文章编号:1007-5321(2018)03-0113-06 **DOI**:10.13190/j.jbupt.2017-220

云计算环境下支持高效撤销的新型属性基加密方案

陈红松[1,2], 沈强磊[1]

(1. 北京科技大学 计算机与通信工程学院, 北京 100083; 2. 材料领域知识工程北京市重点实验室, 北京 100083)

摘要: 为了解决开放云计算环境下用户属性变化导致的用户权限撤销及变更问题,提出一种基于代理重加密和密钥分割技术的属性基代理重加密方案,该方案支持用户权限的即时撤销,当发生用户撤销时,只需要更新云存储服务器中的密文组件以及代理服务器中未撤销用户的属性无关私钥组件. 当发生用户属性撤销时,只需更新用户属性撤销列表,解密时根据用户属性撤销列表控制撤销属性用户的访问,可减少密文更新和用户私钥更新的计算量,提高系统撤销用户权限的执行效率,保护用户属性的隐私信息.

关 键 词: 属性基加密; 用户-属性撤销; 访问控制; 代理重加密

中图分类号: TP309 **文献标志码**: A

Attribute-Based Encryption Scheme With High Efficient Revocation in Cloud Computing Environment

CHEN Hong-song[1,2], SHEN Qiang-lei[1]

(1. School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China; 2. Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing, 100083, China)

Abstract: In order to solve the problem of user authorization revocation caused by user attributes change in open cloud computing environment, an attribute-based proxy re-encryption scheme based on proxy re-encryption and key segmentation is proposed. Instant revocation of user authorization is supported in the scheme. When user revocation occurs, it is only necessary to update the ciphertext component in the cloud storage server and unrevoked users' attribute independence private key component in the proxy server. When the user attribute revocation occurs, it is necessary to update the user attribute revocation list from which the user authorization can be limited. Then the workload of updating ciphertext and the user private key is lessen to improve the execution efficiency of re-encryption and access control scheme, at the same time, the privacy information of user attributes are protected.

Key words: attribute-based encryption; user-attribute revocation; access control; proxy re-encryption

2006 年首次提出云计算概念,经过多年发展,云计算已成为信息技术领域的热点;同时,其安全问题成为了云计算发展的阻碍. 2005 年,基于身份加密技术,Sahai 等[1]首先提出了基于属性的加密机制,用户可根据自身属性获得相应私钥,进而获得对应的解密能力及权限,从而实现对云存储密文数据的访问控制. 但是属性基加密算法需要解决由用户撤销或者用户某个属性被撤销时,及时变更用户权限的问题. 在传统的属性撤销机制中,当一个用户的某个属性被撤销时,需要对整个系统含有这个属性的未撤销用户私钥进行更新,计算量和通信量太大. 还有一些方案,发生用户属性撤销时,只对密文

收稿日期: 2017-11-06

基金项目: 中央高校基本科研业务费专项资金项目(FRF-GF-17-B27)

作者简介: 陈红松(1977—), 男, 副教授; 沈强磊(1992—), 男, 硕士生, E-mail:chenhs@ustb.edu.cn.

中与撤销属性相关的密文组件进行更新. 然而,被撤销用户可使用上次解密得到的部分中间结果来组合并解密新密文,无法保证重加密的安全性.

1 相关工作

Yu 等[2]运用代理重加密技术与版本号标记法来实现密文策略属性基加密(CP-ABE, ciphertext policy attribute based encryption)方案中的属性即时撤销. Chen 等[3]提出了分散多机构属性基加密方案,该方案采用多个授权中心负责用户密钥的分发,无需中央授权中心的协调. Naruse 等[4]利用代理重加密技术让云服务提供商来负责属性撤销工作,减轻了授权机构的工作. Zhang 等[5]提出了支持外包解密的方案,能够减少用户的解密开销. Takeru 等[6]提出的方案利用可信机构代理重加密技术对密文和用户私钥的更新,但是可信机构重加密的计算量较大. Chen 等[7]提出了基于多标签的大数据安全访问控制模型,数据所有者可以根据实际安全需求添加安全标签并实施安全访问控制. Sahai 等[8]在 2012 年的 CRYPTO 美密会上提出了采用重加密方法来收回撤销用户对加密数据的解密权限.

