云计算中基于时间和隐私保护的可撤销可追踪的数据共享方案

张嘉伟，马建峰，马卓，李腾
（西安电子科技大学网络与信息安全学院，陕西 西安 710071）

摘 要：传统的密文策略属性基加密方案为云计算数据共享服务提供细粒度访问控制功能的同时，其访问策略中的明文属性会导致隐私和敏感数据泄露，而且根据恶意用户泄露的解密密钥对其进行高效追踪并撤销是一个挑战性问题，同时，大多数现有可撤销方案中都存在着撤销列表过长、效率过低等缺陷。针对这些问题，基于密文策略属性基加密方法，提出一种可撤销可追踪的基于时间并具有隐私保护的云数据共享方案。通过隐藏访问策略的属性值，所提方案支持单调且部分隐藏的访问策略和大规模属性空间，并使用层级的基于身份加密技术设置用户密钥有效期从而实现基于时间限制的数据访问控制。在此基础上，利用白盒追踪和二叉树技术，所提方案实现了高效的用户追踪和具有较短用户撤销列表的直接用户撤销，并使用在线/离线和可验证外包解密技术提高整体效率。最后，在判定性 q-BDHE 假设下，所提方案被证明是安全的。理论分析和实验结果显示，所提方案在时间和存储开销方面具有较高的性能。

关键词：密文策略属性基加密；云计算；基于时间的访问控制；白盒追踪；直接用户撤销
中图分类号：TP309
文献标识码：A
DOI: 10.11959/j.issn.1000−436x.2021206

Time-based and privacy protection revocable and traceable data sharing scheme in cloud computing

ZHANG Jiawei, MA Jianfeng, MA Zhuo, LI Teng
School of Cyber Engineering, Xidian University, Xi’an 710071, China

Abstract: General ciphertext-policy attribute-based encryption (CP-ABE) provides fine-grained access control for data sharing in cloud computing, but its plaintext formed access policy may cause leakage of private and sensitive data. And revoking a malicious user by accurately tracing the identity according to a leaked decryption key is a huge challenge. Moreover, most of existing revocable schemes incur long user revocation list and low efficiency. To solve these problems, a time-based and privacy preserving revocable and traceable data sharing scheme was proposed based on CP-ABE to support expressive monotonic and partial hidden access policy, large attribute universe by conceal the attribute values in access policy. Time-limited data access control using hierarchical identity-based encryption was achieved to set key valid period for users. Moreover, with the approaches of white-box tracing and binary tree, efficient user tracing and direct revocation with shorter revocation list was realized together with high efficiency via online/offline and verifiable outsourced decryption techniques. Furthermore, the scheme was secure under decisional q-BDHE assumption. Theoretical analysis and extensive experiments demonstrate its advantageous performance in computational and storage cost.

Keywords: CP-ABE, cloud computing, time-based access control, while-box tracing, direct user revocation

收稿日期：2021−07−22；**修回日期**：2021−09−15
通信作者：马卓，mazhuo@mail.xidian.edu.cn
基金项目：国家自然科学基金资助项目（No.61902291）；中国博士后基金资助项目（No.2019M653567）；陕西省自然科学基金资助项目（No.2019JM-425）；中央高校基本科研业务费专项资金资助项目（No.JB191507）

Foundation Items: The National Natural Science Foundation of China (No.61902291), China Postdoctoral Science Foundation (No.2019M653567), The Natural Science Foundation of Shaanxi Province (No.2019JM-425), The Fundamental Research Funds for the Central Universities (No.JB191507)

1 引言

由于云计算技术[1]能够减轻本地设备的计算和存储负担，越来越多的个人和组织用户将其数据外包并通过云计算提供方便的数据共享服务，用户可以随时随地访问并获取有用数据[2]。然而，共享在云计算的数据包含了大量的隐私和机密，一旦被非法访问，可能造成隐私泄露等重大事故。因此，数据的机密保护和访问控制成为云计算共享数据安全的关键[3]。密文策略属性基加密（CP-ABE, ciphertext-policy attribute-based encryption）方案[4-6]可以同时提供数据机密性和数据的细粒度访问控制，因此非常适于为云环境下的数据共享服务提供数据保护。目前，大量针对 CP-ABE 的研究工作提出了适于云计算环境下各种应用场景的安全数据共享[7-12]，并在效率、安全性、访问策略可表示性和属性空间规模等方面进行提升。然而，这些方案在隐私保护、用户追踪和撤销等方面还存在许多挑战性问题。

在多数 CP-ABE 方案中，访问策略往往以明文形式和密文一起托管在公有云中。拥有共享密文访问权限的用户都能够获取与其关联的访问策略。然而，访问策略属性中的敏感信息可能被泄露。例如，在智慧医疗系统中，如果一个医疗记录的访问策略为“癌症∧（医生∨护士）”，则对应患者的医疗记录等隐私信息存在泄露风险。因此，访问策略中包含敏感信息的属性必须被保护。为了解决这个问题，文献[13]提出了一种方案来隐藏访问策略属性值。该方案不仅能实现隐私保护，还支持灵活的可表示性和大规模属性空间，而且在标准模型上达到完全安全。但其基于合数阶群的双系统构造所带来的高复杂度和低性能不适合在实际中使用。为了解决效率问题，文献[14]基于素数阶群构建属性完全隐藏的方案，但其需要对访问策略和用户属性集合同时进行处理，因此效率也较低。为了提高实际运行效率，文献[15]基于素数阶群构造属性值隐藏的 CP-ABE 方案。

除此之外，在 CP-ABE 方案中存在许多恶意用户共享其解密密钥给非授权用户，从而牟取非法利益，而解密密钥仅取决于用户的属性，因此根据泄露的解密密钥追踪恶意用户的身份是一个难题。文献[16]在文献[13]的基础上提出了一种基于白盒追踪的 CP-ABE 方案，但其在合数阶群基础上的构造导致效率很低。同时，仅用户追踪对于 CP-ABE 系统是不够的，还需要有效的用户撤销机制，而用户撤销是 CP-ABE 的一个长期存在的热点问题。针对用户撤销，文献[17-18]提出了不同的基于素数阶群构建的可撤销 CP-ABE 方案，但是这些方案的效率都比较低。之后，文献[19-21]结合在线/离线技术和可验证外包解密方法极大地提高了 CP-ABE 中用户撤销的效率。与此同时，文献[22]则将用户撤销功能加入可追踪 CP-ABE 方案中。针对该方案的隐私保护、串谋攻击等问题，文献[23]在其基础上实现了隐私保护、用户追踪和撤销的功能，但是仍然存在低效率等不足。而且，上述可撤销 CP-ABE 方案在用户撤销列表上也有较高的开销，为了缩短用户撤销列表，文献[24]引入了基于时间的用户密钥和密文解密周期，从而在用户撤销列表中通过移除密钥失效的用户来缩短列表长度。但是该方案的效率很低且不具备可追踪和隐私保护的特性。

为了同时解决现存的基于 CP-ABE 且具有隐私保护的数据共享方案中存在的用户追踪和撤销以及相应的效率较低和列表开销较高等问题，本文提出一种基于时间和隐私保护的云数据共享方案。该方案在现有研究基础上，同时实现了 CP-ABE 的隐私保护、高效用户追踪和撤销以及较短的撤销列表，其效率也超过了相关工作。表 1 列出了本文方案和现存的一些相关工作的特性对比。其中，F1 表示用户撤销，F2 表示短撤销列表，F3 表示隐私保护，F4 表示用户追踪，F5 表示基于时间访问控制，F6 表示高效率，F7 表示大属性空间，F8 表示素数运算域。

表 1 本文方案和现存的一些相关工作的特性对比

方案	F1	F2	F3	F4	F5	F6	F7	F8
文献[19]	√	×	×	×	×	√	√	√
文献[20]	√	×	×	×	√	√	√	√
文献[24]	√	√	×	×	√	×	×	√
文献[13]	×	×	√	×	×	×	√	×
文献[16]	√	×	√	√	×	×	√	×
文献[23]	√	×	√	√	×	×	√	√
本文方案	√	√	√	√	√	√	√	√

本文主要的研究工作如下。

1) 本文提出一种基于时间并具有隐私保护的云数据共享方案。该方案针对外包在云计算中的共享数据提供了基于时间的细粒度访问控制，而且可

以有效保护共享数据的访问策略中包含的用户隐私。同时，故意泄露解密密钥的恶意用户可以被高效追踪并进行短撤销列表的直接撤销。

2) 为了实现云数据共享服务中的基于时间的细粒度访问控制，本文基于素数域构造设计了根据时间控制的密文策略属性基加密方案，通过引入时间周期的形式化表示方法，控制对用户密钥和共享数据的有效使用期限。同时，为了保护密文访问策略中的用户隐私，该方案通过隐藏访问策略的属性值从而实现用户隐私保护。

3) 本文方案不仅实现了对泄露解密密钥的恶意用户的高效追踪和低开销的直接用户撤销，还优化了用户端计算效率，利用在线/离线加密技术提升用户加密的效率，而且通过可验证外包解密技术将繁杂的用户解密操作卸载到云端，极大减少了用户解密的时耗。

4) 本文方案通过安全分析证明其在选择明文攻击下的不可区分性，即 IND-CPA（indistinguish ability under chosen-plaintext attack）攻击类型。同时，大量的实验结果展示了本文方案较高的时间效率与空间占用方面较好的效能。与当前相关方案的对比也验证了本文方案在实际运行中的高效性和实用性。

2　背景知识

2.1　访问结构

定义 1　访问结构。假设存在一个 n 元素集合 $\boldsymbol{Q}=\{Q_1,Q_2,\cdots,Q_n\}$，则 $\boldsymbol{Q}$ 上的访问结构 $\boldsymbol{P}$ 是一个集合，其元素为 $\boldsymbol{Q}$ 的非空子集，即 $\boldsymbol{P}\subset 2^{\boldsymbol{Q}}\setminus\{\varnothing\}$。如果存在 2 个集合 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 满足 $\boldsymbol{A}\in\boldsymbol{P},\boldsymbol{A}\subseteq\boldsymbol{B}$，有 $\boldsymbol{B}\in\boldsymbol{P}$，则称访问结构 $\boldsymbol{P}$ 为单调的。如果存在一个集合 $\boldsymbol{C}\in\boldsymbol{P}$，则称其为授权集合，否则称为非授权集合。

