. 若等式不成立，BC忽略此请求，否则，BC查询节点QN执行以下操作：

为了保护访问请求$req=\{oid,H(fid),OK\}$中的敏感信息，查询节点QN需要对访问请求req进行加密. QN计算$key_1=e(PK_{\mathrm{BC}},H_3(oid))^{aSK_2}$，生成$req$密文$C_{req}=SEnc_{key_1}(req\|ts)$，其中$ts$表示当前时间戳. C_{req}以交易的形式存储在新建区块中，QN基于该区块编号bn生成编号标签ntg_{bn}并将其插入到ftg_{fid}所对应的索引列表$list_{fid}$中，其中$ntg_{bn}=F(e(H_3(oid),h^{aSK_2})^{SK_{\mathrm{BC}}}\|bn)$；QN基于访问请求$req$计算文件名标签$ftg_{fid}=F(e(H_3(oid),h^{aSK_2})^{SK_{\mathrm{BC}}}\|H(fid))$，在BC中查询得到$ftg_{fid}$对应的辅助解密信息$aux=\{h^{\gamma_1+\gamma_2},g^{\xi_2}\}$，计算$h^{(\gamma_1+\gamma_2)SK_{\mathrm{BC}}}$，将$\{h^{(\gamma_1+\gamma_2)SK_{\mathrm{BC}}},g^{\xi_2}\}$发送给CSP.

3）CSP从DU_j发送的访问请求$req=\{oid,H(fid),OK\}$中获取其外包解密密钥OK，并基于BC发送的$\{h^{(\gamma_1+\gamma_2)SK_{\mathrm{BC}}},g^{\xi_2}\}$调用算法$OutDec(CT,OK,aux)\to CT_{\mathrm{out}}$得到外包解密密文$CT_{\mathrm{out}}$，该算法具体流程为：

CSP计算$\xi_1=H(h^{(\gamma_1+\gamma_2)SK_{\mathrm{BC}}}h^{(\gamma_1+\gamma_2)SK_{\mathrm{CSP}}})$，$g^{\eta+\xi}=ok_8g^{\xi_2}g^{-\xi_1}=(g^{\eta+\xi_2})g^{-\xi_1}$；基于线性秘密共享方案的子秘密还原过程，CSP构造常数$\{\omega_x\in\mathbb{Z}_p\}_{x\in[l]}$和$\{\beta_y\in\mathbb{Z}_p\}_{y\in[L]}$，使其满足$\sum\omega_x\boldsymbol{M}_x=(1,0,\cdots,0)$和$\sum\beta_y\boldsymbol{A}_y=(1,0,\cdots,0)$，其中$\boldsymbol{M}_x$代表矩阵$\boldsymbol{M}$的第$x$行、$\boldsymbol{A}_y$代表矩阵$\boldsymbol{A}$的第$y$行；CSP分别计算$E_1,E_2,E_3,E_4$，具体构造为：

$$E_1=e(ok_1,g^{\eta+\xi})=e((g^{\alpha/(a+z_j)}g_1^r)^{\eta},g^{\eta})e((g^{\alpha/(a+z_j)}g_1^r)^{\eta},g^{\xi}),$$

$$E_2=\prod_x[e(ok_4^{ok_2}ok_3,C_{1,x})e(ok_5,C_{2,x})e(ok_6,C_{3,x})]^{\omega_x}=\prod_x[e(g^{\eta rz_j}g^{\eta ar},g_1^{\lambda_x}g_4^{t_x})e(g^{\eta r_\tau},(g_2^{\delta(x)}g_3)^{-t_x})\times e((g_2^{A_\tau}g_3)^{\eta r_\tau}g_4^{-(a+z_j)r\eta},g^{t_x})]^{\omega_x}=\prod_x[e(g,g_1)^{\eta r(a+z_j)\lambda_x}]^{\omega_x}=e(g,g_1)^{\eta rs(a+z_j)},$$

$$E_3=\prod_y[e(g_1^{ok_2}h,C_{4,y})e(ok_7,C_{5,y})e(g_2^{ok_2},C_{6,y})]^{\beta_y}=\prod_y[e(g_1^{z_j}g^a,g^{\alpha_{\rho(y)}t_y})e(g_1^{\alpha_{\rho(y)}(a+z_j)},g^{-t_y})\times e(g^{z_j},g^{t_y}g_2^{\theta_y})]^{\beta_y}=\prod_y e(g^{z_j},g_2^{\theta_y})^{\beta_y}=1,$$

$$E_4=e(ok_1,C_0^{ok_2}C_0')=e((g^{\alpha/(a+z_j)}g_1^r)^{\eta},g^{sz_j}g^{as}g^{\xi})=e((g^{\alpha/(a+z_j)}g_1^r)^{\eta},g^{s(z_j+a)})e((g^{\alpha/(a+z_j)}g_1^r)^{\eta},g^{\xi})=e(g^{\alpha s},g)^{\eta}e(g_1^{\eta r},g^{s(z_j+a)})e((g^{\alpha/(a+z_j)}g_1^r)^{\eta},g^{\xi}),$$

CSP向DU_j发送外包解密算法$OutDec()$，输出结果$CT_{\mathrm{out}}=E_4/(E_1E_2E_3)=e(g,g)^{\alpha s\eta}/e((g^{\frac{\alpha}{a+z_j}}g_1^r)^{\eta},g^{\eta})$.

4）DU_j对CT_{out}调用算法$Dec(CT_{\mathrm{out}})\to m/\perp$得到明文$m$，算法具体流程为：

$$key=C/(CT_{\mathrm{out}}{}^{\eta^{-1}}e(ok_1,g))=key\times e(g,g)^{\alpha s}/e(g,g)^{\alpha s}.$$

DU_j验证$m\|H(m)=SDec_{key}(C_m)$是否成立，若该等式成立，则表示DU_j成功完成密文的访问；若不成立，则输出$\perp$.

3.5 访问行为审计

在该阶段，DO通过BLS聚合签名[31]对审计目标fid_{t}发起审计查询T_{Q}；BC查询节点QN收到T_{Q}后，调用查询匹配算法$QryInt()$对索引词典元素进行隐私集合求交，其结果为审计目标fid_{t}在索引词典中的元素ftg_{t}；QN根据ftg_{t}指向的索引列表$list_{fid_{\mathrm{t}}}$找到审计目标fid_{t}的访问记录C_{req}所在的区块编号，并调用身份追踪算法$Trace()$对访问记录中数据访问者身份进行抽取，并将结果返回给DO，从而实现密文场景下的用户访问行为身份审计，具体过程为：

1）DO调用算法$QryGen(fid_{\mathrm{t}},oSK)\to T_{\mathrm{Q}}$生成审计查询$T_{\mathrm{Q}}$，DO随机选择$\varphi\in G$，令$tq_1=e(PK_{\mathrm{BC}},H_3(ftg_{\mathrm{t}})\varphi)$，$tq_2=e(PK_{\mathrm{BC}},oSK\varphi)$；DO将审计查询$T_{\mathrm{Q}}=\{oid,tq_1,tq_2\}$发送给BC查询节点QN.

2）查询节点QN调用算法$QryInt(T_{\mathrm{Q}},idx)\to\{bn\}$得到区块编号集合$\{bn\}$. QN通过倒排索引$idx$的审计词典$dict_{oid}$得到候选向量$\boldsymbol{cand}=(e(PK_{\mathrm{BC}},H_3(ftg_i)))_{i\in[dict]}$，并基于$T_{\mathrm{Q}}$计算$tq_2'=tq_2/e(PK_{\mathrm{BC}},H_3(oid))^{aSK_{\mathrm{QN},2}}$，进而通过求交运算$\boldsymbol{cand}\cap(tq_1/tq_2')$得到索引列表$list_{fid_{\mathrm{t}}}$，QN通过$list_{fid_{\mathrm{t}}}$存储的区块编号标签$tag_{bn}$定位到BC

的第bn号区块，进而获得区块中存储的文件fid的访问记录密文$C_{req}=SEnc_{key_1}(req\|ts)$并对其解密：QN 计算$key_1=e(PK_{\mathrm{BC}},oPK)^{aSK_{\mathrm{QN},2}}$，得到访问记录$(req\|ts)=SDec_{key_1}(C_{req})$，并从$req$中抽取出 DU 的外包解密密钥$OK$.