由以上相关工作可知,尽管目前的属性基加密机制可以支持用户和属性撤销,但仍存在以下问题.

1) 计算和通信量巨大

如果发生用户撤销属性,可信第三方机构更新密文和非撤销用户的用户私钥. 但一个系统中有大量的用户,只要一个用户的某个属性发生撤销,系统中所有的含有这个属性的用户私钥都要更新,计算量和通信量太大,效率不高.

2) 存在用户隐私泄露的风险

通常把用户私钥的更新工作交由第三方机构来做,用户私钥中与用户属性相关的私钥组件也能被第三方机构查看,用户属性信息存在隐私泄露风险.

3) 重加密密文解密中间值利用漏洞

在已有的重加密方案中,如发生用户撤销属性事件,在重加密密文时只对密文中和撤销属性相关的密文组件进行更新,被撤销用户可能利用被撤销属性之前解密得到的中间值对密文进行解密攻击.

2 支持用户-属性撤销的新型属性基加密方案

2.1 方案设计思想

本方案的主要设计思想:可信中心为每个合法的用户根据其属性生成私钥,然后用密钥分割的思想把用户的私钥分成 2 部分,一部分和用户属性相关的私钥组件发送给用户;另一部分的私钥组件发送给代理服务器. 如果发生用户撤销事件,可信中心把撤销用户的 ID 从用户列表中删除,并生成代理重密钥发送给云存储服务器和代理服务器. 云存储服务器对密文进行代理重加密. 代理服务器查看可信中心里的用户列表,如果用户在用户列表中则更新用户的私钥组件. 如果发生用户属性撤销,可信中心把撤销用户的 ID 加入到属性撤销列表中. 如果撤销了用户的这个属性,解密时不计算该属性.

2.2 算法描述

2.2.1 Setup(k):系统初始化

可信机构选择生成元为 g,阶为素数 p 的双线性映射 G_0,Z_p为模 p 的整数群,随机选取 $\alpha,\beta \in Z_p$,可信机构生成系统公钥 $P_K = (G_0, g, h = g^{\beta}, f = g^{\frac{1}{\beta}}, e(g,g)^{\alpha})$,主密钥 $W_K = (\beta, g^{\alpha})$. 可信机构获取全部的系统属性集 U,为每个属性 $a_{tt} \in U$ 初始化一个空的 ArrayList,ArrayList 用来存放撤销属性用户的 ID. 初始化的属性撤销列表格式是 map〈a_{tt}, ArrayList〈Integer〉〉,并存入到属性撤销列表文件中.

2.2.2 密钥生成与分割

1) KeyGen(W_K, S):生成用户私钥

可信机构获得每个用户的属性集合 S,输出被 S 所标记的用户密钥. 首先,算法选择随机数 $r \in Z_p$;然后,对每一个属性 j 选择随机数 $r_j \in Z_p$;最后,计算出用户私钥 $S_K = (D_1 = g^{\frac{(\alpha - r)}{\beta}}, D_2 = g^r, \forall j \in S: D_j = g^r H(j)^{r_j}, D_{j'} = g^{r_j})$,其中 $H(j)$为每个属性对应的散列值. 把每个新用户加入到用户列表中,用户列表的形式是映射 map〈username, userid〉,username 和 userid 分别存放的是用户名和用户 ID. 新建用户时要保证用户的用户名和用户 ID 都唯一,并把映射 map 存入到用户列表文件中.

2) Split(S_K):分割用户私钥

可信机构将每个私钥拆成 2 部分,跟用户属性无关的私钥组件 $D_b = (D_1, D_2)$发送给代理服务器,跟用户属性相关的私钥组件 $D_a = (\forall j \in S: D_j, D_{j'})$发送给用户.

