doi:10.11959/j.issn.1000-436x.2015142

面向云存储的基于属性加密的多授权中心访问控制方案

关志涛，杨亭亭，徐茹枝，王竹晓
（华北电力大学 控制与计算机工程学院，北京 102206）

摘 要：已有基于属性加密的访问控制研究多是基于单授权中心来实现，该种方案在授权方不可信或遭受恶意攻击的情况下可能会造成密钥泄露。提出一种基于属性加密的多授权中心访问控制模型PRM-CSAC。基于CP-ABE方法，设计多授权中心的属性加密方案以提高密钥安全性；设计最小化属性分组算法，使用户访问文件时，能够按需分配密钥，减少不必要的属性密钥分配，降低重加密属性数量，提高系统效率；增加读写属性加强加密方对文件的访问控制，使访问控制策略更加完善。安全性分析及仿真实验表明，相比已有方案，PRM-CSAC 对用户访问请求的响应时间更短，开销较小，且能够提供很高的安全性。

关键词：云存储；多授权中心；访问控制；CP-ABE
中图分类号：TP393.08 **文献标识码**：A

Multi-authority attribute-based encryption access control model for cloud storage

GUAN Zhi-tao, YANG Ting-ting, XU Ru-zhi, WANG Zhu-xiao
(School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China)

Abstract: The existing attribute-based encryption access control studies are mostly based on single authority, and this scheme is apt to be under attack to cause exposure of secret keys. Thus, a multi-authority access control model PRM-CSAC is proposed. Based on CP-ABE method, a multi-authority attribute-based encryption scheme is designed to improve security level. Minimized attribute grouping algorithm is designed to distribute keys to users according to needs, which can reduce unnecessary attribute key distribution and decrease the amount of re-encryption attributes. The read and write attribute are added to strengthen the control of owners. The analysis shows that the proposed scheme can meet the security requirement of access control in cloud, and it also has less response time and system cost.

Key words: cloud storage; multi-authority; access control; CP-ABE

1 引言

云服务具有便捷的云端存储、大量的开放软件服务、强大的云计算支撑平台、终端配置要求低[1]、可扩展性高等特点。随着云存储的兴起，越来越多的用户享受到了云计算带来的更大的存储空间和更便捷的存储服务[2]。但云存储中用户数据存储在云服务器上，与用户分离，数据的安全性和完整性难以得到保障，用户对其数据的控制难度也大大增加。

在云存储中要设计行之有效的密文访问控制方案应对下述几个方面进行综合考虑：1)细粒度访问控制：访问策略尽可能可区分更多不同权限的访问用户；2)隐私保护：对在云服务器上存储的用户数据、访问控制信息予以保护；3)系统效率：尽量减少整个方案特别是用户的计算、存储开销。

2 相关研究

Shamir 和 Boneh 等提出并实现了身份加密(IBE, identity-based encryption)机制，首次将用户密钥与

收稿日期：2014-07-02；**修回日期**：2014-11-19
基金项目：国家自然科学基金资助项目（61402171, 61300132）；中央高校面上基金资助项目（JB2014075）
Foundation Items: The National Natural Science Foundation of China (61402171,61300132); The Central Government University Foundation (JB2014075)

用户身份信息相关联[3]。在此基础上，Sahai 和 Waters[4]提出基于属性加密方案(ABE, attribute-based encryption)，以门限值作为访问控制策略，将用户身份信息用多个属性来标识。为表示更加灵活的访问策略，Goyal 提出密钥策略的基于属性加密方案(KP-ABE, key-policy attribute-based encryption)[5]，用关于文件的描述性信息作为属性，采用树形访问控制结构来描述访问控制策略[6]，并将访问树嵌入密钥中。文献[7]采用 KP-ABE 加密算法实现细粒度的访问控制，并结合代理重加密算法和懒惰加密算法降低系统开销[8]。但 KP-ABE 中加密方对访问结构缺乏控制。Benthcourt 等提出密文策略的基于属性加密方案(CP-ABE, ciphertext-policy attribute- based encryption)[9]，以用户身份信息为属性，由加密方构建访问树。文献[10]结合层次化基于身份加密（HIBE）和 CP-ABE 算法提出了一种层次化的属性加密访问控制方案。

以上方案都是由单个授权中心完成密钥的计算分发工作，Chase [11,12]提出了多授权中心的 ABE 方案，将密钥的计算、分配任务交由多个授权中心完成，每个中心都只负责部分密钥的计算工作而不能看到完整的用户信息，既提高了用户信息的安全性，又降低了单个授权中心的计算负担。Chase 的方案使用仅支持门限的 ABE 算法，在加密策略方面缺少灵活性。文献[13]以个人医疗信息记录为例，提出了分域的多授权中心基于属性加密方案，但前提是需要准确地知道访问者的逻辑访问域，同一个文件在每个访问域中都要进行一次加密。文献[14]的多授权中心访问控制方案中实现了对用户身份信息的隐私保护，并用多棵访问树分别对应不同的文件操作权限，但多访问树的设计无疑增加了文件加密的复杂性和存储、传输的负担。

文献[15]提出了代理重加密的思想，可将一密钥下的密文在不解密的情况下经计算转换为另一密钥下的密文。文献[7]实现了基于 KP-ABE 的代理重加密算法。多授权中心机制具有较好的安全性，但在多授权中心场景下使用 CP-ABE 算法实现访问控制却缺少好的代理重加密方案来提高其整个系统的效率。

针对上述问题，本文提出一个安全高效的、支持代理重加密的多授权中心云存储访问控制方案 PRM-CSAC(proxy re-encrypt mulit-authority cloud storage access control)，基于 CP-ABE 算法，将单授权中心扩展至多授权中心提高密钥安全性，并实现 CP-ABE 的可代理重加密。

3　预备知识

3.1　双线性对

定义 1　设 q 是一个大素数，G_1 是阶为 q 的循环群，生成元为 g。双线性映射 e: $G_1 \times G_1 \rightarrow G_2$ 为一个双线性对[16]，$\forall m,n \in G_1$，选择任意的 $a,b \in Z_p$(Z_p 为素数 p 阶循环群),e 具有以下性质。

1) 双线性：有 $e(m^a,n^b) = e(m,n)^{ab}$；

2) 对称性：$\forall m,n \in G_1, e(m,n)=e(n,m)$；

3) 非退化性: $e(g,g) \neq 1$。

由上述性质可知，对生成元 g 有 $e(g^a,g^b)= e(g,g)^{ab}= e(g^b, g^a)$。

3.2　访问结构

基于 CP-ABE 算法，以 Visitor 描述性身份信息作为属性，构建属性全集[17] $P=\{P_1,P_2,\cdots,P_n\}$。每一个 Visitor 作为 Owner 文件的访问用户，具有身份信息以属性集 A 表示，是以上属性全集的非空子集，$A \subseteq \{P_1,P_2,\cdots,P_n\}$。如可构建属性全集 P={北京，上海，一中，二中，学生，教师，教务管理员}，Visitor1 属性集 A={北京，二中，学生}[18]。