3）查询节点 QN 调用算法$Trace(OK,aSK_i)\rightarrow uid$得到数据访问者身份信息，算法具体流程为：

QN 计算$(o_{j,1}\|o_{j,2})=SDec_{aSK_{\mathrm{QN},2}}(ok_2)$，访问请求$req$所对应的$DU_j$的身份信息$uid_j=SDec_{aSK_{\mathrm{QN},2}}(o_{j,1})$，将$C_{uid}=SEnc_{tq_2'}(uid_j)$发送给 DO.

4）DO 计算$SDec_{e(PK_{\mathrm{BC}},\varphi)}(C_{uid})$得到$DU_j$的身份信息$uid_j$.

4 方案分析

4.1 安全性证明

根据 2.3 节所述，本文方案的安全性证明包括选择性安全、审计正确性和审计过程的隐私安全. 在选择性安全证明部分，将证明本文方案的安全性和文献[23]的安全性在q-type 假设下是等价的. 令Σ_{RW}和Σ_{our}分别代表文献[23]和本文的 CP-ABE 加密方案.

定理 1. 若q-type 假设成立，则Σ_{our}在 2.3 节的游戏模型下是选择性安全的.

证明. 假设存在多项式时间敌手$\mathcal{A}$对于挑战矩阵$\boldsymbol{M}^*$拥有$Adv_{\mathcal{A},\Sigma_{\mathrm{our}}}$的优势选择性攻破本文方案$\Sigma_{\mathrm{our}}$，其中$\boldsymbol{M}^*$为$l\times n$矩阵且满足约束$l,n\leqslant q$. 则能够构造多项式算法$\mathcal{B}$，该算法拥有$Adv_{\mathcal{B},\Sigma_{\mathrm{RW}}}$的优势选择性攻破文献[23]方案$\Sigma_{\mathrm{RW}}$.

1）初始化. 敌手$\mathcal{A}$将挑战访问策略$(\boldsymbol{M}^*,\delta^*)$发送给算法$\mathcal{B}$和方案$\Sigma_{\mathrm{RW}}$，其中$\boldsymbol{M}^*$为$l\times n$矩阵且满足约束$l,n\leqslant q$，函数$\delta^*:\{0,1\}^l\rightarrow\mathbb{Z}_p$.

2）参数建立. Σ_{RW}将公共参数$pp_{\Sigma_{\mathrm{RW}}}$发送给$\mathcal{B}$，其构造为：

$$g=g,\ g_1=g^d,$$

$$g_2=g^{\tilde{b}}\prod_{(j,k)\in[l,n]}\left(g^{d^k/b_j^2}\right)^{M^*_{j,k}},$$

$$g_3=g^{\tilde{c}}\prod_{(j,k)\in[l,n]}\left(g^{d^k/b_j^2}\right)^{-\delta^*(j)M^*_{j,k}},$$

$$g_4=g^{\tilde{d}}\prod_{(j,k)\in[l,n]}\left(g^{d^k/b_j}\right)^{M^*_{j,k}},\ v=g^{d^{q+1}+\tilde{\alpha}}.$$

$\mathcal{B}$随机选择$a\in\mathbb{Z}_p^*$，计算$h=g_1^a$和$f=g^a$，通过$pp_{\Sigma_{\mathrm{RW}}}$得到$pp_{\Sigma_{\mathrm{our}}}=\{g,g_1,g_2,g_3,g_4,h,f,v,v^{-1}\}$，$\mathcal{B}$将$pp_{\Sigma_{\mathrm{our}}}$发送给敌手$\mathcal{A}$.

3）查询阶段 1. 敌手$\mathcal{A}$向算法$\mathcal{B}$提交(uid,S)以获取对应的解密密钥，$\mathcal{B}$将(uid,S)发送给Σ_{RW}以获得对应的解密私钥：

$$\hat{K}_0=g^{\tilde{\alpha}}(g^d)^{\tilde{r}}\prod_{i=2}^{n}(g^{d^{q+2-i}})^{\omega_i},$$

$$\hat{K}_1=g^{\tilde{r}}\prod_{i\in[n]}(g^{d^{q+2-i}})^{\omega_i},$$

$$\hat{K}_{\tau,2}=g^{\tilde{r}_\tau}\prod_{i'\in[l],\delta^*(i')\notin S}(g^{b_{i'}})^{\frac{\tilde{r}}{A_\tau-\delta^*(i')}}\times\prod_{i,i'\in[n,l],\delta^*(i')\notin S}(g^{b_{i'}d^{q+1-i}})^{\frac{\omega_i}{A_\tau-\delta^*(i')}},$$

$\hat{K}_{\tau,3}=\hat{\Psi}\hat{\Phi}$，其中

$$\hat{\Psi}=(g_2^{A_\tau}g_3)^{\tilde{r}_\tau}(\hat{K}_{\tau,2}/g^{\tilde{r}_\tau})^{\tilde{u}A_\tau+\tilde{h}}\times\prod_{(i',j,k)\in[l,l,n],\delta^*(i')\notin S}\left(g^{\frac{b_{i'}d^k}{b_j^2}}\right)^{\frac{\tilde{r}(A_\tau-\delta^*(j))\boldsymbol{M}^*_{j,k}}{A_\tau-\delta^*(i')}}\times\prod_{\substack{(i,i',j,k)\in[n,l,l,n],\\ \delta^*(i')\notin S,(i'\neq j\vee i\neq k)}}\left(g^{\frac{b_{i'}d^{q+1+k-i}}{b_j^2}}\right)^{\frac{\tilde{r}(A_\tau-\delta^*(j))\omega_i\boldsymbol{M}^*_{j,k}}{A_\tau-\delta^*(i')}},$$

$$\hat{\Phi}=g_4^{-\tilde{r}}\prod_{i\in[n]}(g^{d^{q+1-i}})^{-\tilde{b}\omega_i}\prod_{(i,j,k)\in[n,l,n],i\neq k}\left(g^{\frac{d^{q+1+k-i}}{b_j}}\right)^{-\omega_i\boldsymbol{M}^*_{j,k}}.$$

$\mathcal{B}$随机选择$z\in\mathbb{Z}_p^*$，令$r=\tilde{r}/(a+z)$和$dk_2=z$，随机选择$g'\in G$，计算$dk_1=(\hat{K}_0)^{1/(a+z)}$，$dk_2=z$，$dk_3=(\hat{K}_1)^{a/(a+z)}g'$，$dk_4=(\hat{K}_1)^{1/(a+z)}$，$dk_5=\{\hat{K}_{\tau,2}\}$，$dk_6=\{\hat{K}_{\tau,3}\}$，$dk_7=aPK^{a+z}\cdot g_2^z$.

$\mathcal{B}$将(uid,S)对应的解密密钥$DK=\{dk_i\}_{i\in[7]}$发送给$\mathcal{A}$.

4）挑战阶段. 敌手$\mathcal{A}$将 2 个等长的明文信息(key_0,key_1)发送给算法$\mathcal{B}$，$\mathcal{B}$将其提交给方案Σ_{RW}并获得其挑战密文：

$$\{\hat{C}=key_\beta T'e(g,g^s)^{\tilde{\alpha}},\ \hat{C}_0=g^s,$$

$$\hat{C}_{i,1}=g_1^{\tilde{\lambda}_i}(g^{s\cdot b_i})^{-\tilde{d}}\prod_{(j,k)\in[l,n],j\neq i}(g^{sd^kb_i/b_j})^{-\boldsymbol{M}^*_{j,k}},$$

$$\hat{C}_{i,2}=(g^{sb_i})^{-(\tilde{u}\delta^*(i)+\tilde{c})}\times\prod_{(j,k)\in[l,n],j\neq i}(g^{sd^kb_i/b_j^2})^{-(\delta^*(i)-\delta^*(j))\boldsymbol{M}^*_{j,k}},$$

$$\hat{C}_{i,3}=g^{t_i}=(g^{sb_i})^{-1}\},$$

其中T'是挑战项，g^s对应Σ_{RW}中的安全性假设.

$\mathcal{B}$基于返回密文构造$CT'=\{C=\hat{C},C_0=\hat{C}_0,\ C_0'=\hat{C}_0^a,\{C_{1,x}=\hat{C}_{i,1},C_{2,x}=\hat{C}_{i,2},C_{3,x}=\hat{C}_{i,3}\}_{x\in[l]}\}$，并将$\{C_{4,y},C_{5,y},C_{6,y}\}_{y\in[L]}$加入到$CT'$得到$CT$，然后$\mathcal{B}$将挑战密文$CT$发送给敌手$\mathcal{A}$.

