云存储环境下无密钥托管可撤销属性基加密方案研究

赵志远*[①] 朱智强[①②] 王建华[①③] 孙 磊[①]
[①](信息工程大学三院 郑州 450001)
[②](郑州信大先进技术研究院 郑州 450001)
[③](空军电子技术研究所 北京 100195)

摘 要：属性基加密因其细粒度访问控制在云存储中得到广泛应用。但原始属性基加密方案存在密钥托管和属性撤销问题。为解决上述问题，该文提出一种密文策略的属性基加密方案。该方案中属性权威与中央控制通过安全两方计算技术构建无密钥托管密钥分发协议解决密钥托管问题。通过更新属性版本密钥的方式达到属性级用户撤销，同时通过中央控制可以实现系统级用户撤销。为减少用户解密过程的计算负担，将解密运算过程中复杂对运算外包给云服务商，提高解密效率。该文基于 q-Parallel BDHE 假设在随机预言机模型下对方案进行了选择访问结构明文攻击的安全性证明。最后从理论和实验两方面对所提方案的效率与功能性进行了分析。实验结果表明所提方案无密钥托管问题，且具有较高系统效率。

关键词：云存储；属性基加密；无密钥托管；撤销；解密外包
中图分类号： TP309 **文献标识码**： A **文章编号**：1009-5896(2018)01-0001-10
DOI: 10.11999/JEIT170317

Revocable Attribute-based Encryption with Escrow-free in Cloud Storage

ZHAO Zhiyuan[①] ZHU Zhiqiang[①②] WANG Jianhua[①③] SUN Lei[①]
[①](*The Third College*, *Information Engineering University*, *Zhengzhou* 450001, *China*)
[②](*Zhengzhou Xin Da Advanced Technology Research Institute*, *Zhengzhou* 450001, *China*)
[③](*Electronic Technology Institute of Air Force*, *Beijing* 100195, *China*)

Abstract: Attribute-Based Encryption (ABE) scheme is widely used in cloud storage, which can achieve fine-grained access control. However, the original attribute-based encryption schemes have key escrow and attribute revocation problems. To solve these problems, this paper proposes a ciphertext-based ABE scheme. In the scheme, the key escrow problem could be solved by escrow-free key issuing protocol, which is constructed using the secure two-party computation between the attribute authority and the central controller. By updating the attribute version key, the scheme can achieve attribute-level user revocation. And by central controller, the scheme can achieve system-level user revocation. In order to reduce the user's computational burden of decryption, this scheme outsources the complicated pair operation to cloud service providers. Based on the assumption of q-Parallel BDHE, the scheme is proved that is the security of the chosen plaintext attack in the random oracle model. Finally, the efficiency and function of this scheme are analyzed theoretically and experimentally. The experimental results show that the proposed scheme does not have key escrow problem and has the higher system efficiency.

Key words: Cloud storage; Attribute-Based Encryption (ABE); Escrow-free; Revocation; Outsourced decryption

1 引言

云存储是基于云计算建立起来的一种新型的网络存储技术，通过按需付费等方式向广大用户提供存储服务，免去用户管理资源和花费大量资金购买硬件等负担。云存储在为人们带来巨大便利的同时，也为用户的信息资产安全和隐私保护带来了巨大的冲击和挑战[1]。密码学作为信息安全的基石，可以提供信息的完整性、机密性、不可抵赖性、可控性及可用性[2]，也是解决当前云存储安全问题的关键支撑技术之一。

在云存储模式下，数据脱离了用户控制域，用

收稿日期：2017-04-11；改回日期：2017-07-07；网络出版：2017-08-28
*通信作者：赵志远 zzy_taurus@foxmail.com
基金项目：国家重点研发计划(2016YFB0501900)，国家 973 计划项目(2013CB338000)
Foundation Items: The National Key Research Program of China (2016YFB0501900), The National 973 Program of China (2013CB338000)

户与云服务商之间缺乏信任机制，现阶段普遍观点认为要实现用户数据的隐私保护，最直接有效的方法是将数据加密后再存储。但是在云存储模式下，这种方法也牺牲了用户对数据的细粒度访问控制。传统的对称加密技术和公钥加密技术难以应对云存储这种具有海量用户的复杂情况(密钥管理、密文副本数量多等问题)。

密文策略的属性基加密方案(Ciphertext-Policy Attribute-Based Encryption, CP-ABE)[3]可以在密文中嵌入访问控制策略，提供了一种灵活的访问控制方法，是云存储环境下实现基于密码技术的访问控制的关键技术。CP-ABE 提供灵活访问控制的同时也带来了众所周知的密钥托管问题，密钥产生中心(KGC)能够通过用户的属性集产生私钥解密任何密文，这将给云中数据的机密性和隐私性带来严重的安全挑战；另一个安全挑战就是属性撤销问题，因为在系统运行过程中一些用户的相关属性会改变，或者一些私钥有可能被泄露，因此撤销或更新每一个属性的私钥组件对于系统安全至关重要。属性基加密中因为每一个属性有可能被多个用户共享，这意味着撤销任何属性或用户都有可能影响其它用户，因此属性撤销是一个非常困难的问题[4]。

针对上述问题，Pirretti 等人[5]于 2006 年最先提出 ABE 属性撤销方案，其通过对每一个属性设定一个有效期，授权机构周期性地更新属性版本，通过撤销某个属性的最新版本以此达到用户属性撤销的目的。该方案中，由于授权中心密钥更新过程中的计算量与用户数量的多少成线性关系，因此效率不高。2008 年，Boldyreva 等人[6]提出一个有效撤销属性的 IBE 方案，其利用二叉树构建一个数据结构。但是这个方案不适用于 CP-ABE。这种通过给每个属性设定时间周期来达到撤销目的的方法是一种粗粒度的撤销方法，其不能实现属性或用户的立即撤销。然而这种撤销方案主要有以下两个问题：第 1 个问题就是前向安全(forward secrecy)和后向安全问题(backward secrecy)[7]，其存在一个不受控制的时期被称为脆弱性窗口。另一个问题是可扩展性问题，授权中心需要通过单播方式在脆弱性窗口期发布一个密钥更新组件以确保所有未撤销用户能够更新他们的私钥。这可能是授权机构和所有未撤销用户的一个瓶颈。

为解决上述问题，Ibraimi 等人[8]和 Yu 等人[9]提出基于 CP-ABE 的立即属性撤销方案，但是这种方案仍然没有实现数据外包环境下的细粒度访问控制。2011 年，Hur 等人[10]提出一种具有属性和用户撤销能力的 CP-ABE 方案，该方案增强了用户访问控制的前向安全和后向安全，具有属性级别的属性撤销能力。同时当用户没及时更新私钥的情况下，仍可以通过一个二叉树解决这种状态丢失接收问题。但是该方案不能抵抗用户合谋攻击。2013 年，Yang 等人[11]提出一种云存储环境下支持细粒度属性撤销的属性基加密方案，该方案不需要服务器支持任何协作的访问控制，数据拥有者也不需要实时在线，在效率方面较 Hur 等人方案[10]有所提高。但是该方案只是在随机预言机下证明其安全性。2014 年，Zu 等人[12]提出一种云存储环境下具有有效撤销能力的 CP-ABE 方案，该方案中的访问结构是具有强表现能力的线性密钥共享方案。Qian 等人[13]提出一种云环境下用于电子医疗记录的隐私保护多授权中心的 CP-ABE 方案，该方案撤销属性过程中采用懒惰重加密技术更新密文。但该方案在生成用户私钥时需要每个授权中心通过安全两方计算技术，这带来严重的通信开销问题。王等人[14]构造了具有两个可撤销属性列表的密钥策略的属性基加密方案，该方案是对含有单个属性撤销列表方案的推广。Vaanchig 等人[15]提出一种细粒度访问控制的属性撤销 CP-ABE 方案，该方案允许未撤销用户根据自己唯一的更新密钥更新自己的私钥，云服务商利用密钥更新密文，同时该方案降低了对云服务商的信任程度。无论如何，上述方案没有考虑密钥托管问题。