2.2.3 Encrypt(P_K, M, τ):加密算法

数据属主加密数据,加密算法用访问结构树 τ 加密消息 M. 算法从根节点 R 开始选择随机数 $s_1 \in Z_p$,并设置 $q_R(0) = s_1$. 然后,通过算法随机选择 d_R

个点来完全定义多项式 q_R. 对于其他节点 x，令 $q_x(0)=q_{\text{parent}(x)}(\text{index}(x))$，随机选择其他 d_x 个节点来完全定义 q_x. 用户选择随机数 $s_2\in Z_p$，且 $s=s_1+s_2$. 用随机数 s 计算加密密文组件 $\widetilde{C}=Me(g,g)^{\alpha s}$，计算 $C_1=h^s$，$C_2=g^{s_2}$. 设 τ 中所有叶子节点的集合为 Y，那么在给定的树形访问结构 τ 下计算出来的密文为

$$C_T=(\tau,\widetilde{C}=Me(g,g)^{\alpha s},C_1=h^s,C_2=g^{s_2},\forall y\in Y:C_y=g^{q_y(0)},C_{y'}=H(a_{tt}(y))^{q_y(0)}) \tag{1}$$

其中：y 为访问结构树的每个叶子节点，也就是访问树的属性；$q_y(0)$ 为多项式的常数项；$a_{tt}(y)$ 为属性值；$H(a_{tt}(y))$ 为属性值对应的散列值. 数据属主把加密好的密文上传到云存储服务器.

2.2.4　用户私钥合成与解密

1）Combine(D_b，D_a)：合成用户私钥

用户向代理服务器请求用户私钥组件，代理服务器给未撤销用户分发其用户私钥组件 D_b. 用户获得私钥组件后，跟自己所拥有的私钥组件 D_a 一起合成其完整私钥：

$$S_K=(D_1=g^{\frac{(\alpha-r)}{\beta}},D_2=g^r,\forall j\in S:D_j=g^rH(j)^{r_j},D_{j'}=g^{r_j}) \tag{2}$$

然后用私钥开始解密运算.

2）Decrypt(P_K，C_T，S_K)：解密算法

定义一个递归算法 DecryptNode(C_T，S_K，x)，其中 C_T 是密文，S_K 是与用户属性集合 S 关联的私钥，x 是访问结构树 τ 中的节点. 如果用户拥有的属性满足访问策略，则可以得到

$$A=\text{DecryptNode}(C_T,S_K,R)=e(g,g)^{rq_R(0)}=e(g,g)^{rs_1} \tag{3}$$

然后解密计算：

$$\frac{\widetilde{C}}{e(C_1,D_1)e(C_2,D_2)A}=\frac{Me(g,g)^{\alpha s}}{e(g,g)^{(\alpha-r)s}e(g,g)^{rs_2}e(g,g)^{rs_1}}=\frac{Me(g,g)^{\alpha s}}{e(g,g)^{(\alpha-r)s}e(g,g)^{rs}}=M \tag{4}$$

解密成功，得到明文 M.

2.2.5　用户撤销

如果发生用户撤销事件，需要撤销用户的全部解密能力，撤销的用户的私钥不能解密任何加密文件. 这里需要对密文和未撤销用户的私钥进行代理重加密，涉及另外 3 个算法：ReKeyGen、ReKey 和 ReEnc. 首先可信机构生成代理重密钥，服务器拿到代理重密钥之后分别对密文和用户私钥组件进行更新. 具体的更新过程如下：

1）ReKeyGen：生成代理重密钥

可信机构选取随机数 $t\in Z_p$，作为代理重密钥发送给代理服务器和云存储服务器. 可信机构把撤销用户的用户名和 ID 从用户列表中删除.

