标准模型下可证明安全的支持大规模属性集与属性级用户撤销的 CP-ABE 方案

王建华[①②] 王光波[*②] 徐开勇[②]
[①](电子技术研究所 北京 100195)
[②](信息工程大学 郑州 450000)

摘 要：密文策略属性加密方案，特别是不受某个特定值限制的大规模属性集下的密文策略属性加密方案在云存储中得到了越来越广泛的应用，它能够实现细粒度的访问控制。但是在原始的属性加密方案中，解决动态的用户与属性撤销，是当前面临的重要挑战。为了解决这一问题，该文提出一个标准模型下可证明安全的支持大规模属性集的密文策略属性加密方案，该方案能够实现属性级的用户撤销，即若用户的某个属性被撤销，不会影响该用户其他合法属性的正常访问。为了实现撤销，将密钥分为两部分：为用户生成的私钥以及为云存储中心生成的授权密钥。在该方案中，若用户的属性被撤销，那么该属性对应的密文将进行更新，只有该属性没有被撤销的用户才能够成功地进行密钥更新而解密密文。该文基于 q-type 假设在标准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。最后对方案进行了性能分析与实验验证，实验结果表明，与已有相关方案相比，虽然为了实现属性撤销，增加了存储中心的计算负载，但是不需要属性中心的参与，因此降低了属性中心的计算负载，而且用户除了密钥外不需要其它额外参数来实现属性撤销，因此大大节省了存储空间。

关键词：密文策略属性加密；数据外包；大规模属性集；属性级的用户撤销
中图分类号： TP393.08 **文献标识码**： A **文章编号**：1009-5896(2017)12-3013-10
DOI：10.11999/JEIT170199

Ciphertext Policy Attribute-based Encryption Scheme Supporting Attribute Level User Revocation Under Large Universe

WANG Jianhua[①②] WANG Guangbo[②] XU Kaiyong[②]
[①](*Electronic Technology Institute*, *Beijing* 100195, *China*)
[②](*Information Science and Technology University*, *Zhengzhou* 450000, *China*)

Abstract: Ciphertext-Policy Attribute-Based Encryption (CP-ABE), especially large universe CP-ABE that is not bounded with the attribute set, is getting the more and the more extensive application to the cloud storage. However, there exists an important challenge in original large universe CP-ABE, namely dynamic user and attribute revocation. In this paper, a large universe CP-ABE scheme with efficient attribute level user revocation is proposed, namely the revocation to an attribute of some user can not influence the common access of other legitimate attributes. To achieve the revocation, the master key is divided into two parts: delegation key and secret key, which are sent to the cloud provider and user separately. In this scheme proposed, if an attribute is revoked, then the ciphertext corresponding to this attribute should be updated so that only persons who are not revoked will be able to carry out key updating and decrypt the ciphertext successfully. Note that, the proposed scheme is proved selectively secure in the standard model under "q-type" assumption. Finally, the performance analysis and experimental verification are carried out in this paper, and the experimental results show that, compared with the existing revocation schemes, although the proposed scheme increases the Computational load of Storage service Provider (CSP) in order to achieve the attribute revocation, it does not need the participation of Attribute Authority (AA), which reduces the computational load of AA. Moreover, the user does not need any additional parameters to achieve the attribute revocation except of the private key, thus saving the storage space greatly.

Key words: Ciphertext-Policy Attribute-Based Encryption (CP-ABE); Outsourced decryption; Large universe; Attribute level user revocation

收稿日期：2017-03-06；改回日期：2017-08-06；网络出版：2017-11-01
*通信作者：王光波 691759571@qq.com
基金项目：国家 973 计划项目(2013CB338001)
Foundation Item: The National 973 Program of China (2013 CB338001)

1 引言

Sahai 等人[1]在 2005 年提出了属性加密(Attribute-Based Encryption, ABE)的概念，将密文与密钥与一系列的属性相关联，通过定义访问结构，指定能够解密数据的属性集合，实现细粒度的访问控制，属性加密方案凭借其灵活的访问结构在云存储中得到了广泛的应用。最初的 ABE 只能实现门限操作，策略表达不够丰富。因此，有学者提出了基于密文策略[2−4](Ciphertext-Policy, CP)和密钥策略[5,6](Key-Policy, KP)的 ABE 机制，实现丰富的属性操作，因此，能够支持更加灵活的访问控制策略。

然而，所有这些 ABE 方案，都存在一个共同的缺陷，即仅支持小规模的属性集，其属性规模受某个特定值限制，需要在系统初始化时进行设定，因此限制了 ABE 的广泛应用。为了解决这一问题，Lewko 等人[7]第 1 次提出将 ABE 方案进行分类：小规模属性集的 ABE 和大规模属性集的 ABE。在小规模属性集的 ABE 中，系统初始设定的公钥参数随着属性集成线性增长，而在大规模属性集的 ABE 中，属性集可以设定为任意数值，且公钥参数保持不变。随后，Rouselakis 等人[8]提出了基于素数阶双线性群构造的两个大属性集下的 ABE 方案：CP-ABE 和 KP-ABE，该方案被证明为标准模型 q-type 假设下安全的。然而该方案并没有涉及动态的属性和用户撤销问题，而该问题在云存储应用中至关重要。因为，云存储环境下存在大量的用户，而 ABE 中不同的用户可能共享相同的属性。若是某个用户的某个属性被撤销，如何保证在不影响其他正常用户访问的前提下，对该用户实现相应访问权限的撤销，成为亟待解决的问题。

近来，在 ABE 的实际应用中，用户撤销的重要性引起了人们的重视，Ostrovsky 等人[9]提出了一种可实现用户撤销的 ABE 方案。该方案通过对撤销用户的身份进行 AND 的“非”操作来实现撤销，但是效率太低。随后，Staddon 等人[10]提出了一种用户可撤销的 KP-ABE 方案，但是该方案只能在满足密文相关属性正好为整个属性集的一半时才能被使用，因此限制太高，不符合实际应用。Liang 等人[11]提出了一种利用二叉结构来实现用户撤销的 CP-ABE 方案，由属性中心生成更新密钥实现撤销，但是效率较低，而且大大增加了属性中心的负担。

需要注意的是，以上几种方案都只能实现系统级的用户撤销，即一旦某个用户的某个属性被撤销，其失去了系统中所有其他属性对应的访问权限。在属性级用户撤销方面，文献[12~14]使用为每个属性设置有效期来实现属性撤销，但是我们称这种方式为粗粒度的撤销，因为其不能实现实时撤销。Yang 等人[15]提出了一种云存储下的 CP-ABE 方案，该方案为每个属性生成两个对应的公开参数，当进行属性撤销时，由属性中心更新需要撤销属性对应的公开参数，并为用户更新密钥，因此不仅加重了属性中心的计算负载，而且增大了属性中心与用户间的通信负载。

虽然以上方案可以实现属性级的用户撤销，但是方案仅适用于小规模属性集环境。即方案的属性集合受某个特定值的限制，为一个关于系统安全参数的多项式规模，需要在系统初始化时进行设定，而且公钥参数随着属性集合的大小成线性增长，因此，方案的灵活性较差，限制了方案的广泛应用。Hur 等人[16]提出了一种基于密钥加密密钥结构来实现属性撤销的 CP-ABE 方案，该方案不仅能够实现属性级的用户撤销，而且支持大规模属性集合。在该方案中，用户需要额外存储 $\lg(n_u+1)$ 长度的密钥加密密钥，其中 n_u 表示系统内的所有用户，而且该方案被证明为通用群模型下的安全性，而很多通用群模型下安全的方案被证明在实际应用中并不安全。本文针对这一问题展开研究，提出了一种标准模型下可证明安全的支持大规模属性集合和属性级用户撤销的 CP-ABE 方案，在该方案中，属性集合不受某个特定值的限制，可以为关于系统安全参数的指数规模，而且公钥参数与属性集合的大小无关，为常数。本文方案结合代理重加密实现属性撤销，即将大部分本由属性中心执行的撤销操作转移到云存储中心执行，大大降低了属性中心的计算负载。另外，该方案的属性中心不仅需要为用户生成解密密钥，而且需要为云存储中心生成授权密钥。一旦用户的某个属性被撤销，云存储中心将基于授权密钥和广播加密方案更新该属性对应的密文，只有未被撤销的用户才能够成功地进行密钥更新从而解密密文。