2.2　线性秘密共享方案

定义 2　线性秘密共享方案（LSSS, linear secret sharing scheme）。假设属性集合为 $\boldsymbol{U}_a$，$\overline{\boldsymbol{A}}=(a_1,a_2,\cdots,a_n)$ 是其中属性名称的集合，每个属性 $a_i\in\overline{\boldsymbol{A}}$ 对应一个属性值的集合 $\boldsymbol{V}_i=\{v_{i,1},v_{i,2},\cdots,v_{i,n_i}\}$，其中，$n_i$ 是属性值集合 $\boldsymbol{V}_i$ 的阶数。在 LSSS 中，$\boldsymbol{U}_a$ 上访问策略表示为 $\boldsymbol{A}=(\hat{\boldsymbol{A}},\rho,\boldsymbol{V})$，其中，$\hat{\boldsymbol{A}}$ 是一个 $l\times n$ 的秘密生成矩阵，ρ 将 $\hat{\boldsymbol{A}}$ 的每一行 $\hat{\boldsymbol{A}}_i$ 映射到属性名称索引的映射，$\boldsymbol{V}=(v_{\rho(1)},v_{\rho(2)},\cdots,v_{\rho(l)})$ 是访问策略中的属性值集合。对于一个秘密值 $s\in\mathbb{Z}_p$，随机选取一个向量 $\boldsymbol{b}=(s,b_2,\cdots,b_n)$，则 $\lambda_i=\hat{\boldsymbol{A}}_i\boldsymbol{b}$ 是属性 $a_{\rho(i)}$ 所对应的 s 的一个分享值。假设 $\boldsymbol{P}$ 是访问策略 $\boldsymbol{A}$ 的一个授权集合且 $\boldsymbol{I}=\{i:\rho(i)\in\boldsymbol{P}\}\subseteq[l]$，则存在一个常数组合 $\{\omega_i\in\mathbb{Z}_p\}$ 满足 $\sum_{i\in\boldsymbol{I}}\omega_i\hat{\boldsymbol{A}}_i=(1,0,\cdots,0)$ 从而恢复秘密值 s。假设用户 u 所关联的属性集合为 $\boldsymbol{S}_u=\{\boldsymbol{I}_u,\hat{\boldsymbol{S}}_u\}$，其中，$\boldsymbol{I}_u\subseteq\mathbb{Z}_p$ 为属性名称索引集合，$\hat{\boldsymbol{S}}_u=\{\alpha_i\}_{i\in\boldsymbol{I}_u}$ 为属性值集合。若 $\boldsymbol{S}_u$ 匹配 $\boldsymbol{A}$，则存在集合 $\boldsymbol{I}'\subseteq\mathbb{Z}_p$ 满足 $\{\rho(i)\,|\,i\in\boldsymbol{I}'\}\subseteq\boldsymbol{I}_u$，且对于每个 $i\in\boldsymbol{I}'$ 有 $v_{\rho(i)}=\alpha_{\rho(i)}$。

2.3　用户二叉树

假设 $\boldsymbol{U}$ 表示全体系统用户集合，**RL** 表示用户撤销列表，T_u 表示用户二叉树，用于管理系统用户，它具有以下特性。

1) 每个叶子节点和一个用户相关联。由于系统用户总数为 $|\boldsymbol{U}|$，则 T_u 的节点总数为 $2|\boldsymbol{U}|-1$ 且每个节点按照深度优先遍历的方式编号为 $0\sim 2|\boldsymbol{U}|-2$。

2) 路径 $\mathrm{path}(\eta)$ 表示从 T_u 的根节点到节点 η 的路径上的节点集合。

3) 最小覆盖集合 $\mathrm{cover}(\mathbf{RL})$ 表示可以覆盖用户撤销列表 **RL** 之外的所有未撤销用户所关联叶子节点的最小节点集合。

根据用户二叉树 T_u 的这些特征，如果一个用户 u 未被撤销且与 T_u 中的叶子节点 η_u 关联，则存在唯一的节点 $\eta_i=\mathrm{path}(\eta_u)\bigcap\mathrm{cover}(\mathbf{RL})$。

2.4　时间周期树

用户私钥或者密文解密的有效期一般可表示成几个时间段，为了减少有效期时间表示的空间和时间消耗，本文方案使用时间周期树的方法。

假设 T_t 表示深度为 d 的时间周期树，它的每个节点可以拥有最多 m 个子节点。根节点被赋值为 1，每个节点的子节点从 1 开始从左向右依次赋值。因此，每个节点可以被一个 m 进制序列表示（从根节点到当前节点的路径的序列表示），即 $\boldsymbol{\sigma}=(\sigma_1,\sigma_2,\cdots,\sigma_{l_\sigma}),l_\sigma\leqslant d,\sigma_i\in[m]$。同时，每个节点（根节点除外）都表示一个具体的时间周期（例如，天、周、月、年等）。如果一个有效期包含了多个时间周期，可以使用最小覆盖集合的方法对所有的有效时间周期进行表示，这样比枚举方法更简洁高效。

图 1 表示一个深度为 4 的时间周期树，其第一层叶子节点表示年，第二层叶子节点表示月，第三

层叶子节点表示天。如果一个用户的密钥在 2020 年 10 月 1 日到 2021 年 12 月 31 日期间有效，则该用户密钥的有效期可以表示为

$T_v = \{(2020,\text{Oct}),(2020,\text{Nov}),(2020,\text{Dec}),(2021)\}$

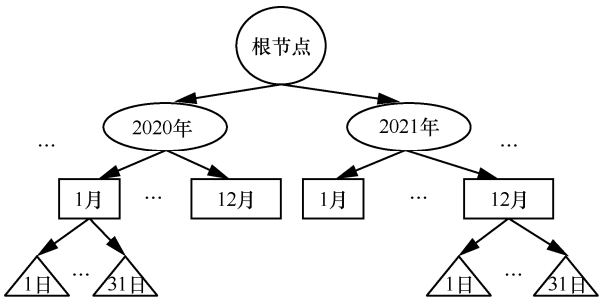


图 1 时间周期数

2.5 双线性群和复杂性问题假设

定义 3 双线性映射。假设 $\boldsymbol{G}_1$ 和 $\boldsymbol{G}_2$ 为 2 个阶为大素数 p 的乘法交换群，g 是 $\boldsymbol{G}_1$ 的一个生成元，则双线性映射 $\hat{e}:\boldsymbol{G}_1\times\boldsymbol{G}_1\to\boldsymbol{G}_2$ 满足以下条件。

1) 双线性。对于任意元素 $u,v\in\boldsymbol{G}_1, x,y\in\mathbb{Z}_p$，有 $\hat{e}(u^x,v^y)=\hat{e}(u,v)^{xy}$。

2) 非退化性。$\hat{e}(g,g)\neq 1$。

3) 可计算性。对于任意元素 $u,v\in\boldsymbol{G}_1$，存在一个高效的算法计算 $\hat{e}(u,v)$。同时称元组 $(\boldsymbol{G}_1,\boldsymbol{G}_2,p,\hat{e})$ 为一个素数双线性群。

定义 4 q-BDHE 假设。假设 $(\boldsymbol{G}_1,\boldsymbol{G}_2,p,\hat{e})$ 为素数双线性群及生成元 $g\in\boldsymbol{G}_1$，随机选择 $a,b\in\mathbb{Z}_p$ 和向量 $\boldsymbol{\gamma}=(g,g^b,g^a,\cdots,g^{a^q},g^{a^{q+2}},\cdots,g^{a^{2q}})$，则判定性 q-BDHE 问题为区分 $\hat{e}(g,g)^{a^{q+1}b}\in\boldsymbol{G}_2$ 和一个随机元素 $Y\in\boldsymbol{G}_2$。一个算法 $\prod$ 解决判定性 q-BDHE 问题的优势可以定义为

$$\text{Adv}=|\Pr[\prod(\boldsymbol{\gamma},\hat{e}(g,g)^{a^{q+1}b})]-\Pr[\prod(\boldsymbol{\gamma},Y)]|$$

判定性 q-BDHE 假设成立的必要条件是当且仅当不存在一个多项式时间算法以不可忽略的优势解决判定性 q-BDHE 问题。

3 系统定义和安全模型

3.1 系统模型和威胁模型

基于时间和隐私保护的可撤销可追踪数据共享的系统架构如图 2 所示，该系统包括属性机构、公有云、数据所有者和数据使用者 4 个实体。其中，属性机构负责管理用户属性、产生和分配用户密钥以及用户追踪和撤销，同时负责将更新密钥和最新的用户撤销列表发送至公有云进行密文更新；公有云向用户提供数据外包和共享服务并向授权用户提供外包解密服务，同时，当有用户被撤销时，公有云根据最新的用户撤销列表以及更新密钥对相应的密文进行更新，从而保证被撤销用户无法访问

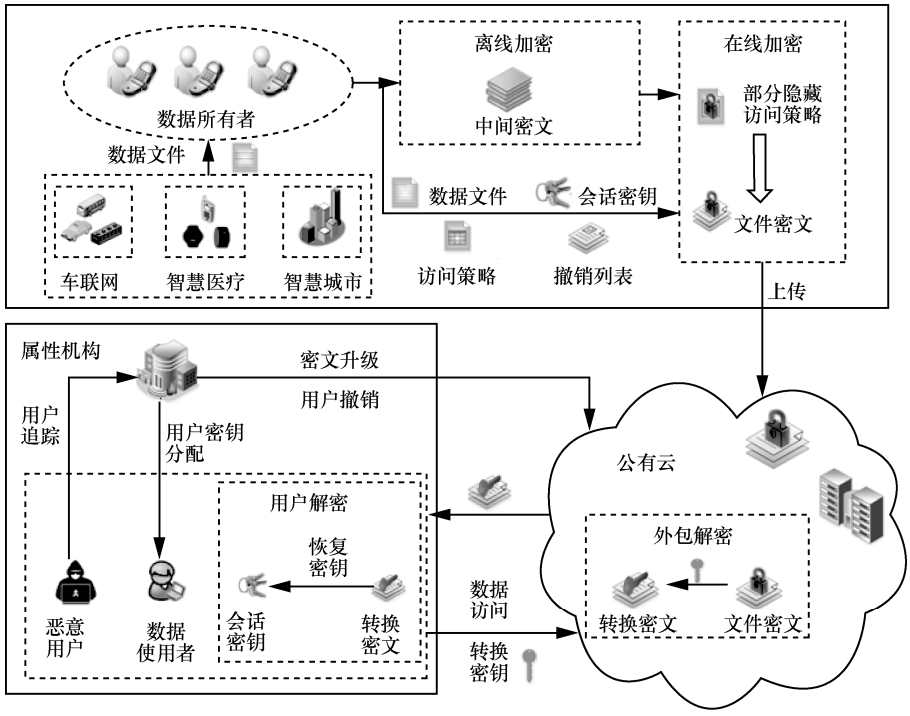


图 2 系统架构

撤销前加密的密文；数据所有者将其产生的数据经过离线和在线加密后上传到公有云进行数据共享；被授权的数据使用者向公有云请求所需数据并得到其转换密文，在经过用户解密后得到明文数据。

在系统运行中，公有云被认为是“诚实且好奇”的，该实体按照指定协议提供服务，但是可能会通过分析用户数据获取隐私和机密信息。数据所有者被认为是可信的且不与服务器进行串谋。权威机构被认为是可信的并且不与任何一方串谋。数据使用者被认为是不可信任的，存在一些恶意的用户非法访问共享数据并且泄露访问策略中包含的敏感和隐私信息，甚至篡改数据文件从而对用户隐私带来极大威胁。同时，一些数据使用者试图将其解密密钥和非授权用户分享从而获取额外利益。