5）查询阶段 2. 该阶段和查询阶段 1 相同.

6）猜测阶段. 敌手$\mathcal{A}$将挑战密文的猜测结果β'发送给算法$\mathcal{B}$，$\mathcal{B}$将β'发送给方案Σ_{RW}. 由于方案Σ_{our}的

公共参数、解密密钥和挑战密文与方案Σ_{RW}拥有相同的分布，因此有$Adv_{\mathcal{B},\Sigma_{RW}}=Adv_{\mathcal{A},\Sigma_{our}}$，即存在多项式时间敌手$\mathcal{A}$能够以$Adv_{\mathcal{A},\Sigma_{our}}$优势攻破本文方案，则多项式算法$\mathcal{B}$能够以$Adv_{\mathcal{B},\Sigma_{RW}}$优势攻破方案$\Sigma_{RW}$，与文献[23]相矛盾. 因此，若$q$-type假设成立，本文方案是选择性安全的. 证毕.

定理 2. 如果l-SDH假设成立，则本文方案在$q<l$条件下是可追踪的.

证明. 假设存在多项式时间敌手$\mathcal{A}$在与挑战者C的O次交互后能够以不可忽略的优势$Adv_{\mathcal{A}}$赢得2.3节的追踪游戏，则存在多项式算法$\mathcal{B}$能够通过敌手$\mathcal{A}$攻破l-SDH假设. 为不失普遍性，在下面的证明过程中，我们将省略授权代理AA_i和数据使用者DU_j的对应下标(i,j).

1)参数建立阶段. 算法$\mathcal{B}$对于给定的l-SDH实例$\{G,G_T,e,p,\bar{g},\bar{g}^x,\bar{g}^{x^2},\cdots,\bar{g}^{x^l}\}$构造系统参数.

2)查询阶段. 敌手$\mathcal{A}$构造O次密钥查询，挑战者C对所有密钥查询依次响应并发送对应的解密密钥给敌手$\mathcal{A}$.

3)挑战阶段. 敌手$\mathcal{A}$将挑战密钥DK发送给挑战者C，若DK能够通过密钥检测公式：$e(dk_6,g)e(uPK_2,dk_4)=e(dk_5^{A_T},g_2)e(dk_5,g_3)$，则敌手$\mathcal{A}$能够成功伪造有效的解密密钥，并实现解密密钥的不可追踪性，同样地，算法$\mathcal{B}$能够通过DK伪造$(z^*,g^*)\in\mathbb{Z}_p^*\times G$使其满足$g^*=\bar{g}^{1/(a+z)}$攻破$l$-SDH假设；由于外包解密密钥是由解密密钥转换得到且$dk_2=ok_2$，同时外包解密密钥OK通过Gamma签名与用户的公钥绑定，因此敌手$\mathcal{A}$同样无法伪造外包解密密钥.

由于l-SDH假设对于多项式时间算法是不可解的，多项式时间敌手$\mathcal{A}$无法以不可忽略的优势$Adv_{\mathcal{A}}$赢得追踪游戏，因此基于本文方案构造的解密密钥是可追踪的. 证毕.

定理 3. 本文方案构造的倒排索引结构及查询过程和返回结果是语义安全的.

证明. 多项式时间敌手$\mathcal{A}$选择2个包含相同文件数量的文档集合Π_0,Π_1，并将其发送给挑战者C. C随机选择$\beta=\{0,1\}$，对Π_β中的文件名$\{fid_i\}_{i\in[|\Pi_\beta|]}$依次生成文件名标签$\{ftg_i\}_{i\in[|\Pi_\beta|]}$并构造索引词典$dict$. 敌手$\mathcal{A}$被允许访问查询生成算法和索引匹配算法，若$\mathcal{A}$无法以高于1/2的概率正确地猜测$\beta$的结果，则本文构造的倒排索引结构、查询过程和返回结果是语义安全的.

1)索引隐私. 倒排索引idx由索引词典$dict$和索引列表$list$组成，由于$dict$和$list$中的元素均由伪随机函数F生成，且生成参数包含Diffie-Hellman密钥交换参数，因此敌手$\mathcal{A}$无法以关键词猜测攻击的方式构造$dict$和$list$中包含的文件名标签ftg和区块编号标签ntg，因此$\mathcal{A}$无法以不可忽略的优势正确地猜测出β的结果.

2)查询隐私. 在审计查询$T_Q=\{oid,tq_1,tq_2\}$中，tq_1由ftg_t和随机元素φ生成，tq_2由DO私钥oSK和随机元素φ生成，根据Decisional Bilinear Diffie-Hellman假设和BLS签名安全性证明，敌手$\mathcal{A}$无法从审计查询T_Q推测出DO的审计目标明文；并且由于随机元素φ的存在，为相同审计目标的查询输出引入了随机性.

3)访问模式隐私. DO的审计结果以密文形式获得，由于加密密钥$e(PK_{BC},\varphi)$中包含了随机元素φ，使得相同文件的审计结果在不同的审计查询中的返回结果也并不相同. 因此，敌手$\mathcal{A}$无法从返回结果中推测出DO的审计目标.

综上所述，本文方案构造的倒排索引结构及查询过程和返回结果是语义安全的. 证毕.

4.2 功能分析

功能性和方案开销是评估CP-ABE密文数据共享方案是否实用的2项重要指标. 在本节中，本文方案与现有相关研究[4,6,13,19,28,32]分别在功能性与方案开销方面进行对比.

本文方案主要关注如何在跨域密文数据共享服务下实现用户访问行为的可审计功能，因此，本文方案主要在可扩展性、跨域共享、可追踪性和行为审计4个应用和安全功能进行对比. 对比结果如表2所示.

1)可扩展性是指方案的初始化过程采用大属性域技术构造，生成的系统公共参数规模固定，当系统中访问属性集合中的元素数量发生变化时，系统无需对公共参数进行更新. 根据3.1节中*Setup*()算法的具体流程可知，本文方案的系统公共参数pp由9个双线性群G的元素构成，其参数规模不会跟随用户

Table 2 Features Comparison with Other Schemes

表 2 与其他方案的功能性对比

方案	可扩展性	跨域共享	可追踪性	行为审计
文献[4]	√	×	×	×
文献[6]	√	×	√	×
文献[13]	×	×	√	×
文献[19]	√	×	×	√
文献[28]	√	√	×	×
文献[32]	×	×	√	√
本文	√	√	√	√

访问属性规模的增长而变化，因此本文方案具有较好的可扩展性，尤其当系统用户规模较大、需要更多用户属性对其进行描述时，本文方案支持用户属性的动态变化，且方案公共参数的存储开销不会受其影响. 此外，对比文献 [4, 6, 19, 28] 中的公共参数规模固定，具备可扩展性，对比文献 [13, 32] 中的公共参数中包含了访问属性集合所构成的参数，因此对比文献 [13, 32] 在属性规模变化时需要重新生成公共参数，不具备可扩展性.

2）跨域共享是指方案支持多个授权代理在不同的管理域中对用户的访问属性进行管理，并为该管理域中的用户生成解密密钥，在用户访问密文时，无需考虑该用户的解密密钥是由哪个授权代理生成，只需要该解密密钥对应的访问属性满足密文中内嵌的访问策略就能够完成解密，跨域共享功能可以解决单密钥生成中心存在的性能与安全瓶颈，是一种更符合现实场景的应用功能. 在 3.2 节的 *KeyGen*() 算法中，数据访问者 uid_j 所属的 AA_i 基于全体 AA 的公钥 *aPK* 构造对应的解密密钥 $dk_{i,j,7}$. 在 3.3 节的 *Enc*() 算法中，DO 通过构建 2 个访问结构 $(\boldsymbol{M},\delta)$ 和 $(\boldsymbol{A},\rho)$ 分别对访问属性对应的秘密 s 和授权代理对应的秘密 0 进行线性共享，并把生成的子秘密 λ_x 和 θ_y 参与到数据加密过程，其中密文 $C_{4,y}$, $C_{5,y}$, $C_{6,y}$ 对应授权代理的子秘密 θ_y. 在 3.4 节的 *OutDec*() 算法中，CSP 基于线性秘密共享机制对授权代理的子秘密 θ_y 进行还原，其具体过程对应 3.4 节 *OutDec*() 算法中参数 E_3 的计算. 总体来说，通过在密钥生成算法、加密算法和外包解密算法中引入线性秘密共享机制使得 DU 能够采用跨域访问策略对数据进行加密，并允许不同授权代理管理的授权用户访问外包密文. 因此，本文方案具有跨域共享性质，出于相似的构造过程，文献 [28] 同样具有跨域共享性质，且文献 [4, 6, 13, 19, 32] 不具备该性质.