2 相关技术

在方案提出前，首先对文中将用到的相关技术进行简单介绍，包括双线性群及确定性 q-type 假设。

2.1 双线性群

定义 1(双线性群) 令 ψ 是一个群生成算法，以安全参数 λ 为输入，输出 $(p,\mathbb{G},\mathbb{G}_T,e)$。其中 p 为素数，由安全参数 λ 决定，$\mathbb{G}$ 和 $\mathbb{G}_T$ 是两个阶为 p 的循环

群，$e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$ 是一个满足下面条件的映射：

(1) 双线性：$\forall u,v\in\mathbb{G},a,b\in\mathbb{Z}_p,e(u^a,v^b)=e(u,v)^{ab}$。

(2)非退化性：$\exists g\in\mathbb{G}$ 使得 $e(g,g)$ 在 $\mathbb{G}_T$ 中的阶是 p。

(3)计算性：存在有效的对运算算法。

2.2 确定性 q-type 假设

定义 2(q-type 假设)　令 $\mathbb{G}$ 表示阶为 p 的双线性群，$a,s,b_1,b_2,\cdots,b_q$ 为 $\mathbb{Z}_p$ 内随机选择的参数，g 为 $\mathbb{G}$ 的生成元。若攻击者给定参数 $\boldsymbol{y}$：

$$\boldsymbol{y}=\begin{cases} g^{a^i},g^{b_j},g^{sb_j},g^{a^ib_j},g^{a^i/b_j^2}, & \forall(i,j)\in[q,q]\\ g^{a^ib_j/b_{j'}^2}, & \forall(i,j,j')\in[2q,q,q],\ j\neq j'\\ g^{a^i/b_j}, & \forall(i,j)\in[2q,q],\ i\neq q+1\\ g^{sa^ib_j/b_{j'}},g^{sa^ib_j/b_{j'}^2}, & \forall(i,j,j')\in[2q,q,q],\ j\neq j' \end{cases}$$

算法 $\mathfrak{B}$ 通过输出 $\beta\in\{0,1\}$ 来进行猜测，定义其拥有优势 ε 来解决群 $\mathbb{G}$ 下的 q-type 假设，若：$\left|\Pr\left[\mathcal{B}\left(\boldsymbol{y},e(g,g)^{a^{q+1}s}\right)=0\right]-\Pr\left[\mathcal{B}(\boldsymbol{y},R)=0\right]\right|\geq\varepsilon$。

3　大规模属性集下支持属性级用户撤销的 CP-ABE 方案

本节首先对提出的大规模属性集下支持属性级用户撤销的 CP-ABE 方案进行构造，接着对其进行了安全性证明。

3.1 方案构造

3.1.1 系统初始化　系统初始化阶段，属性中心生成系统的相关参数，包括公开密钥与主密钥。

初始化算法：$\text{Setup}\left(1^\lambda\right)\to(\text{PK},\text{MSK})$。

属性中心以安全参数 1^λ 为输入，运行群生成函数 ψ 获得系统参数 $D=(p,\mathbb{G},\mathbb{G}_T,e)$，其中 p 为素数，$\mathbb{G}$ 和 $\mathbb{G}_T$ 是 p 阶循环群，e 是一个双线性映射。$g\in\mathbb{G}$ 为群 $\mathbb{G}$ 的生成元。属性集合定义为 $\mathcal{U}=\mathbb{Z}_p$。

然后，算法随机选择参数 $g,u,h,w,v\in\mathbb{G}$ 和 $\alpha_1,\alpha_2\in\mathbb{Z}_p$，并且满足 $\alpha_1+\alpha_2=\alpha \bmod p$。最后系统公开密钥参数 PK 设置为：$\text{PK}=(g,u,h,w,v,e(g,g)^\alpha)$，主密钥 MK 设置为：$\text{MK}=(\alpha_1,\alpha_2)$。

3.1.2 密钥生成　为了实现外包解密，提高效率，生成密钥如下：

密钥生成算法：$\text{KeyGen}_{\text{out}}(\text{PK},\text{MK},S=\{s_1,s_2,\cdots,s_k\}\subseteq\mathbb{Z}_p)\to(\text{SK}_1,\text{SK}_2)$。

算法以系统公开密钥 PK、主密钥 MK 和用户属性集合 $S\subset\mathcal{U}$ 为输入，然后算法随机选择 $k+1$ 个指数 $r',r_1',r_2',\cdots,r_k'\in\mathbb{Z}_p$，并为用户生成相应的密钥为

$$\text{SK}_1'=\left(K_0',K_1',\left\{K_{\sigma,2}',K_{\sigma,3}'\right\}_{\sigma=1}^k\right)，\text{其中，}$$

$$K_0'=g^{\alpha_1}w^{r'},\ K_1'=g^{r'}$$

$$\left\{K_{\sigma,2}'=g^{r_\sigma'},K_{\sigma,3}'=\left(u^{s_\sigma}h\right)^{r_\sigma'}v^{-r'}\right\}_{\sigma=1}^k$$

接着，算法使用主密钥 MK 的另一参数 α_2 为 CSP 生成授权密钥 $\text{SK}_2=g^{\alpha_2}$。

需要注意的是，用户接收到密钥 SK_1' 后，算法随机选择参数 $z\in\mathbb{Z}_p^*$，并计算：

$$K_0=\left(K_0'\right)^{1/z}=\left(g^{\alpha_1}w^{r'}\right)^{1/z},\ K_1=\left(K_1'\right)^{1/z}=\left(g^{r'}\right)^{1/z}$$

$$\left\{K_{\sigma,2}=\left(K_{\sigma,2}'\right)^{1/z}=\left(g^{r_\sigma'}\right)^{1/z},\right.$$

$$\left.K_{\sigma,3}=\left(K_{\sigma,3}'\right)^{1/z}=\left(\left(u^{s_\sigma}h\right)^{r_\sigma'}v^{-r'}\right)^{1/z}\right\}_{\sigma=1}^k$$

令 $r=r'/z,r_1=r_1'/z,r_2=r_2'/z,\cdots,r_k=r_k'/z$，得到密钥为

$$K_0=g^{\alpha_1/z}w^r,\quad K_1=g^r$$

$$\left\{K_{\sigma,2}=g^{r_\sigma},K_{\sigma,3}=\left(u^{s_\sigma}h\right)^{r_\sigma}v^{-r}\right\}_{\sigma=1}^k$$

因此，算法设置外包密钥为 $\text{TK}=\left(K,\widetilde{K},L,\{K_i=h_i^r\}_{i\in S}\right)$，密钥 $\text{SK}_1=(z,\text{TK})$。

3.1.3 数据加密　当用户想要将数据 m 放到 CSP 时，他首先定义访问控制策略 $(\boldsymbol{M},\rho)$，然后运行加密算法 $\text{Encrypt}(\text{PK},m,(\boldsymbol{M},\rho))$ 对 m 进行加密。

加密算法：$\text{Encrypt}(\text{PK},m,(\boldsymbol{M},\rho))\to\text{CT}$。

算法以系统公开密钥 PK、明文消息 m 和访问控制策略 $(\boldsymbol{M},\rho)$ 为输入，其中 $\boldsymbol{M}$ 是一个 $l\times n$ 矩阵，然后算法随机选择指数 $t_1,t_2,\cdots,t_l\in\mathbb{Z}_p$ 和参数 $s,v_2,v_3,\cdots,v_n\in\mathbb{Z}_p$，并定义向量 $\boldsymbol{v}=(s,v_2,v_3,\cdots,v_n)$，对 $\boldsymbol{M}$ 的每一行 $\boldsymbol{M}_i$，计算内积 $\lambda_i=\boldsymbol{M}_i\cdot\boldsymbol{v}$，并随机选择 $r_i\in\mathbb{Z}_p$，算法输出密文：

$$\text{CT}=\left((\boldsymbol{M},\rho),C=m\cdot e(g,g)^{\alpha s},C_0=g^s,\right.$$

$$\left.\left\{C_{i,1}=w^{\lambda_i}v^{t_i},C_{i,2}=\left(u^{\rho(i)}h\right)^{-t_i},C_{i,3}=g^{t_i}\right\}_{i=1}^l\right)$$