根据以上威胁，本文方案需要满足以下要求。

1) 隐私保护：密文中的访问策略所包含的用户隐私等信息需要得到保护。

2) 高效用户追踪：对于泄露的解密密钥，需要能够高效并精确追踪到恶意用户的身份。

3) 短撤销列表用户撤销：在直接撤销恶意用户时，需要缩短用户撤销列表以节省开销。

4) 大规模属性空间：需要能够支持大规模属性空间，节省属性管理复杂度，提高灵活性。

3.2　系统定义

基于时间和隐私保护的可撤销可追踪数据共享方案由以下多项式时间算法组成。

$\text{SysSetup}(\kappa, d)$。该算法由属性机构执行。输入安全参数 κ 和时间树深度 d，输出系统公共参数 **PP** 和主密钥 **MSK**。

$\text{KeyGen}(\mathbf{PP}, \mathbf{MSK}, \boldsymbol{S}_u, T_v)$。该算法由属性机构执行。输入系统公共参数 **PP**、主密钥 **MSK**、用户 u 的属性集合 $\boldsymbol{S}_u=(\boldsymbol{I}_u, \hat{\boldsymbol{S}}_u)$ 及其密钥有效时间期限 T_v，输出用户 u 的私钥 $\boldsymbol{D}_u$。

$\text{TKeyGen}(\mathbf{PP}, \boldsymbol{D}_u)$。该算法由数据使用者执行。输入系统公共参数 **PP** 和用户 u 的私钥 $\boldsymbol{D}_u$，输出用户转换密钥 $\mathbf{TK}_u$ 和恢复密钥 $\mathbf{RK}_u$。

$\text{Encrypt}_{\text{off}}(\mathbf{PP})$。该算法由数据所有者执行。输入系统公共参数 **PP**，输出 2 个加密元素池。

$\text{Encrypt}_{\text{on}}(\mathbf{PP}, M, T_d, \mathbf{RL}, \boldsymbol{A})$。该算法由数据所有者执行。输入系统公共参数 **PP**、要加密的消息数据 M、密文可解密时间段 T_d、用户撤销列表 **RL** 以及一个指定的数据，输出相应的密文 **CT**。

$\text{Decrypt}_{\text{out}}(\mathbf{PP}, \mathbf{CT}, \mathbf{TK}_u)$。该算法由公有云执行。输入系统公共参数 **PP**、密文 **CT** 以及用户 u 的转换密钥 $\mathbf{TK}_u$，输出转换密文 $\mathbf{CT}^*$。

$\text{Decrypt}_u(\mathbf{PP}, \mathbf{CT}^*, \boldsymbol{D}_u', \mathbf{RK}_u)$。该算法由数据使用者执行。输入系统公共参数 **PP**、转换密文 $\mathbf{CT}^*$ 以及用户 u 的部分私钥 $\boldsymbol{D}_u'$ 和恢复密钥 $\mathbf{RK}_u$，输出解密验证后的明文消息 M'。

$\text{Trace}(\mathbf{PP}, \mathbf{RL}, \boldsymbol{D}_u)$。该算法由属性机构执行。输入系统公共参数 **PP**、用户撤销列表 **RL** 和用户私钥 $\boldsymbol{D}_u$，输出被追踪用户的身份 u 和更新后的用户撤销列表 $\mathbf{RL}'$。

$\text{CTUpdate}(\mathbf{PP}, \mathbf{RL}', \mathbf{CT})$。该算法由属性机构和公有云之间的交互来完成。输入更新后的用户撤销列表 $\mathbf{RL}'$，输出更新后的密文 $\mathbf{CT}'$。

3.3　安全模型

基于时间和隐私保护的可撤销可追踪数据共享方案的 IND-CPA 安全可以通过下述挑战者 C 和敌手 A 之间的安全游戏进行描述。

初始化。敌手 A 提交规模为 $l^* \times n^*$ 的挑战访问策略 $\boldsymbol{A}^* = (\hat{\boldsymbol{A}}^*, \rho^*, \boldsymbol{V}^*)$，用户撤销列表 $\mathbf{RL}^*$ 和解密时间周期 T_d^*，其中，访问策略的每个属性只出现一次，其属性值集合为 $\boldsymbol{V}^* = \{v_{\rho^*(i)}\}_{i\in[1,l^*]}$。

系统建立。C 运行 SysSetup 算法，生成系统公共参数和主密钥，并将系统公共参数发送给敌手 A。

询问 1。敌手 A 适应性地向挑战者 C 提交针对 q 个包含元组 $(u_1, \boldsymbol{S}_1, T_1), \cdots, (u_q, \boldsymbol{S}_q, T_q)$ 的用户密钥询问。若 $\boldsymbol{S}_i \notin \boldsymbol{A}^*$，或 $u_i \in \mathbf{RL}^*$，或 $T_d^* \notin T_i$，即属性集合 $\boldsymbol{S}_i$ 不满足挑战访问策略 $\boldsymbol{A}^*$，或用户 u_i 未被撤销，或挑战解密时间 T_d^* 不被密钥有效期 T_i 完全覆盖，则 C 计算对应的用户密钥并返回给敌手 A。

挑战者应答。敌手 A 输出 2 个等长消息 m_0 和 m_1 给挑战者 C。C 随机选择 $b \in \{0,1\}$，根据挑战访问策略 $\boldsymbol{A}^*$ 加密消息 m_b 得到密文 CT^* 并返回给 A。

询问 2。敌手 A 重复询问 1。

猜测。敌手 A 输出对 c 的猜测 c'，如果 $c = c'$，则敌手 A 赢得该安全游戏。其优势定义为

$$\text{Adv} = \left| \Pr[c = c'] - \frac{1}{2} \right|$$

定义 5　如果任意随机多项式时间（PPT, probabilistic polynomial time）敌手最多只能以一个可忽略的优势赢得上述安全游戏，则本文方案在选

择明文攻击下是不可区分性安全的。

4 具体方案构造

在本文中，每个属性包括属性名称和属性值两部分，系统时间使用时间周期树进行描述。访问策略可表示为 $\boldsymbol{A}=(\hat{\boldsymbol{A}},\rho,\boldsymbol{V})$，其中 $\hat{\boldsymbol{A}}$ 是一个 $l\times n$ 的秘密生成矩阵，ρ 是一个从 $\hat{\boldsymbol{A}}$ 的每一行到属性名称索引的映射，$\boldsymbol{V}$ 是访问策略中的属性值集合。以下是本文方案的具体构造。

SysSetup(κ,d)。属性机构首先生成一个双线性群 $(\boldsymbol{G}_1,\boldsymbol{G}_2,\hat{e},p)$ 并随机选择一个生成元 $g_0\in\boldsymbol{G}_1$ 以及几个元素 $\mu,\nu\in_{\mathrm{R}}\boldsymbol{G}_1$ 和 $x,y\in_{\mathrm{R}}\mathbb{Z}_p$，选取一个随机对称加密算法 (E_s,D_s) 和一个随机对称密钥 k_s。同时，创建一个用户二叉树 T_u 和深度为 d 的系统时间树 T_t，其中，用户二叉树 T_u 的每个叶子节点和一个用户 u 相关联并赋予一个唯一值 $v_u\in_{\mathrm{R}}\mathbb{Z}_p$。对于 T_u 中的每个节点 η_j，选择 $\xi_j\in_{\mathrm{R}}\mathbb{Z}_p$ 得到 $\{\xi_j\}_{j=0}^{2|U|-2}$ 并且计算 $\{\zeta_j=g_0^{\xi_j}\}_{j=0}^{2|U|-2}$。接着，随机选取元素 $L_0,L_1,\cdots,L_d\in_{\mathrm{R}}\boldsymbol{G}_1$ 并输出系统公共参数 **PP** 和主密钥 **MSK**

$$\begin{aligned}\mathbf{PP}=&\{\boldsymbol{G}_1,\boldsymbol{G}_2,\hat{e},p,g_0,\mu,\nu,\zeta_0,\cdots,\zeta_{2|U|-2},\\&L_0,L_1,\cdots,L_d,\hat{e}(g_0,g_0)^y,g_0{}^x,(E_s,D_s)\}\\\mathbf{MSK}=&\{g_0{}^y,x,k_s,\boldsymbol{\xi}=\{\xi_j\}_{j=0}^{2|U|-2}\}\end{aligned}\tag{1}$$

KeyGen$(\mathbf{PP},\mathbf{MSK},\boldsymbol{S}_u=\{\boldsymbol{I}_u,\boldsymbol{S}_u\},T_v)$。属性机构为合法用户 u 按照如下步骤生成其私钥。

首先，假设 T_u 中和用户 u 相关联的叶子节点 η_u 的路径为 $\mathrm{path}(\eta_u)=\{\eta_0,\eta_1,\cdots,\eta_u\}$，其中 η_0 为 T_u 的根节点，选择随机数 $\tau_u\in\mathbb{Z}_p$ 和 $\{\xi_j\}_{\eta_j\in\mathrm{path}(\eta_u)}$，同时计算 $\{\hat{\xi}_j=\xi_j\tau_u\}_{\eta_j\in\mathrm{path}(\eta_u)}$ 作为用户密钥的一部分并计算 $\gamma=E_s(k_s,v_u)$ 和 $D_0=g_0^{r_u/\xi_{\eta_u}}$。

其次，选择随机数 $r_u\in\mathbb{Z}_p$，计算 $D_1=g_0^{xr_u}$，$D_2=g_0^{r_u},D_3=\gamma$。对于用户 u 的每个属性值 $\alpha_i\in\hat{\boldsymbol{S}}_u$，计算 $D_i=g_0^{\alpha_i r_u}\nu^{-(x+\gamma)r_u}$。

再次，假设用户密钥有效期限 T_v 的覆盖子集为 $\boldsymbol{T}$，其中元素 $\boldsymbol{\sigma}\in\boldsymbol{T}$ 可以表示为 $\boldsymbol{\sigma}=\left(\sigma_1,\sigma_2,\cdots,\sigma_{l_\sigma}\right),l_\sigma\leqslant d$。对于每个元素 $\sigma\in\boldsymbol{T}$，选择一个随机数 $\tau_\sigma\in\mathbb{Z}_p$ 并且计算 $D_{\sigma,1}=g_0^{\tau_\sigma}$，$D_{\sigma,2}=g_0^{\frac{y}{x+\gamma}}\mu^{r_u}(L_0\prod_{j=1}^{l_\sigma}L_j^{\sigma_j})^{\tau_\sigma}$，$D_{\sigma,3}=g_0^{x\tau_\sigma}$，$D_{\sigma,k}=L_k^{\tau_\sigma}$，$k\in\{l_\sigma+1,\cdots,d\}$。

最后，该算法输出用户 u 的私钥

$$\begin{aligned}\boldsymbol{D}_u=&\{\boldsymbol{S}_u,T_v,D_0,D_1,D_2,D_3,\{D_i\}_{i\in\boldsymbol{I}_u},\{D_{\sigma,1},\\&D_{\sigma,2},D_{\sigma,3},D_{\sigma,l_k+1},\cdots,D_{\sigma,d}\}_{\sigma\in\boldsymbol{T}},\{\hat{\xi}_\eta\}_{\eta\in\mathrm{path}(\eta_u)}\}\end{aligned}\tag{2}$$