3）可追踪性是指方案的用户解密密钥基于白盒可追踪机制生成，系统在用户解密密钥生成过程中，将用户的身份信息内嵌入解密密钥当中，当系统发现解密密钥发生泄露后，能够从解密密钥中还原得到该用户的身份信息. 在 3.2 节的 *KeyGen*() 算法中基于白盒可追踪机制将用户的身份信息内嵌于解密密钥中，即数据访问者的身份信息 uid_j 被加密为随机数 z_j，并参与解密密钥 $dk_{i,j}$ 的生成. 在 3.3 节密文数据访问过程的外包密钥 *OK* 的生成过程中保证了 $dk_{i,j,2}=ok_{i,j,2}=z_j$. 在 3.5 节的 *Trace*() 算法中，AA_i（查询节点 QN）能够利用私钥 $aSK_{i,2}$ 对 z_j 解密得到该密钥所对应的 DU 身份信息 uid_j. 因此，该方案和其他使用白盒可追踪机制的对比文献 [6, 13, 32] 具有解密密钥的可追踪性，对比文献 [4, 19, 28] 则不具有可追踪性.

4）行为审计是指系统能够对授权用户的每一次合法访问操作进行身份审计，系统通过 BC 中记录的用户访问请求来获取该访问行为所对应的外包解密密钥，并通过白盒可追踪性从外包解密密钥中获取该密文访问行为所对应的用户身份信息，从而完成访问行为身份审计，需要注意的是文献 [32] 的行为审计功能仅能够对数据更新操作进行身份审计，而本文行为审计是针对授权用户的所有数据访问行为进行身份审计操作，具有更强的适用性. 本文方案的行为审计基于解密密钥的可追踪性实现，在 3.4 节密文数据访问过程中，BC 对数据访问者的访问请求 *req* 加密后进行存储，由于 *req* 中包含用户的外包解密密钥 *OK*，因此在 3.5 节的访问行为审计过程中，AA_i（查询节点 QN）能够利用私钥 $aSK_{i,2}$ 对其解密得到该密钥所对应的数据使用者身份信息 uid_j，进而实现授权用户访问行为的身份审计. 对比文献 [19, 32] 具备行为审计性质，对比文献 [4, 6, 13, 28] 不具备访问行为性质.

方案开销对比结果如表 3 所示，方案开销对比包括公共参数、解密密钥和密文的存储开销，以及加密算法、解密密钥生成算法和外包解密算法的计算开销. 符号 $|U|$ 表示整个属性域的数量，$|S|$ 表示用户属性集合的数量；$|l|$ 表示访问控制矩阵 $\boldsymbol{M}$ 的行数，即用户属性的数量；$|L|$ 表示访问控制矩阵 $\boldsymbol{A}$ 的行数，即授权集合 AA 的数量；$|G|$, $|G_T|$, $|\mathbb{Z}_p|$ 分别表示双线性群 G, G_T 和随机数所需的存储空间；σ 表示文献 [19] 中允许用户下载文件的最大次数；t_G 和 t_{GT} 分别表示双线性群 G 和 G_T 上一次指数运算的时间开销，t_e 表示双线性映射 $e:G\times G\to G_T$ 的时间开销. 表 4 给出了本文方案 11 个算法的具体开销，其中包括参数的存储开销和算法的计算开销，而忽略了哈希算法、对称加密算法和伪随机函数算法的计算开销. 其中，$|dict|$ 表示索引词典中元素个数.

4.3 倒排索引存储与更新开销

根据 3.3 节所述，DO 在数据加密阶段，调用 *IdxGen*() 算法将待加密上传的明文数据 m 所对应的文件名 *fid* 生成文件名标签 *ftg*，基于文件名标签生成倒排索引的索引词典 *dict* 并发送给 BC 查询节点 QN 进行存储，其中 $dict[i]=ftg_i$. 根据表 4 所示，DO 执行 *IdxGen*() 算法的计算开销为 $|dict|t_G$，存储开销为 $|dict||G|$，QN 处的存储开销为 $|dict||G|$.

Table 3 Comparison of Storage Overhead and Computation Overhead

表 3 存储开销与计算开销对比

方案	存储开销			计算开销		
	公共参数	解密密钥	密文	加密算法	解密密钥生成算法	外包解密算法
文献 [4]	$4\|G\|+2\|G_T\|$	$(6+3\|S\|)\|G\|$	$(3+3\|l\|)\|G\|+\|G_T\|$	$(3+3\|l\|)t_G+t_{GT}$	$(6+3\|S\|)t_G$	$(3+6\|l\|)t_G+6t_e$
文献 [6]	$6\|G\|+\|G_T\|$	$(3+2\|S\|)\|G\|$	$(2+3\|l\|)\|G\|+\|G_T\|$	$(2+3\|l\|)t_G+t_{GT}$	$(3+2\|S\|)t_G$	$(2+3\|l\|)t_e$
文献 [13]	$(12+2\|U\|)\|G\|+2\|G_T\|$	$(8+2\|S\|)\|G\|$	$(6+4\|l\|)\|G\|+\|G_T\|$	$(4+6\|l\|)t_G$	$(5+\|S\|)t_G$	$(1+2\|l\|)t_e$
文献 [19]	$7\|G\|+\|G_T\|$	$(2+2\|S\|)\|G\|$	$(2+3\|l\|+\sigma)\|G\|+\|G_T\|$	$(2+3\|l\|+\sigma)t_G+t_{GT}$	$(2+2\|S\|)t_G$	$(1+6\|l\|)t_G+3(1+\|l\|)t_e$
文献 [28]	$4\|G\|+\|G_T\|$	$(3+\|S\|)\|G\|$	$(1+4\|l\|)\|G\|+\|G_T\|$	$(1+6\|l\|)t_G+t_{GT}$	$(3+3\|S\|)t_G$	$(1+4\|l\|)t_e$
文献 [32]	$(7+\|U\|)\|G\|+\|G_T\|$	$(2+\|S\|)\|G\|$	$(3+\|l\|)\|G\|+\|G_T\|$	$(3+\|l\|)t_G+t_{GT}$	$(3+2\|S\|)t_G+(1+\|S\|)t_{GT}$	$(3+6\|l\|)t_e$
本文	$9\|G\|$	$(3+2\|S\|+\|L\|)\|G\|+\|\mathbb{Z}_p\|$	$(2+3\|l\|+3\|L\|)\|G\|+\|G_T\|$	$(2+3\|l\|+3\|L\|)t_G+t_{GT}$	$(6+2\|S\|+\|L\|)t_G$	$t_G+3(1+\|l\|+\|L\|)t_e$

Table 4 Storage Overhead and Computation Overhead for Parameters of Each Algorithm in Our Scheme

表 4 本文方案中各算法的参数存储开销与算法计算开销

算法	参数	参数存储开销	算法计算开销
Setup()	pp, msk	$9\|G\|$, $\|\mathbb{Z}_p\|$	$4t_G$
AASetup()	aPK, aSK	$\|L\|\|G\|$, $2\|L\|\|\mathbb{Z}_p\|$	$\|L\|t_G$
EntSetup()	PK_{BC}, SK_{BC}	$\|G\|$, $\|\mathbb{Z}_p\|$	t_G
	PK_{CSP}, SK_{CSP}	$\|G\|$, $\|\mathbb{Z}_p\|$	t_G
KeyGen()	oPK, oSK	$\|G\|$, $\|G\|$	t_G
	uPK, uSK, uDK	$2\|G\|$, $\|\mathbb{Z}_p\|$, $(3+2\|l\|+\|L\|)\|G\|+\|\mathbb{Z}_p\|$	$2t_G$, t_G, $(6+2\|S\|+\|L\|)t_G$
Enc()	aux, CT	$2\|G\|$, $(2+3\|l\|+3\|L\|)\|G\|+\|G_T\|$	$3t_G$, $(2+3\|l\|+3\|L\|)t_G+t_{GT}$
IdxGen()	idx	$\|dict\|\|G\|$	$\|dict\|t_G$
OutDec()	OK, CT_{out}	$(4+2\|l\|+\|L\|)\|G\|+\|\mathbb{Z}_p\|$, $\|G_T\|$	$4+2\|l\|t_G$, $t_G+3(1+\|l\|+\|L\|)t_e$
Dec()			$3t_e$
QryGen()	T_{Q}	$2\|G_T\|$	$2(t_G+t_e)$
QryInt()			$(2+\|dict\|)t_e$
Trace()			$0t_G+0t_{GT}+0t_e$

根据 3.4 节所述，DU 在密文数据访问阶段，向 BC 发送访问请求 $req=\{oid,H(fid),OK\}$，BC 查询节点 QN 根据访问请求 req 生成访问请求密文 C_{req}，并将 C_{req} 以交易事务的形式存储在新建区块之中，并基于该区块编号 bn 生成编号标签 ntg_{bn}，并将 ntg_{bn} 插入到索引词典 $dict$ 中元素 ftg_{fid} 所对应的索引列表 $list_{fid}$ 中，该更新过程由 QN 完成，QN 的计算开销为 $|list|t_G$，存储开销为 $|dict||list|$. 由于 DU 需要生成访问请求 req 以访问云端密文数据，因此，倒排索引的更新过程不会给 DU 带来额外的计算、存储开销. 因此，在 QN 处倒排索引构建和更新过程存储开销为 $(|dict|+|dict||list|)|G|$.