2）ReKey：更新未撤销用户私钥

代理服务器获取到可信机构发来的代理重密钥，查看用户列表中用户的 ID 是否被删除，如果没有被删除，则对用户的私钥组件 D_b 进行更新. 代理服务器更新之后的私钥组件为

$$D_b'=(D_1'=g^{\frac{(\alpha-r)}{\beta t}},D_2=g^r) \tag{5}$$

3）ReEnc：重加密密文

云存储服务器产生一个随机数 $\varepsilon\in Z_p$，使得 $s_2'=s_2+\varepsilon$，$s'=s_1+s_2'=s+\varepsilon$（其中 $s=s_1+s_2$），并用可信机构发来的代理重密钥 t 更新密文组件，根据式(6)执行重加密操作.

$$\widetilde{C}'=\widetilde{C}e(g,g)^{\alpha\varepsilon}=e(g,g)^{\alpha s}e(g,g)^{\alpha\varepsilon}=e(g,g)^{\alpha(s+\varepsilon)}$$
$$C_1'=C_1h^{t\varepsilon}=h^{st}h^{\varepsilon t}=h^{(s+\varepsilon)t}$$
$$C_2'=C_2g^{\varepsilon}=g^sg^{\varepsilon}=g^{(s+\varepsilon)} \tag{6}$$

更新后的密文为

$$C_T'=(\tau,\widetilde{C}'=Me(g,g)^{\alpha(s+\varepsilon)},C_1'=h^{(s+\varepsilon)t},C_2'=g^{(s_2+\varepsilon)},\forall y\in Y:C_y=g^{q_y(0)},C_{y'}=H(a_{tt}(y))^{q_y(0)}) \tag{7}$$

4）Decrypt(P_K，C_T，S_K)：解密密文

代理服务器将用户私钥组件 D_b' 分发给未撤销用户，由用户将私钥组件 D_b' 跟自己所拥有的私钥组件 D_a 一起合成其完整私钥.

$$S_K'=(D_1'=g^{\frac{(\alpha-r)}{\beta t}},D_2=g^r,\forall j\in S:D_j=g^rH(j)^{r_j},D_{j'}=g^{r_j}) \tag{8}$$

然后用新的私钥开始解密.

用户的属性满足访问结构树，即可以分别计算出：

$$e(C_1',D_1')=e(g^{\beta(s+\varepsilon)t},g^{\frac{(\alpha-r)}{\beta t}})=e(g,g)^{(\alpha-r)(s+\varepsilon)}$$
$$e(C_2',D_2)=e(g^{(s_2+\varepsilon)},g^r)=e(g,g)^{r(s_2+\varepsilon)}$$
$$A=F_R=\text{DecryptNode}(C_T',S_K',R)=e(g,g)^{rq_R(0)}=e(g,g)^{rs_1} \tag{9}$$

因此，解密计算明文：

$$\frac{\widetilde{C}'}{e(C_1',D_1')e(C_2',D_2)A}=\frac{Me(g,g)^{\alpha(s+\varepsilon)}}{e(g,g)^{(\alpha-r)(s+\varepsilon)}e(g,g)^{r(s_2+\varepsilon)}e(g,g)^{rs_1}}=$$

$$\frac{Me(g,g)^{\alpha(s+\varepsilon)}}{e(g,g)^{\alpha(s+\varepsilon)}}=M \tag{10}$$

解出明文 M,解密成功.

2.2.6 用户属性撤销

如果发生用户属性撤销,需要撤销用户属性对应的权限,用户不再具备那个属性的解密能力. 用户发生属性撤销时,把用户的 ID 加入到对应的属性撤销列表 ArrayList 中. 如果 ArrayList 没有用户的 userid,表明该用户属性 i 没有被撤销,那么就输出 $\mathrm{DecryptNode}(C_T,S_K,x)=e(g,g)^{rq_x(0)}$,否则输出 $\mathrm{DecryptNode}(C_T,S_K,x)=\perp$,从而实现用户单个属性撤销的功能. 如果用户现有未撤销的属性满足访问结构树,则能够解密文件.

3 安全性证明

本方案基于判定双线性问题(DBDH, decision bilinear Diffie-Hellman,):给定(g,g^a,g^b,g^c,Z),其中 $a,b,c,\theta\in Z_p$,$Z=e(g,g)^{\theta}$,计算 $e(g,g)^{abc}$,判断等式 $Z=e(g,g)^{abc}$是否成立. 一个概率性多项式时间算法 B 能够以优势 ε 求解 DBDH 问题,当且仅当满足:

$$\Pr[B(g,g^a,g^b,g^c,e(g,g)^{abc})=1]-\Pr[B(g,g^a,g^b,g^c,e(g,g)^{\theta})=1]\geqslant\varepsilon \tag{11}$$

定理 1 假设 DBDH 难题成立,如果没有敌手在多项式时间内选择访问结构树 τ 攻破支持撤销的 CP-ABE 方案,则此方案就是在选择属性和明文攻击下的不可区分性(IND-sAtt-CPA,indistinguishability under selective-attribute and chosen-plaintext attack)下的安全模型.