TKeyGen$(\mathbf{PP},\boldsymbol{D}_u)$。数据使用者 u 首先选择一个随机数 $z_u\in\mathbb{Z}_p$ 作为其恢复密钥 RK_u 并按照如下方式生成其转换密钥。

$$\begin{aligned}\mathbf{TK}_u=&\{K_1,K_2,K_3,\{K_i\}_{i\in\boldsymbol{I}_u},\{K_{\sigma,1},\\&K_{\sigma,2},K_{\sigma,3},K_{\sigma,l_k+1},\cdots,K_{\sigma,d}\}_{\boldsymbol{\sigma}\in\boldsymbol{T}}\}\end{aligned}\tag{3}$$

其中，

$$\begin{aligned}&K_1=D_1^{1/z_u},K_2=D_2^{1/z_u},K_3=D_3\\&\{K_i=D_i^{1/z_u}\}_{i\in\boldsymbol{I}_u},\{K_{\sigma,1}=D_{\sigma,1}{}^{1/z_u}\\&K_{\sigma,2}=D_{\sigma,2}{}^{1/z_u},K_{\sigma,3}=D_{\sigma,3}{}^{1/z_u}\\&K_{\sigma,l_k+1}=D_{\sigma,l_k+1}{}^{1/z_u},\cdots,K_{\sigma,d}=D_{\sigma,d}{}^{1/z_u}\}_{\boldsymbol{\sigma}\in\boldsymbol{T}}\end{aligned}\tag{4}$$

最后，输出用户恢复密钥 RK_u 和转换密钥 $\mathbf{TK}_u$。

$\mathrm{Encrypt}_{\mathrm{off}}(\mathbf{PP})$。数据所有者按照如下步骤生成中间密文。

首先，选择一个随机元素 $R_u\in G_1$ 和一个随机秘密值 $s\in\mathbb{Z}_p$，计算会话密钥 $k_u=H_0(R_u)$ 和 $C_0=R_u\hat{e}(g_0,g_0)^{ys}$，$C_1=g_0{}^s$，$\hat{C}_1=g_0{}^{xs}$。同时，对系统公共参数中的 ζ_j 计算 $\{\zeta_j{}^s\}_{j=0}^{2|U|-2}$ 且构造主密文模块 $\mathbf{IT}_m=\{(s,R_u,k_u,C_0,C_1,\hat{C}_1,\{\zeta_j{}^s\}_{j=0}^{2|U|-2})\}$。

其次，对于 $j\in[|\boldsymbol{P}|]$，选择 $\lambda_j{}',\beta_j,\nu_j\in_{\mathrm{R}}\mathbb{Z}_p$，计算 $C_{j,1}{}'=\mu^{\lambda_j{}'}\nu^{\beta_j},C_{j,2}{}'=g_0{}^{-\beta_j\nu_j}g_0{}^{\lambda_j{}'},C_{j,3}{}'=g_0{}^{\beta_j}$ 并生成属性密文模块

$$\mathbf{IT}_a=\{(\lambda_j{}',\beta_j,\nu_j,C_{j,1}{}',C_{j,2}{}',C_{j,3}{}')\}\tag{5}$$

最后，该算法输出中间密文 $\mathbf{IT}=\{\mathbf{IT}_m,\mathbf{IT}_a\}$。

$\mathrm{Encrypt}_{\mathrm{on}}(\mathbf{PP},M,\mathbf{IT},T_d,\mathbf{RL},\boldsymbol{A})$。数据所有者按照如下步骤计算密文。

首先，从中间密文的主密文模块 $\mathbf{IT}_m$ 中随机选择一个元组 $(s,R_u,k_u,C_0,C_1,\hat{C}_1,\{\zeta_j{}^s\}_{j=0}^{2|U|-2})$，计算消息验证码 $V_m=H(R_u,M)$ 并对消息数据 M 加密得到密文元素 $C_s=E_s(k_u,M)$。

其次，将密文解密周期 T_d 按照时间周期树的方法表示为 $\boldsymbol{\sigma}_d=(\sigma_1,\cdots,\sigma_{l_d})$，其中 $\boldsymbol{\sigma}_d$ 是一个 m 进制序列且 $l_d<d$ 并计算 $C_2=(L_0\prod_{j=1}^{l_d}L_j^{\sigma_j})^s$，从而将 T_d 嵌入密文中。

接着，选择一个随机向量 $\boldsymbol{b}=(s,b_2,\cdots,b_n)$，对于矩阵 $\hat{\boldsymbol{A}}$ 的每一行 $\hat{\boldsymbol{A}}_x$，计算 $\lambda_x=\hat{\boldsymbol{A}}_x\boldsymbol{b}$，随机选择元组 $(\lambda_j^{'},\beta_j,v_j,C_{j,1}^{'},C_{j,2}^{'},C_{j,3}^{'})\in\mathbf{IT}_a$ 并计算如下密文元素。

$$\begin{aligned}&C_{x,1}=C_{x,1}^{'},C_{x,2}=C_{x,2}^{'},C_{x,3}=C_{x,3}^{'},\\&C_{x,4}=\lambda_x-\lambda_x^{'},C_{x,5}=\beta_x(v_{\rho(x)}-v_x^{'})\end{aligned}\tag{6}$$

其中，$v_{\rho(x)}$ 是访问策略中属性 $\rho(x)$ 对应的属性值。

再次，通过用户二叉树算法 cover(**RL**) 得到撤销列表的最小覆盖集合，根据该集合中的各节点选择密文元素 $\{E_j=\zeta_j^{s}\}_{j\in\text{cover}(\mathbf{RL})}$。

最后，得到最终密文

$$\begin{aligned}\mathbf{CT}=\{&\bar{\boldsymbol{A}},\mathbf{RL},T_d,V_m,C_s,C_0,C_1,\hat{C}_1,C_2,\{C_{x,1},\\&C_{x,2},C_{x,3},C_{x,4},C_{x,5}\}_{x\in[l]},\{E_j\}_{j\in\text{cover}(\mathbf{RL})}\}\end{aligned}\tag{7}$$

其中，$\bar{\boldsymbol{A}}=(\hat{\boldsymbol{A}},\rho)$ 为隐藏属性值后的访问策略。该密文 **CT** 生成后被上传到公有云。

$\text{Decrypt}_{\text{out}}(\mathbf{PP},\mathbf{CT},\mathbf{TK}_u)$。公有云首先检测用户 u 是否在密文 **CT** 的用户撤销列表 **RL** 中，如果在，则算法终止，返回失败；否则，根据用户属性集合和密文访问策略，获取一个行索引集合 $\boldsymbol{I}_r=\{i:\rho(i)\in\boldsymbol{I}_u\wedge i\in[l]\}$ 和一个常量集合 $\{\omega_i\in\mathbb{Z}_p\}_{i\in\boldsymbol{I}_r}$ 使当 $\{\lambda_i\}$ 有效时，$\sum_{i\in\boldsymbol{I}_r}\omega_i\lambda_i=s$ 成立。同时，如果用户私钥中嵌入的属性值 α_i 和密文中嵌入的访问策略值 $v_{\rho(i)}$ 一致（$v_{\rho(i)}=\alpha_{\rho(i)}$），可以得到

$$\begin{aligned}&P_1=\hat{e}(K_2^{K_3}K_1,C_{i,1}\mu^{C_{i,4}})\cdot\\&\hat{e}(K_2,C_{i,2}g_0^{C_{i,4}-C_{i,5}})\hat{e}(K_{\rho(i)},C_{i,3})=\\&\hat{e}(g_0^{(x+\gamma)r_u/z_u},\mu^{\lambda_i}v^{\beta_i})\hat{e}(g_0^{r_u/z_u},g_0^{-\beta_i v_{\rho(i)}}g_0^{\lambda_i})\cdot\\&\hat{e}(g_0^{\alpha_{\rho(i)}r_u/z_u}v^{-(x+\gamma)r_u/z_u},g_0^{\beta_i})=\\&\hat{e}(g_0,\mu)^{(x+\gamma)r_u\lambda_i/z_u}\hat{e}(g_0,g_0)^{r_u\lambda_i/z_u}\\&P_2=\prod_{i\in\boldsymbol{I}_c}(P_2)^{\omega_i}=\\&\hat{e}(g_0,\mu)^{(x+\gamma)r_u s/z_u}\hat{e}(g_0,g_0)^{r_u s/z_u}\end{aligned}\tag{8}$$

其次，如果用户转换密钥有效期 T_v 完全覆盖了密文解密有效期 T_d，则有 $\boldsymbol{\sigma}_d=\{\sigma_1,\cdots,\sigma_l\}\in\boldsymbol{T}$ 并获取 $K_{\sigma_d,2}$，否则，搜索 $\boldsymbol{\sigma}_d$ 的一个前缀节点 $\boldsymbol{\sigma}_d^{'}=\{\sigma_1,\cdots,\sigma_{l'}\},l'<l$ 满足 $\boldsymbol{\sigma}_d^{'}\in\boldsymbol{T}$，根据 $\boldsymbol{\sigma}_d^{'}$ 获取 $K_{\sigma^{'},2}$ 并计算 $K_{\sigma_d,2}=K_{\sigma_d^{'},2}\prod_{k=l'+1}^{l}K_{\sigma_d^{'},k}^{\sigma_k}$。然后计算如下中间变量。

$$\begin{aligned}&P_3=\hat{e}(K_{\sigma,2},C_1^{K_3}\hat{C}_1)=\\&\hat{e}(g_0^{y/(x+\gamma)z_u}\mu^{r_u/z_u}(L_0\prod_{j=1}^{l_\sigma}L_j^{\sigma_j})^{\tau_\sigma/z_u},g_0^{(x+\gamma)s})=\\&\hat{e}(g_0,g_0)^{ys/z_u}\hat{e}(g_0,\mu)^{(x+\gamma)r_u s/z_u}\cdot\\&\hat{e}(g_0,(L_0\prod_{j=1}^{l_\sigma}L_j^{\sigma_j}))^{(x+\gamma)\tau_\sigma s/z_u}\\&P_4=\hat{e}(K_{\sigma,1}^{K_3}K_{\sigma,3},C_2)=\\&\hat{e}(g_0,(L_0\prod_{j=1}^{l}L_j^{\sigma_j}))^{(x+\gamma)\tau_\sigma s/z_u}\end{aligned}\tag{9}$$

再次，计算转换密文元素

$$C_{\text{t}}=\frac{P_3}{P_2P_4}=\frac{\hat{e}(g_0,g_0)^{ys/z_u}}{\hat{e}(g_0,g_0)^{r_u s/z_u}}\tag{10}$$

最后，该算法输出最终的转换密文 $\mathbf{CT}^{*}=\{\mathbf{RL},V_m,C_t,C_s,C_0,\{E_j\}_{j\in\text{cover}(\mathbf{RL})}\}$。

$\text{Decrypt}_u(\mathbf{PP},\mathbf{CT}^{*},\boldsymbol{D}_u^{'},\text{RK}_u)$。假设用户 u 未被撤销，其首先根据用户二叉树中与用户 u 相关联的叶子节点 η_u 获取一个唯一的节点 $\eta_j=\text{path}(\eta_u)\cap\text{cover}(\mathbf{RL})$。