访问结构 T 是属性全集 P 的非空子集，T 代表一个属性判断条件，在 T 中的集合得到授权，不在 T 中的集合无授权。只有授权用户的密钥可解密文件。

以访问树来描述访问结构，n 个属性作为访问树的 n 个叶节点，一个 Write 节点和一个 Read 节点也作为叶子节点实现加密方对用户的读写权限的控制。每个非叶子节点代表一个关系函数，关系函数可以是 AND(n of n)、OR(1of n)以及 n of m(m>n)门限等[9]。

3.3　安全性假设

1) 假设云服务提供方是不完全可信的[19]，即云服务方不会主动泄露用户信息。但是存取方案中的漏洞可能造成用户文件、访问权限等信息的泄露，而这些信息可能会被恶意攻击者或云服务方内部恶意员工、外部攻击者利用。

2) 在多授权中心的环境下，假设当 n 个授权中心中的 m 个受到攻击时，攻击者得到的密钥不足以解密文件，但当 m+1 个受到攻击后攻击者获得的密钥能够解密文件，则认为此方案最多能够抵挡 m 合谋攻击[14]。

3.4 符号使用及定义

本文中所用符号及定义如表 1 所示。

表 1 符号定义及说明

符号	说明
PK, *MK*	Public Key 公钥, Master Key 主密钥
SK, *RK*	用户私钥, 重加密密钥
Ver_i^j,Ver_{F1}^k	属性 *i* 的版本号 *j*，文件 File1 的版本号 *k*
Atti，$Atti_1$	属性名 *i*，属性名 *i* 的第 1 个候选属性值
T, *LT*	树形访问结构，访问树 *T* 的叶子节点集合
v=*Att*(*x*)	*T* 中的叶子节点 *x* 对应的属性值 *v*
AMS	属性管理服务器
CS	云服务器
AA	属性授权中心
M→*CT*	明文 *M* 加密后形成密文 *CT*
$\delta_{O.F}$	Owner 对文件 File 的权限签名

4 基于请求的多授权中心访问控制

4.1 PRM-CSAC 模型描述

本方案整体模型如图 1 所示，引入属性管理服务器（AMS, attribute management sever）为用户属性分配授权中心。整体流程概述如下。

Step1 加密：1) Owner 根据文件加密需求，将 Owner加密用到的属性名集按最小化方式分成*N*个不相交的子集，由 AMS 交给 *N* 个 AA 分别管理；2) Owner 将文件与相关访问树加密后存储到 CS 上。

Step2 文件访问：1) 新的 Visitor 向 CS 发出访问 Owner1 的文件 File1 的申请，CS 验证后向 AMS 申请密钥。2) AMS 根据申请的文件查找 AA 管理表，将密钥计算任务交给管理此属性的 AA。AA 计算密钥，经由 AMS,CS 传递给 Visitor。传递过程中密钥以密文形式传送，AMS 和 CS 都不能看到密钥内容。

Step3 用户撤销：由 Owner 计算重加密密钥交给 CS 来完成代理重加密，使 CS 可在不解密原文的情况下完成密钥更新操作。

PRM-CSAC 模型的特点：1) 给出多授权中心环境下的最小化属性分组方案，降低用户撤销时需要重加密属性的数量，提高重加密效率；2) 在云服务器上以属性名形式存储属性全集信息，保护属性值信息；3) 将读写节点引入访问树，实现读写权限控制；4) 增加重加密参数，实现多授权中心的可代理重加密云存储访问控制方案。

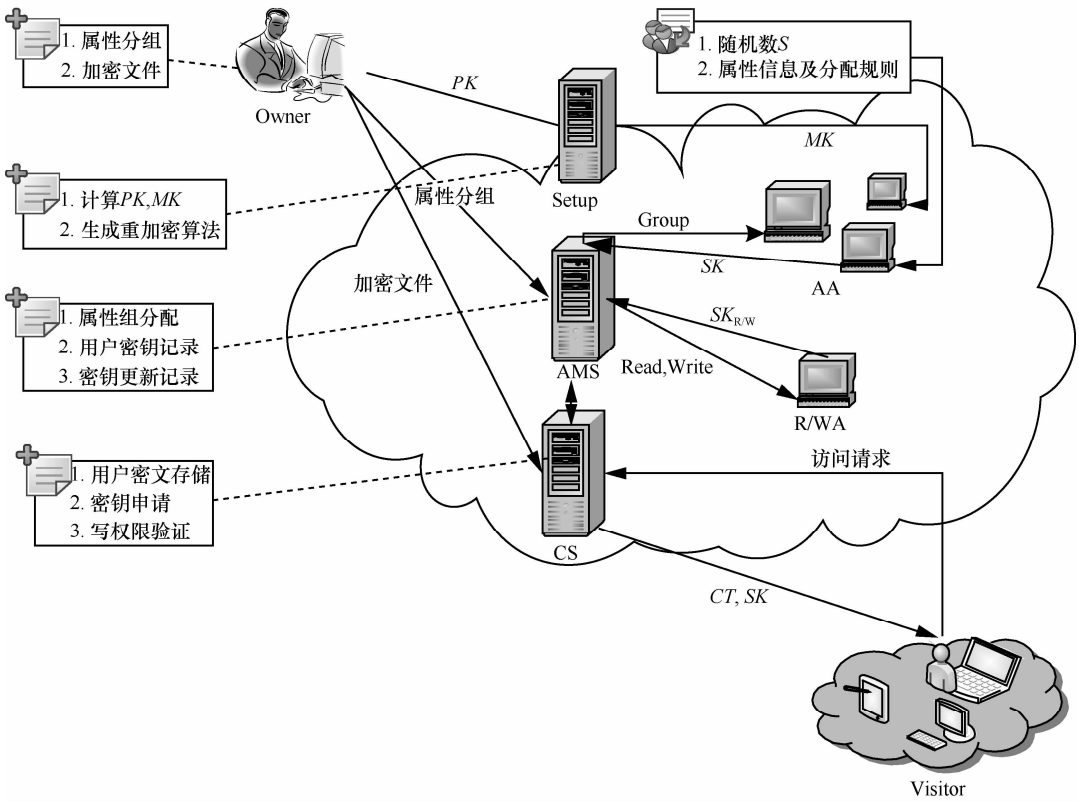


图 1 PRM-CSAC 系统模型

4.2　系统初始化操作

4.2.1　文件属性集划分

本方案中将属性全集按照其属性值划分为多个互不相交子集，每个子集取一个属性名，每个子集中的元素为此属性名的候选值。如上述属性全集可划分为 3 个子集：地市{北京，上海}；学校{一中，二中}；身份{学生，教师，教务管理员}。Owner 访问树的每个叶子节点对应一个属性名，其具体的值为属性名中一个候选值。除计算密钥之外的云服务器上处理属性信息均使用属性名，保护 Owner 隐私信息，提高安全性。