3.1.4 数据重加密　当用户集 RL_x 的属性 x 被撤销时，为了撤销该属性对应的访问权限，使用广播属性加密进行更新如下：

重加密算法：$\text{Re-encrypt}(\text{PK},\text{CT},\text{SK}_2,\text{RL}_x)\to\text{RCT},\text{SK}_2'$。

算法以系统公开密钥 PK、密文 $\text{CT}=\left(C,C_0,\left\{C_{i,1},C_{i,2},C_{i,3}\right\}_{i=1}^l\right)$、授权密钥 SK_2 和属性 x 的撤销用户集 RL_x 为输入，令 ID_i 表示用户 i 的身份。

(1)若无属性被撤销，即 $\text{RL}_x=\Phi$，那么 CSP 随机选择参数 $k\in\mathbb{Z}_p$ 并重新加密密文 CT 如下：

$$\mathrm{CT}'=\Big(C'=C=m\cdot e(g,g)^{\alpha s},\ C_0'=C_0=g^s,\ C_1'=g^{s/k},\ \forall i=1,2,\cdots,l: C_{i,1}'=w^{\lambda_i}v^{t_i}v^k,\ C_{i,2}'=\left(u^{\rho(i)}h\right)^{-t_i}\left(u^{\rho(i)}h\right)^{-k},\ C_{i,3}'=g^{t_i}g^k\Big)$$

因此，重加密密文被设置为 $\mathrm{RCT}=\mathrm{CT}'$。另外，重加密算法将更新相应的授权密钥为 $\mathrm{SK}_2'=\left(g^{\alpha_2}\right)^k$。

(2)若用户 ID_j 的属性 x 被撤销，即 $\mathrm{RL}_x=\Phi$，那么算法首先随机选择指数 $v_x\in\mathbb{Z}_p$，并使用文献[17]中定长密文与定长密钥的广播加密方案为对 v_x 进行加密生成相应的密文头 CH_x。然后算法同样随机选择参数 $k\in\mathbb{Z}_p$ 并重新加密密文 CT 如下：

$$\mathrm{CT}'=\Big(C'=C=m\cdot e(g,g)^{\alpha s},\ C_0'=C_0=g^s,\ C_1'=g^{s/k},\ \forall i=1,2,\cdots,l,\ C_{i,1}'=w^{\lambda_i}v^{t_i}v^k,\ C_{i,2}'=\left(u^{\rho(i)}h\right)^{-t_i}\left(u^{\rho(i)}h\right)^{-k},\ \text{for } \rho(i)\neq x: C_{i,3}'=g^{t_i}g^k,\ \text{for } \rho(i)=x:\ C_{i,3}'=\left(g^{t_i}g^k\right)^{1/v_x}\Big)$$

因此，重加密密文被设置为 $\mathrm{RCT}=(\mathrm{CH}_x,\mathrm{CT}')$。

3.1.5 部分解密 为了实现用户的外包解密，用户需要将外包密钥 TK 发送给 CSP，然后 CSP 代为进行部分解密如下：

部分解密算法： $\mathrm{Transform}(\mathrm{TK},\mathrm{SK}_2',\mathrm{RCT})\rightarrow\mathrm{TCT}$。

算法以外包密钥 $\mathrm{TK}=(K_0,K_1,\{K_{\sigma,2},K_{\sigma,3}\}_{\sigma=1}^k)$、授权密钥 SK_2' 和重加密密文 RCT 为输入。

(1)无属性被撤销，即 $\mathrm{CH}_x=\Phi$。

$\mathrm{RCT}=\left(C',C_0',C_1',\left\{C_{i,1}',C_{i,2}',C_{i,3}'\right\}_{i=1}^l\right)$，若与外包密钥 TK 有关的用户属性集合 S 满足密文 RCT 中的访问策略 $(\boldsymbol{M},\rho)$，则 CSP 能在多项式时间计算 $\{w_i\in\mathbb{Z}_p\}_{i\in I}$ 使等式成立：$\sum_{i\in I}w_i\boldsymbol{M}_i=(1,0,\cdots,0)$。然后计算：

$$B=\prod_{i\in I}\left(e\left(C_{i,1}',K_1\right)e\left(C_{i,2}',K_{i,2}\right)e\left(C_{i,3}',K_{i,3}\right)\right)^{w_i}=\prod_{i\in I}\left(e\left(w^{\lambda_i}v^{t_i+k},g^r\right)e\left(\left(u^{\rho(i)}h\right)^{-t_i-k},g^{r_i}\right)\cdot e\left(g^{t_i+k},\left(u^{A_i}h\right)^{r_i}v^{-r}\right)\right)^{w_i}=e(g,w)^{rs}$$

$$D=e\left(C_0',K_0\right)=e\left(g^s,g^{\alpha_1/z}w^r\right)=e(g,g)^{\alpha_1 s/z}e(g,w)^{rs}$$

$$E=e\left(\mathrm{SK}_2',C_1'\right)=e\left(\left(g^{\alpha_2}\right)^k,g^{s/k}\right)=e(g,g)^{\alpha_2 s}$$

$$F=D/B=e(g,g)^{\alpha_1 s/z}e(g,w)^{rs}\Big/e(g,w)^{rs}=e(g,g)^{\alpha_1 s/z}$$

部分解密完成后，CSP 将 $\mathrm{TCT}=(C',E,F)$ 发送给用户进行最后解密。

(2)用户集 RL_x 的属性 x 被撤销时，即 $\mathrm{CH}_x\neq\Phi$。

此时，$\mathrm{RCT}=(\mathrm{CH}_x,\mathrm{CT}')$，$\mathrm{CT}'=\left(C',C_0',C_1',\left\{C_{i,1}',C_{i,2}',C_{i,3}'\right\}_{i=1}^l\right)$，接下来对密文 CT' 实施部分解密如下：

$$\rho(i)\neq x: B_i=e\left(C_{i,1}',K_1\right)e\left(C_{i,2}',K_{i,2}\right)e\left(C_{i,3}',K_{i,3}\right)=e(g,w)^{r\lambda_i}$$

$\rho(i)=x$: $C_{i,1}',C_{i,2}',C_{i,3}'$ are kept unchanged

$$D=e\left(C_0',K_0\right)=e\left(g^s,g^{\alpha_1/z}w^r\right)=e(g,g)^{\alpha_1 s/z}e(g,w)^{rs}$$

$$E=e\left(\mathrm{SK}_2',C_1'\right)=e\left(\left(g^{\alpha_2}\right)^k,g^{s/k}\right)=e(g,g)^{\alpha_2 s}$$

因此算法设置部分解密后的密文为：$\mathrm{TCT}'=\left(C',\{B_i\}_{\rho(i)\neq x},\left\{C_{i,1}',C_{i,2}',C_{i,3}'\right\}_{\rho(i)=x},D,E\right)$，并将其发送给用户进行最后解密。

3.1.6 解密 用户得到部分解密密文后，进行最后解密如下：

解密算法： $\mathrm{Decrypt}(\mathrm{TCT},\mathrm{SK}_1)\rightarrow m$。

算法以部分解密密文 TCT 与用户密钥 SK_1 为输入，然后解密如下：

(1)无属性被撤销，即 $\mathrm{TCT}=(C',E,F)$。此时用户计算：

$$C'/(E\cdot F^z)=m\cdot e(g,g)^{\alpha s}\Big/\left(e(g,g)^{\alpha_2 s}\cdot\left(e(g,g)^{\alpha_1 s/z}\right)^z\right)=m\cdot e(g,g)^{\alpha s}/e(g,g)^{\alpha s}=m$$