针对上述所提出问题，本文提出一种可撤销属性基加密方案。该方案通过两方安全计算解决了密钥托管问题；在解密过程中，为减少用户的计算负担而将复杂的计算外包给云服务商。本文主要贡献总结如下：

(1)为解决密钥托管问题，本文方案将属性中心拆分为属性权威(Attribute Authority, AA)和中央控制(Central Controller, CC)，属性权威和中央控制二者运行两方安全计算产生用户私钥。

(2)为解决属性撤销问题，本文方案引入属性版本密钥 $\mathrm{VK}_{\mathrm{att}}$，通过更新版本密钥达到属性撤销的目的，同时该方案满足前向安全和后向安全要求。本文方案还支持用户撤销，通过中央控制直接撤销某一个用户。

(3)为提高系统效率，在属性撤销过程中本文只更新关联被撤销属性的密文组件和拥有撤销属性但未被撤销用户的私钥组件。同时为减少用户(数据拥有者和数据用户)的计算量，本文将密文更新过程外包给云服务商(数据拥有者)，将解密过程的部分计算外包给云服务商(数据用户)。

2　理论基础

定义 1(双线性群)　双线性群是密码系统中重要的关键技术。令 ψ 是一个群生产算法，以安全参数 λ 作为输入，输出 (p,G,G_T,e)。其中 p 为由安全参数 λ 决定的素数，G 和 G_T 是阶为素数 p 的循环群。双线性映射 $e:G\times G\to G_T$ 满足下列性质：(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：$\exists g\in G$ 使得 $e(g,g)$ 在 G_T 中的阶是 p；(3)可计算性：对于 $\forall u,v\in G$，可以有效计算 $e(u,v)$。

定义 2(决策性 q-Parallel Bilinear Diffie-Hellman Exponent (BDHE)假设)　令 G 表示阶为 p 的双线性群，$a,s,b_1,\cdots,b_q\in Z_p$ 为群 G 内随机选择的参数，g 为 G 的生成元。若攻击者给定参数

$$\begin{aligned}\boldsymbol{y}=\Big(&g,g^s,g^a,\cdots,g^{a^q},g^{a^{q+2}},\cdots,g^{a^{2q}},\forall 1\le j\le q:\\&g^{sb_j},g^{a/b_j},\cdots,g^{a^q/b_j},g^{a^{q+2}/b_j},\cdots,g^{a^{2q}/b_j},\\&\forall 1\le j,k\le q,k\ne j:g^{asb_k/b_j},\cdots,g^{a^qsb_k/b_j}\Big)\end{aligned}\tag{1}$$

对攻击者来说，要区分 $e(g,g)^{a^{q+1}s}$ 与群 G_T 中的随机元素 R 来说是困难的。

算法 $\mathcal{B}$ 通过输出 $z\in\{0,1\}$ 进行猜测，如果 $\left|\Pr\left[\mathcal{B}\left(\boldsymbol{y},T=e(g,g)^{a^{q+1}s}\right)=0\right]-\Pr\left[\mathcal{B}(\boldsymbol{y},T=R)=0\right]\right|\ge\varepsilon$，则定义其拥有优势 ε 来解决群 G 下的 q-Parallel BDHE 假设。若无多项式时间算法以不可忽略的优势来解决 q-Parallel BDHE 问题，那么我们就说假设 q-Parallel BDHE 在群 G 和 G_T 中是成立的。

3　系统及安全模型

3.1 系统模型

本文提出一种具有用户及属性立即撤销功能的属性基加密方案，其主要包括中央控制、属性权威、云服务商、数据拥有者和数据用户 5 部分组成。该方案通过中央控制和属性权威之间的两方安全计算解决了密钥托管问题；在解密阶段，用户将部分解密计算分别外包给中央控制和云服务商来达到减少用户计算量的目的。另外，本文假设中央控制、属性权威和云服务商是诚实并好奇的(honest but curious)，即它们会诚实地按照指示正确地执行步骤，但是由于好奇心，其会在工作过程中窥探数据中的隐私。同时假设中央控制和属性权威不能够合谋。本文通过中央控制和属性权威之间的两方安全计算技术完成方案设计，弱化了目前存在方案的假设条件，解决了密钥托管问题。

本文所提方案 EFR-ABE(Revocable ABE with Escrow-Free)包含以下 5 个阶段：

(1)系统初始化：该阶段包含 GlobalSetup, CCSetup 和 AASetup 3 个多项式时间算法。由中央控制负责产生全局公共参数，中央控制和属性权威分别产生自己的公私钥对。

$\text{GlobalSetup}(1^\lambda)\to\text{PP}$：该算法以隐含安全参数 λ 作为输入，输出系统公共参数 PP。

$\text{CCSetup}(\text{PP})\to(\text{PK}_{\text{CC}},\text{MSK}_{\text{CC}})$：中央控制运行该算法进行初始化，该算法以公共参数 PP 作为输入，输出中央控制的公钥 PK_{CC} 和主私钥 MSK_{CC}。

$\text{AASetup}(\text{PP})\to(\text{PK}_{\text{AA}},\text{MSK}_{\text{AA}},\text{PK}_{\text{att}})$：属性权威运行该算法进行初始化，该算法以公共参数 PP 作为输入，输出属性权威的公钥 PK_{AA}，主私钥 MSK_{AA} 和属性公钥 PK_{att}。

考虑简洁因素，以下算法输入中省略公共参数 PP。

(2)私钥生成：该阶段采用 Hur 的安全密钥产生协议[16]构建无密钥托管问题的私钥生成算法，主要包括以下两个阶段。

$\text{EF - Key}(\text{MSK}_{\text{AA}},u)\leftrightarrow\text{EF - Key}(\text{MSK}_{\text{CC}},u)$：当属性权威授权用户 u 私钥时，属性权威和中央控制通过两方安全计算协议产生部分私钥组件 CK_u 存储于中央控制处。

$\text{KeyGen}(S,t)\to(\text{SK}_u)$：属性权威根据用户属性集合 S 产生用户属性部分私钥 SK_u 并传送给数据用户。

(3)数据加密：$\text{Encrypt}(\text{PK}_{\text{AA}},\text{PK}_{\text{CC}},(\boldsymbol{M},\rho),m)\to(\text{CT}_k,\text{CT}_m)$：数据拥有者首先用对称密钥 k 加密数据 m。然后数据拥有者指定一个基于属性的访问策略 $(\boldsymbol{M},\rho)$ 用于加密对称密钥 k。最终获得密钥密文 CT_k 和 CT_m。只有当数据用户的属性满足访问策略时，才能正确解密密钥密文获得 k，然后用对称密钥 k 解密内容密文获得明文 m。

(4)数据解密：数据解密阶段涉及到数据用户、中央控制和云服务商。当数据用户的属性满足数据拥有者的访问结构时，可以通过以下协议计算获得明文。

$\text{ReqGen}(\text{SK}_u,\text{CT}_k)\to\text{RT}$：数据用户和云服务商之间通过一系列协议计算获得请求令牌 RT。

$\text{TKGen}(\text{RT},\text{CK}_u)\to\text{CT}_k''$：中央控制运行该算法，以请求令牌 RT 和 CK_u 作为输入，输出中间密文 CT_k''。

$\text{Decrypt}\left(\text{CT}_k'',\text{SK}_u'\right)\to k$：数据用户运行该算法，以中间密文 CT_k'' 和私钥 SK_u' 作为输入，获得对称密钥 k。然后用该密钥 k 解密内容密文获得最终明文数据 m。