然后，根据节点 η_j 和用户 u 的部分私钥 $\boldsymbol{D}_u^{'}=\{D_0,\{\hat{\xi}_\eta\}_{\eta\in\text{path}(\eta_u)}\}$，得到密钥元素 $\hat{\xi}_{\eta_u}$ 和 $\hat{\xi}_{\eta_j}$ 以及 $\delta=\hat{\xi}_{\eta_u}/\hat{\xi}_{\eta_j}=\xi_{\eta_u}/\xi_{\eta_j}$ 并计算如下中间变量。

$$P_5=\hat{e}(D_0,E_j)^{\delta}=\hat{e}(g_0^{r_u/\xi_{\eta_u}},(g_0^{\xi_{\eta_j}})^{s})^{\xi_{\eta_u}/\xi_{\eta_j}}=\hat{e}(g_0,g_0)^{r_u s}\tag{11}$$

最后，计算 $R_u^{'}=\frac{C_0}{(C_t)^{z_u}P_5}=\frac{C_0}{\hat{e}(g_0,g_0)^{ys}}$ 并且得到消息对称会话密钥 $k_s^{'}=H_1(R_u^{'})$。使用 $k_s^{'}$ 恢复消息明文为 $M^{'}=D_s(k_s^{'},C_s)$。如果 $V_m=H(k_s^{'},M^{'})$ 成立，则消息明文 $M^{'}$ 有效。

Trace(**PP**,**RL**,$\boldsymbol{D}_u$)。该算法首先对用户私钥 $\boldsymbol{D}_u$ 执行如下检查。

$$\begin{aligned}&D_3\in\mathbb{Z}_p,D_0,D_1,D_2,D_i,D_{\sigma,k}\in\boldsymbol{G}_1\\&\hat{e}(g_0,D_1)=\hat{e}(g_0^{x},D_2)\neq1\\&\hat{e}(D_{\sigma,2},g_0^{x}g_0^{D_3})=\hat{e}(g_0,g_0)^{y}\hat{e}(D_1D_2^{D_3},\mu)\cdot\\&\qquad\hat{e}(L_0\prod_{j=1}^{l_\sigma}L_j^{\sigma_j},D_{\sigma,1}^{D_3}D_{\sigma,3})\neq1\\&\exists i\in I_u,\text{s.t. }\hat{e}(D_i,g_0)\hat{e}(D_1D_2^{D_3},v)=\hat{e}(D_2,g_0)^{\alpha_i}\neq1\end{aligned}\tag{12}$$

如果用户私钥 $\boldsymbol{D}_u$ 通过该检查，则计算 $v_u=D_s(k_s,D_3)$。根据 v_u 在 T_u 中进行搜索得到对应的用户身份 u，并将其加入用户撤销列表中，得到 $\mathbf{RL}^{'}=\mathbf{RL}\cup\{u\}$。

CTUpdate($\mathbf{PP},\mathbf{RL}',\mathbf{CT}$)。首先，属性机构随机选择$\pi\in\mathbb{Z}_p$，计算更新密钥$\xi'=\{\pi\xi_j\}_{j=0}^{2|\boldsymbol{U}|-2}$并将其通过安全信道发送给公有云。

其次，公有云根据更新后的用户撤销列表$\mathbf{RL}'$得到其最小覆盖集合cover($\mathbf{RL}'$)。对于该集合中的每个节点$\eta_{j'}\in\text{cover}(\mathbf{RL}')$，对以下 2 种情况升级对应的密文$\mathbf{CT}$。

情况 1 如果对于升级前的用户撤销列表$\mathbf{RL}$，存在一个节点$\eta_j\in\text{cover}(\mathbf{RL})$使$\eta_{j'}=\eta_j$，则设置$\hat{E}_{j'}=E_j$。

情况 2 如果对于升级前的用户撤销列表$\mathbf{RL}$，存在一个节点$\eta_j\in\text{cover}(\mathbf{RL})$为$\eta_{j'}$的祖先节点，即$\text{path}(\eta_{j'})=\text{path}(\eta_j)\cup\{\eta_{j+1},\cdots,\eta_{j'}\}$，设置一个中间变量$Y_j=\zeta_j$，并按照递归的方式计算$Y_{k+1}=(Y_k)^{\frac{\xi_{k+1}'}{\xi_k'}}=\zeta_{k+1}^{s}$，其中，$k=j+1,\cdots,j'$，设置$\hat{E}_{j'}=Y_{j'}$。

最后，更新后的密文为

$$\begin{aligned}\mathbf{CT}=\{&\bar{\boldsymbol{A}},\mathbf{RL}',C_s,C_0,C_1,C_2,C_3,\{C_{x,1},\\&C_{x,2},C_{x,3},C_{x,4},C_{x,5}\}_{x\in[l]},\{\hat{E}_{j'}\}_{j'\in\text{cover}(\mathbf{RL}')}\}\end{aligned}\tag{13}$$

5 安全性分析

在本文方案中，IND-CPA 安全可以规约到判定性 q-BDHE 困难性问题上。

定理 1 如果判定性 q-BDHE 困难性假设成立，任意 PPT 敌手选择访问策略和选择明文攻击下最多只能以一个可忽略的优势攻破本文方案，其中，$q>2|\boldsymbol{U}|-2$。

定理 2 如果存在一个 PPT 敌手 A 可以以优势ε攻破本文方案，则可以构建一个模拟器 B 以$\varepsilon/2$的优势解决 q-BDHE 困难问题。该模拟过程描述如下。

初始化。模拟器 B 生成一个双线性群$\boldsymbol{D}=(\boldsymbol{G}_1,\boldsymbol{G}_2,p,\hat{e},g_0)$，其中，$g_0\in\boldsymbol{G}_1$。已知一个向量$\boldsymbol{z}=(g_0,g_0^a,g_0^b,\cdots,g_0^{b^q},g_0^{b^{q+2}},\cdots,g_0^{b^{2q}})$，B 随机选择$c\in\{0,1\}$，如果$c=1$，设置$Y=\hat{e}(g,g)^{a^{q+1}b}$；否则，设置$Y=Z$，其中，$Z\in_{\mathrm{R}}\boldsymbol{G}_2$。敌手 A 选择要挑战的$l^*\times n^*$的访问策略$\boldsymbol{A}^*=(\hat{\boldsymbol{A}}^*,\rho^*,\boldsymbol{V}^*)$，用户撤销列表$\mathbf{RL}^*$和解密时间周期$T_d^*$，其中，访问策略的每个属性只出现一次，其属性值集合为$\boldsymbol{V}^*=\{v_{\rho^*(i)}\}_{i\in[1,l^*]}$。

系统建立。挑战者 B 随机选择$x,y'\in\mathbb{Z}_p$计算g_0^x并设置$\hat{e}(g_0,g_0)^y=\hat{e}(g_0,g_0)^{y'}\hat{e}(g_0^a,g_0^{a^q})$，$\mu=g_0^b$以及$\nu=g_0^{b^q}$，其中，$y=y'+g_0^{a^{q+1}}$。设置$\boldsymbol{I}_{\mathbf{RL}^*}=\{\eta_j\in\text{path}(u)\,|\,u\in\mathbf{RL}^*\}$并随机选择$\{x_i\in\mathbb{Z}_p\}_{i\in\{0,2|\boldsymbol{U}|-1\}}$。若$\eta_j\in\boldsymbol{I}_{\mathbf{RL}^*}$，则$\xi_j=x_j+b^j$；否则，$\xi_j=x_j+b^q$。根据每一个$\xi_j$计算对应的$\zeta_j=g_0^{\xi_j}$。最后，挑战者 B 发送系统公共参数$\mathbf{PP}=\{\boldsymbol{D},\mu,\nu,\hat{e}(g_0,g_0)^\alpha,g_0^x,\{\zeta_j\}_{j=0}^{2|\boldsymbol{U}|-2}\}$给 A。

询问 1。敌手 A 适应性地向挑战者 B 提交q个询问用户密钥的元组$(u_i,\boldsymbol{S}_i=\{\boldsymbol{I}_i,\hat{\boldsymbol{S}}_i\},T_{d,i})$。对于每个属性值$\alpha_j\in\boldsymbol{I}_i$和$\forall k\in\{1,\cdots,l^*\}$，如果$\alpha_j=v_{\rho^*(k)}$，则$\iota_j=\alpha_j+\sum_{m=1}^{n^*}b^m\hat{\boldsymbol{A}}_{j,m}^*$；否则，设置$\iota_j=\alpha_j$。如果$u_i\in\mathbf{RL}^*$，则对于$\forall\eta_j\in\text{Path}(u_i)$有$\eta_j\in\boldsymbol{I}_{\mathbf{RL}^*}$，此时设置$\xi_{\eta_u}=x_{\eta_u}+b^{\eta_u}$；否则，设置$\xi_{\eta_u}=x_{\eta_u}+b^q$。

其次，如果$\boldsymbol{S}_{i'}\in\boldsymbol{A}^*$，则随机选择$\gamma\in\mathbb{Z}_p$并计算$r_u=-\frac{b^q}{x+\gamma}+\frac{b^{q-1}}{x+\gamma}\frac{\boldsymbol{A}_{i,1}^*}{\boldsymbol{A}_{i,2}^*}$；否则，对$\forall\rho(i)\in\boldsymbol{I}_{i'}$，存在一个随机向量$\boldsymbol{b}^*=(b_1,\cdots,b_{n^*})\in\mathbb{Z}_p^{n^*}$使$b_1=-1$且$\boldsymbol{A}_i^*\boldsymbol{b}=\boldsymbol{0}$成立。此时，随机选择$\gamma,t\in\mathbb{Z}_p$且设置$r_u=\frac{1}{x+\gamma}(t+b_1b^q+\cdots+b_{n^*}b^{q-n^*+1})$。同时，对于$\forall k\in\boldsymbol{I}_i$存在$i\in[l^*],\rho^*(i)=k,\alpha_k=v_k$，此时按照之前提到的方式设置$\iota_k=\alpha_k+\sum_{m=1}^{n^*}b^m\hat{\boldsymbol{A}}_{j,m}^*$；否则，设置$\iota_k=\alpha_k$。

B 根据已知消息选择的参数按照 KeyGen 和 TKeyGen 算法分别生成用户解密密钥和转换密钥并返回给敌手 A。

挑战者应答。敌手 A 向 B 提交 2 个长度相同的消息m_0和m_1。模拟器 B 随机选择$\hat{c}\in\{0,1\}$并计算$C_0=m_{\hat{c}}Y\hat{e}(g_0^a,g_0^{y'}),C_1=g_0^a,C_2=(g_0^a)^x$。然后，选择随机值$r_2,\cdots,r_{n^*}\in\mathbb{Z}_p$，并且计算向量$\boldsymbol{b}=(a,ab+r_2,\cdots,ab^{n^*-1}+r_{n^*})$。根据该向量计算

$$C_{i,1}=\prod_{j=2}^{n^*}(g_0^{br_j})^{\hat{\boldsymbol{A}}_{i,j}^*}\prod_{j=1}^{n^*}(g_0^{ab^j})^{\hat{\boldsymbol{A}}_{i,j}^*}g_0^{-xb^{q+i}}$$

$$C_{i,2}=(g_0^{v_{\rho^*(i)}})^{-xb^i}\prod_{j=2}^{n^*}(g_0^{b^j\hat{\boldsymbol{A}}_{i,j}^*})^{-xb^i}\prod_{j=2}^{n^*}(g_0^{r^j})^{\hat{\boldsymbol{A}}_{i,j}^*}\prod_{j=2}^{n^*}(g_0^{ab^{j-1}})^{\hat{\boldsymbol{A}}_{i,j}^*}$$

$$C_{i,3}=g_0^{-xb^i}$$

其次，对于 $\forall \eta \in \text{cover}(\mathbf{RL}^*)$，由于 $\eta \notin \boldsymbol{I}_{\mathbf{RL}^*}$，可以得出 $\xi_j = x_j + b^q, E_j = \zeta_j^a = (g_0^a)^{x_j + b^q}$。最后，B 得到最终完整的密文 CT^* 并将其返回给敌手 A。

$$\mathbf{CT} = \{C_0, C_1, \hat{C}_1, C_2, \{C_{x,1}, C_{x,2}, C_{x,3}\}_{x\in[l]}, \{E_j\}_{j\in\text{cover}(\mathbf{RL})}\}$$