根据 3.5 节所述，DO 在访问行为审计阶段，调用 *QryGen*() 算法对审计目标 fid_t 发起审计查询 T_{Q}，BC 查询节点 QN 收到 T_{Q} 后调用 *QryInt*() 算法通过隐私集合求交技术对审计查询 T_{Q} 和倒排索引词典 $dict$ 进行求交，进而得到存储审计目标 fid_t 访问信息密文 C_{req} 的区块编号标签 ntg. 根据表 4 所示，DO 执行 *QryGen*() 算法的计算开销为 $2(t_G+t_e)$，存储开销为 $2|G_T|$，QN 执行 *QryInt*() 算法的计算开销为 $(2+|dict|t_e)$.

本文引入倒排索引的目的是针对遍历区块中存储访问请求密文数据的审计操作而设计的优化方法，根据上述描述，倒排索引的存储和更新过程大部分由 BC 中的查询节点 QN 完成，不会给 DO 和 DU 带来较大的计算、存储开销. 同时根据表 5 所示，当索引词典规模 $|dict|=20$ 和索引列表长度 $|list|=50$ 时，查询节点 QN 对于单个用户的倒排索引存储、维护和查询开销是可接受的.

Table 5 Inverted Index Storage and Calculation Overheads

表 5 倒排索引存储与计算开销

索引各个阶段	客户端		查询节点 QN 端	
	存储开销/kb	计算开销/ms	存储开销/kb	计算开销/ms
索引生成		229.2	20	
索引更新			1 020	573
索引查询	2	25.9		120.25

5 实　　验

5.1 实验参数设置

本文及对比方案在 Windows 10 操作系统上（AMD Ryzen 4800H 2.90 GHz CPU, 16 GB 内存）进行了仿真实验，所有方案均基于 jPBC(Java pairing based cryptography)[33] 实现，BC 功能通过 Caliper[34] 工具在 Docker 环境中部署 Fabric Blockchain 模拟平台，其中包含 2 个主节点、1 个排序节点和 1 个查询节点，并设置 AA 的数量为 4，使用 A 类椭圆曲线构建双线性群 G 和 G_T，其中用户下载最大次数 σ=20，$|\mathbb{Z}_p|$=160 b，$|G|=|G_T|$=1 024 b. 伪随机函数和对称加密算法采用 AES-128.

5.2 存储开销对比

本节将对本文方案和对比文献 [4, 6, 13, 19, 28, 32] 进行公共参数、解密密钥和密文的存储开销对比.

1）公共参数存储开销对比. 进行公共参数存储开销对比的目的是验证方案的可扩展性，即验证当系统中访问属性域的规模增长时公共参数规模的变化情况. 根据系统初始化算法 *Setup*() 所示，本文方案公共参数包含 9 个双线性群 G 元素、需要 9 216 b 的存储开销，公共参数存储开销为常数. 本文方案与对比方案的公共参数存储开销如图 3 所示，当用户属性 $|S|$=100 时，本文方案公共参数规模为 9.22 kb，文献 [4, 6, 13, 19, 28, 32] 公共参数规模分别为 6.14 kb, 7.17 kb, 219.14 kb, 8.19 kb, 5.12 kb, 24.19 kb.

本文方案与文献 [4, 6, 19, 28] 中的公共参数均使用大属性域技术构建，如图 3 所示，上述方案的公共参数规模不会跟随横坐标的属性数量的增长而变化. 此外根据表 2 所示，本文方案相较于文献 [4, 6, 19, 28] 在功能性方面覆盖更为全面，因此，在 3.1 节参数构建部分额外设置必要的参数来支持系统功能. 例如，相较于文献 [6]，为了实现跨域数据共享和访问行为审计，在公共参数构建过程中增加了参数 h, v, v^{-1}. 相较于文献 [28]，为了实现用户密钥的可追踪性和访问行为审计，在公共参数构建过程中增加了参数 g_4, h, f, v, v^{-1}，该参数分别在 3.2 节和 3.3 节中参与了 *KeyGen*() 算法、辅助解密信息生成过程和 *Enc*() 算法. 如上所述，本文方案在实用性和存储开销方面进行了一定的平衡，使得本文方案在功能性方面获得一定优势，但在公共参数存储开销方面要略高于文献 [4, 6, 19, 28]；同时，由于文献 [13, 32] 的公共参数构建过程并未采用大属性域技术，其公共参数规模随着属性数量的增长而扩大，本文方案的公共参数存储开销要优于文献 [13, 32].

2）解密密钥存储开销对比. 进行解密密钥存储开销对比的目的是验证方案中授权代理生成用户解密密钥时所需的通信代价以及 DU 使用访问密钥而付出的存储代价. 根据用户密钥生成算法 *KeyGen*() 所示，用户的解密密钥包含 3+2$|S|$+$|L|$个双线性群 G 元素和 1 个常数项，其中 $|S|$ 表示用户访问属性的数量，$|L|$ 表示系统中 AA 的数量. 本文方案与对比方案的解密密钥存储开销如图 4 所示，当用户属性 $|S|$=100 时，本文方案解密密钥规模为 211.96 kb，文献 [4, 6, 13, 19, 28, 32] 解密密钥规模分别为 313.34 kb, 207.87 kb, 212.99 kb, 206.85 kb, 105.47 kb, 104.45 kb.

从图 4 可以看出，本文方案与其他对比方案的解密密钥规模均伴随用户访问属性规模的增长而提升. 造成本文方案解密密钥存储开销较大的原因在于本文方案为了支持跨域共享功能，需要在用户解密密钥中额外加入 $|L|$ 个授权代理的属性，该属性对应解密密钥的 $dk_{i,j,7}$. 从表 3 可以看出，若本文方案不再支持跨域共享功能，则解密密钥将包含 3+2$|S|$个双线性