假设加密用户 Owner1 文件加密时与属性可能的关联关系如图 1 所示。按照最小化原则，将每次加密都一起使用的属性为一组，如 *Att*1 和 *Att*2 总是在一起加密，则将其归为一组；*Att*3 在 2 种不同加密文件中都用到，单独归为一组，*Att*6 单独归为一组；*Att*4 和 *Att*5 共同使用且只在此文件加密情况下使用，则 *Att*4 和 *Att*5 为一组。Read 和 Write 节点为一组。则图 2 所示情况分组如表 2 所示。

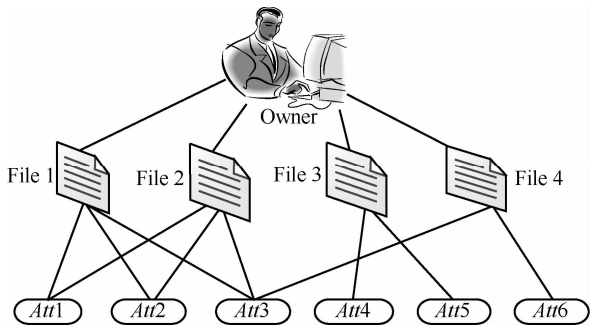


图 2　用户文件加密属性示意

4.2.2　属性集分配

Owner1 将属性按最小化分组之后，将分组数据交给 AMS。AMS 根据用户分组表，将每个 Group 交给一个 AA，其中 R/WA 专门用于管理 Read-Write 节点。如 AA1 管理 Group1，则将元组{Owner1, Group1, *Ver*1:*Att*1,*Att*2}交给 AA1。分配结束后将表 2 后加上分配信息后形成表 3 形式交给 Owner1。

Owner 了解到 AA 相应信息后可直接将属性名对应的候选属性值及分配规则用 AA_k 的 PK_{AA_k} 加密交给 AA_k。如 *Att*1 为“地市”这一身份属性，其属性值集合为{北京，上海，*}，其中*表示无匹配信息分配给新用户时的选项。属性分配规则规定了怎样为用户进行上述属性值的选择。R/WA 只计算读写请求密钥，不分配属性值。

整个过程中的信息都以加密形式传递，AMS 将只存储属性名而不能看到具体的属性值信息，有效保护了 Owner 隐私信息。

表 2　　**用户属性分组**

Owner	Group	Ver	Attributes
Owner1	Group1	Ver_{G1}^1	*Att*1, *Att*2
	Group2	Ver_{G2}^1	*Att*3
	Group3	Ver_{G3}^1	*Att*6
	Group4	Ver_{G4}^1	*Att*4, *Att*5
	Group5	—	Read/Read, Write

表 3　　***Att*-AA 分配**

Owner	Group	Ver	Attributes	AA	PK
Owner1	Group1	Ver_{G1}^1	*Att*1, *Att*2	AA_1	PK_{AA_1}
	Group2	Ver_{G2}^1	*Att*3	AA_2	PK_{AA_2}
	Group3	Ver_{G3}^1	*Att*6	AA_3	PK_{AA_3}
	Group4	Ver_{G4}^1	*Att*4, *Att*5	AA_4	PK_{AA_4}
	GroupRW	—	*Att*R, *Att*W	R/WA	$PK_{R/WA}$

4.2.3　初始参数设置

如图 1 所示，使用 Setup Sever 服务器为 Owner 完成密钥初始化工作，设置相关参数，具体为执行 Setup 操作。

Setup→*PK*,*MK*,*RK*：

$$PK = G0, g, h = g^{\beta}, e(g,g)^{\alpha} \tag{1}$$

$$MK=(\beta, g^{\alpha}) \tag{2}$$

$$RK=\{rk_1, rk_2, \cdots, rk_m\} \tag{3}$$

其中，*G*0 为素数 *p* 阶双线性群，*g* 为生成元，选择随机数 $\alpha, \beta \in Z_p$ (*p* 阶循环群)[9]。

预设版本号最大值为 *m*，对第 *k* 个版本 Ver^k，$1 \leqslant k \leqslant m$,选择随机数 $rk_k \in Z_p$，得到重加密密钥 *RK*。

操作执行结束后，Setup Sever 将 *MK*，*RK* 分发给每个 AA，*PK*，*RK* 交给请求 Setup 操作的 Owner。

4.3　文件加密

4.3.1　访问树（Tree）构建

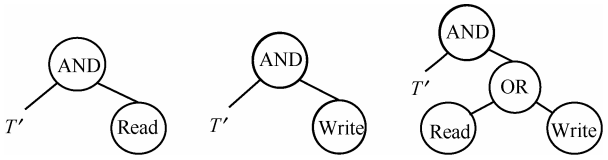
Owner 先对要加密的文件 File1 按照以下规则构建访问树，将对用户身份的描述性属性信息构建属性全集作为叶子节点，AND(*n* of *n*)、OR(1of *n*)以及 *n* of *m*(*m*> *n*)门限作为非叶子节点。

本方案中在访问树中增加读/写节点来控制对 Visitor 的读写请求授权。若 Owner 以属性构建的访问树为 *T*’，将 AND 作为 *T* 的根节点，*T*’作为 *T* 的

左子树，将对文件的访问请求分为 Read 和 Write，Read 和 Write 作为叶子节点构成读写访问权限子树控制读写访问。那么，对于 File1 的访问策略 T'：

1) 若 Owner 只允许满足 T' 的用户读或写文件，将节点 Read/Write 作为 T 的右子树,读/写权限访问树如图 3(a)、图 3(b)所示；

2) 若允许满足 T' 的用户写文件，用 OR 节点连接 Read 和 Write 这 2 个叶子节点，将 OR 子树作为 T 的右子树，如图 3(c)所示。



(a) 读权限访问树 (b) 写权限访问树 (c) 读写权限访问树

图 3 访问树构建

4.3.2 文件加密

文件加密是将文件与访问策略共同加密，执行 Encryption 操作。对于允许写操作的文件，Owner 在文件末尾加入个人数字签名 δ_{O1F1}，与文件一起加密。能够正确解密的用户可拥有此签名，作为申请写操作时的验证。对于明文 m,若允许对其进行 Write 操作，在 m 后加上个人签名 δ_{O1F1}；若只允许 Read 操作则在此位置加上*，表示无写权限签名，即此文件不允许写操作，m 与签名形成明文 M。Owner1 执行 Encrypt(PK, M, T)，生成密文 CT。