(5)属性撤销：该阶段属性撤销分为用户撤销和用户部分属性撤销两种情况。

用户撤销：当需要直接撤销某个用户时，中央控制在解密过程中直接拒绝为该用户返回中间密文 CT_k''，从而达到撤销用户的目的。

属性撤销：属性撤销主要包括以下 3 个算法。

$\mathrm{UpKeyGen}(\mathrm{MSK}_{\mathrm{AA}}, v_{\mathrm{att}'}) \to (\overline{v}_{\mathrm{att}'}, \mathrm{UK}_{\mathrm{att}'})$：该算法以属性权威的主私钥 $\mathrm{MSK}_{\mathrm{AA}}$ 和撤销属性 att' 的版本密钥 $v_{\mathrm{att}'}$ 作为输入，输出撤销属性 att' 的新版本密钥 $\overline{v}_{\mathrm{att}'}$ 和更新密钥 $\mathrm{UK}_{\mathrm{att}'}$。

$\mathrm{UpSK}(\mathrm{SK}_u, \mathrm{UK}_{\mathrm{att}'}) \to \overline{\mathrm{SK}}_u$：该算法以用户私钥 SK_u 和撤销属性 att' 的更新密钥 $\mathrm{UK}_{\mathrm{att}'}$ 作为输入，输出新的私钥 $\overline{\mathrm{SK}}_u$。

$\mathrm{UpCT}(\mathrm{CT}, \mathrm{UK}_{\mathrm{att}'}) \to \overline{\mathrm{CT}}$：该算法以 CT 和撤销属性 att' 的更新密钥 $\mathrm{UK}_{\mathrm{att}'}$ 作为输入，输出新的密文 $\overline{\mathrm{CT}}$。

3.2 安全模型

该系统中，本文假设云服务商是诚实并好奇的，并允许未授权的用户访问云中的数据资源。数据用户是不诚实的，并且用户之间允许进行合谋解密密文。通过挑战者和敌手之间的博弈游戏描述 EFR-ABE 方案的安全模型，具体过程如下：

系统建立：挑战者 $\mathcal{C}$ 运行 Setup 算法，将公共参数 PK 传递给敌手 $\mathcal{A}$。

查询阶段 1：敌手 $\mathcal{A}$ 可以询问一系列属性集合 $S_1, S_2, \cdots, S_{ql}$ 的私钥 SK 和更新密钥 UK。

挑战阶段：敌手 $\mathcal{A}$ 提交两个等长的消息 m_0 和 m_1，同时敌手 $\mathcal{A}$ 提交一个访问结构 $\mathbb{A}^*$，并且查询阶段 1 询问的任何属性集合都不允许满足访问结构 $\mathbb{A}^*$。然后挑战者 $\mathcal{C}$ 随机选择 $b \in \{0,1\}$，并在访问结构 $\mathbb{A}^*$ 下加密 m_b，产生密文 CT^*，并将其发送给敌手 。

查询阶段 2：类似查询阶段 1，敌手 $\mathcal{A}$ 继续向挑战者 $\mathcal{C}$ 提交一系列属性集合 $S_{ql+1}, S_{ql+2}, \cdots, S_q$，其限制与查询阶段 1 相同。

猜测阶段：敌手 $\mathcal{A}$ 输出一个值 $b' \in \{0,1\}$ 作为对 b 的猜测。如果 $b' = b$，我们称敌手 $\mathcal{A}$ 赢得了该游戏。敌手 $\mathcal{A}$ 在该游戏中的优势定义为：$\mathrm{Adv}_{\mathcal{A}} = |\Pr[b' = b] - 1/2|$。

定义 3 若无多项式时间算法以不可忽略的优势来攻破上述安全模型，那么我们就说本文提出可撤销属性加密方案是安全的。

4 可撤销方案

4.1 具体方案

(1)系统初始化：$\mathrm{GlobalSetup}(1^\lambda) \to \mathrm{PP}$：中央控制运行该算法，选择两个阶为素数 p 的乘法循环群 G 和 G_T，g 是循环群 G 的生成元，并且存在有效的双线性映射 $e: G \times G \to G_T$。选择一个将属性映射到群 G 上元素的哈希函数 $H: \{0,1\}^* \to G$。输出系统的公共参数 $\mathrm{PP} = (p, g, G, G_T, e, H)$。

$\mathrm{CCSetup}(\mathrm{PP}) \to (\mathrm{PK}_{\mathrm{CC}}, \mathrm{MSK}_{\mathrm{CC}})$：中央控制运行该算法进行初始化，随机选择 $\alpha \in Z_p$ 作为中央控制的主私钥 $\mathrm{MSK}_{\mathrm{CC}}$，然后计算 $e(g,g)^\alpha$ 作为中央控制的公钥 $\mathrm{PK}_{\mathrm{CC}}$。

$\mathrm{AASetup}(\mathrm{PP}) \to (\mathrm{PK}_{\mathrm{AA}}, \mathrm{MSK}_{\mathrm{AA}}, \mathrm{PK}_{\mathrm{att}})$：属性权威运行该算法进行初始化，随机选择 $a, \beta \in Z_p$，计算 g^a 和 g^β，对于 $\forall \mathrm{att}$，选择版本密钥 $\mathrm{VK}_{\mathrm{att}} = v_{\mathrm{att}} \in Z_p$，然后计算属性公钥 $\mathrm{PK}_{\mathrm{att}} = g^{\beta v_{\mathrm{att}}}$。最后输出属性权威主私钥 $\mathrm{MSK}_{\mathrm{AA}} = (a, \beta, v_{\mathrm{att}})$、属性权威公钥 $\mathrm{PK}_{\mathrm{AA}} = (g^a, g^\beta)$ 和属性公钥 $\mathrm{PK}_{\mathrm{att}}$。

考虑简洁因素，以下算法输入中省略公共参数 PP。

(2)私钥生成：$\mathrm{EF\text{-}Key}(\mathrm{MSK}_{\mathrm{AA}}, u) \leftrightarrow \mathrm{EF\text{-}Key}(\mathrm{MSK}_{\mathrm{CC}}, u)$：属性权威为一个合法授权用户 u 随机选择唯一的 $t \in Z_p$。然后属性权威和中央控制运行两方安全计算协议，在两方没有泄露任何私有信息的情况下中央控制获得隐私输出 $x = (\alpha + at)\beta$；中央控制随机选择 $\mu \in Z_p$ 并且计算 $A = g^{x/\mu}$，将 A 传送给属性权威；属性权威接收到 A 后，计算 $B = A^{1/\beta^2}$ 并将 B 传送给中央控制；中央控制接收到 B 后，为用户 u 计算控制密钥 $\mathrm{CK}_u = B^\mu = g^{(\alpha+at)/\beta}$，并且中央控制保留 CK_u。协议流程如图 1 所示。每一步骤中 PoK 代表秘密值的知识证明，为简洁起见，证明过程省略，可以参考文献[16]。

$\mathrm{KeyGen}(S, t) \to (\mathrm{SK}_u)$：属性权威根据用户属性集合 S 计算 $K = g^t, \forall \mathrm{att} \in S: K_{\mathrm{att}} = \left(g^{\beta v_{\mathrm{att}}}\right)^t H(\mathrm{att})^t$。将用户私钥 $\mathrm{SK}_u = (K, \forall \mathrm{att} \in S: K_{\mathrm{att}})$ 传送给用户。

(3)数据加密：$\mathrm{Encrypt}(\mathrm{PK}_{\mathrm{AA}}, \mathrm{PK}_{\mathrm{CC}}, (\boldsymbol{M}, \rho), k) \to \mathrm{CT}$：数据拥有者首先用对称密钥 k 加密数据 m。然后数据拥有者指定一个基于属性的访问策略 $(\boldsymbol{M}, \rho)$ 用于加密对称密钥 k。最终获得密钥密文 CT_k 和 CT_m。为简洁起见，下文不再考虑对称加密