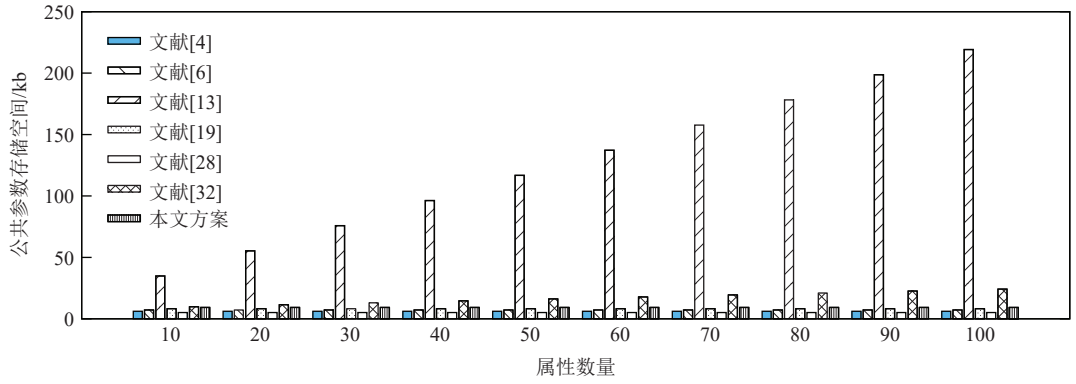

Fig. 3　Storage comparison of public parameter

图 3　公共参数存储对比

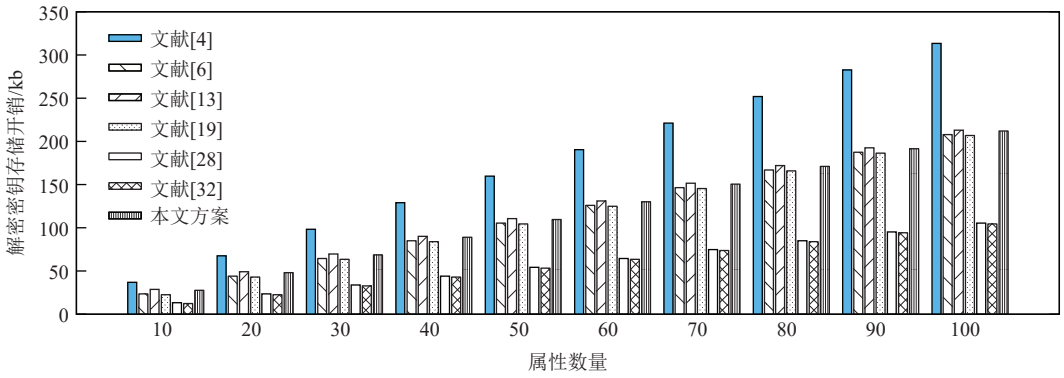

Fig. 4 Storage comparison of decryption key

图 4 解密密钥存储对比

群 G 元素和 1 个常数项, 其规模则略高于文献 [28,32], 但在功能性方面依然优于文献 [28,32].

3)密文存储开销对比. 进行密文存储开销对比的目的是验证方案的加密和上传过程给 DO 带来的通信开销以及给 CSP 带来的存储开销. 根据加密算法 *Enc*() 所示, 本文方案密文包含 $(2+3|I|+3|L|)|G|+|G_T|$ 个双线性群 G 和 G_T 元素, 其中 $|I|$ 表示数据拥有者构造访问策略所需的属性数量. 本文方案与对比方案的密文存储开销如图 5 所示, 当访问属性 $|I|$=100、授权代理数量 $|L|$=20 时, 本文方案密文规模为 322.56 kb, 文献 [4, 6, 13, 19, 28, 32] 密文规模分别为 311.30 kb, 310.27 kb, 416.77 kb, 330.75 kb, 411.65 kb, 106.50 kb.

从图 5 可以看出, 本文方案与其他对比方案的密文规模均伴随用户访问属性规模的增长而提升. 造成本文方案密文存储开销逊于文献 [4, 6, 32] 的原因在于本文方案为了让来自不同授权代理所属的管理域中的授权用户能够正常完成解密操作, 需要在加密过程中在密文内额外增加 $3|L|$ 个双线性群 G 元素, 对应密文 $C_{4,y}$, $C_{5,y}$, $C_{6,y}$, 该部分与解密密钥的 $dk_{i,j,7}$ 相对应. 从表 3 可以看出, 本文方案的密文规模受访问属性数量和授权代理数量影响, 当授权代理数量发生变化时, 本文方案的密文存储开销也会发生变化, 由于在实际应用中, 授权代理的数量一般保持不变, 因此本文方案密文规模也相对稳定.

5.3 计算开销对比

1)加密算法计算开销对比. 进行加密算法计算开销对比的目的是验证 DO 在加密明文数据时在本地带来的时间开销, 由于加密算法涉及大量的双线性群上的指数运算和双线性运算, 因此加密算法计算开销越小则表明加密过程所需的等待时间越小, 给 DO 端带来的计算开销也越小. 根据表 3 所示, 本文方案加密算法主要包含 $2+3|I|+3|L|$ 个双线性群 G 上的指数运算和 1 个双线性群 G_T 上的指数运算. 当访问属性 $|I|$=100、授权代理数量 $|L|$=20 时, 本文方案加密算法所需的时间开销为 3.60 s, 文献 [4, 6, 13, 19, 28, 32] 加密算法所需的时间开销分别为 1.19 s, 3.46 s, 6.92 s, 6.01 s, 6.89 s, 1.18 s.

从图 6 可以看出, 本文方案与其他对比方案的加密算法时间开销均伴随用户访问属性规模的增长而提升. 造成本文方案加密算法计算开销高于文献 [4, 6, 32] 的原因在于本文方案为了实现跨域共享功能, 在加密过程中需要额外计算 $3|L|$ 个双线性群 G 元素, 对应

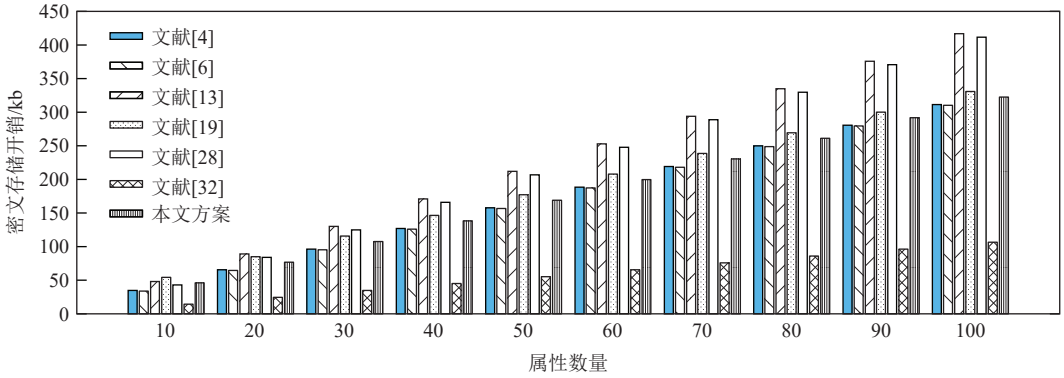

Fig. 5 Storage comparison of ciphertext

图 5 密文存储对比

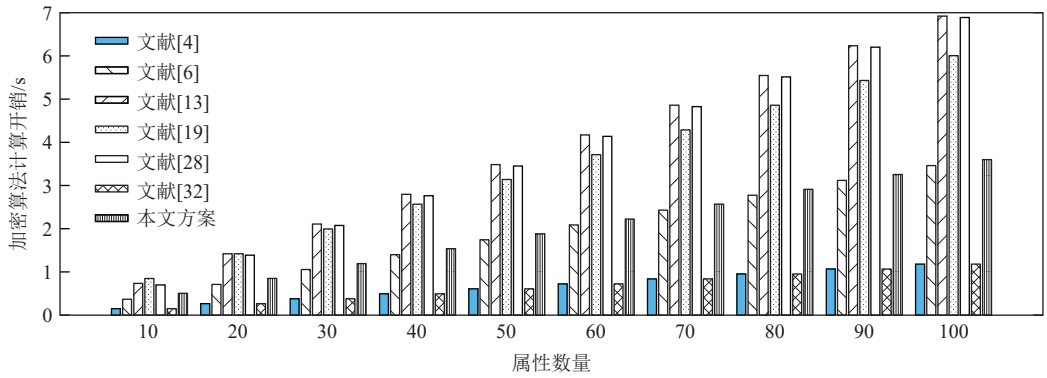

Fig. 6 Computation overhead comparison of *Enc*()

图 6 *Enc*() 计算开销对比

密文 $C_{4,y}$, $C_{5,y}$, $C_{6,y}$. 从表 3 可以看出，本文方案的加密算法时间开销受访问属性数量和 AA 数量影响，当授权代理数量发生变化时，本文方案的加密算法时间开销也会发生变化，由于在实际应用中，AA 的数量一般保持不变，因此本文方案加密算法的时间开销也相对稳定.

2）解密密钥生成算法计算开销对比. 进行解密密钥生成算法计算开销对比的目的是验证 DU 在向 AA 注册时所需要的响应时间. 根据表 3 所示，本文方案解密密钥算法主要包含 6+2|*S*|+|*L*|个双线性群 *G* 上的指数运算. 当用户属性|*S*|=100、授权代理数量|*L*|=20 时，本文方案解密密钥生成算法所需的时间开销为 2.38 s，方案 [4, 6, 13, 19, 28, 32] 解密密钥生成算法所需的时间开销分别为 3.51 s, 2.33 s, 1.20 s, 2.31 s, 3.47 s, 2.48 s.

从图 7 可以看出，本文方案与其他对比方案的解密密钥生成算法时间开销均伴随用户访问属性规模的增长而提升. 造成本文方案加密算法计算开销高于文献 [6, 13, 19] 的原因在于本文方案为了实现跨域共享功能，在解密密钥生成过程中需要额外计算|*L*|个双线性群 *G* 元素，该元素对应解密密钥的 $dk_{i,j,7}$.