CT=Encrypt(PK,M,T)：

$$CT=(T,\ \tilde{C}=Me(g,g)^{as},\ C=h^{s},$$
$$\forall y\in Y: C_y=g^{q_y(0)}, C_y'=H(att(y))^{q_y(0)rk_y^k}) \quad (4)$$

T：访问结构；

Y：访问树叶子节点集合；

q_x：节点 x 的多项式，对根节点 R，选择随机数 $s\in Z_p$，$q_R(0)=s$。$H(i)$表示属性 i 转化成二进制值的函数[9]。

Owner1 将选择的随机数 S 分发给管理 Owner1 属性集的 k 个 AA，用于密钥的加密操作。将密文和加密用到的属性组以及当前文件版本号组成 tuple $\{Owner1, CT_{\mathrm{File1}}, \{Att_i\}_{i\in L_T}, C, Ver_{F1}^{1}\}$ 存放到 CS。

4.4 密钥分配

1) 新用户请求

新用户 Visitor1 要访问 Owner1 的 File1（加密属性如图 2 所示），对其申请执行 Write 操作。CS 验证 Visitor1 ID 后检查其请求记录发现此用户是新加入此系统，则为其申请属性值和密钥。过程如图 4 所示。

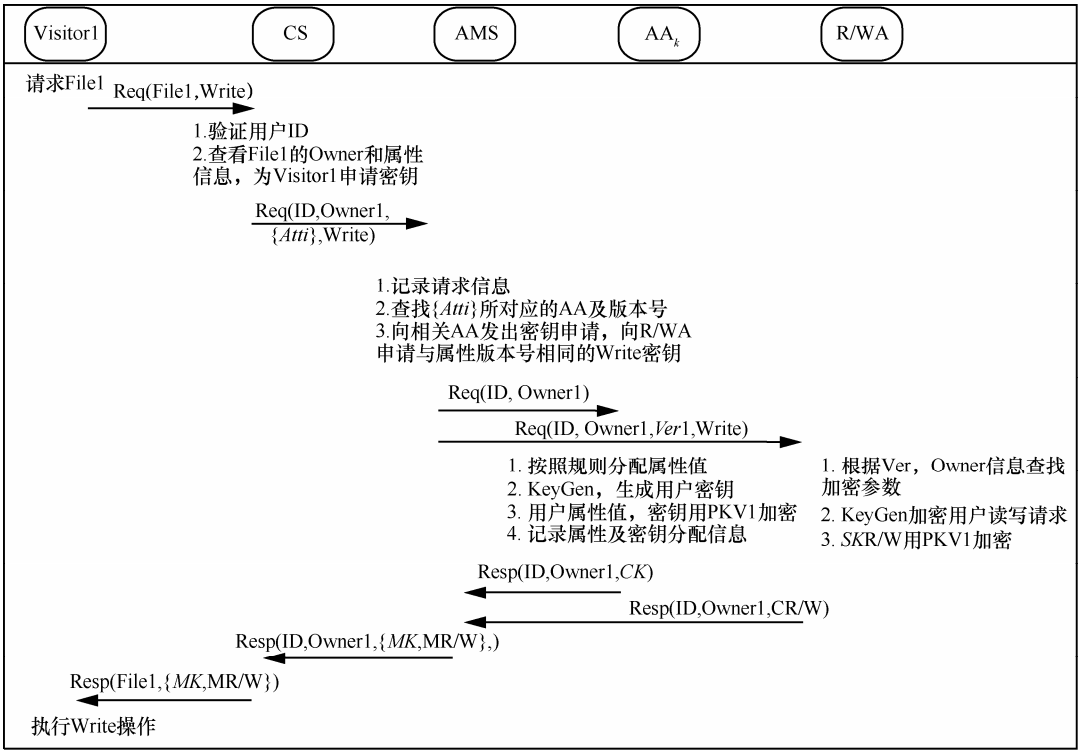


图 4 用户请求响应流程

密钥生成算法输入主密钥 *MK*，重加密密钥 *RK*，属性集 *I*，生成私钥 *SK*。任选一 AA 计算 *D* 值，所有相关 AA 均执行

KeyGen(*MK*,*RK*,*I*) →*SK*：

SK=($D = g^{(\alpha+r+rk^k)/\beta}$,

$$\forall j \in I : D_j = g^{r+rk^k} H(j)^{r_j rk_j^k}, D_j' = g^{r_j}) \quad (5)$$

其中，随机数 $r \in Z_p$，对集合 *I* 中的每个属性 $j \in I$，选择随机数 $r_j \in Z_p$。

在为每个用户生成密钥时每个 AA 要使用同一个随机数 *r*，使任一 AA 选择随机数 *r*，并采用如图 5 所示过程来完成各个 AA 中 *r* 值的传递。

2) 已有用户请求新文件

若 Visitor1 在申请 Owner1 的文件 File1 之后又发出访问 File4 的申请，CS 向 AMS 发出为 Visitor1 获取 Owner1 的密钥 *Att*3，*Att*6 的申请。AMS 检查分配记录发现已经为 Visitor1 分配 *Att*3 的密钥，且密钥版本尚未更新，则只需为其申请 *Att*6 的密钥，仍按图 4 所示流程完成。这样就不需对已分配的密钥再次加密分配，减少不必要的加密操作，有效降低计算开销。

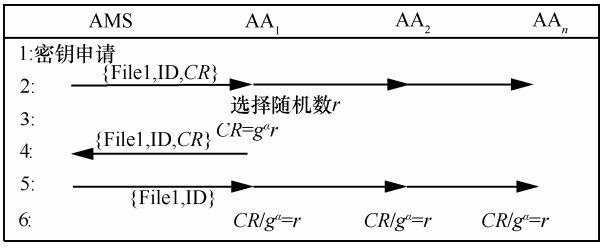


图 5　AA 间 *r* 值传递

4.5　用户解密

Visitor1 申请到 File1 和相应密钥之后执行 Decrypt 算法进行解密操作，即可访问明文 *M*。

Decrypt[9] (*CT*,*SK*)→*M*:

x 为访问树中节点，*i* 为其对应属性，为叶子节点解密定义非递归算法 DecryptNodeL(*CT*,*SK*,*x*)，非叶子节点定义递归算法 DecryptNodeNL(*CT*,*SK*,*x*)。