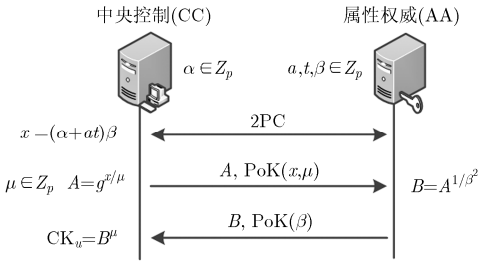


图 1 控制密钥产生协议

算法，直接将对称密钥 k 定义为密文。数据拥有者选择一个随机加密指数 $s \in Z_p$ 和一个随机向量 $\boldsymbol{v}=(s,y_2,\cdots,y_n)$，$y_2,\cdots,y_n \in Z_p$ 随机选择用来隐藏 s。对于矩阵 $\boldsymbol{M}$ 中的每一行 $i=1,2,\cdots,l$，计算 $\lambda_i=\boldsymbol{M}_i\boldsymbol{v}$，其中 $\boldsymbol{M}_i$ 是矩阵 $\boldsymbol{M}$ 的第 i 行。然后计算 $C=ke(g,g)^{\alpha s}, C'=g^{\beta s}$。随机选择 $r_1,r_2,\cdots,r_n \in Z_p$，然后计算 $\forall i=1,2,\cdots,l: C_{i,1}=g^{a\lambda_i}\left(g^{\beta v_{\rho(i)}}\right)^{-r_i} \cdot H(\rho(i))^{-r_i}$, $C_{i,2}=g^{r_i}$。输出密文 $\mathrm{CT}=(C,C',C_{i,1},C_{i,2})_{i=1,2,\cdots,l}$。只有当数据用户的属性满足访问策略时，才能正确解密密钥密文获得 k。

(4)数据解密：　$\mathrm{ReqGen}(\mathrm{SK}_u,\mathrm{CT}_k)\to \mathrm{RT}$：设 $S \in \mathcal{A}$ 是一个访问授权集合，参与者下标集合 $I \subset \{1,2,\cdots,l\}$ 被定义为 $I=\{i:\rho(i)\in S\}$，那么可以在多项式时间内找到一组常数 $\{w_i \in Z_p\}_{i\in I}$，如果 $\{\lambda_i\}$ 是对秘密 s 的有效共享份额，则等式 $\sum w_i\lambda_i = s$ 成立，其中 $i \in I$。首先用户随机选择 $z \in Z_p$，令 $\mathrm{SK}_u^{'}=z$，然后计算 $\mathrm{TK}_u=\mathrm{SK}_u^{1/z}$ 传送给云服务商，自己保留 $\mathrm{SK}_u^{'}$。云服务商接收到 TK_u 后，计算

$$\mathrm{RT}_1=\prod_{i\in I}\left(e\left(C_{i,1},K^{1/z}\right)e\left(C_{i,2},K_{\rho(i)}^{1/z}\right)\right)^{w_i}=e(g,g)^{ats/z} \quad (2)$$

然后将 $\mathrm{CT}'=(C,C',\mathrm{RT}_1)$ 传送给用户。用户计算 $\mathrm{RT}_2=C'^{1/z}=g^{\beta s/z}$，最后将 $\mathrm{RT}=(\mathrm{RT}_1,\mathrm{RT}_2)$ 传送给中央控制。

$\mathrm{TKGen}\left(\mathrm{RT},\mathrm{CK}_u\right)\to \mathrm{CT}_k^{''}$：中央控制运行该算法，计算 $\mathrm{CT}''=e\left(\mathrm{RT}_2,\mathrm{CK}_u\right)/\mathrm{RT}_1=e(g,g)^{\alpha s/z}$ 传送给用户。

$\mathrm{Decrypt}\left(\mathrm{CT}_k^{''},\mathrm{SK}_u^{'}\right)\to k$：数据用户接收到 $\mathrm{CT}_k^{''}$ 后，计算

$$k=\frac{C}{\left(\mathrm{CT}''\right)^{\mathrm{SK}_u'}}=\frac{ke(g,g)^{\alpha s}}{\left(e(g,g)^{\alpha s/z}\right)^z} \quad (3)$$

获得对称密钥 k，解密成功。

(5)属性撤销：　用户撤销：设置撤销列表 $\mathrm{RL}=\varnothing$，当用户 u 离开系统后，该用户应该不能再解密任何存储于服务器里的数据，也就是说用户 u 的访问权限应该被撤销，即用户撤销。本文方案中，当需要直接撤销用户 u 时，中央控制设置 $\mathrm{RL}=\mathrm{RL}\cup\{u\}$，在解密过程中直接拒绝为撤销列表中的用户返回中间密文 $\mathrm{CT}_k^{''}$，从而达到撤销用户的目的。

属性撤销：属性撤销即撤销用户属性集合中的某一个或某些属性，撤销后该用户失去该属性所对应的权限，但不影响其它属性的权限。同时也不能影响未撤销该属性但拥有该属性的用户的访问权限。为达到这样要求，本文的属性撤销包括以下 3 个算法。假设撤销用户 u 的属性 att'。

$\mathrm{UpKeyGen}\left(\mathrm{MSK}_{\mathrm{AA}},v_{\mathrm{att}'}\right)\to\left(\overline{v}_{\mathrm{att}'},\mathrm{UK}_{\mathrm{att}'}\right)$：当撤销一个属性时，属性权威运行该算法，以属性权威的主私钥 $\mathrm{MSK}_{\mathrm{AA}}$ 和撤销属性 att' 的版本密钥 $v_{\mathrm{att}'}$ 作为输入，随机选择 $\overline{v}_{\mathrm{att}'}\in Z_p\left(\overline{v}_{\mathrm{att}'}\neq v_{\mathrm{att}'}\right)$ 作为撤销属性 att' 的新版本密钥 $\overline{\mathrm{VK}}_{\mathrm{att}'}$。然后属性权威计算并输出撤销属性 att' 的更新密钥 $\mathrm{UK}_{\mathrm{att}'}=\beta\left(v_{\mathrm{att}'}-\overline{v}_{\mathrm{att}'}\right)$。然后通过安全信道将 $\mathrm{UK}_{\mathrm{att}'}$ 发送给云服务商用于更新关联 att' 的密文。同时更新属性 att' 的属性公钥 $\overline{\mathrm{PK}}_{\mathrm{att}'}=g^{\beta\overline{v}_{\mathrm{att}'}}$，然后通知所有用户 att' 的属性公钥已经更新。

$\mathrm{UpSK}\left(\mathrm{SK}_u,\mathrm{UK}_{\mathrm{att}'}\right)\to\overline{\mathrm{SK}}_u$：拥有该属性但未撤销该属性的用户向属性权威提交自己的私钥 SK_u。属性权威接收到该私钥后运行该算法计算 $\overline{K}_{\mathrm{att}'}=K_{\mathrm{att}'}\cdot K^{-\mathrm{UK}_{\mathrm{att}'}}$，并传送给用户。用户获得新的私钥 $\overline{\mathrm{SK}}_u=\left(K,\overline{K}_{\mathrm{att}'},\forall\mathrm{att}\in S\setminus\{\mathrm{att}'\}:K_{\mathrm{att}}\right)$。只有包含撤销属性的组件被更新，其它组件均保持不变。

$\mathrm{UpCT}\left(\mathrm{CT},\mathrm{UK}_{\mathrm{att}'}\right)\to\overline{\mathrm{CT}}$：为确保新加入的用户在满足访问结构的情况下依然能够正确解密先前发布的密文，所有关联撤销属性 att' 的密文应该被更新至最新版本。为减少数据拥有者的计算量，本文将密文更新的操作外包给云服务商，这样能够大大削减数据拥有者的计算量，同时减少数据拥有者和云服务商之间的通信量。本文采用代理重加密技术，这意味着云服务商在更新密文之前不需要知道密文的内容。云服务商接收到 $\mathrm{UK}_{\mathrm{att}'}$ 后运行该算法更新关联属性 att' 的密文。该算法以密文 CT 和撤销属性 att' 的更新密钥 $\mathrm{UK}_{\mathrm{att}'}$ 作为输入，如果 $\rho(i)=\mathrm{att}'$，计算 $\overline{C}_{i,1}=C_{i,1}\cdot\left(C_{i,2}\right)^{\mathrm{UK}_{\mathrm{att}'}}$。最后云服务商更新关联 att' 的密文为

$$\left.\begin{aligned}\overline{\mathrm{CT}}=\Big(&\overline{C}=C,\overline{C}'=C',\forall i=1,2,\cdots,l:\overline{C}_{i,2}=C_{i,2},\\&\text{if } \rho(i)\neq\mathrm{att}':\ \overline{C}_{i,1}=C_{i,1}\\&\text{if } \rho(i)=\mathrm{att}':\ \overline{C}_{i,1}=C_{i,1}\cdot\left(C_{i,2}\right)^{\mathrm{UK}_{\mathrm{att}'}}\Big)\end{aligned}\right\} \quad (4)$$