询问 2。敌手 A 重复询问 1。

猜测。敌手 A 输出对 $\hat{c}$ 的猜测 $\hat{c}'$，如果 $\hat{c} = \hat{c}'$，则挑战者 B 输出对 c 的猜测 $c' = 1$，此时，A 得到合法的密文，优势为 $\varepsilon = |\Pr[\hat{c} = \hat{c}' \mid c = 1] - 1/2|$，即 $\Pr[c = c' \mid c = 1] = \varepsilon + 1/2$；否则，B 猜测 $c' = 0$，A 得到一个随机密文，即 $\Pr[\hat{c} \neq \hat{c}' \mid c = 0] = 1/2$，此时 B 输出 $c' = 1$，于是有 $\Pr[c \neq c' \mid c = 0] = 1/2$。因此，可以得到 B 解决 q-BDHE 问题的优势为 $|\Pr[c = c' \mid c = 0]/2 + \Pr[c = c' \mid c = 1]/2 - 1/2| = \varepsilon/2$。因此，定理 1 得以证明。

其次，对已撤销用户和未撤销用户的合谋攻击进行简单的分析。假设密文 $\mathbf{CT}$ 及其访问策略 $\overline{\boldsymbol{A}}$，一个已被撤销的用户 u_0 及其满足 $\overline{\boldsymbol{A}}$ 的私钥 $\boldsymbol{D}_{u_0}$ 和一个未被撤销的用户 u_1 及其不满足 $\overline{\boldsymbol{A}}$ 的私钥 $\boldsymbol{D}_{u_1}$，如果用户 u_1 和用户 u_0 合谋访问密文 $\mathbf{CT}$，则需要使用已撤销用户 u_0 的私钥中的属性相关元素 K_i 生成转换密钥并发送给公有云进行外包解密，并生成 $C_{\mathrm{t}} = \dfrac{\hat{e}(g_0, g_0)^{ys/z_{u_0}}}{\hat{e}(g_0, g_0)^{r_{u_0}s/z_{u_0}}}$。在本地用户解密过程中，如果使用 u_0 的部分私钥 $\boldsymbol{D}_{u_0}'$，则由于 $\varnothing = \text{path}(\eta_{u_0}) \cap \text{cover}(\mathbf{RL})$ 而无法得到 $P_5 = \hat{e}(D_0, E_j)^{\delta} = \hat{e}(g_0, g_0)^{r_{u_0}s}$。如果使用 u_1 的部分私钥 $\boldsymbol{D}_{u_1}'$，通过 $\eta_k = \text{path}(\eta_{u_1}) \cap \text{cover}(\mathbf{RL})$ 可以获取密文中的元素 $E_k = (g_0^{\xi_{\eta_k}})^s$，但是只能得到 $P_5 = \hat{e}(D_0', E_j)^{\delta} = \hat{e}(g_0, g_0)^{r_{u_1}s}$，这里的随机数 r_{u_1} 和 C_{t} 的 r_{u_0} 不同，因此无法完成本地用户解密。如果使用 u_0 的部分私钥 $\boldsymbol{D}_{u_0}'$ 中的 D_0，则在计算 P_5 中使用参数 δ 时需要用到 u_0 的部分私钥 $\boldsymbol{D}_{u_0}'$ 中的 $\hat{\xi}_{\eta_{u_0}}$ 和 u_1 的部分私钥 $\boldsymbol{D}_{u_1}'$ 中的 $\hat{\xi}_{\eta_k}'$（注：根据用户二叉树的特性，此时 u_0 的部分私钥 $\boldsymbol{D}_{u_0}'$ 中不存在 $\hat{\xi}_{\eta_k}'$），由于本文方案中对每个用户私钥中的 $\{\xi_\eta\}_{\eta\in\text{path}(\eta_u)}$ 元素进行了不同的随机化，因此使用 u_0 的 $\hat{\xi}_{\eta_{u_0}}$ 和 u_1 的 $\hat{\xi}_{\eta_k}'$ 无法计算得到 $\xi_{\eta_{u_0}} / \xi_{\eta_k}$，即 $\delta = \hat{\xi}_{\eta_{u_0}} / \hat{\xi}_{\eta_k}' \neq \xi_{\eta_{u_0}} / \xi_{\eta_k}$，从而无法得知 $P_5 = \hat{e}(g_0, g_0)^{r_{u_0}s}$。因此，此种类型的合谋攻击是无效的。

另外，本文方案的可追踪性采用文献[23]中的机制，因此保持了相同的强可追踪性，其安全性证明也相近，此处因篇幅限制不做专门证明。

6　性能分析

本节通过将本文方案与文献[13]方案、文献[16]方案和文献[23]方案在计算开销和存储开销方面进行对比，给出本文方案的理论性能分析。其中，$|S|$ 表示用户属性集大小，$|I|$ 表示访问策略复杂度，$E_1(E_2)$、$M_1(M_2)$、P 分别表示素数阶群 $\boldsymbol{G}_1(\boldsymbol{G}_2)$ 上的指数、乘法运算和双线性映射运算，$|\boldsymbol{G}_1|$、$|\boldsymbol{G}_2|$、$|\mathbb{Z}_p|$ 分别表示素数阶群 $\boldsymbol{G}_1$、$\boldsymbol{G}_2$、$\mathbb{Z}_p$ 上的元素长度，$E_i(E_T)$、$M_{ij}(M_T)$、P_{ij} 分别表示合数阶群 $\boldsymbol{G}_i(\boldsymbol{G}_T)$ 上的指数、乘法运算和双线性映射运算，$|\boldsymbol{G}_{ij}|$、$|\boldsymbol{G}_T|$、$|\mathbb{Z}_N|$ 表示合数阶群 $\boldsymbol{G}_i$、$\boldsymbol{G}_T$、$\mathbb{Z}_N$ 上的元素长度，$|C|$ 表示撤销列表在用户二叉树中最小覆盖集合大小，$|P|$ 表示用户二叉树中路径长度。

表 2 给出了本文方案和文献[13,16,23]方案的计算开销对比。在加密算法中，本文方案实现了最小的加密开销，由于引入了在线/离线技术，加密过程中省去了文献[23]方案中的访问策略相关开销；文献[13]方案和文献[16]方案的复杂度高于本文方案而且其是在合数阶群上进行的设计，因此效率很低。在解密过程中，由于外包解密的引入，本文方案仅需要一次双线性映射操作，访问策略相关操作都外包至公有云，因此开销远小于文 献[23]方案；在文献[13]方案和文献[16]方案中，解密过程也需要大量合数阶群上的指数和双线性运算，效率很低。

表 3 给出了本文方案和文献[13,16,23]方案的存储开销对比。由于支持用户撤销，本文方案和文献[23]方案都引入了额外的用户空间相关的公共参数元素，另外，本文方案还引入了时间周期树的相关元素，因此具有更长的公共参数，但是，以上全部方案的公共参数均为常数，因此都支持大属性空间。在用户密钥长度上，本文方案由于采用了外包解密机制，仅需要保存恢复密钥和用户密钥中与用户二叉树路径相关的部分。而其他方案则需要保存较长的密钥，特别是文献[13]方案和文献[16]方案的用户密钥在合数阶群上的长度更长。在密文长度方面，

表 2 本文方案和相关方案的计算开销比较

方案	加密	解密	密钥生成
文献[13]方案	$(7\|I\|+2)E_i+2E_T+(7\|I\|+1)M_{ij}+M_T$	$\|I\|E_T+(2\|I\|+1)P_{ij}+2\|I\|M_{ij}$	$(2\|S\|+3)E_i+(2\|S\|+3)M_{ij}$
文献[16]方案	$(7\|I\|+4)E_i+2E_T+(7\|I\|+2)M_{ij}+M_T$	$(2\|I\|+1)E_i+\|I\|E_T+(2\|I\|+1)P_{ij}+(2\|I\|+1)M_{ij}$	$(2\|S\|+4)E_i+(2\|S\|+4)M_{ij}$
文献[23]方案	$(4\|I\|+\|C\|+2)E_1+E_2+\|I\|M_1+M_2$	$(2\|I\|+1)E_1+(2\|I\|+4)M_1+(3\|I\|+1)P$	$(2\|S\|+5)E_1+(\|S\|+1)M_1$
本文方案	$(l_d+\|C\|+1)E_1+l_dM_1$	$E_1+E_2+2M_1+P$	$(2\|S\|+d+8)E_1+(l_\sigma+\|S\|)M_1$

表 3 本文方案和相关方案的存储开销对比

方案	公共参数长度	用户密钥长度	密文长度
方案[13]方案	$4\|\boldsymbol{G}_i\|+\|\boldsymbol{G}_T\|$	$(\|S\|+2)\|\boldsymbol{G}_{ij}\|$	$(3l+2)\|\boldsymbol{G}_{ij}\|+2\|\boldsymbol{G}_{\mathrm{T}}\|$
方案[16]方案	$4\|\boldsymbol{G}_i\|+\|\boldsymbol{G}_T\|$	$(\|S\|+3)\|\boldsymbol{G}_{ij}\|+\|\mathbb{Z}_N\|$	$(3l+4)\|\boldsymbol{G}_{ij}\|+2\|\boldsymbol{G}_{\mathrm{T}}\|$
方案[23]方案	$(2\|U\|+3)\|G_1\|+\|G_2\|$	$(\|S\|+4)\|\boldsymbol{G}_1\|+(\|P\|+1)\|\mathbb{Z}_p\|$	$(3l+\|C\|+2)\|G_1\|+\|G_2\|$
本文方案	$(2\|U\|+d+3)\|G_1\|+\|G_2\|$	$\|\boldsymbol{G}_1\|+(\|P\|+1)\|\mathbb{Z}_p\|$	$(3l+\|C\|+3)\|G_1\|+\|G_2\|+2l\|\mathbb{Z}_p\|$

本文方案由于采用了在线/离线加密技术，因此比文献[23]方案有更多的开销，但是由于所增加的长度为素数阶群$\mathbb{Z}_p$上的元素长度，因此额外开销很少。