(2)用户集 RL_x 的属性 x 被撤销时，即 $\mathrm{TCT}=\left(\mathrm{CH}_x,C',\{B_i\}_{\rho(i)\neq x},\left\{C_{i,1}',C_{i,2}',C_{i,3}'\right\}_{\rho(i)=x},D,E\right)$。

若满足 $\mathrm{ID}\notin\mathrm{RL}_x$，那么用户可以解密广播密文得到相应的指数 v_x，并且接着计算：

$$B_i=e\left(C_{i,1}',K_1\right)e\left(C_{i,2}',K_{i,2}\right)e\left(C_{i,3}',\left(K_{i,3}\right)^{v_x}\right)=e(g,w)^{r\lambda_i}$$

若用户的属性集合能够满足访问控制策略 $(\boldsymbol{M},\rho)$，那么 CSP 能在多项式时间计算 $\left\{w_i\in\mathbb{Z}_p\right\}_{i\in I}$ 使以下等式成立：$\sum_{i\in I}w_i\boldsymbol{M}_i=(1,0,\cdots,0)$。然后计算：

$$B=\prod_{i\in I}(B_i)^{w_i}=\prod_{i\in I}\left(e(g,w)^{r\lambda_i}\right)^{w_i}=e(g,w)^{rs}$$

$$F=(D/B)^z=\left(e(g,g)^{\alpha_1 s/z}e(g,w)^{rs}\Big/e(g,w)^{rs}\right)^z=\left(e(g,g)^{\alpha_1 s/z}\right)^z=e(g,g)^{\alpha_1 s}$$

$$C'/(E\cdot F)=m\cdot e(g,g)^{\alpha s}\Big/\left(e(g,g)^{\alpha_2 s}\cdot e(g,g)^{\alpha_1 s}\right)=m\cdot e(g,g)^{\alpha s}/e(g,g)^{\alpha s}=m$$

由此可见，本文所提方案中初始化算法生成的公钥参数为 $\mathrm{PK}=(g,u,h,w,v,e(g,g)^{\alpha})$，其规模与属性集合的大小无关，只包含了 6 个群元素，为常数。实际上，这些参数构造了两个不同“层”来实现大

属性集下的 CP-ABE 方案。在“属性层”，参数 u 和 h 提供了一个 Boneh-Boyen 形式[17]的哈希函数，在“密钥共享层”，参数 w 在密钥生成算法中拥有随机性 r，而在加密算法中则共享随机性 s。另外，参数 v 将两“层”绑定在一起。参数 g 和 $e(g,g)^{\alpha}$ 用来引进主密钥功能，并且实现正确的数据解密。

3.2 安全证明

引理 1　若确定性的 q-type 假设在群 $\mathbb{G}$ 与 $\mathbb{G}_T$ 中成立，那么没有多项式时间的攻击者能选择性地攻破本文提出的 CP-ABE 方案，其中挑战矩阵为 $\boldsymbol{M}^*\left(l^*\times n^*\right)$，且 $l^*,n^*\le q$。

证明　假设攻击者 $\mathcal{A}$ 能以不可忽略的优势 $\varepsilon=\mathrm{Adv}_{\mathcal{A}}$ 选择性地攻破本文方案，而且假设其挑战矩阵为 $\boldsymbol{M}^*\left(l^*\times n^*\right)$，且 $l^*,n^*\le q$。接下来，我们将构造仿真器 $\mathcal{B}$ 来攻破确定性的 q-type 假设。

选择阶段：仿真器 $\mathcal{B}$ 以 q-type 假设挑战 $\boldsymbol{y},T$ 为输入。并且攻击者 $\mathcal{A}$ 给定访问控制 $\left(\boldsymbol{M}^*,\rho^*\right)$ 与属性 x^* 的撤销列表 RL_{x^*}，其中 $\boldsymbol{M}^*$ 有 n^* 列。

参数设置阶段：仿真器 $\mathcal{B}$ 随机选择指数 $\alpha',\alpha''\in\mathbb{Z}_p$，并通过计算 $e(g,g)^{\alpha}=e\left(g^a,g^{a^q}\right)\cdot e(g,g)^{\alpha'}\cdot e(g,g)^{\alpha''}$ 来隐含地设置 $\alpha_1=\alpha'+a^{q+1}$, $\alpha_2=\alpha''$, $\alpha=\alpha'+a^{q+1}+\alpha''$。然后 $\mathcal{B}$ 随机选择指数 $u',v',h'\in\mathbb{Z}_p$，并利用 q-type 假设实例构造如下参数：

$$u=g^{u'}\cdot\prod_{(j,k)\in[l,n]}\left(g^{a^k/b_j^2}\right)^{\boldsymbol{M}^*_{j,k}}$$

$$h=g^{h'}\cdot\prod_{(j,k)\in[l,n]}\left(g^{a^k/b_j^2}\right)^{-\rho^*(j)\boldsymbol{M}^*_{j,k}},w=g^a$$

$$v=g^{v'}\cdot\prod_{(j,k)\in[l,n]}\left(g^{a^k/b_j}\right)^{\boldsymbol{M}^*_{j,k}}$$

$$e(g,g)^{\alpha}=e\left(g^a,g^{a^q}\right)\cdot e(g,g)^{\alpha'}\cdot e(g,g)^{\alpha''}$$

最后仿真者 $\mathcal{B}$ 发送给攻击者 $\mathcal{A}$ 公开密钥参数为

$$\mathrm{PK}=\left(g,u,h,w,v,e(g,g)^{\alpha}\right)$$

查询阶段 1：$\mathcal{A}$ 向 $\mathcal{B}$ 进行如下查询：密钥生成查询 $\mathcal{O}_{\mathrm{kg}}$ 与密文重加密查询 $\mathcal{O}_{\mathrm{ree}}$。

(1) $\mathcal{A}$ 向 $\mathcal{B}$ 进行用户身份 ID_j 和用户属性集合 S_j 的密钥生成查询 $\mathcal{O}_{\mathrm{kg}}$，若 $\mathrm{ID}_j\notin\mathrm{RL}_{x^*}$，则设置 $S_j^{'}=S_j$，若 $\mathrm{ID}_j\in\mathrm{RL}_{x^*}$，则设置 $S_j^{'}=S_j\setminus\left\{x^*\right\}$。若 $S_j^{'}$ 满足访问控制 $\left(\boldsymbol{M}^*,\rho^*\right)$，输出 $\perp$。否则生成用户密钥如下：

仿真器 $\mathcal{B}$ 首先计算向量 $\boldsymbol{w}=\left(w_1,w_2,\cdots,w_{n^*}\right)\in\mathbb{Z}_p^n$，其中 $w_1=-1$，且对于所有的 $\rho^*(i)\in S_j^{'}$ 满足 $M_i^*\boldsymbol{w}^{\mathrm{T}}=0$。

然后仿真器随机选择 $t\in\mathbb{Z}_p$，并定义 r 为

$$\begin{aligned}r&=t+w_1a^q+w_2a^{q-1}+\cdots+w_{n^*}a^{q+1-n^*}\\&=r+\sum_{i\in[n^*]}w_ia^{q+1-i}\end{aligned}$$

接下来，计算密钥组件 $K_1^{'}$ 为

$$\begin{aligned}K_1^{'}&=g^r=g^{\left(t+w_1a^q+w_1a^{q+1}+\cdots+w_{n^*}a^{q+1-n^*}\right)}\\&=g^t\prod_{i=1,\cdots,n^*}\left(g^{a^{q+2-i}}\right)^{w_i}\end{aligned}$$

通过 r 的定义及 $w_1=-1$, w^r 包含了 $g^{-a^{q+1}}$ 项，而 $g^{-a^{q+1}}$ 在假设中并没有给出，但是在生成密钥 $K_0^{'}$ 时，由于隐含地设置 $\alpha_1=\alpha'+a^{q+1}$，因此 $g^{-a^{q+1}}$ 能通过与 $g^{\alpha_1}=g^{\alpha'}g^{a^{q+1}}$ 相乘而被取消：

$$\begin{aligned}K_0^{'}&=g^{\alpha_1}w^r=g^{\alpha'}g^{a^{q+1}}g^{at}\prod_{i\in[n^*]}\left(g^{a^{q+2-i}}\right)^{w_i}\\&=g^{\alpha'}\left(g^a\right)^t\prod_{i=2}^{n^*}\left(g^{a^{q+2-i}}\right)^{w_i}\end{aligned}$$