如果 $i \in I$，那么

$$\text{DecryptNodeL}(CT,SK,x) = \frac{e(D_i, C_x)}{e(D_i', C_x')}$$

$$= \frac{e(g^{r+rk^k} H(i)^{q_x(0) rk_i^k}, g^{q_x(0)})}{e(g^{r_i}, H(i)^{q_x(0) rk_i^k})}$$

$$= e(g,g)^{(r+rk^k) q_x(0)} \quad (6)$$

如果 $i \notin I$，令 DecryptNodeNL(*CT*,*SK*,*x*)=⊥ 。

当 *x* 是非叶子节点时调用递归算法 Decrypt NodeNL(*CT*, *SK*,*x*)：

对 *x* 所有的孩子节点 *z* 调用 DecryptNodeL(*CT*, *SK*,*z*)并将输出存储为 F_z。让 I_x 是子节点 *z* 的 k_x(非叶子节点的门限值)大小的任意集合，使 $F_z \neq \perp$。如果没有此集合存在，那么节点就会不满足函数，返回⊥。

否则做如下计算并返回结果。

$$F_x = \prod_{z \in I_x} F_z^{\Delta_{i,I_x'(0)}}, i = index(z), I_x' = \{index(z) : z \in I_x\}$$

$$= \prod_{z \in I_x} (e(g,g)^{(r+rk^k) q_z(0)})^{\Delta_{i,I_x'(0)}}$$

$$= \prod_{z \in I_x} (e(g,g)^{(r+rk^k) q_{parent(z)}(index(z))})^{\Delta_{i,I_x'(0)}}$$

$$= \prod_{z \in I_x} (e(g,g)^{(r+rk^k) q_x(i)})^{\Delta_{i,I_x'(0)}}$$

$$= e(g,g)^{(r+rk^k) q_x(0)} \quad (7)$$

index(*z*)为节点 *z* 作为 *x* 子节点的唯一索引值。

解密时对 *T* 的根节点 *r* 调用函数，如果 *I* 能满足访问树，最终可得

A = DecryptNodeNL(*CT*,*SK*,*r*)

$$= e(g,g)^{(r+rk^k) q_R(0)} = e(g,g)^{(r+rk^k) s} \quad (8)$$

再经过以下计算如下解密

$$\tilde{C} / (e(C,D) / A)$$

$$= \tilde{C} / (e(h^s, g^{(\alpha+r+rk^k)/\beta}) / e(g,g)^{(r+rk^k) s}) = M \quad (9)$$

Visitor1 解密得到 File1 的写权限签名 δ_{O1F1}，可以执行对 File1 的写操作。

4.6　用户撤销及更新

4.6.1　访问策略更新

CS 为文件 File1 存储元组 tuple，访问策略更新流程如图 6 所示。经过更新后 CS 存储新版本的 tuple 信息，Visitor 获得更新后的密文。

4.6.2　用户撤销

当某 Visitor 离开系统后,为防止已撤销用户用所持密钥重新访问文件，需要对文件进行重加密操作。重加密流程如图 7 所示。

重加密主要有文件重加密和密钥重加密 2 部分。文件重加密主要是对访问树相关属性的重加密。为实现便捷的代理重加密，在加密时将属性的密钥版本号分别嵌入访问树属性信息与用户密钥