本文方案只更新关联撤销属性 att' 的密文组件，其它密文组件不变。这种方式能够节约大量计算资源，提高属性撤销的效率。

4.2 安全证明

本文基于第 3.2 小节中定义的安全模型证明定理 1。

定理 1　若 Decisional q-Parallel BDHE 假设在群 G 和 G_T 中成立，那么没有多项式时间敌手能够选择性地攻破本文方案，其中挑战矩阵 $\boldsymbol{M}^*\left(l^*\times n^*\right)$，$n^*\leq q$。

证明　假设敌手 $\mathcal{A}$ 能以不可忽略的优势 $\varepsilon=\mathrm{Adv}_{\mathcal{A}}$ 选择性地攻破本文方案，设其挑战矩阵为

$\boldsymbol{M}^*\left(l^*\times n^*\right)$，并且 $n^*\le q$。然后，我们能够构造挑战者 $\mathcal{C}$ 以不可忽略的优势攻破 Decisional q-Parallel BDHE 假设。

系统初始化：挑战者 $\mathcal{C}$ 以 q-Parallel BDHE 挑战 $\boldsymbol{y}$ 和 T 作为输入。敌手 $\mathcal{A}$ 提交访问策略 $\left(\boldsymbol{M}^*,\rho^*\right)$。

系统建立：挑战者 $\mathcal{C}$ 随机选择 $\alpha'\in Z_p$，计算 $e(g,g)^{\alpha}=e(g,g)^{\alpha'}e\left(g^a,g^{a^q}\right)$，这意味着 $\alpha=\alpha'+a^{q+1}$。随机选择 $\beta\in Z_p$，计算 g^{β}。选择一个随机预言机 H 并建立一个列表。当调用 H 时，如果 $H(x)$ 已经存在列表中，则直接返回结果；如果 $H(x)$ 不存在列表中，则随机选择 $z_x\in Z_p$，设 X 是满足 $\rho^*(i)=x$ 这一条件的 i 的集合，则设置 $H(x)=g^{z_x}\prod_{i\in X}g^{a\boldsymbol{M}^*_{i,1}/b_i}\cdot g^{a^2\boldsymbol{M}^*_{i,2}/b_i}\cdots g^{a^{n^*}\boldsymbol{M}^*_{i,n^*}/b_i}$。如果 $X=\varnothing$，则设置 $H(x)=g^{z_x}$。

查询阶段 1：该阶段挑战者 $\mathcal{C}$ 回答敌手 $\mathcal{A}$ 的私钥询问。因为本文采用两方安全计算生成用户私钥已经被证明是安全且不会泄露任何信息给敌手[16]，所以本文挑战者 $\mathcal{C}$ 以整体形式回答敌手 $\mathcal{A}$ 的私钥询问。假设挑战者 $\mathcal{C}$ 回答敌手 $\mathcal{A}$ 的私钥询问属性集合 S，并且 S 不能够满足 $\boldsymbol{M}^*\left(l^*\times n^*\right)$。

挑战者 $\mathcal{C}$ 随机选择 $v_x,r\in Z_p$，然后计算向量 $\boldsymbol{w}=\left(w_1,w_2,\cdots,w_{n^*}\right)\in Z_p^{n^*}$，其中 $w_1=-1$。对于所有的 $i,\rho^*(i)\in S$ 满足 $\boldsymbol{w}\boldsymbol{M}^*_i=0$。通过 LSSS 的定义，这样的向量能在多项式时间内计算得到。挑战者 $\mathcal{C}$ 隐含定义 $t=r+w_1a^q+w_2a^{q-1}+\cdots+w_{n^*}a^{q-n^*+1}$，然后计算 $K=g^r\prod_{i=1,2,\cdots,n^*}\left(g^{a^{q+1-i}}\right)^{w_i}=g^t$。通过 t 的定义和 $w_1=-1$，g^{at} 包含 $g^{-a^{q+1}}$ 项，而 $g^{-a^{q+1}}$ 在假设中并没有给出，但是在设置阶段，隐含设置 $\alpha=\alpha'+a^{q+1}$，因此 $g^{-a^{q+1}}$ 能通过与 $g^{\alpha}=g^{\alpha'}g^{\alpha^{q+1}}$ 相乘而被消去。挑战者 $\mathcal{C}$ 能够计算

$$\begin{aligned}\mathrm{CK}_u&=\left(g^{\alpha'}g^{a^{q+1}}g^{ar}g^{-a^{q+1}}\prod_{i=2,3,\cdots,n^*}\left(g^{a^{q+2-i}}\right)^{w_i}\right)^{1/\beta}\\&=\left(g^{\alpha'}g^{ar}\prod_{i=2,3,\cdots,n^*}\left(g^{a^{q+2-i}}\right)^{w_i}\right)^{1/\beta}\end{aligned}\tag{5}$$

如果不存在 i 使得 $\rho^*(i)\in S$，然后挑战者 $\mathcal{C}$ 能够计算 $K_x=K^{\beta v_x}K^{z_x}$；否则，挑战者 $\mathcal{C}$ 随机选择版本密钥 v_x，然后能够计算

$$K_x=K^{\beta v_x}K^{z_x}\cdot\prod_{i\in X}\prod_{j=1,2,\cdots,n^*}\left(g^{\left(a^j/b_i\right)r}\prod_{\substack{k=1,2,\cdots,n^*\\k\ne j}}\left(g^{a^{q+1+j-k}/b_i}\right)^{w_k}\right)^{\boldsymbol{M}^*_{i,j}}\tag{6}$$

然后将私钥发送给敌手 $\mathcal{A}$。

密钥更新询问阶段，假设敌手 $\mathcal{A}$ 提交用户 u 的一个属性 x'，如果属性 x' 有一个新版本密钥 $\overline{v}_{x'}$，并且 u 的属性 x' 没有被撤销，则回答密钥更新询问 $K_{x'}=K^{\beta\overline{v}_x}K^{z_x}$

$$\cdot\prod_{i\in X}\prod_{j=1,2,\cdots,n^*}\left(g^{\left(a^j/b_i\right)r}\prod_{\substack{k=1,2,\cdots,n^*\\k\ne j}}\left(g^{a^{q+1+j-k}/b_i}\right)^{w_k}\right)^{\boldsymbol{M}^*_{i,j}}\tag{7}$$