接下来，仿真器 $\mathcal{B}$ 将计算密钥 $K_{\sigma,2}^{'},K_{\sigma,3}^{'},\forall\sigma\in S_j^{'}$，为了构造密钥，$\mathcal{B}$ 首先设置通用项 v^{-r} 如下：

$$\begin{aligned}v^{-r}&=v^{-t}\left(g^{v'}\prod_{(j,k)\in\left[l^*,n^*\right]}g^{a^k\boldsymbol{M}^*_{j,k}/b_j}\right)^{-\sum_{i\in[n^*]}w_ia^{q+1-i}}\\&=v^{-t}\prod_{i\in\left[n^*\right]}\sum\left(g^{a^{q+1-i}}\right)^{-v'w_i}\\&\quad\cdot\prod_{(i,j,k)\in\left[n^*,l^*,n^*\right]}g^{-w_i\boldsymbol{M}^*_{j,k}a^{q+1+k-i}/b_j}\\&=v^{-t}\prod_{i\in\left[n^*\right]}\left(g^{a^{q+1-i}}\right)^{-v'w_i}\\&\quad\cdot\prod_{(i,j,k)\in\left[n^*,l^*,n^*\right],i\ne k}\left(g^{a^{q+1+k-i}/b_j}\right)^{-w_i\boldsymbol{M}^*_{j,k}}\\&\quad\cdot\prod_{(i,j)\in\left[n^*,l^*\right]}\left(g^{a^{q+1}/b_j}\right)^{-w_i\boldsymbol{M}^*_{j,i}}\end{aligned}$$

令

$$v^{-t}\prod_{i\in\left[n^*\right]}\left(g^{a^{q+1-i}}\right)^{-v'w_i}\prod_{(i,j,k)\in\left[n^*,l^*,n^*\right],i\ne k}\left(g^{a^{q+1+k-i}/b_j}\right)^{-w_i\boldsymbol{M}^*_{j,k}}=\varphi$$

那么可以得出

$$\begin{aligned}v^{-r}&=\varphi\cdot\prod_{(i,j)\in[n,l]}\left(g^{a^{q+1}/b_j}\right)^{-w_i\boldsymbol{M}^*_{j,i}}\\&=\varphi\cdot\prod_{j\in[l],\rho^*(j)\notin S_j^{'}}\left(g^{-<w,\boldsymbol{M}_j^*>}\right)^{a^{q+1}/b_j}\end{aligned}$$

需要注意的是，$\mathcal{B}$ 可以通过使用给出的 q-type 假设实例计算 φ，而其它项则可以通过与 $(u^{s_\sigma}h)^{r_\sigma}$ 项相乘而被取消。因此，对于每个属性 $s_\sigma \in S_j^{'}$，$\mathcal{B}$ 选择随机参数 $r_\sigma^{'} \in \mathbb{Z}_p$ 并隐含设置

$$\begin{aligned} r_\sigma &= r_\sigma^{'} + r\cdot \sum_{i'\in[l],\,\rho^*(i')\notin S_j'} b_{i'} \Big/ \left(s_\sigma - \rho^*(i')\right) \\ &= r_\sigma^{'} + t\cdot \sum_{i'\in[l],\,\rho^*(i')\notin S_j'} b_{i'} \Big/ \left(s_\sigma - \rho^*(i')\right) \\ &\quad + \sum_{(i,i')\in[n,l],\,\rho^*(i')\notin S_j'} w_i a^{q+1-i} b_{i'} \Big/ \left(s_\sigma - \rho^*(i')\right) \end{aligned}$$

接下来，$\mathcal{B}$ 计算密钥组件 $K_{\sigma,3}^{'}$ 的 $(u^{s_\sigma}h)^{r_\sigma}$ 项为 $\left(u^{s_\sigma}h\right)^{r_\sigma}$

$$\begin{aligned} &=\left(u^{s_\sigma}h\right)^{r_\sigma'}\cdot\left(K_{\sigma,2}\Big/ g^{r_\sigma'}\right)^{u's_\sigma+h'} \\ &\cdot \prod_{(i',j,k)\in[n,l,n],\rho^*(i')\notin S_j'} \left(g^{t\left(s_\sigma-\rho^*(j)\right)\boldsymbol{M}_{j,k}^* b_{i'}a^k \Big/ \left(s_\sigma-\rho^*(i')b_j^2\right)}\right) \\ &\cdot \prod_{(i,i',j,k)\in[n,l,l,n],\rho^*(i')\notin S_j'} \left(g^{\left(s_\sigma-\rho^*(j)\right)w_i\boldsymbol{M}_{j,k}^* b_{i'}a^{q+1+k-i} \Big/ \left(s_\sigma-\rho^*(i')b_j^2\right)}\right) \\ &= \varphi\cdot \prod_{j\in[l],\rho^*(j)\notin S_j'} g^{<w,\boldsymbol{M}_j^*>a^{q+1}/b_j} \end{aligned}$$

其中，φ 包含了乘积的剩余项，φ 和 $K_{\sigma,2}^{'}$ 则可以通过使用给出的 q-type 假设实例计算得到。而 $(u^{s_\sigma}h)^{r_\sigma}$ 的第 2 项则可以通过与 v^{-r} 相乘被取消。因此，$\mathcal{B}$ 可以成功地计算密钥组件 $K_{\sigma,2}^{'}$ 和 $K_{\sigma,3}^{'}$。

密钥生成后，$\mathcal{B}$ 随机选择参数 $z\in\mathbb{Z}_p^*$，并且设置外包密钥 TK 为

$$\begin{aligned} &\mathrm{TK} = K_0 = \left(K_0^{'}\right)^{1/z}, K_1 = \left(K_1^{'}\right)^{1/z}, \\ &\left\{K_{\sigma,2} = \left(K_{\sigma,2}^{'}\right)^{1/z}, K_{\sigma,3} = \left(K_{\sigma,3}^{'}\right)^{1/z}\right\}_{\sigma\in S_j'} \end{aligned}$$

因此，$\mathcal{B}$ 最终设置私钥为：$\mathrm{SK}_1 = (z,\mathrm{TK})$，最后将 TK 发送给 $\mathcal{A}$。

(2) $\mathcal{A}$ 向 $\mathcal{B}$ 进行属性 x 的撤销列表 RL_x 与密文 $\mathrm{CT} = \left((\boldsymbol{M},\rho), C, C_0, \left\{C_{i,1}, C_{i,2}, C_{i,3}\right\}_{i\in[l]}\right)$ 的重加密查询 $\mathcal{O}_{\mathrm{ree}}$。算法生成重加密密文如下：

(a)若无属性被撤销，即 $\mathrm{RL}_x = \Phi$，那么 CSP 随机选择参数 $k\in\mathbb{Z}_p$ 并重新加密密文 CT 如下：

$$\begin{aligned} C' = C &= m\cdot e(g,g)^{\alpha s}, C_0^{'} = C_0 = g^s, C_1^{'} = \left(C_0\right)^{1/k} \\ &= g^{s/k}, \forall i = 1,2,\cdots,l: C_{i,1}^{'} = C_{i,1}\cdot v^k \\ &= w^{\lambda_i}v^{t_i}v^k, C_{i,2}^{'} = C_{i,2}\cdot\left(u^{\rho(i)}h\right)^{-k} \\ &= \left(u^{\rho(i)}h\right)^{-t_i}\left(u^{\rho(i)}h\right)^{-k}, C_{i,3}^{\prime} = C_{i,3}\cdot g^k = g^{t_i}g^k \end{aligned}$$

因此，重加密密文被设置为 $\mathrm{RCT} = \left(C', C_0^{'}, C_1^{'}, \left\{C_{i,1}^{'}, C_{i,2}^{'}, C_{i,3}^{'}\right\}_{i=1}^{l}\right)$。另外，重加密算法将更新相应的授权密钥为 $\mathrm{SK}_2^{'} = \left(g^{\alpha_2}\right)^k$。