证明 如果存在一个攻击者 A 可以攻破本方案,则存在一个算法 B 可以攻破 DBDH 问题,即输入$(g,g^a,g^b,g^c,Z=e(g,g)^{\theta})$,算法 B 决定等式$Z=e(g,g)^{abc}$是否成立.

攻击者 A 和挑战者 B 按照如下操作,执行 IND-sAtt-CPA 游戏.

步骤 1 初始化

攻击者 A 选择访问结构树 τ、用户撤销列表 R 发送给挑战者 B.

步骤 2 系统设置

挑战者 B 执行方案中的 Setup(k)算法:

① 选择随机数 $x'\in Z_p$,设置 $e(g,g)^{\alpha}=e(g,g)^{ab}e(g,g)^{x'}$,使得 $\alpha=ab+x'$. 此外,设定$g^{\beta}=g^b$.

② 计算得到 $P_K=(G_0,g,h=g^b,f=g^{\frac{1}{b}},e(g,g)^{ab+x'},W_K=(b,g^{ab+x'})$.

③ 挑战者 B 发送给攻击者 A 公钥 P_K,挑战者 B 自己保存主密钥 W_K.

步骤 3 属性私钥查询阶段 1

攻击者 A 选择属性集 $w=\{a_j\mid a_j\notin\tau\}$ 和 $\mathrm{ID_A}\notin R$,并向挑战者 B 发出属性私钥查询请求.

① 挑战者 B 随机选择 $r'\in Z_p$,计算 $D_1=g^{\frac{x'-r'}{b}}$,因为 $x'=\alpha-ab$,所以 $D_1=g^{\frac{x'-r'}{b}}=g^{\frac{\alpha-ab-r'}{b}}=g^{\frac{\alpha-(ab+r')}{b}}$,可以得出 $r=ab+r'$,$D_2=g^{ab+r'}$.

② 对每个属性 $j\in w$,选择随机数 $r_j\in Z_p$,因为 $r=ab+r'$,计算得 $D_j=g^{ab+r'}H(j)^{r_j}$,$D_{j'}=g^{r_j}$. 生成的私钥为

$$S_K=(D_1=g^{\frac{x'-r'}{b}},D_2=g^{ab+r'},\forall j\in w:D_j=g^{ab+r'}H(j)^{r_j},D_{j'}=g^{r_j})$$

③ 挑战者 B 把用户私钥 S_K传送给攻击者 A.

步骤 4 挑战阶段

攻击者 A 向挑战者 B 传送 2 个一样长的信息 m_0 和 m_1,挑战者 B 执行公平的掷硬币游戏选取 $b\in\{0,1\}$,对信息加密,操作如下.

① 计算 $C_1=h^c$,

$$\widetilde{C}=m_be(g,g)^{\alpha c}=m_be(g,g)^{(ab+x')c}=m_be(g,g)^{abc}e(g,g)^{x'c}=m_bZe(g^c,g^{x'})$$

② 挑战者 B 选择随机数 $s_2\in Z_p$,计算 $C_2=g^{s_2}$,那么 $s_1=c-s_2$,用 s_1 作为访问结构树 τ 根节点 R 所对应多项式 q_R 的常数项值,随机选择 d_R 个点来完全定义 q_R. 对于其他顶点 x,令 $q_x(0)=q_{\mathrm{parent}(x)}(\mathrm{index}(x))$,随机选择其他 d_x 个顶点来完全定义 q_x. 设 τ 中所有叶子节点的集合为 Y,则在给定的树形访问结构 τ 下计算得到 $\forall y\in Y$:$C_y=g^{q_y(0)}$,$C_{y'}=H(a_{tt}(y))^{q_y(0)}$.