挑战阶段：敌手 $\mathcal{A}$ 提交两个等长的消息 m_0 和 m_1，挑战者 $\mathcal{C}$ 随机选择参数 $b\in\{0,1\}$ 生成挑战密文 $C=m_bTe\left(g^s,g^{\alpha'}\right)$ 和 $C'=g^{\beta s}$。挑战者 $\mathcal{C}$ 随机选择 $y_2',y_3',\cdots,y_{n^*}'\in Z_p$，并隐含地通过向量 $\boldsymbol{v}=\left(s,sa+y_2',sa^2+y_3',\cdots,sa^{n^*-1}+y_{n^*}'\right)\in Z_p^{n^*}$ 来共享密钥 s。另外，挑战者 $\mathcal{C}$ 选择随机参数 $r_1',r_2',\cdots,r_l'\in Z_p$。对于 $i=1,2,\cdots,n^*$，定义 R_i 为所有满足 $\rho^*(i)=\rho^*(k)$ 这一条件的 $k\ne i$ 的集合。因此挑战密文可以被计算

$$\begin{aligned}C_{i,2}&=g^{-r_i'}g^{-sb_i},\ C_{i,1}=\left(g^{\beta v_{\rho^*(i)}}H\left(\rho^*(i)\right)\right)^{r_i'}\\&\cdot\left(\prod_{j=2,3,\cdots,n^*}\left(g^a\right)^{\boldsymbol{M}^*_{i,j}\cdot y_j'}\right)\left(g^{b_i\cdot s}\right)^{\left(\beta v_{\rho^*(i)}-z_{\rho^*(i)}\right)}\\&\cdot\left(\prod_{k\in R_i}\prod_{j=1,2,\cdots,n^*}\left(g^{a^j\cdot s\cdot(b_i/b_k)}\right)^{\boldsymbol{M}^*_{k,j}}\right)\end{aligned}\tag{8}$$

最后将密文发送给敌手 $\mathcal{A}$。

查询阶段 2：类似查询阶段 1，敌手 $\mathcal{A}$ 继续向挑战者 $\mathcal{C}$ 提交一系列属性集合，其限制与查询阶段 1 相同。

猜测阶段：敌手 $\mathcal{A}$ 输出一个值 $b'\in\{0,1\}$ 作为对 b 的猜测。如果 $b'=b$，挑战者 $\mathcal{C}$ 输出 0 表示猜测 $T=e(g,g)^{a^{q+1}s}$；否则输出 1 表示猜测 T 为群 G_T 中的随机元素。当 $T=e(g,g)^{a^{q+1}s}$ 时，挑战者 $\mathcal{C}$ 能够提供一个有效的仿真。因此得出：$\Pr\left[\mathcal{C}\left(\boldsymbol{y},T=e(g,g)^{a^{q+1}s}\right)=0\right]=1/2+\mathrm{Adv}_{\mathcal{A}}$；当 T 为群 G_T 中的随机元素时，m_b 对于敌手来说是完全随机的，因此得出：$\Pr\left[\mathcal{C}(\boldsymbol{y},T=R)=0\right]=1/2$。因此，挑战者 $\mathcal{C}$ 能以不可忽略的优势攻破决策性 q-Parallel BDHE 假设。

定理 2 本文方案满足前向安全和后向安全。

证明 前向安全：被撤销属性的用户在下一个时间周期到来之前，仍然能够访问关联该属性的密文，即使他不再持有该属性。后向安全：新加入的用户在用最新的更新密钥通过周期性的重加密密文

之前，也许能够访问其加入以前的密文。这样一个不受控制的时期被称为脆弱窗口。对于属性撤销方案，前向安全和后向安全是方案最基本的安全需求。由于我们方案是立即属性撤销，所以不存在脆弱窗口。下面我们将说明本文方案能够满足前向安全和后向安全。

当发生用户撤销时，中央控制拒绝为撤销列表中的用户返回中间密文 CT_k''，其直接不能够解密任何密文，且不涉及其他用户，所以满足前向安全和后向安全。

当发生属性撤销时，AA 便会为被撤销的属性 att' 产生一个新的版本密钥 $\overline{\mathrm{VK}}_{\mathrm{att}'}$，并为未撤销该属性的用户升级密钥 $\overline{\mathrm{SK}}_u$，而撤销该属性的用户不能进行密钥升级，同时为涉及该撤销属性 att' 的密文进行升级。若被撤销属性 att' 的用户仍用自己先前的密钥去解密更新后的密文，根据数据解密阶段相关算法，最终解密结果为 $k=k\cdot e(g,g)^{\beta t r_{\mathrm{att}} w_{\mathrm{att}}\left(v_{\mathrm{att}'}-\overline{v}_{\mathrm{att}'}\right)}=k\cdot e(g,g)^{t r_{\mathrm{att}} w_{\mathrm{att}}\mathrm{UK}_{\mathrm{att}'}}$。而根据安全模型，敌手 $\mathcal{A}$ 不能获得 $t\in Z_p$，$r_{\mathrm{att}}\in Z_p$，$w_{\mathrm{att}}\in Z_p$ 和 $\mathrm{UK}_{\mathrm{att}'}=\beta\left(v_{\mathrm{att}'}-\overline{v}_{\mathrm{att}'}\right)$，所以无法解密获得 k，这确保了方案的前向安全。对于新加入的用户，他们的私钥是根据最新版本的属性密钥 $\overline{v}_{\mathrm{att}'}\in Z_p$ 生成。假设其属性集合满足密文的访问策略，此时试图用该私钥解密之前的密文，其可以获得 $k=k\cdot\prod_{i\in I}e(g,g)^{t\beta r_i w_i\left(\overline{v}_{\mathrm{att}'}-v_{\mathrm{att}'}\right)}=k\cdot\prod_{i\in I}e(g,g)^{-t r_i w_i\mathrm{UK}_{\mathrm{att}'}}$。而根据安全模型，$\mathcal{A}$ 不能获得 $t\in Z_p$，$r_i\in Z_p$，$w_i\in Z_p$ 和 $\mathrm{UK}_{\mathrm{att}'}=\beta\left(v_{\mathrm{att}'}-\overline{v}_{\mathrm{att}'}\right)$，所以无法解密获得 k，这确保了方案的后向安全。

5　方案分析及实验验证

5.1 理论分析

本节主要在功能性、存储成本、通信成本和计算效率方面将本文方案与已有几种撤销方案进行对比。对比过程中所使用描述符定义如下：$|p|$ 表示 Z_p 中数据元素的长度；$|g|$ 表示 G 中数据元素的长度；$|g_T|$ 表示 G_T 中数据元素的长度；$|C_k|$ 表示 Hur 方案中使用的密钥 KEK 的长度；n_c 表示与密文有关的属性个数；n_k 表示用户密钥中属性的个数；n_a 表示整个系统中属性的总个数；n_u 表示整个系统中用户的总个数。

5.1.1 功能比较　从表 1 可以看出，文献[9]方案使用了较弱的 DBDH 假设，但是该方案的表现能力差；所有的方案都实现了属性级的立即撤销能力，但是本文在满足这种撤销能力的同时，还可以实现系统级的用户撤销，一旦用户被撤销，那么该用户就失去了系统内所有的访问权限，这在实际也有应用之处；本文在实现属性撤销的同时还解决了密钥托管问题，在解密过程中将部分计算外包给云服务商，减少用户的计算负担。

5.1.2 存储成本　表 2 将本文方案与其它相关方案进行了存储成本的对比。本文方案将授权中心拆分为中央控制 CC 和属性权威 AA(相当于其它方案的授权中心)，它们的存储成本和其它方案授权中心的存储成本主要来自于主密钥。本文由于 CC 要存储密钥 α 和控制密钥 CK_u，所以存储量与用户量 n_u 成正比。文献[10]与文献[13]方案使用了较少的主密钥，文献[9,11]与本文方案的 AA 的存储量随着属性总数 n_a 成线性增长。数据拥有者的存储成本主要来自于公钥。文献[10]和文献[13]方案使用了最少的公钥，文献[9,11]与本文方案的公钥与属性总数 n_a 线性正相关。云服务商 CSP 的存储成本主要来自于密文与密文头。文献[10]方案中，数据拥有者将密文发送给 CSP 后，CSP 为每个属性群生成相应的密文头，因此其存储包括密文及密文头，其密文长度与密文相关属性个数 n_c 线性正相关，而密文头长度与密文相关属性个数 n_c 及用户总数 n_u 的乘积线性正相关。文献[9,13]与本文方案的密文长度与密文相关属性个数 n_c 线性正相关。数据用户的存储成本主要来自于其拥有的密钥。文献[10]方案中，每个用户都要存储一定的 KEK 来解密相应的指数进行密钥更新，因此其密钥长度不仅与用户拥有的属性个数 n_k 线性正相关，而且与整个系统中用户的总个数 n_u 成对数增长关系。本文和其它方案的密钥长度较短，只与用户拥有的属性个数 n_k 线性正相关。