3）外包解密算法计算开销对比. 进行外包解密算法计算开销对比的目的是验证 DU 在访问云端密文数据时所需要等待的时间开销，由于外包解密算法涉及大量的双线性群上的指数运算和双线性运算，因此加密算法计算开销越小则表明外包解密过程所需的计算量越小，给云端带来的计算开销也越小. 根据表 3 所示，本文方案外包解密算法主要包含 1 个双线性群 *G* 上的指数运算和 3(1+|*I*|+|*L*|) 个双线性运算. 当访问属性|*I*|=100、授权代理数量|*L*|=20 时，本文方案外包解密算法所需要的时间开销为 1.72 s，文献 [4, 6, 13, 19, 28, 32] 外包解密算法所需的时间开销分别为 6.94 s, 1.65 s, 1.10 s, 8.54 s, 2.19 s, 3.30 s.

从图 8 可以看出，本文方案与其他对比方案的外包解密算法时间开销均伴随用户访问属性规模的增长而提升. 造成本文方案外包解密算法计算开销高于文献 [6, 13] 的原因是在外包解密过程中需要额外计算 3|*I*|个双线性运算将密文的 $C_{4,y}$, $C_{5,y}$, $C_{6,y}$ 部分和解密密钥中的 $dk_{i,j,7}$ 相互抵消. 本文方案的外包解密算法计算开销受访问属性数量和 AA 数量的影响，当 AA 数量发生变化时，本文方案的外包解密算法时间开销也会发生变化，由于在实际应用中，AA 的数

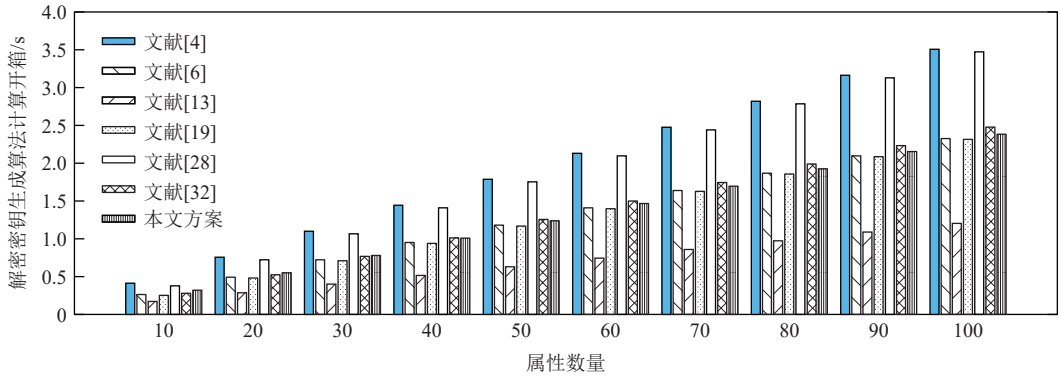

Fig. 7 Computation overhead comparison of *KeyGen*()

图 7 *KeyGen*() 计算开销对比

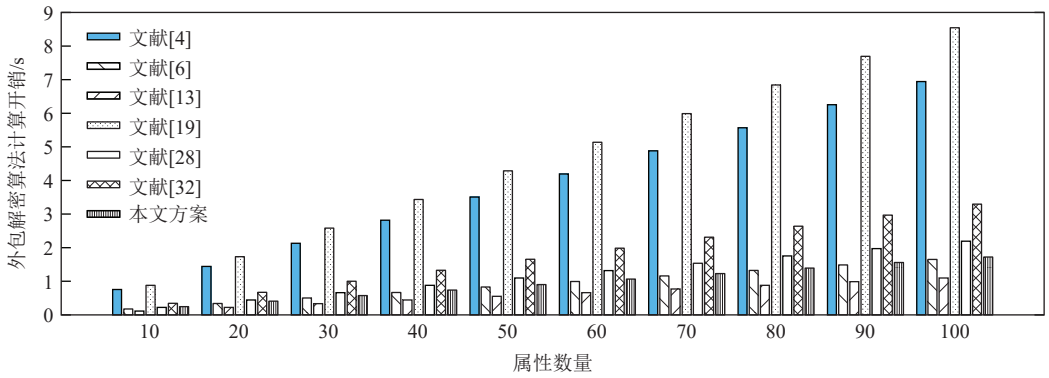

Fig. 8 Computation overhead comparison of *OutDec*()

图 8 *OutDec*() 计算开销对比

量一般保持不变, 因此本文方案外包解密算法计算开销也相对稳定.

5.4 倒排索引存储与计算开销

为了评估加密倒排索引对 BC 上遍历区块中密文内容的优化效果, 我们将审计目标文件的访问记录以随机分布的方式存储于 BC 的区块中, 令$|S|$=10, $|L|$=4, 令 Hyperledger Fabric 区块大小为 2 MB, 交易事务大小为 3 KB, 在考虑区块头部和 Merkle Tree 结构的存储开销情况下, 一个区块中最多能够存储 668 个交易事务; 假设目标文件访问记录的数量为区块数量的 5%, 依据分布情况构造对应的加密倒排索引, 其中倒排索引词典包含的最大元素数量$|dict|$=20, 传统的 BC 数据查询方法即基准方法(baseline), 需要对 BC 中每个区块进行遍历并对其内容进行解密, 本文的审计查询方法的时间开销则包括索引生成算法 *IdxGen*()、审计生成算法 *QryGen*()、查询匹配算法 *QryInt*() 和身份追踪算法 *Trace*() 的时间开销总和.

根据表 4 可知, 本文方案审计查询方法与区块规模无关, 仅与 DO 外包文件数量和文件访问次数成正比, 在不同的区块数量规模下, 索引的生成、维护与查询的开销较小. 在索引词典规模$|dict|$=20 和索引列表长度$|list|$=50 的约束条件下, 表 5 列出了倒排索引在生成、更新、查询过程中在客户端和 BC 查询节点 QN 端的存储与计算开销, 可以看出本文提出的倒排索引对客户端用户(DO 和 DU)是资源消耗友好的.

图 9 给出了基准方法与本文方法在时间开销上的对比, 可以看出基准方法的时间开销随着区块数量的增长而线性增长, 而本文的基于索引查询方法的时间开销则随着区块数量的增长以较低趋势增长, 即本文所提出的加密倒排索引结构对于区块中密文数据的遍历优化是有效的.

根据表 2、表 3 的理论分析和图 3~8 的实验结果

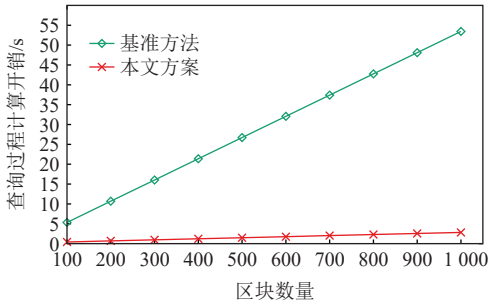

Fig. 9 Time overhead comparison of block traversal

图 9 区块遍历时间开销对比

对比可知, 本文方案在付出可接受的存储开销和计算开销代价的情况下, 在基于属性加密的密文共享领域提出了一个实用且安全合规的跨域密文数据共享方案, 本文方案支持密文场景的访问行为身份追踪, 这是现有研究所没有实现的; 同时实验证明了引入的加密倒排索引结构能够有效提升 BC 中密文数据的遍历效率. 因此, 本文方案是实用且高效的.

6 结 论

被广泛应用于云环境数据共享的属性基加密机制因其访问授权的匿名性, 在敏感数据应用场景(例如 EHR 数据共享)还存在安全性与合规性问题. 本文基于密文策略属性基加密(CP-ABE)机制设计了支持访问行为身份追踪的跨域密文共享方案, 并引入倒排索引结构优化访问行为身份追踪效率. 在后续的工作中, 我们将考虑通过可追踪属性签名[32]技术对访问请求的构造过程进一步优化, 实现访问权限的撤销功能和设计支持复杂审计功能的索引结构也是未来的研究方向.

作者贡献声明: 申远提出了方法思路和实验方

案设计，并撰写论文；宋伟负责改进方案并修改论文；赵常胜完成实验；彭智勇提出指导意见并修改论文. 宋伟(songwei@whu.edu.cn)和彭智勇（peng@whu.edu.cn）为通信作者.