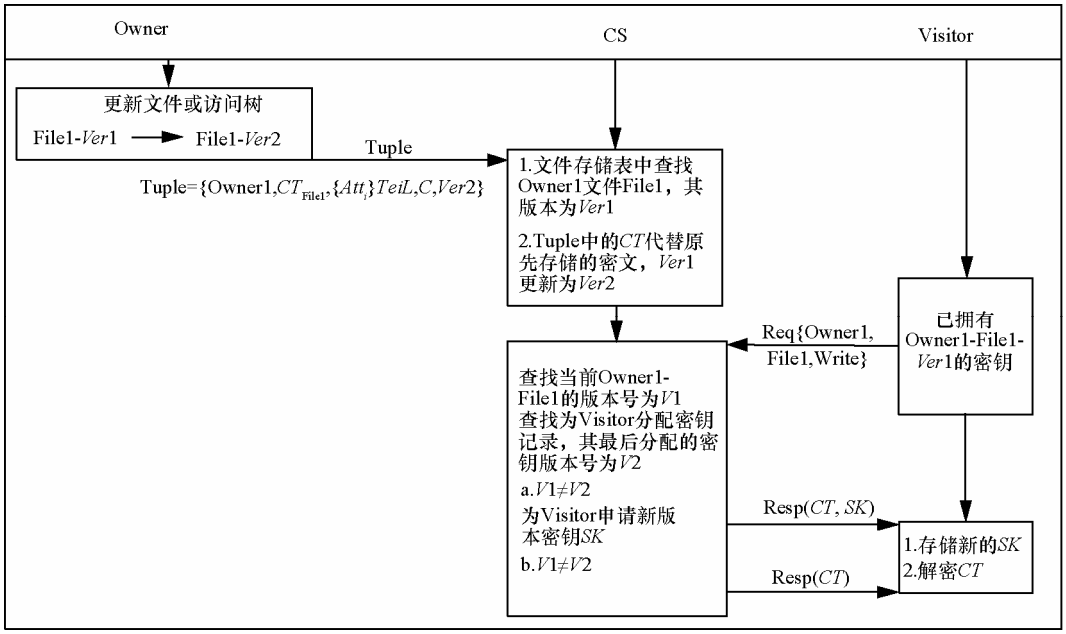


图 6 文件及访问策略更新流程

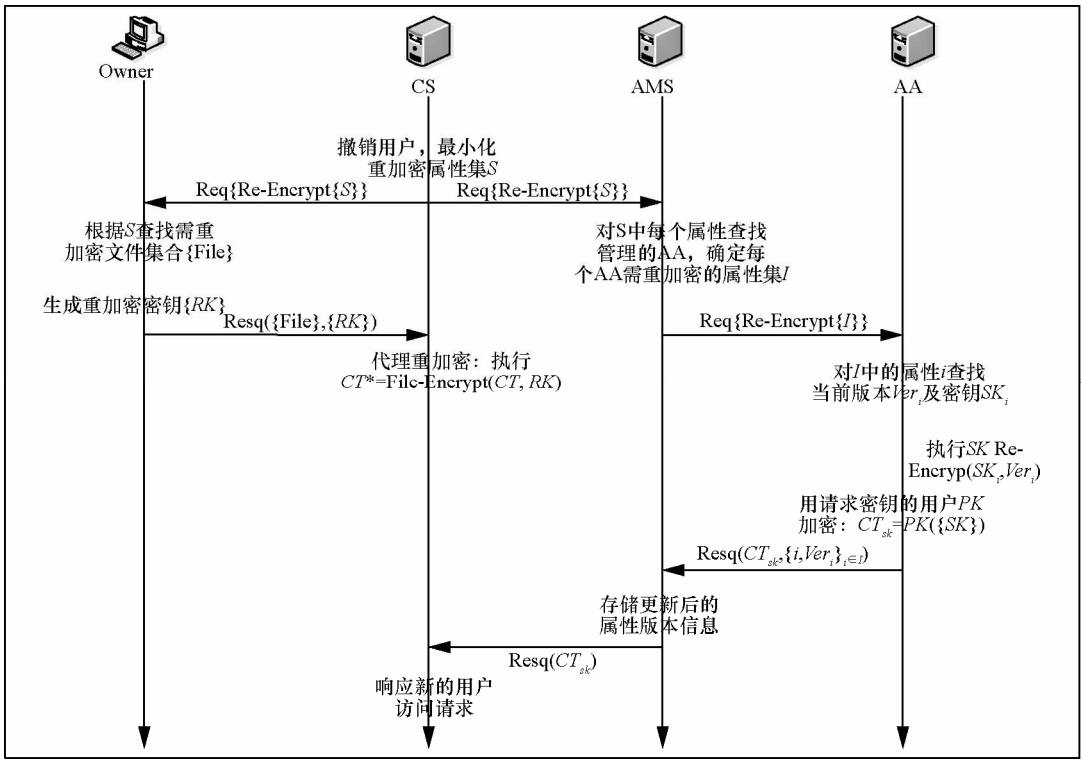


图 7 用户撤销处理流程

中，并将重加密密钥与版本号相关联。重加密操作定义 File Re-Encrypt 和 SK Re-Encrypt 这 2 个算法，分别用于文件重加密和密钥重加密。

File Re-Encrypt(CT, RK)：

由 CS 执行，输入文件 File1 密文 $CT(T, \tilde{C}, C, C_x, C_x')$，以及每个属性 i 的重加密密钥 RK_i,更新后的属性密文。

对于 File1 加密的访问树 T 的叶子节点 x，$\forall i = Att(x), i \in S$，即 S 为访问树 T 的叶子节点属性集合。$\forall i \in S$，将属性版本号 Ver_i^k 更新为 Ver_i^{k+1}，CT 的版本号为 k 的密文表示为 $C_i'^k$，一次版本更新后为 $C_i'^{k+1}$，则计算方法如下

$$C_i^{'k+1}=C_i^{'k\,rk_i^{k+1}/rk_i^k}=(H(i)^{q_x(0)rk_i^k})^{rk_i^{k+1}/rk_i^k}=H(i)^{q_x(0)rk_i^{k+1}} \tag{10}$$

即重加密密钥

$$RK_{k\to k+1}=rk_i^{k+1}/rk_i^k \tag{11}$$

SK Re-Encrypt(*SK*,*Ver*,*RK*):

由 AA 执行，输入一个属性集 S 的密钥集合 $SK(D,D_j,D_j^{'})$，以及需重加密属性 i（版本号为的版本号 Ver_i^k）的重加密密钥,输出版本号为 Ver_i^{k+1} 的密钥 SK_i^{k+1}。

对于属性 $\forall j,SK_j\in SK$，执行以下操作。

$$\begin{aligned}D_j^{k+1}&=D_j^kH(j)^{r_j(rk_j^{k+1}-rk_j^k)}g^{r+rk_j^{k+1}-rk_j^k}\\&=g^{r+rk^k}H(j)^{r_jrk_j^k}H(j)^{r_j(rk_j^{k+1}-rk_j^k)}g^{r+rk^{k+1}-rk^k}\\&=g^{r+rk^{k+1}}H(j)^{q_x(0)rk_i^{k+1}}\end{aligned} \tag{12}$$

经过以上版本更新的重加密操作后，文件中的属性密文与密钥中的信息均改变，使已撤销用户持有的密钥已不能再与密文中的属性信息相匹配，解密时无法解出正确的 $e(g,g)^{(r+rk^k)q_x(0)}$。

5　安全性分析

5.1　整体安全性分析

多授权中心安全性特点：每个 AA 都只计算部分属性的密钥，不能看到用户的全部密钥。

本方案中 AMS 要保存所有 Owner 的属性名，但不能获得任何一个属性的属性值和密钥，只是为 Owner 完成属性分配管理的任务，没有泄露用户隐私信息。CS 也同样能看到文件加密相关属性名但是不能获得用户属性值和密钥。

重加密操作中将属性密文、密钥与版本号相关联，保证用户在撤销后其密钥不能再解密密文。

5.2　合谋攻击抵御分析

1) 假设一个文件加密属性由 N 个 AA 管理，则敌手若要解密此文件必须要得到 N 个 AA 提供的全部密钥，若有一个 AA 不与其他合谋，泄露密钥信息，所得的密钥也不能满足访问树加密要求。即本方案至多能够抵御 $N-1$ 服务器合谋攻击。

2) 若有不同权限的用户进行合谋时，若要解密出明文，攻击者必须恢复出 $e(g,g)^{(r+rk^k)s}$ 的值。这需要合谋的攻击者提供足够的属性密钥来满足访问树 T。但若满足 T 中每个属性的属性密钥来自不同的密钥，由于加密时采用的随机数不同，在解密时无法满足多项式插值，不能恢复出明文。因此来自不同密钥的密钥成分联合不能解密其他文件，从而能够抵御用户的合谋攻击。

3) 本方案在访问树中引入读写访问控制节点，并与原访问树用 AND 根节点连接。读写节点在访问树中同样作为叶子节点出现，并且只有当读写请求与访问权限同时满足时才能解密密文。若有攻击者持有读写节点密钥和不能满足原访问权限的部分叶子节点密钥，无法解密。对于能够满足访问权限的用户若提出超出其读权限的写请求，也无法完成其请求操作。因此读写节点的引入加强了 Owner 对文件的访问控制，同时没有降低其安全性。