(b)若用户 ID_j 的属性 x 被撤销，即 $\mathrm{RL}_x \neq \Phi$，那么算法首先随机选择指数 $\mathrm{RL}_x \neq \Phi$，并使用文献[18]中定长密文与定长密钥的广播加密方案为对 v_x 进行加密生成相应的密文头 CH_x。然后算法同样随机选择参数 $k\in\mathbb{Z}_p$ 并重新加密密文 CT 如下：

$$\begin{aligned} &C' = C = M\cdot e(g,g)^{\alpha s}, C_0^{'} = C_0 = g^s, C_1^{'} = g^{s/k}, \\ &\quad \forall i = 1,2,\cdots,l \\ &C_{i,1}^{'} = w^{\lambda_i}v^{t_i}v^k, C_{i,2}^{'} = \left(u^{\rho(i)}h\right)^{-t_i}\left(u^{\rho(i)}h\right)^{-k} \\ &\text{for } \rho(i)\neq x: C_{i,3}^{'} = g^{t_i}g^k \\ &\text{for } \rho(i) = x: C_{i,3}^{'} = \left(g^{t_i}g^k\right)^{1/v_x} \end{aligned}$$

因此，重加密密文被设置为 $\mathrm{RCT} = \left(\mathrm{CH}_x, C', C_0^{'}, C_1^{'}, \left\{C_{i,1}^{'}C_{i,2}^{'}C_{i,3}^{'}\right\}_{i=1}^{l}\right)$。同样，重加密算法将更新相应的授权密钥为 $\mathrm{SK}_2^{'} = \left(g^{\alpha_2}\right)^k$。

挑战阶段：攻击者 $\mathcal{A}$ 向 $\mathcal{B}$ 提交相同长度的密文 m_0 与 m_1。$\mathcal{B}$ 随机选择参数 $\beta\in\{0,1\}$，生成挑战密文为：$C^* = m_\beta\cdot T\cdot e\left(g^s, g^{\alpha'}\right)\cdot e\left(g^s, g^{\alpha''}\right), C_0^* = g^s$，然后 $\mathcal{B}$ 随机选择 $y_2^{'}, y_3^{'}, \cdots, y_{n^*}^{'}\in\mathbb{Z}_p$，并隐含地通过向量 $\boldsymbol{v} = \left(s, sa + y_2^{'}, sa^2 + y_3^{'}, \cdots, sa^{n-1} + y_{n^*}^{'}\right)\in\mathbb{Z}_p^{n^*}$ 来共享密钥 s。因为 $\lambda = \boldsymbol{M}^*\boldsymbol{v}$，因此可以得出：

$$\lambda_\tau = \sum_{i\in[n]}\boldsymbol{M}_{\tau,i}^* sa^{i-1} + \sum_{i=2}^{n}\boldsymbol{M}_{\tau,i}^* y_i^{'}$$

令 $\lambda_\tau^{'} = \sum_{i=2}^{n}\boldsymbol{M}_{\tau,i}^* y_\tau^{'}$，并且 $\mathcal{B}$ 已知 $\lambda_\tau^{'}$。对于矩阵的每一行，$\mathcal{B}$ 隐含地设置 $t_\tau = -sb_\tau$。接下来，$\mathcal{B}$ 继续计算：

$$\begin{aligned} C_{\tau,1} &= w^{\lambda_\tau}v^{t_\tau} = w^{\lambda_\tau^{'}}\cdot\prod_{i\in[n]} g^{\boldsymbol{M}_{\tau,i}^* sa^{i-1}}\cdot\left(g^{sb_\tau}\right)^{-v'} \\ &\quad\cdot\prod_{(j,k)\in[l,n]} g^{-\boldsymbol{M}_{j,k}^* a^k sb_\tau/b_j} = w^{\lambda_\tau^{'}}\cdot\left(g^{sb_\tau}\right)^{-v'} \\ &\quad\cdot\prod_{(j,k)\in[l,n],j\neq\tau}\left(g^{a^k sb_\tau/b_j}\right)^{-\boldsymbol{M}_{j,k}^*} \\ C_{\tau,2} &= \left(u^{\rho^*(\tau)}h\right)^{-t_\tau} = \left(g^{sb_\tau}\right)^{-\left(u'\rho^*(\tau)+h'\right)} \\ &\quad\cdot\left(\prod_{(j,k)\in[l,n]} g^{\left(\rho^*(\tau)-\rho^*(j)\right)\boldsymbol{M}_{j,k}^* a^k/b_j^2}\right)^{-sb_\tau} \\ &= \left(g^{sb_\tau}\right)^{-\left(u'\rho^*(\tau)+h'\right)} \\ &\quad\cdot\prod_{(j,k)\in[l,n],j\neq\tau}\left(g^{sb_\tau a^k/b_j^2}\right)^{-\left(\rho^*(\tau)-\rho^*(j)\right)\boldsymbol{M}_{j,k}^*} \\ C_{\tau,3} &= g^{t_\tau} = \left(g^{sb_\tau}\right)^{-1} \end{aligned}$$

最后，$\mathcal{B}$ 将挑战密文 $\mathrm{CT}^* = \left(\left(\boldsymbol{M}^*,\rho\right), C, C_0, \left\{C_{\tau,1}, C_{\tau,2}, C_{\tau,3}\right\}_{\tau\in[l^*]}\right)$ 发送给攻击者 $\mathcal{A}$。

查询阶段 2：如查询阶段 1，$\mathcal{A}$ 向 $\mathcal{B}$ 进行密钥生

成查询 $\mathcal{O}_{\text{kg}}$ 与密文重加密查询 $\mathcal{O}_{\text{ree}}$。

猜测阶段：攻击者 $\mathcal{A}$ 最终输出对 β 的猜测 β'。若 $\beta=\beta'$，$\mathcal{A}$ 输出 0 表示猜测 $T=e(g,g)^{\alpha^{q+1}s}$。否则输出 1 表示猜测 T 为群 $\mathbb{G}_T$ 中的随机元素。　　证毕

4　方案分析与实验验证

现将本文提出的方案与已有几种撤销方案进行对比，包括功能性、存储成本与计算效率。其中所使用的描述符如下：$|C_1|$ 表示 $\mathbb{G}$ 中数据元素的长度；$|C_T|$ 表示 $\mathbb{G}_T$ 中数据元素的长度；$|C_p|$ 表示 $\mathbb{Z}_p$ 中数据元素的长度；$C_{\mathcal{T}}$ 表示密文中访问结构的元素个数；$|C_k|$ 表示 Hur 方案[16]中使用的密钥 KEK 的长度；t 表示与密文有关的属性个数；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；n_m 表示撤销用户的个数。

在进行实际比较前，本文首先对用到的文献[18]提出的定长密钥与定长密文广播加密方案进行具体分析，其公钥长度为 $(2n_u+1)|C_1|$，主密钥长度为 $|C_p|$，用户密钥长度为 $|C_1|$，密文长度为 $2|C_1|$。

4.1 功能对比

表 1 可以看出，Liang 方案[11]实现了系统级的用户撤销，一旦用户的某个属性被撤销，那么该用户就失去了系统内所有其他合法属性的访问权限，这不符合实际应用环境。而本文提出的方案与 Hur 方案[16]、Yang 方案[15]实现了属性级的用户撤销，如果用户的某个属性被撤销，不影响其他合法属性的正常访问。另外，本文方案和 Hur 方案支持大规模属性集合，灵活性更强，但是 Hur 方案仅为通用群模型下可证明安全的，而一般群模型下的安全性被认为是启发式的安全，并不是可证明安全，很多一般群模型下可证明安全的方案在实际应用中被发现并不安全。Yang 方案实现了随机预言模型下的可证明安全性，虽然随机预言模型下的安全性是可证明安全，但是模型进行了理想化的假设，安全性较低，而且该方案只支持小规模属性集合。本文方案不仅支持大规模属性集合，而且实现了标准模型下的可证明安全性，安全性较高。