③ 计算得到 $C_T=(\tau,\widetilde{C},C_1,C_2,\forall y\in Y:C_y,C_{y'})$.

④ 挑战者 B 把生成好的密文 C_T传送给攻击者 A.

步骤 5 属性私钥查询阶段 2

攻击者 A 在查询阶段 1 的限制条件下继续向挑战者 B 发出属性私钥询问.

步骤 6 猜测阶段

攻击者 A 输出猜测 $b'\in\{0,1\}$. 如果 $b'=b$,则挑战者 B 输出 1,表示 DBDH 成立,$Z=e(g,g)^{abc}$;

否则输出 0，表明 $Z=e(g,g)^{\theta}$.

因此，如果

$$\Pr[B(g,g^a,g^b,g^c,e(g,g)^{abc})=1]-\Pr[B(g,g^a,g^b,g^c,e(g,g)^{\theta})=1]\geqslant\varepsilon \quad (12)$$

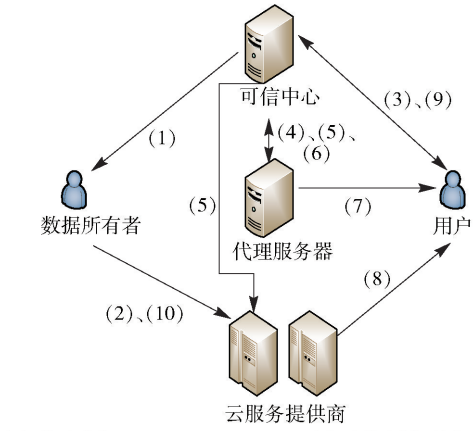
成立，即假定攻击者 A 能以 ε 优势求解 DBDH 难题.

综上所述，假设 DBDH 难题成立，则没有攻击者可以在多项式时间内选择访问结构树 τ 和用户撤销列表 R 攻破支持撤销的 CP-ABE 方案，那么该方案是 IND-sAtt-CPA 安全的.

4　新方案在云环境下访问控制的应用

4.1　系统模型建立与实现

新方案在云环境下访问控制系统如图 1 所示.



(1) 获取公钥P_K　(6) 查询用户列表
(2) 上传密文C_T　(7) 获取最新的私钥组件D_b
(3) 获取用户属性相关私钥组件D_a　(8) 获取最新密文C_T
(4) 获取用户属性无关私钥组件D_b　(9) 查询属性撤销列表
(5) 分发代理重密钥t　(10) 删除文件C_T

图 1　新方案在云环境下的访问控制系统模型

图 1 所示为一个以云环境下的属性基加密访问控制模型，包括数据所有者（数据提供者）、云存储服务器、代理服务器、用户和可信中心. 系统按照图 1 所列流程进行加密与解密等操作. 本实验使用 基于 Java 配对的密码算法库，对基于 Java 语言版本的 Bethencourt-Sahai-Waters CP-ABE（BSW CP-ABE，）进行改进，采用面向对象的方法，通过软件再工程实现了所提方案. 算法的核心类是 RuaCPABE 类，其中各个方法所实现的功能：Setup 方法用于系统参数的初始化，并为系统属性生成属性撤销列表；KeyGen 方法用于生成用户私钥，并把用户加入到用户列表中；Encrypt 方法用于加密文件；SplitSK 方法实现了可信中心对用户私钥分割的功能；RevokeUser 方法实现了可信中心撤销用户的功能；ReKeyGen 方法实现了生成代理重密钥的功能；UpdateSK 方法实现了更新私钥的功能；UpdateCT 方法实现了代理重加密密文的功能；RevokeUserAtt 方法实现了对用户属性撤销的功能；CombineSK 方法实现了用户私钥合并的功能；Decrypt 方法实现了解密密文，并且保证用户解密时判断用户的单个属性是否撤销.

4.2　方案的实验结果分析

本实验采用 Myeclipse 工具使用 Java 语言开发. 所提方案与 BSW CP-ABE 方案用户私钥更新时间随用户属性数量变化关系的对比如图 2 所示，所提方案密钥更新时间更短.