5.3　算法安全性证明

文献[9]对 CP-ABE 的安全性予以了证明，本文基于 CP-ABE 算法实现了代理重加密，算法安全性并未降低，下面给出证明。

定义 2　原 CP-ABE 算法是安全的。

定理 1　本系统模型算法安全性不低于原算法。

证明　与原 CP-ABE 相比，本模型在算法上增加了重加密参数 RK 来实现代理重加密，增加此参数后并未降低系统安全性。

由上文可知，在式(4)和式(5)中，对于 CT，令 $q_y(0)_1=q_y(0)rk_y^k$，因为 $q_y(0)$ 与 rk_y^k 均为随机数，所以 $q_y(0)_1$ 也为随机数。得到

$$C_{y1}^{'}=C_y^{'}=H(att(y))^{q_y(0)_1}$$

$$CT1=(T,\tilde{C},C,C_y,C_{y1}^{'})$$

对于 SK，令 $r_1=r+rk^k$，$r_{j1}=r_jrk_j^k$，因为 r,rk^k,r_j 均为随机数，所以 r_1,r_{j1} 也为随机数，则有

$$D_1=D=g^{(\alpha+r+rk^k)/\beta}=g^{(\alpha+r_1)/\beta} \tag{13}$$

$$D_{j1}=D_j=g^{r+rk^k}H(j)^{r_jrk_j^k}=g^{r_1}H(j)^{r_{j1}} \tag{14}$$

得到 $SK_1=(D_1,D_{j1},D_j^{'})$。

至此可得，解密本方案算法的安全性不低于 CP-ABE 算法。

6　开销评估

最小化属性集分组每次只对用户需要的密钥进行分配，暂时不需要的属性不分配密钥，减少了一部分密钥计算操作。特别是当需要重加密时，避免了用户持有而未使用密钥的重加密。用户属性集分组数量与用户属性数量和加密属性的数量相关，

最差情况下每个属性分为一组由一个 AA 管理。

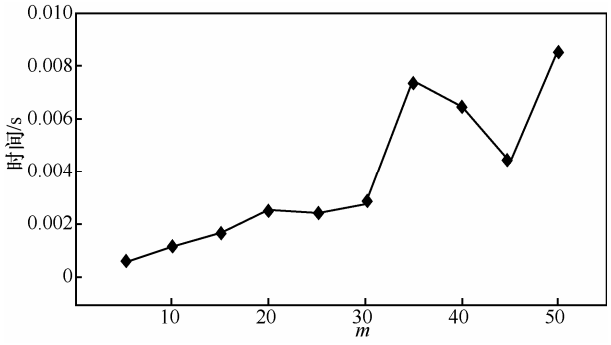
下面对以下几个主要操作进行计算复杂度的分析。

1) 系统初始化 Setup：此算法计算公钥 *PK*、主密钥 *MK* 和重加密参数 *RK*，选择 2 个随机数，进行 3 个幂的计算：$h=g^{\beta}, e(g,g)^{\alpha}, g^{\alpha}$。设定版本号最大值 m，并为每个版本号选择一个随机数。时间复杂度为 $O(m)$。Setup 操作时间开销随 m 变化如图 8(a)所示。

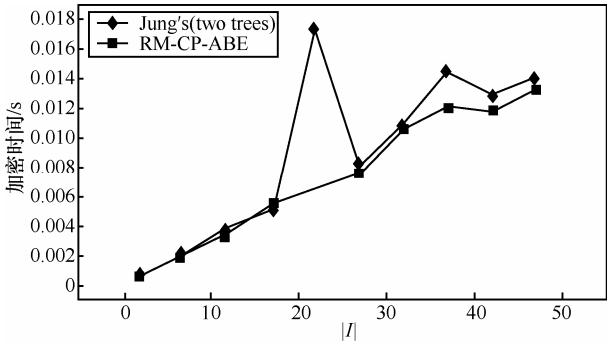
2) 文件创建：Encrypt 算法中对明文只进行一个乘的操作，加密开销只与明文长度相关。加密访问树需要对每个属性进行 2 次幂运算，计算开销与属性的数量成正比，若 File1 中访问树相关的属性集合为 I(可用$|I|$表示属性数量)，则文件创建操作的时间复杂度与计算开销可表示为 $O(|I|)$。

文献[13]在多授权中心环境下通过不同访问树实现了对文件读写权限的控制，但每个权限的实现都需要原文及访问树的一次加密操作。同样实现读写 2 种访问控制权限时，这方案需要 2 棵访问树 2 次加密，PRM-CSAC 方案能够通过一棵访问树实现，与文献[13]方案相比加密开销更低(如图 8(b)所示)。

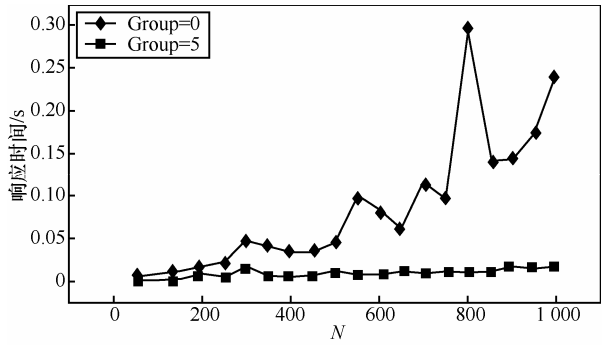
3) 用户新访问请求授权：此操作主要是为 Visitor 新的访问请求申请密钥。最坏情况下，需为其申请 File1 访问相关的全部 I 个属性的密钥。



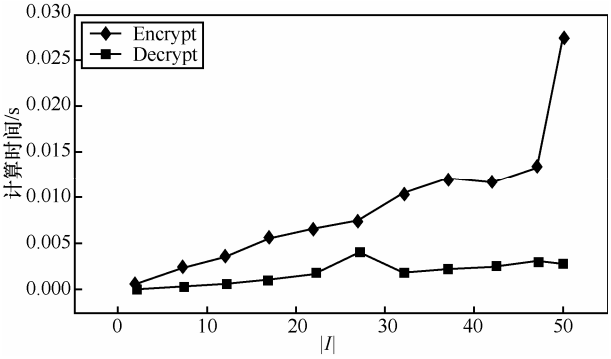
(a) Setup 时间开销与版本号数量 m 关系



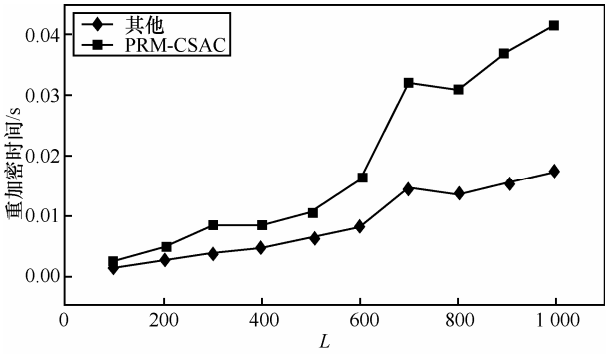
(b) 实现读写权限控制的加密开销比较



(c) 多授权中心密钥响应时间比较



(d) PRM-CSAC 加密解密时间开销



(e) 重加密时间开销比较

图 8 开销评估

当此 Visitor 是已存在的用户时，根据 AMS 记录，对于已分配过密钥且其版本尚未更新的属性不再进行密钥的计算，减少了计算开销，特别是对于重复访问较多的用户申请会在更大程度上减少密钥计算量。密钥计算由 N 个 AA 共同完成，N 的数量与属性分组数相关。由于每个 AA 完成密钥计算后需要利用用户的 PK 进行加密，增加了 N 次加密计算，但对每个 AA 而言只是增加一次加密操作。