4.2 存储成本

表 2 将本文方案与其他相关方案进行了存储成本的对比。属性中心 AA 的存储成本主要来自于主密钥，本文方案与 Hur 方案使用了较少的主密钥。而 Liang 方案中，主密钥随着用户总数 n_u 成线性增长，Yang 方案则随着属性总数 n_a 成线性增长。数据拥有者 O 的存储成本主要来自于公钥。Hur 方案使用了最短的公钥，Yang 方案公钥随着属性总数 n_a 成线性增长，Liang 方案公钥随着属性总数 n_a 与访问矩阵列向量 $C_{\mathcal{T}}/t$ 成互为斜率的线性增长，而本文方案中，虽然属性加密产生了 $5|C_1|+|C_T|$ 定长的公钥长度，但是用到的广播加密方案则生成了 $(2n_u+1)|C_1|$ 长度的公钥，其随着用户总数 n_u 成斜率

表 1　功能对比

方案	撤销粒度	属性集	模型	假设
Liang 方案[11]	系统级的用户撤销	小规模	标准模型	DBDH 假设
Hur 方案[16]	属性级的用户撤销	大规模	通用群模型	–
Yang 方案[15]	属性级的用户撤销	小规模	随机预言模型	q-parallel BDHE 假设
本文方案	属性级的用户撤销	大规模	标准模型	q-type 假设

表 2　存储成本对比

实体	Liang 方案[11]	Hur 方案[16]	Yang 方案[15]	本文方案
AA	$\|C_1\|+\left(2^{(\log n_u+1)}+1\right)\|C_p\|$	$\|C_p\|+\|C_1\|$	$(4+n_a)\|C_p\|$	$3\|C_p\|$
O	$\left(\frac{C_{\mathcal{T}}}{t}n_a+6\right)\|C_1\|+\|C_T\|+\|C_p\|$	$2\|C_1\|+\|C_T\|$	$(2n_a+4)\|C_1\|+\|C_T\|$	$(2n_u+6)\|C_1\|+\|C_T\|$
CSP	$(C_{\mathcal{T}}+3)\|C_1\|+\|C_T\|$	$(2t+1)\|C_1\|+\|C_T\|+\frac{t\cdot n_u}{2}\|C_p\|$	$(3t+1)\|C_1\|+\|C_T\|$	$(3t+5)\|C_1\|+\|C_T\|$
U	$\left(k+3+\frac{C_{\mathcal{T}}}{t}\right)(\log n_u+1)\|C_1\|+2(n_u-n_m)\cdot\log\frac{n_u}{n_u-n_m}\|C_1\|$	$(2k+1)\|C_1\|+(\log n_u+1)C_k$	$(k+2)\|C_1\|$	$(2k+3)\|C_1\|+\|C_p\|$

为常数的线性增长。云存储提供商 CSP 的存储成本主要来自于密文与密文头。因为 Liang 只实现了用户撤销，在该方案中，利用子集覆盖进行密钥更新，不需要对密文进行更新，但其密文长度随着访问结构 C_T 成线性增长。Yang 方案通过属性中心与用户交互为用户进行密钥更新，并对被撤销属性对应的密文进行更新，因此，密文长度只与密文相关属性个数 t 成线性增长。Hur 方案中，数据拥有者将密文发送给 CSP 后，CSP 为每个属性组生成相应的密文头，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 t 成斜率为常数的线性增长，而密文头长度与密文相关属性个数 t 及用户总数 n_u 成互为斜率的线性增长。本文方案中，若发生属性改变，CSP 为该属性对应的密文重新选择指数进行密文更新，并将指数进行加密，生成相应的密文头。因此其存储也包括密文及密文头，且密文的长度为 $(3t+3)|C_1|+|C_T|$，随着密文相关属性个数 t 成斜率为常数的线性增长，而密文头为 $2|C_1|$ 的广播加密密文。数据访问者的存储成本主要来自于其拥有的密钥。本文方案与 Yang 方案中，密钥长度较短，只与用户拥有的属性个数 k 成线性增长，Liang 方案利用二叉结构来生成用户密钥，其密钥长度与密钥属性个数 k、访问矩阵列向量 C_T/t 与用户个数 n_u 都相关。而属性撤销时，使用子集覆盖进行密钥更新，其更新密钥长度与最小覆盖集成正增长。而 Hur 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 k 成线性增长，而且与整个系统中用户的总个数 n_u 成对数增长。

4.3 计算效率

实验环境为 64 bit Ubuntu 14.04 操作系统、Intel® Core™ i7-3770CPU (3.4 GHz)、内存 4 G，实验代码基于 Pairing-based Cryptography Library (PBC-0.5.14)[19]与 cpabe-0.11[20]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 20 次所得的平均值。在实验中，PBC 库计算对运算的时间大约为 5.3 ms，$\mathbb{G}_1$ 与 $\mathbb{G}_T$ 的运算时间大约为 6.2 ms 与 0.6 ms。另外，通过使用 Ubuntu 14.04 操作系统中的/dev/urandom 来选择 $\mathbb{G}_1$ 与 $\mathbb{G}_T$ 中随机元素的时间大约为 14 ms 与 1.4 ms。

本文对几种方案在参数生成时间、密钥生成时间、加密时间、解密时间与重加密时间方面进行了比较，结果如图 1–图 5 所示，其中取 $C_T/t=6$，$n_u=8$。

如图 1 所示，由于本文所提方案和 Hur 方案支持大规模属性集合，参数生成时间与属性集合的大小无关，为常数。而 Liang 方案和 Yang 方案仅支持小规模属性集合，因此，其参数生成时间随着属性集合的大小成线性增长。如图 2 所示，密钥生成时间与用户属性个数成线性增长，本文方案密钥生成时间略高于 Yang 方案，但优于 Hur 方案与 Liang

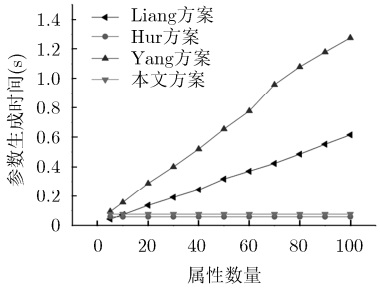


图 1 参数生成时间

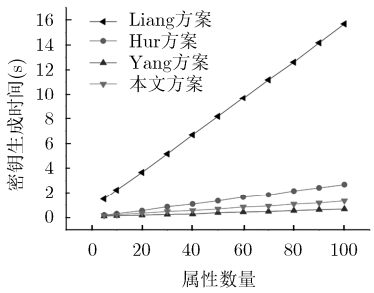


图 2 密钥生成时间

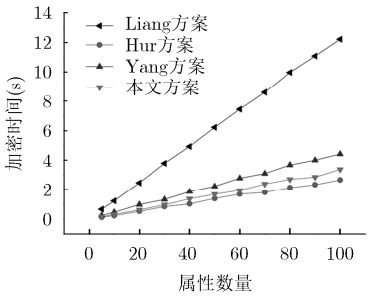


图 3 加密时间

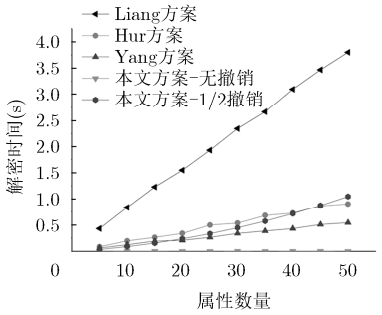
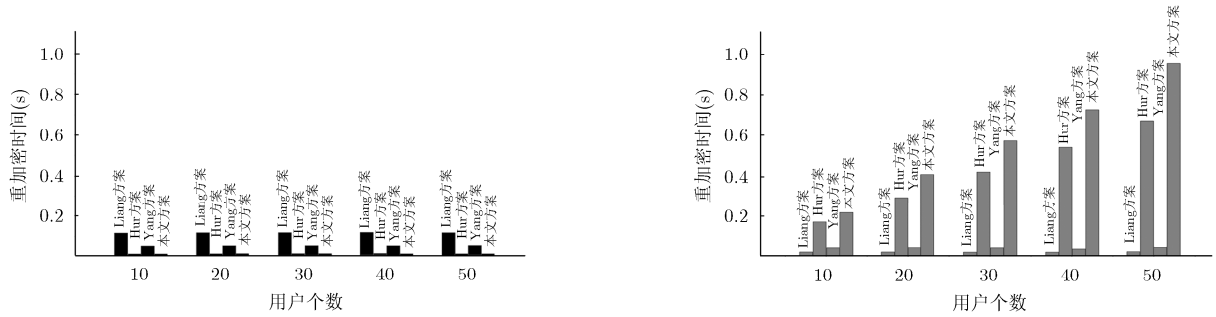