表 1 属性撤销方案功能对比

方案	访问策略	安全假设	密钥托管	撤销粒度	撤销机制	外包解密
文献[9]方案	AND	DBDH	否	属性撤销	立即撤销	否
文献[10]方案	Tree	–	否	属性撤销	立即撤销	否
文献[11]方案	LSSS	q-parallel BDHE	否	属性撤销	立即撤销	否
文献[13]方案	LSSS	q-parallel BDHE	否	属性撤销	立即撤销	否
本文方案	LSSS	q-parallel BDHE	是	属性/用户撤销	立即撤销	是

表 2 存储成本对比

方案	中央控制	属性权威	数据拥有者	云服务商	数据用户
文献[9]方案	–	$(1+5n_a)\mid p\mid$	$(1+3n_a)\mid g\mid+\mid g_T\mid$	$3n_c\mid g\mid+\mid g_T\mid+2n_k\mid p\mid$	$(1+2n_k)\mid g\mid$
文献[10]方案	–	$\mid p\mid+\mid g\mid$	$2\mid g\mid+\mid g_T\mid$	$(2n_c+1)\mid g\mid+\mid g_T\mid$ $+(n_c\cdot n_u/2)\mid C_k\mid$	$(2n_k+1)\mid g\mid+\lg(n_u+1)\mid C_k\mid$
文献[11]方案	–	$(4+n_a)\mid p\mid$	$(4+2n_a)\mid g\mid+\mid g_T\mid$	$(3n_c+1)\mid g\mid+\mid g_T\mid$	$(2+n_k)\mid g\mid$
文献[13]方案	–	$3\mid p\mid$	$2\mid g\mid+\mid g_T\mid$	$(2n_c+3)\mid g\mid+\mid g_T\mid+n_c\mid p\mid$	$(2+n_k)\mid g\mid$
本文方案	$(1+n_u)\mid p\mid$	$(2+n_a)\mid p\mid$	$(2+n_a)\mid g\mid+\mid g_T\mid$	$(2n_c+1)\mid g\mid+\mid g_T\mid$	$(1+n_k)\mid g\mid$

5.1.3 通信成本 本文方案将授权中心拆分为中央控制 CC 和属性权威 AA(相当于其它方案的授权中心)。在生成密钥阶段 CC 与 AA 两方安全计算生成控制密钥，CC 与数据拥有者 O 之间传递 $\mathrm{PK}_{\mathrm{CC}}$ 用于数据加密，CC 与数据用户 U 之间传递解密信息产生的额外通信量是为了解决密钥托管问题和实现系统级用户撤销。这部分额外开销在其它方案中并没有涉及(故下文比较中没有给出这部分)，但是无论如何，本文在功能性方面较其它方案有较大优势。

表 3 将本文方案与其它相关方案进行了通信成本的对比。通信成本主要是由密钥与密文产生的。AA 与 U 之间的通信成本主要是由密钥产生的。AA 与 O 之间的通信成本主要是由公钥产生的。本文方案中撤销一个属性时，O 需要获得撤销属性的属性公开密钥。CSP 与 U 的通信成本主要是由密文产生的。而文献[10]方案中，CSP 不仅要发送密文，还要生成用户的 KEK 密钥而产生 $\lg(n_u+1)\big|C_k\big|$ 的通信成本，另外还要发送相应 $(n_c\cdot n_u)/2\big|C_k\big|$ 的密文头。CSP 与 O 之间的通信主要是由 O 生成的密文产生的。

5.2 实验分析

实验环境为 64 bit Ubuntu 14.04 操作系统、Intel® CoreTM i5-6200U(2.3 GHz)、内存 8 G，实验代码基于 Pairing-based Cryptography Library (PBC-0.5.14)[17]与 cpabe-0.11[18]进行修改与编写，并且使用基于 512 bit 有限域上的超奇异曲线 $y^2=x^3+x$ 中的 160 bit 椭圆曲线群。实验数据取运行 30 次所得的平均值。通过实验统计得知：PBC 库计算对运算的时间大约为 5.4 ms，G 与 G_T 的运算时间大约为 5.2 ms 与 0.8 ms。

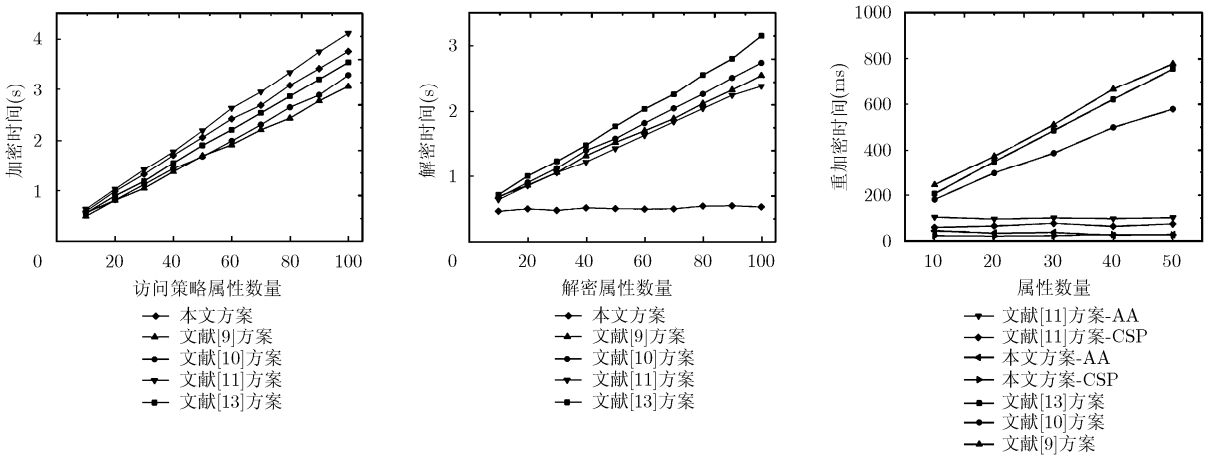
本文方案与其它几种方案分别在“加密时间”、“解密时间”和“重加密时间”3 个方面进行对比，测试出每一种方案中“加密时间”、“解密时间”和“重加密时间”与属性数量的关系，如图 2 所示。

图 2(a)描述了数据拥有者加密数据过程中加密时间与访问策略中属性数量的关系，从图 2(a)中可以看出加密时间与访问结构中的属性数量成线性增长关系。即当属性从 10 变化至 100，每一种方案的加密时间都近似成线性增长。但是通过几种方案对比，本文方案加密时间略高于文献[9,10,13]方案，优于文献[11]方案。需要注意的是，文献[10]方案加密过程中，中间节点的多项式操作涉及了适当数量的乘法，但是乘法运算所需时间很短，所以整体上表现较好。

图 2(b)描述了数据用户在解密过程中解密时间与解密所需属性数量的关系，从图 2(b)可以看出解密时间与解密属性数量成线性增长关系。即当属性从 10 变化至 100，每一种方案的解密时间都近似成线性增长。这种时间花费将给用户带来严重的计算负担，尤其是移动终端的用户带来难以承受的计算负担。然而，本文方案将复杂的解密计算外包给云服务商，完成解密过程中所需的大部分计算工作。