此外，本文通过实现 4 个方案并对其进行仿真实验和对比分析真实运行数据来展示本文方案的实际性能。使用 Java 编程语言和 JPBC 库[25]对 4 个方案进行实现，采用 JPBC 库的 Type A 曲线$E(F_q):y^2=x^3+x$生成 2 个阶为素数 p 的乘法循环群$\boldsymbol{G}_1,\boldsymbol{G}_2$，其中，$p=80$，$q=160$。因此，$|\boldsymbol{G}_1|=|\boldsymbol{G}_2|=320$，$|\mathbb{Z}_p|=160$。所有实验都在一台硬件配置为 Core i5-6500 CPU @ 2.60 GHz、6 GB 内存并安装 Windows Server 2013 操作系统的服务器上运行和测试。

图 3 描绘了 4 个方案的文件加密时间和密文长度。图 3(a)和图 3(b)对比了这 4 个方案在不同的撤销列表最小覆盖集设置$|C|$的文件加密时间。可以很明显看出，本文方案在加密过程中需要的时间损耗远小于其他方案，而且随着访问策略复杂度的增长，本文方案所需的加密时间增长较缓慢。图 3(c)和图 3(d)展示了 4 个方案在不同的撤销列表最小覆盖集设置$|C|$下的密文长度。其中，4 个方案的密文大小均随着访问策略的复杂度的增大而增加。由于引入了在线离线技术，本文方案的密文长度要大于文献[23]方案，但是仅限于群$\mathbb{Z}_p$上元素的量级。而文献[13]方案和文献[16]方案由于在合数阶群上构建，因此其实际的开销远大于其他 2 个方案。

图 4 展示了 4 个方案中文件解密时间随着密文个数的变化情况。图 4(a)和图 4(b)分别为在访问策略复杂度$|I|=5$和$|I|=10$的设置下，4 个方案的解密时间对比。从图 4 可以看出，在同样的访问策略复杂度设置下对同样个数的密文进行解密时，本文方案所需的解密时间要远小于其他方案，而且其随文件个数增长的变化非常缓慢。

图 5 展示了 4 个方案在密钥生成时间、用户密钥长度和公共参数长度方面的性能对比。图 5(a)和图 5(b)显示了 4 个方案在不同的时间周期树深度 d 设置下密钥生成时间随着用户属性集合大小的变化情况。可以看出，由于基于合数阶群构建，文献[13]方案和文献[16]方案的密钥生成时间远超过另外 2 个方案。本文方案在密文生成算法引入了基于时间的操作，因此开销比文献[23]方案大。但是，在实际中 d 一般很小，因此本文方案中额外引入的与系统时间周期树深度 d 相关的计算和存储开销也非常小。图 5(c)和图 5(d)显示了 4 个方案在不同路径长度$|P|$设置下用户密钥长度随着用户属性集合大小的变化情况。本文方案由于采用了外包解密机制，需要保存的用户密钥仅为恢复密钥以及和路径相关的解密密钥部分，因此用户密钥长度很小，随着属性集合大小的变化也很小。文献[13]方案和文献[16]方案由于采用合数阶群实现，因此密钥长度远大于另外 2 个方案。图 5(e)和图 5(f)描绘了在不同的时间周期树深度 d 设置下系统公共参数长度随系统属性集合的变化情况。很明显，4 个方案的公共参

数的大小均不受系统属性全集大小影响，即均支持大规模属性集合。文献[23]方案和本文方案由于支持用户撤销，因此需要更多的参数开销，同时本文方案还支持时间有效性而引入了时间相关的公共参数，因此，公共参数长度大于文献[23]方案。

图 6 展示了本文方案和文献[23]方案、文献[16]方案在用户追踪时间和存储方面的真实开销对比。图 6(a)对比评估了 3 个方案的用户追踪时间。由于文献[16]方案基于合数阶群进行构建，而且其追踪用户的算法基于用户列表查找，因此计算开销高于另外 2 个方案。图 6(b)是 3 个方案在存储方面的实际消耗对比。由于本文方案和文献[23]方案在实现用户追踪时不需要用户列表，其用户追踪的存储开销为 0；而文献[16]方案的用户追踪需要维护一个额外的用户列表，因此，文献[16]方案在存储方面消耗较大，而且随着用户个数增多，列表存储开销增大。

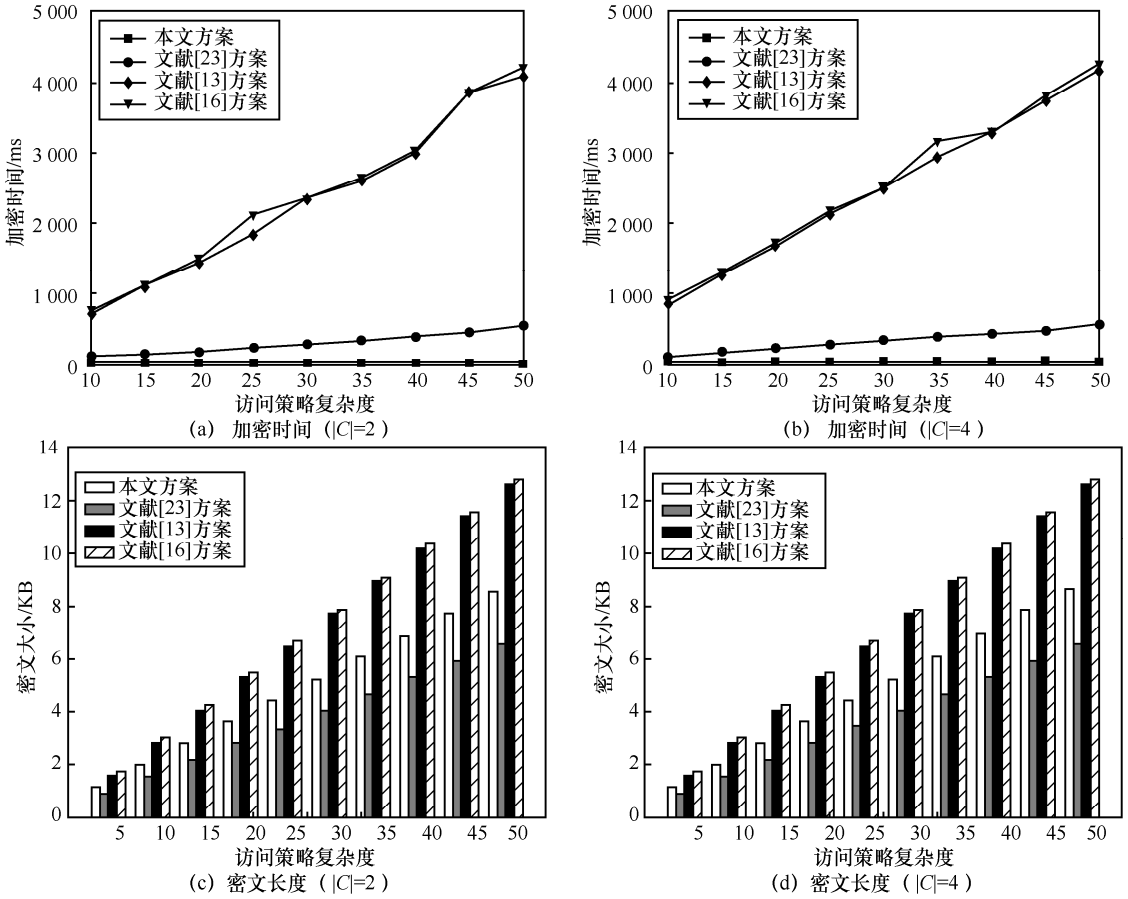


图 3　4 个方案的文件加密时间和密文长度

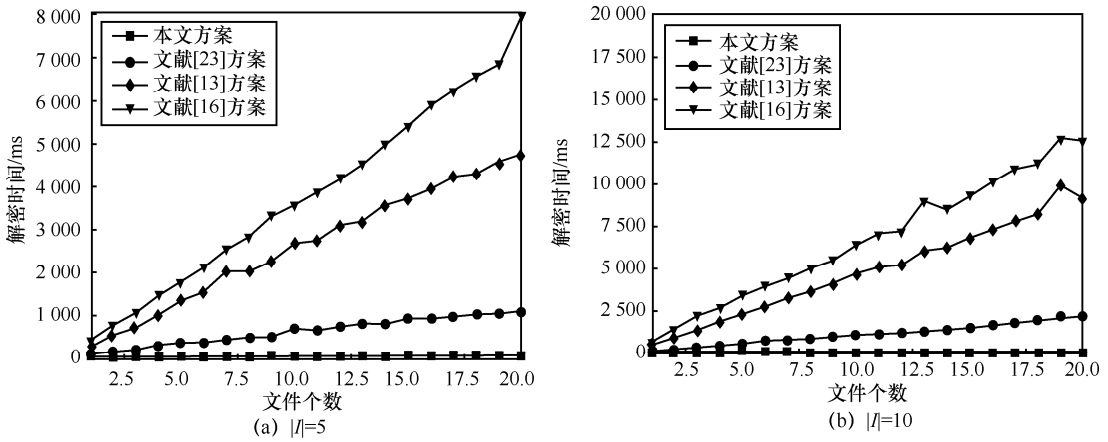
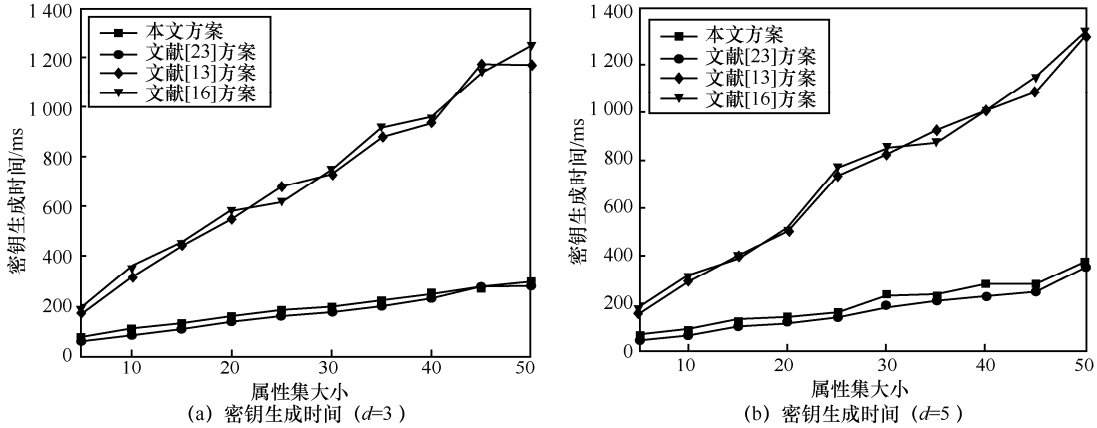
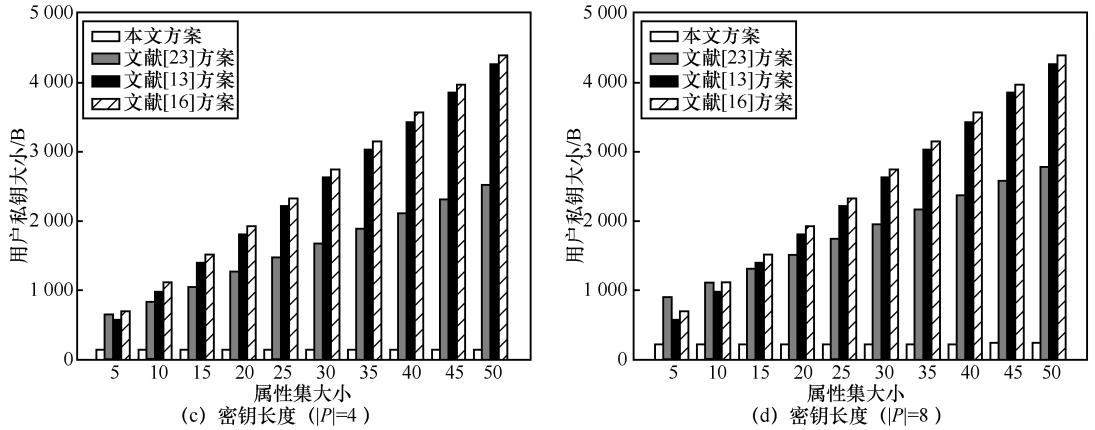