假设每一次密钥计算和加密操作计算量均为 1，且 $|I|$ 个属性平均分为 N 个组由 N 个 AA 进行加密。则本方案中密钥分配操作计算量为 $O(|I|+N)$，所需时间为 $O(|I|/N+1)$。不难看出，当分组数 N 越大时计算开销越大，对用户的响应时间越少。由于密钥的生成是由 AA 完成，则认为云端服务器有足够的计算能力完成要求的操作，因此认为多授权中心的操作对于快速响应用户请求具有时间上的优势。特别是当需要同时响应多个加密请求，密钥计算量较大时能够明显降低计算时间开销（如图 8(c)所示）。

4) 文件访问：在用户得到密文和密钥之后，文件访问开销主要指文件解密操作的开销。而在解密过程中主要的计算开销是每个属性对应的节点的解密操作。则对于文件 File1，其计算量及所需时间均可表示为 $O(|I|)$。加密解密的时间开销如图 8(d)所示。

5) 用户撤销：当合法用户撤销后需要对此用户拥有密钥的属性进行重加密操作，这包括对密文属性的重加密和属性密钥的重加密。

假设用户拥有 L 个属性，最坏情况下这 L 个属性在随后的访问中均需要重新加密。这里的计算开销主要是 File Re-Encrypt、SK Re-Encrypt 这 2 个算法，与需要重加密的属性数量 L 成正比，则重加密计算量和时间开销可表示为 $O(L)$。本方案中由于对用户需要的属性进行响应，避免了用户撤销时部分属性的重加密操作，降低了计算开销（如图 8(e)所示）。

7　结束语

本文提出了一种基于请求的多授权中心属性加密访问控制方案，给出了具体的最小化属性分组方式；实现访问树中的读写控制；实现 CP-ABE 的代理重加密算法；保证用户隐私信息不泄露，即使对必须接触用户信息的 AA 也只能看到用户部分信息。

参考文献：

[1] 李瑞轩, 董新华, 辜希武等. 移动云服务的数据安全与隐私保护综述[J]. 通信学报, 2013,34(12):159-166.
LI R X, DONG X H, GU X W, *et al*. Overview of the data security and privacy preserving of mobile cloud services[J]. Journal on Communications, 2013, 34(12): 159-166.

[2] 冯登国, 张敏, 张妍等.云计算安全研究[J]. 软件学报, 2011, 22(1): 71-83.
FENG D G, ZHANG M, ZHANG Y, *et al*. Study on cloud computing security[J]. Journal of Software, 2011,22(1):71-83.

[3] SHAMIR A. Identity-based cryptosystems and signature schemes[A]. Advances in Cryptology[C]. Springer, 1985. 47-53.

[4] SAHAI A, WATERS B. Fuzzy identity-based encryption[A]. Advances in Cryptology–EUROCRYPT 2005[C]. 2005.557-557.

[5] GOYAL V, PANDEY O, SAHAI A, *et al*. Attribute-based encryption for fine grained access control of encrypted data[A]. CCS[C]. 2006. 89-98.

[6] 苏金树, 曹丹, 王小峰等.属性基加密机制[J].软件学报, 2011, 22(6):1299-1315.
SU J S, CAO D, WANG X F, *et al*. Attribute-based encryption schemes[J]. Journal of Software, 2011, 22(6):1299-1315.

[7] YU S, WANG C, REN K, LOU W. Achieving secure, scalable,and fine-grained data access control in cloud computing[A]. IEEE INFOCOM[C]. 2010. 1-9.

[8] 俞能海, 郝卓, 徐甲甲等. 云安全研究进展综述[J].电子学报, 2013 ,41(2):371-381.
YU N H, HAO Z, XU J J, *et al*. Review of cloud computing security[J]. Acta Electronica Sinica, 2013,41(2):371-381.

[9] BETHENCOURT J, SAHAI A, WATERS B. Ciphertext-policy attribute-based encryption[A]. IEEE S&P[C], 2007.321-334.

[10] WANG G J, LIU Q, WU J. Hierarchical attribute-based encryption for fine-grained access control in cloud storage services[A]. CCS[C]. 2010.735-737.

[11] CHASE M. Multi-authority attribute based encryption[A]. Theory of Cryptography[C]. 2007. 515-534.

[12] CHASE M, CHOW S. Improving privacy and security in multi- authority attribute based encryption[A]. CCS[C]. 2009. 121-130.

[13] LI M, YU S C, ZHENG Y, *et al*. Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption[J]. IEEE Transactions on Parallel and Distributed Systems, 2013, 24(1):131-143.

[14] JUNG T, LI X, WAN Z, WAN M. Privacy preserving cloud data access with multi-authorities[A]. IEEE INFOCOM[C]. 2013.2625-2833.

[15] BLAZE M, BLEUMER G, STRAUSS M. Divertible protocols and atomic proxy cryptography[A]. Proc of EUROCRYPT[C]. 1998. 127-144.

[16] DAN B, MATTHEW K F. Identity-based encryption from the Weil pairing[A]. Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology[C]. 2001. 213-229.

[17] BEIMEL A. Secure Schemes for Secret Sharing and Key Distribution[D]. Haifa, Israel, Israel Institute of Technology, 1996.

[18] 洪澄，张敏，冯登国．AB-ACCS：一种云存储密文访问控制方法[J]．计算机研究与发展,2010,47(z1):259-265.
HONG C, ZHANG M, FENG D G. A cryptographic access control scheme for cloud storage[J]. Journal of Computer Research and Development, 2010,47(z1):259-265.

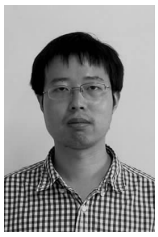
[19] IMERCATI S D C, FORESTI S, JAJODIA S, *et al*. Over-encryption: management of access control evolution on outsourced data[A]. Proc of VLDB'07[C]. 2007.123-134.

作者简介：

关志涛（1979-），男，辽宁沈阳人，博士，华北电力大学讲师，主要研究方向为电力信息安全、云安全、无线传感器网络安全。

杨亭亭（1989-），女，山东淄博人，华北电力大学硕士生，主要研究方向为云安全。

徐茹枝（1966-），女，江西上饶人，博士，华北电力大学副教授，主要研究方向为电力信息安全。



王竹晓（1981-），男，四川自贡人，博士，华北电力大学讲师，主要研究方向为智能电网 Cyber-Physical 系统安全、自治愈技术、知识表示与推理以及分布式动态描述逻辑。