表 3 通信成本对比

方案	AA&U	AA&O	CSP&U	CSP&O
文献[9]方案	$(1+2n_k)\mid g\mid$	$(1+3n_a)\mid g\mid+\mid g_T\mid$	$(3n_c+2n_k)\mid g\mid+\mid g_T\mid$	$3n_c\mid g\mid+\mid g_T\mid$
文献[10]方案	$(1+2n_k)\mid g\mid$	$2\mid g\mid+\mid g_T\mid$	$(2n_c+1)\mid g\mid+\mid g_T\mid+(n_c\cdot n_u/2+\lg(n_u+1))\mid C_k\mid$	$2n_c\mid g\mid+(n_c+1)\mid g_T\mid$
文献[11]方案	$(4+n_k)\mid g\mid$	$(4+2n_a)\mid g\mid+\mid g_T\mid$	$(3n_c+1)\mid g\mid+\mid g_T\mid$	$(3n_c+1)\mid g\mid+\mid g_T\mid$
文献[13]方案	$(2+n_k)\mid g\mid$	$2\mid g\mid+\mid g_T\mid$	$(2n_c+3)\mid g\mid+\mid g_T\mid+n_c\mid p\mid$	$(2n_c+1)\mid g\mid+\mid g_T\mid$
本文方案	$(3+n_k)\mid g\mid$	$(2+n_a)\mid g\mid+\mid g_T\mid$	$(2n_c+1)\mid g\mid+\mid g_T\mid$	$(2n_c+1)\mid g\mid+\mid g_T\mid$



(a)加密时间　　(b)解密时间　　(c)属性撤销中重加密时间

图 2 属性撤销方案仿真时间对比

用户只需要进行少量计算就能够完成最终解密任务，即用户只需要定量地计算就能够完成解密工作，而解密时间不再随着属性数量的增加而增加。这种解密外包方式极大地提高了方案效率，对方案实际应用带来利好。

图 2(c)描述了数据重加密时间与系统属性数量的关系。当属性被撤销时，方案需要对密钥和密文进行更新。通过几种方案对比分析，文献[9,10,13]方案主要对密文进行更新，且随着属性数量逐渐增长，从图中可以发现它们的方案需要的计算时间较长，且重加密过程全部由云服务商独立完成。文献[11]方案与本文方案对密钥和密文进行更新时，只更新涉及到撤销属性的密钥和密文，因此需要的计算时间较少，重加密过程由 AA 和云服务商 CSP 参与。

6　结束语

本文提出一种可撤销属性基加密方案。该方案引入属性版本密钥 $\mathrm{VK}_{\mathrm{att}}$，通过更新版本密钥的方式达到属性撤销的目的，在属性撤销过程中本文只更新关联被撤销属性的密文组件和拥有撤销属性但未被撤销的用户的私钥组件；同时通过中央控制可以直接撤销某一个用户。本文方案将属性中心拆分为属性权威和中央控制，属性权威和中央控制二者运行两方计算产生用户私钥，解决了密钥托管问题。在解密过程中，为减少用户的计算负担而将复杂的计算外包给云服务商，减少用户的计算量。本文基于 q-Parallel BDHE 假设在随机预言机模型下对方案进行了选择访问结构明文攻击的安全性证明。最后对方案进行了理论分析与实验分析，分析结果表明所提方案无密钥托管问题，且具有较高系统效率。

参 考 文 献

[1] 张玉清, 王晓菲, 刘雪峰, 等. 云计算环境安全综述[J]. 软件学报, 2016, 27(6): 1328–1348. doi: 10.13328/j.cnki.jos.005004.
ZHANG Yuqing, WANG Xiaofei, LIU Xuefeng, *et al*. Survey on cloud computing security[J]. *Journal of Software*, 2016, 27(6): 1328–1348. doi: 10.13328/j.cnki.jos.005004.

[2] MOROVATI K, KADAM S, and GHORBANI A. A network based document management model to prevent data extrusion[J]. *Computers & Security*, 2016, 59(c): 71–91. doi: 10.1016/j.cose.2016.02.003.

[3] BETHENCOURT J, SAHAI A, and WATERS B. Ciphertext-policy attribute-based encryption[C]. 2007 IEEE Symposium on Security and Privacy (SP′07), Berkeley, CA, USA, 2007: 321–334. doi: 10.1109/SP.2007.11.

[4] LIU C W, HSIEN W F, YANG C C, *et al*. A survey of attribute-based access control with user revocation in cloud data storage[J]. *International Journal of Network Security*, 2016, 18(5): 900–916.

[5] PIRRETTI M, TRAYNOR P, MCDANIEL P, *et al*. Secure attribute-based systems[C]. Proceedings of the 13th ACM Conference on Computer and Communications Security, Alexandria, Virginia, USA, 2006: 99–112. doi: 10.1145/1180405.1180419.

[6] BOLDYREVA A, GOYAL V, and KUMAR V. Identity-based encryption with efficient revocation[C]. Proceedings of the 15th ACM Conference on Computer and Communications Security, Alexandria, Virginia, USA, 2008: 417–426. doi: 10.1145/1455770.1455823.

[7] HUANG Q, MA Z, YANG Y, *et al*. EABDS: Attribute-based secure data sharing with efficient revocation in cloud computing[J]. *Chinese Journal of Electronics*, 2015, 24(4): 862–868. doi: 10.1049/cje.2015.10.033.

[8] IBRAIMI L, PETKOVIC M, NIKOVA S, *et al*. Mediated ciphertext-policy attribute-based encryption and its application[C]. Information Security Applications: 10th International Workshop, Busan, Korea, 2009: 309–323. doi:

10.1007/978-3-642-10838-9_23.

[9] YU S, WANG C, REN K, *et al*. Attribute based data sharing with attribute revocation[C]. Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, Beijing, China, 2010: 261–270. doi: 10.1145/1755688.1755720.

[10] HUR J and NOH D K. Attribute-based access control with efficient revocation in data outsourcing systems[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2011, 22(7): 1214–1221. doi: 10.1109/TPDS.2010.203.

[11] YANG K, JIA X, and REN K. Attribute-based fine-grained access control with efficient revocation in cloud storage systems[C]. Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, Hangzhou, China, 2013: 523–528. doi: 10.1145/2484313.2484383.

[12] ZU L, LIU Z, and LI J. New ciphertext-policy attribute-based encryption with efficient revocation[C]. IEEE International Conference on Computer and Information Technology, Xi'an, China, 2014: 281–287. doi: 10.1109/CIT.2014.97.

[13] QIAN H, LI J, ZHANG Y, *et al*. Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation[J]. *International Journal of Information Security*, 2015, 14(6): 487–497. doi: 10.1007/s10207-014-0270-9.

[14] 王尚平, 余小娟, 张亚玲. 具有两个可撤销属性列表的密钥策略的属性加密方案[J]. 电子与信息学报, 2016, 38(6): 1406–1411. doi: 10.11999/JEIT150845.

WANG Shangping, YU Xiaojuan, and ZHANG Yaling. Revocable key-policy attribute-based encryption scheme with two revocation lists[J]. *Journal of Electronics & Information Technology*, 2016, 38(6): 1406–1411. doi: 10.11999/JEIT150845.

[15] VAANCHIG N, CHEN W, and QIN Z. Fine-grained access control for cloud data sharing by secure and efficient attribute-revocable ciphertext-policy attribute-based encryption[J]. *International Journal of Security and Its Applications*, 2016, 10(10): 303–320. doi: 10.14257/ijsia.2016.10.10.27.

[16] HUR J. Improving security and efficiency in attribute-based data sharing[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2013, 25(10): 2271–2282. doi: 10.1109/TKDE.2011.78.

[17] LYNN B. The pairing-based cryptography (PBC) library[OL]. http://crypto.stanford.edu/pbc.2006.

[18] BETHENCOURT J, SAHAI A, and WATERS B. Advanced crypto software collection: the cpabetoolkit[OL]. http://acsc.cs.utexas.edu/cpabe.2011.

赵志远： 男，1989 年生，博士生，研究方向为云安全与属性加密.

朱智强： 男，1961 年生，教授，研究方向为云计算与信息安全.

王建华： 男，1962 年生，教授，研究方向为云计算与网络安全.

孙 磊： 男，1973 年生，研究员，研究方向为云计算基础设施可信增强.