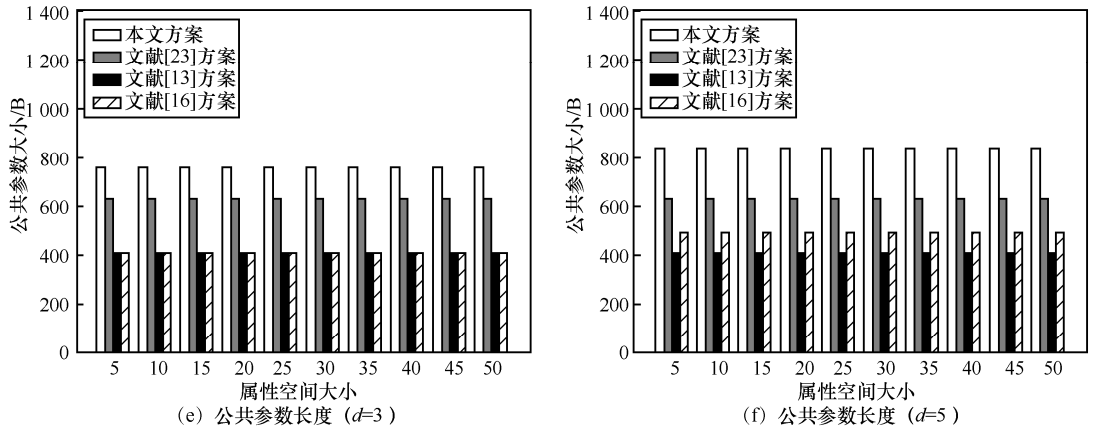
图 4　4 个方案中文件解密时间随着密文个数的变化情况



(a) 密钥生成时间（d=3）　　(b) 密钥生成时间（d=5）

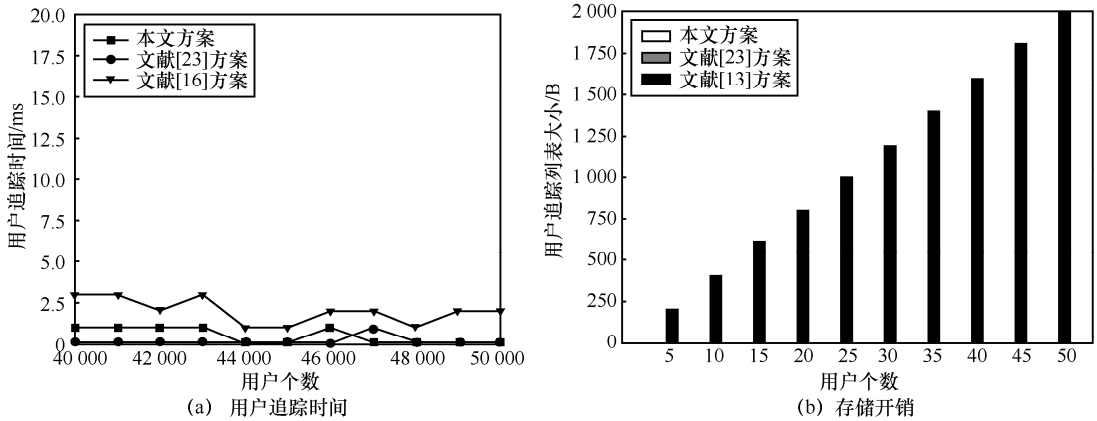


(c) 密钥长度（$|P|$=4）　　(d) 密钥长度（$|P|$=8）



(e) 公共参数长度（d=3）　　(f) 公共参数长度（d=5）

图 5　4 个方案在密钥生成时间、用户密钥长度和公共参数长度方面的性能对比



(a) 用户追踪时间　　(b) 存储开销

图 6　3 个方案在用户追踪时间和存储方面的真实开销对比

综上所述，本文方案由于在系统公共参数中引入时间相关的一些固定参数，因此在公共参数的存储开销中有所增加。然而，本文方案在计算性能和其他存储空间开销方面都远超文献[13,16,23]方案。因此，本文方案更具有实用性和高效性。

7　结束语

为了解决当前 CP-ABE 方案中存在的隐私保护和用户追踪以及撤销等严重问题，本文设计了一种可撤销可追踪的基于时间且具有隐私保护的云数据共享方案，实现了基于时间的细粒度访问控制和访问策略的用户隐私保护。同时，对于恶意泄露密钥的用户，设计了一种高效追踪用户并进行直接用户撤销的机制，该机制具有很短的用户撤销列表。本文方案的加密过程只需要简单的整数计算，而在解密过程中只需要一个双线性映射运算，相比已有方案，整体效率有很大提升。此外，在判定性 q-BDHE 假设下，本文方案是 IND-CPA 安全的，而且支持大规模属性空间，非常适于云环境下数据共享的实际应用。

参考文献：

[1] ZHANG J W, MA J F, MA Z, et al. Efficient hierarchical data access control for resource-limited users in cloud-based e-health[C]// Proceedings of 2019 International Conference on Networking and Network Applications (NaNA). Piscataway: IEEE Press, 2019: 319-324.

[2] MIAO Y B, WENG J, LIU X M, et al. Enabling verifiable multiple keywords search over encrypted cloud data[J]. Information Sciences, 2018, 465: 21-37.

[3] MIAO Y B, DENG R H, LIU X M, et al. Multi-authority attribute-based keyword search over encrypted cloud data[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 18(4): 1667-1680.

[4] ISLAM M A, MADRIA S. Attribute-based encryption scheme for secure multi-group data sharing in cloud[J]. IEEE Transactions on Services Computing, 2020, PP(99): 1.

[5] ZHANG Z T, ZENG P, PAN B F, et al. Large-universe attribute-based encryption with public traceability for cloud storage[J]. IEEE Internet of Things Journal, 2020, 7(10): 10314-10323.

[6] QI S Y, LU Y S, ZHENG Y Q, et al. CPDS: enabling compressed and private data sharing for industrial Internet of things over blockchain[J]. IEEE Transactions on Industrial Informatics, 2021, 17(4): 2376-2387.

[7] HOHENBERGER S, WATERS B. Online/offline attribute-based encryption[C]//International Workshop on Public Key Cryptography. Berlin: Springer, 2014: 293-310.

[8] 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 & Communications Security. New York: ACM Press, 2013: 463-474.

[9] JOSHI M, JOSHI K, FININ T. Attribute based encryption for secure access to cloud based EHR systems[C]//Proceedings of 2018 IEEE 11th International Conference on Cloud Computing (CLOUD). Piscataway: IEEE Press, 2018: 932-935.

[10] LIU Z C, JIANG Z L, WANG X, et al. Practical attribute-based encryption: outsourcing decryption, attribute revocation and policy updating[J]. Journal of Network and Computer Applications, 2018, 108: 112-123.

[11] FAN W J, LI F, CHEN X W, et al. Deploying parallelised ciphertext-policy attributed-based encryption in clouds[J]. International Journal of Computational Science and Engineering, 2018, 16(3): 321.

[12] NING J T, CAO Z F, DONG X L, et al. Auditable $\sigma $ -time outsourced attribute-based encryption for access control in cloud computing[J]. IEEE Transactions on Information Forensics and Security, 2018, 13(1): 94-105.

[13] ZHANG Y H, ZHENG D, 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.

[14] FAN K, XU H Y, GAO L X, et al. Efficient and privacy preserving access control scheme for fog-enabled IoT[J]. Future Generation Computer Systems, 2019, 99: 134-142.

[15] CUI H, DENG R H, LAI J Z, 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.

[16] LI Q, ZHANG Y H, ZHANG T, et al. HTAC: fine-grained policy-hiding and traceable access control in mHealth[J]. IEEE Access, 2020, 8: 123430-123439.

[17] ZHANG P, CHEN Z H, LIANG K T, et al. A cloud-based access control scheme with user revocation and attribute update[C]//Information Security and Privacy. Cham: Springer, 2016: 525-540.

[18] LI J G, YAO W, ZHANG Y C, et al. Flexible and fine-grained attribute-based data storage in cloud computing[J]. IEEE Transactions on Services Computing, 2017, 10(5): 785-796.

[19] ZHANG J W, LI T, OBAIDAT M S, et al. Enabling efficient data sharing with auditable user revocation for IoV systems[J]. IEEE Systems Journal, 2021, PP(99): 1.

[20] ZHANG J W, MA J F, LI T, et al. Efficient hierarchical and time-sensitive data sharing with user revocation in mobile crowdsensing[J]. Security and Communication Networks, 2021, 2021: 1-17.

[21] QIN B D, ZHAO Q L, ZHENG D, et al. (Dual) server-aided revocable attribute-based encryption with decryption key exposure resistance[J]. Information Sciences, 2019, 490: 74-92.

[22] LIU Z H, DUAN S H, ZHOU P L, et al. Traceable-then-revocable

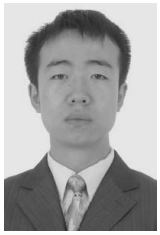
ciphertext-policy attribute-based encryption scheme[J]. Future Generation Computer Systems, 2019, 93: 903-913.

[23] HAN D Z, PAN N N, LI K C. A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection[J]. IEEE Transactions on Dependable and Secure Computing, 2020, PP(99): 1.

[24] LIU J K, YUEN T H, ZHANG P, et al. Time-based direct revocable ciphertext-policy attribute-based encryption with short revocation list[C]//Applied Cryptography and Network Security. Berlin: Springer, 2018: 516-534.

[25] DE CARO A, IOVINO V. JPBC: Java pairing based cryptography[C]//Proceedings of 2011 IEEE Symposium on Computers and Communications (ISCC). Piscataway: IEEE Press, 2011: 850-855.

[作者简介]



张嘉伟（1985− ），男，山西太原人，西安电子科技大学博士生，主要研究方向为网络安全、访问控制、数据安全、云计算安全和区块链等。

马建峰（1963− ），男，陕西西安人，博士，西安电子科技大学教授、博士生导师，主要研究方向为网络安全、系统安全、数据安全和无人机安全等。

马卓（1980− ），男，陕西延安人，博士，西安电子科技大学教授、博士生导师，主要研究方向为人工智能与无人系统安全、无线网络安全等。

李腾（1991− ），男，陕西西安人，博士，西安电子科技大学讲师，主要研究方向为网络安全、系统日志分析、攻击检测、数据安全和隐私保护。