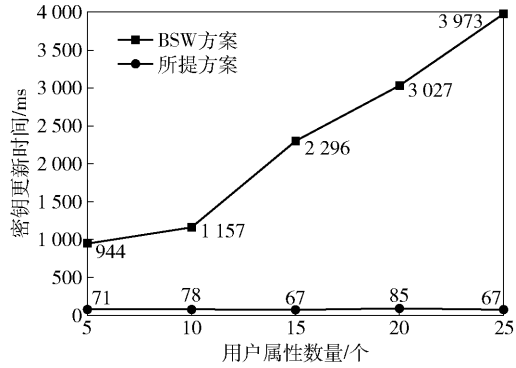


图 2　密钥更新时间与用户属性数量关系

所提方案中，云存储服务器只是对密文组件更新，与访问控制结构树的属性数量无关. 而传统的 BSW CP-ABE 方案中，服务器更新密文的时间与用户属性数量成线性关系. 二者对比如图 3 所示.

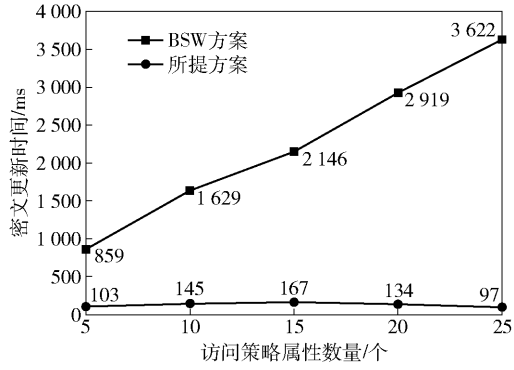


图 3　密文更新时间与访问控制策略属性个数关系

由以上实验可知，所提方案在用户私钥更新时间、密文更新时间等方面相对传统方案更有优势.

5　结束语

针对 CP-ABE 模型中代理重加密计算量巨大、用户属性信息隐私泄露问题，提出了采用密钥分割

和代理重加密技术实现用户撤销以及用户属性的细粒度撤销. 密文和私钥的更新工作由服务器来完成,减少了可信中心和数据属主的计算量. 将用户属性无关的私钥组件存储在代理服务器,从而保护用户属性隐私. 当撤销用户某属性时,可信中心只需更新属性撤销列表,解密前查询属性撤销列表实现用户属性撤销,不需要更新密文和用户私钥,减少了计算量,提高了属性撤销算法执行效率.

参考文献:

[1] Sahai A, Water B. Fuzzy identity-based encryption[M]. Advances in Cryptology-EUROCRYPT 2005. Springer Berlin Heidelberg, 2005: 457-473.

[2] Yu Shucheng, Wang Cong, Ren Kui, et al. Attribute based data sharing with attribute revocation[C]// Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security. [S. l.]: ACM, 2010: 261-270.

[3] Chen Danwei, Wu Qiong, Chen Linling, et al. Attribute-based encryption without central authority: DMA-ABE [J]. Journal of Beijing University of Posts and Telecommunications, 2015, 38(5): 77-80.

[4] Naruse T, Mohri M, 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): 119-125.

[5] Zhang Kai, Ma Jianfeng, Liu Jiajia, et al. Adaptively secure multi-authority attribute-based encryption with verifiable outsourced decryption[J]. Science China Information Sciences, 2016, 59(9): 99-105.

[6] Takeru N, Masami M, Yoshiaki S. 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 Scieneces, 2015 (8): 1.

[7] 陈红松, 韩至. 智慧城市中大数据安全分析与研究[J]. 信息网络安全, 2015(7): 1-6.
Chen Hongsong, Han Zhi. Analysis and research on big data security in smart city[J]. Netinfo Security, 2015 (7): 1-6.

[8] Sahai Amit, Seyalioglu Hakan, Waters Brent. Dynamic credentials and ciphertext delegation for attribute-based encryption[C]// Advances in Cryptology-CRYPTO. California: Springer, 2012: 199-217.