图 4 解密时间



(a)属性授权重 加密　　(b)云存储中心重 加密

图 5 重新加密时间对比

方案。特别是 Liang 方案其密钥生成时间不仅与用户属性个数有关，而且与访问矩阵列向量 C_T/t 与用户个数 n_u 相关，因此其密钥生成时间远远大于其它 3 种方案。如图 3 所示，加密时间与访问结构中的属性成线性增长。本文方案加密时间略高于 Hur 方案，但优于 Yang 方案与 Liang 方案。需要注意的是，Hur 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，但是运行时间很短。而 Liang 方案其加密生成时间不仅与访问结构中的属性个数有关，而且同样与访问矩阵列向量 C_T/t 相关，因此加密时间远远大于其它 3 种方案。进行解密实验时，使用所用的属性进行解密，并且对 Hur 方案进行解密时，使用最简单的二叉结构，所有中间节点均为 (n,n) 门限。而本文提出的方案分别在无属性撤销与 1/2 的属性被撤销两种情况下进行了解密实验。如图 4 所示，Liang 方案、Hur 方案、Yang 方案与本文 1/2 属性被撤销方案的解密时间都随着解密属性数量而增长，而本文无属性撤销方案由于采用了外包解密，其用户只需要进行 1 个 $\mathbb{G}_T$ 的指数操作。另外，由于本文属性撤销方案的解密时间为解密属性的二次函数，但是本文采用了外包解密，大大降低了用户的解密时间，由图 4 可以看出，当属性在某个范围内时，本文撤销方案的解密时间小于其它方案，随着属性个数的增加，其解密时间逐次超过 Yang 方案与 Hur 方案，但在可接受的范围。另外，图 5 表示了重加密时间对比。当属性被撤销时，需要对密钥或密文进行更新。Yang 方案与 Liang 方案主要对密钥进行更新，而 Hur 方案与本文方案主要对密文进行更新，因此图中可以发现 Hur 方案与本文方案需要的计算时间较长，且随着属性数量逐渐增长，但是所需计算完全是由云存储中心实施的，实际应用中云存储中心拥有大量的计算资源。而 Yang 方案与 Liang 方案虽然需要的计算时间较少，但是需要属性授权实施密钥更新，而属性授权的计算资源是有限的，容易成为系统的软肋。

5 结束语

本文提出了一个大规模属性集下的密文策略属性加密方案，该方案能够实现属性级的用户撤销，即若用户的某个属性被撤销，不会影响该用户其他合法属性的正常访问。为了实现撤销，我们将密钥分为两部分：为用户生成的私钥以及为云存储中心生成的授权密钥。在本文方案中，若用户的属性被撤销，那么该属性对应的密文将进行更新，只有该属性没有被撤销的用户才能够成功地进行密钥更新而解密密文。最后对方案进行了性能分析与实验验证，实验结果表明，与已有相关方案相比，虽然为了实现属性撤销，增加了存储中心的计算负载，但是不需要属性中心的参与，因此降低了属性中心的计算负载，而且用户除了密钥外不需要其它额外参数来实现属性撤销，因此大大节省了存储空间。

参 考 文 献

[1] SAHAI A and WATERS B. Fuzzy Identity-Based Encryption [M]. Heidelberg, Berlin, Springer, 2005: 457–473. doi: 10.1007/11426639_27.

[2] YADAV U C. Ciphertext-policy attribute-based encryption with hiding access structure[C]. 2015 IEEE International Advance Computing Conference (IACC), Bangalore, India, 2015: 6–10. doi: 10.1109/IADCC.2015.7154664.

[3] WANG M, ZHANG Z, and CHEN C. Security analysis of a privacy-preserving decentralized ciphertext-policy attribute-based encryption scheme[J]. *Concurrency & Computation Practice & Experience*, 2016, 28(4): 1237–1245. doi: 10.1002/cpe.3623.

[4] NARUSE T, MOHRI M, and SHIRAISHI Y. Provably secure attribute-based encryption with attribute revocation and grant function using proxy re-encryption and attribute key for updating[J]. *Human-centric Computing and Information Sciences*, 2015, 5(1): 1–13. doi: 10.1186/s13673-015-0027-0.

[5] LEWKO A, OKAMOTO T, SAHAI A, *et al*. Fully Secure

Functional Encryption: Attribute-based Encryption and (Hierarchical) inner Product Encryption[M]. Heidelberg, Berlin, Springer, 2010: 62–91. doi: 10.1007/978-3-642-13190-5_4.

[6] RAHULAMATHAVAN Y, VELURU S, HAN J, *et al*. User collusion avoidance scheme for privacy-preserving decentralized key-policy attribute-based encryption[J]. *IEEE Transactions on Computers*, 2016, 65(9): 2939–2946. doi: 10.1109/TC.2015.2510646.

[7] LEWKO A and WATERS B. Unbounded HIBE and attribute-based encryption[C]. International Conference on Theory and Applications of Cryptographic Techniques: Advances in Cryptology, Tallinn, Estonia, 2011: 547–567.

[8] ROUSELAKIS Y and WATERS B. Practical constructions and new proof methods for large universe attribute-based encryption[C]. ACM Sigsac Conference on Computer & Communications Security, Berlin, Germany, 2013: 463–474.

[9] OSTROVSKY R, SAHAI A, and WATERS B. Attribute-based encryption with non-monotonic access structures[C]. CCS 07 ACM Conference on Computer & Communications Security, Alexandria, Virginia, USA, 2007: 195–203.

[10] STADDON J, GOLLE P, *et al*. A content-driven access control system[C]. Proceedings of the 7th Symposium on Identity and Trust on the Internet, Gaithersburg, Maryland, USA, 2008: 26–35.

[11] LIANG X, LU R, and LIN X. Ciphertext policy attribute based encryption with efficient revocation[OL]. https://www.ResearchGate.net/publication/255670422, 2010.

[12] BETHENCOURT J, SAHAI A, and WATERS B. Ciphertext-policy attribute-based encryption[C]. IEEE Symposium on Security and Privacy, Oakland, California, USA, 2007: 321–334.

[13] BOLDYREVA A, GOYAL V, and KUMAR V. Identity-based encryption with efficient revocation[C]. ACM Conference on Computer and Communications Security, Alexandria, Virginia, USA, 2008: 417–426.

[14] PIRRETTI M, TRAYNOR P, MCDANIEL P, *et al*. Secure attribute-based systems[C]. ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 2006: 799–837.

[15] YANG K, JIA X, and REN K. Attribute-based fine-grained access control with efficient revocation in cloud storage systems[C]. ACM Sigsac Symposium on Information, Computer and Communications Security, Denver, Colorado, 2015: 523–528.

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

[17] BONEH D and BOYEN X. Efficient selective-ID Secure identity-based encryption without random oracles[C]. Advancesin Cryptology-EUROCRYPT 2004, Lecture Notes in Computer Science, Berlin, Heidelberg, 2004, 3027: 223–238.

[18] DAN B, GENTRY C, and WATERS B. Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys[M]. Heidelberg, Berlin, Springer, 2005: 258–275.

[19] LYNN B. The Pairing-Based Cryptography (PBC) library [OL]. http://crypto.stanford.edu/pbc,2006.

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

王建华： 男，1962 年生，教授，博士生导师，研究方向为信息安全.

王光波： 男，1987 年生，博士生，研究方向为属性加密、网络信息安全.

徐开勇： 男，1962 年生，研究员，硕士生导师，研究方向为信息安全.