参 考 文 献

[1] Li Fengqi, Liu Kemeng, Zhang Lupeng, et al. EHRChain: A blockchain-based EHR system using attribute-based and homomorphic cryptosystem[J]. IEEE Transactions on Services Computing, 2022, 15(5): 2755−2765

[2] Kshirsagar R, Hsu L Y, Greenberg C H, et al. Accurate and interpretable machine learning for transparent pricing of health insurance plans [C] // Proc of the 35th AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2021: 15127−15136

[3] Shen Jiayan, Zeng Peng, Choo K K R, et al. A certificateless provable data possession scheme for cloud-based EHRs[J]. IEEE Transactions on Information Forensics and Security, 2023, 18: 1156−1168

[4] Qin Xuanmei, Huang Yongfeng, Yang Zhen, et al. LBAC: A lightweight blockchain-based access control scheme for the Internet of things[J]. Information Sciences, 2021, 554: 222−235

[5] Zhang Kai, Ma Jianfeng, Zhang Junwei, et al. Online/Offline traceable attribute-based encryption[J]. Journal of Computer Research and Development, 2018, 55(1): 216−224 (in Chinese)
(张凯, 马建峰, 张俊伟, 等. 在线/离线的可追责属性加密方案[J]. 计算机研究与发展, 2018, 55(1): 216−224)

[6] Ning Jianting, Dong Xiaolei, Cao Zhenfu, 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] Liu Zhenhua, Ding Yingying, Yuan Ming, et al. Black-box accountable authority CP-ABE scheme for cloud-assisted E-health system[J]. IEEE Systems Journal, 2023, 17(1): 756−767

[8] Liu Zhen, Cao Zhenfu, Wong D S. White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures[J]. IEEE Transactions on Information Forensics and Security, 2013, 8(1): 76−88

[9] Ning Jiangting, Cao Zhenfu, Dong Xiaolei, 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, 2018, 15(5): 883−897

[10] Yang Yang, Liu Ximeng, Deng R H, et al. Lightweight sharable and traceable secure mobile health system[J]. IEEE Transactions on Dependable and Secure Computing, 2020, 17(1): 78−91

[11] Shen Yuan, Song Wei, Zhao Changsheng, et al. Secure access control for eHealth data in emergency rescue case based on traceable attribute-based encryption [C] // Proc of the 21st Int Conf on Trust, Security and Privacy in Computing and Communications. Piscataway, NJ: IEEE, 2022: 201−208

[12] Zhandry M. White box traitor tracing [G] // LNCS 12828: Proc of the 41st Annual Int Cryptology Conf. Berlin: Springer, 2021: 303−333

[13] Ziegler D, Marsalek A, Palfinger G. White-box traceable attribute-based encryption with hidden policies and outsourced decryption [C] // Proc of the 20th IEEE Int Conf on Trust, Security and Privacy in Computing and Communications. Piscataway, NJ: IEEE, 2021. 331−338

[14] Liu Zhen, Cao Zhenfu, Wong D S. Blackbox traceable CP-ABE: How to catch people leaking their keys by selling decryption devices on eBay [C] // Proc of ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2013: 475−486

[15] Xu Shengming, Yuan Jiaming, Xu Guowen, et al. Efficient ciphertext-policy attribute-based encryption with blackbox traceability[J]. Information Sciences, 2020, 538: 19−38

[16] Xu Shengming, Huang Xinyi, Yuan Jiaming, et al. Accountable and fine-grained controllable rewriting in blockchains[J]. IEEE Transactions on Dependable and Secure Computing, 2023, 18: 101−116

[17] Ahmad A, Saad M, AlGhamdi M A, et al. BlockTrail: A service for secure and transparent blockchain-driven audit trails[J]. IEEE Systems Journal, 2022, 16(1): 1367−1378

[18] Ruan P, Dinh T T A, Lin Q, et al. LineageChain: A fine-grained, secure and efficient data provenance system for blockchains[J]. The International Journal of Very Large Data Bases, 2021, 30(1): 975−988

[19] Wang Ti, Ma Hui, Zhou Yongbing, et al. Fully accountable data sharing for pay-as-you-go cloud scenes[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 18(4): 2005−2016

[20] Ning Jianting, Cao Zhenfu, Dong Xiaolei, et al. Auditable σ-time outsourced attribute-based encryption for access control in cloud computing[J]. IEEE Transactions on Information Forensics and Security, 2018, 13(1): 94−105

[21] Labadie C, Legner C. Personal data protection inside and out. Integrating data protection requirements in the data lifecycle[J]. Enterprise Modelling and Information Systems Architectures International Journal of Conceptual Modeling, 2020, 15(9): 1−20

[22] Beimel A. Secure schemes for secret sharing and key distribution [D]. Haifa, Israel: Israel Institute of Technology, 1996

[23] Rouselakis Y, Waters B. Practical constructions and new proof methods for large universe attribute-based encryption [C] // Proc of the 20th ACM SIGSAC Conf on Computer & Communications Security. New York: ACM, 2013: 463−474

[24] Goyal V. Reducing trust in the PKG in identity based cryptosystems [G] // LNCS 4622: Proc of the 27th Annual Int Cryptology Conf. Berlin: Springer, 2007: 430−447

[25] Ke Weiliang, Ge Chengyue, Song Wei. Executing efficient retrieval over blockchain medical data based on exponential skip bloom filter [G] // LNCS 13423: Proc of the 6th Web and Big Data Int Joint Conf. Berlin: Springer, 2022: 334−348

[26] Wang Bing, Song Wei, Lou Wenjin, et al. Inverted index based multi-keyword public-key searchable encryption with strong privacy guarantee [C] // Proc of the 34th IEEE Conf on Computer Communications. Piscataway, NJ: IEEE, 2015: 2092−2100

[27] Boneh D, Franklin M K. Identity based encryption from the Weil pairing [G] // LNCS 2139: Proc of the 21st Annual Int Cryptology Conf. Berlin: Springer, 2001: 213−229

[28] Yang Yang, Zheng Xianghan, Guo Wenzhong, et al. Privacy-preserving smart IoT-based healthcare big data storage and self-adaptive access control system[J]. Information Sciences, 2019, 479: 567−592

[29] Wang Jiabei, Zhang Rui, Li Jianhao, et al. Owner-enabled secure authorized keyword search over encrypted data with flexible

metadata[J]. IEEE Transactions on Information Forensics and Security, 2022, 17: 2746−2760

[30] Xiao Yue, Zhang Peng, Liu Yuhong. Secure and efficient multi-signature schemes for fabric: An enterprise blockchain platform[J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 1782−1794

[31] Boneh D, Drijvers M, Neven G. Compact multi-signatures for smaller blockchains [G] // LNCS 11273: Proc of the 24th Int Conf on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2018: 435–464

[32] Hou Huiying, Ning Jianting, Zhao Yunlei, et al. Fine-grained and controllably editable data sharing with accountability in cloud storage[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 19(5): 3448−3463

[33] Caro De A, Iovino V. jPBC: Java pairing based cryptography [C] // Proc of the 16th IEEE Symp on Computers and Communications. Piscataway, NJ: IEEE, 2011: 850−855

[34] Caliper. Hyperledger caliper, version 0.5 [CP/OL]. [2023-10-17]. https://hyperledger.github.io/caliper/

Shen Yuan, born in 1983. PhD, lecturer. Member of CCF. His main research interests include data security, differential privacy, and applied cryptography.

申　远, 1983 年生. 博士, 讲师. CCF 会员. 主要研究方向为数据安全、差分隐私、应用密码学.

Song Wei, born in 1978. PhD, associate professor. Senior member of CCF. His main research interests include big data management and analysis, applied cryptography, privacy protection, cloud security, and AI security.

宋　伟, 1978 年生. 博士, 副教授. CCF 高级会员. 主要研究方向为大数据管理与分析、应用密码学、隐私保护、云安全、人工智能安全.

Zhao Changsheng, born in 1999. Master. His main research interests include data security and applied cryptography.

赵常胜, 1999 年生. 硕士. 主要研究方向为数据安全、应用密码学.

Peng Zhiyong, born in 1963. PhD, professor. Fellow of CCF. His main research interests include database, big data management and analysis, trusted data management, and complex data management.

彭智勇, 1963 生. 博士, 教授. CCF 会士. 主要研究方向为数据库、大数据管理与分析、可信数据管理、复杂数据管理.