云存储中基于MA-ABE的访问控制方案

李谢华 周茂仁 刘 婷
(湖南大学信息科学与工程学院 长沙 410082)

摘 要 针对云存储中跨域数据访问控制的安全性和有效性问题,提出了一种基于MA-ABE的高效的、细粒度的访问控制方案。新方案通过使用密钥分割技术和代理重加密技术,在权限撤销时保证用户密钥的安全性,并将大部分密文重加密工作转移到云端,以降低数据属主的计算代价。利用数据属主和授权机构分别产生和分发属性私钥组件,将用户全球唯一标识(GID)和用户私钥相分离,避免了授权机构间的联合攻击,有效地保护了用户身份信息。最后,通过理论分析表明了新方案的安全性,并实验验证了该方案在权限撤销时的高效性。

关键词 多授权机构,MA-ABE,数据访问控制,访问结构树
中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.02.028

MA-ABE Access Control Scheme in Cloud Storage

LI Xie-hua ZHOU Mao-ren LIU Ting
(School of Information Sciences and Engineering, Hunan University, Changsha 410082, China)

Abstract In order to improve the security and efficiency of cross-domain data access in cloud storage, this paper proposesd a multi-authority attribute-based encryption (MA-ABE) access control scheme. The new scheme uses split-key to guarantee the security of users' secret key. In addition, proxy re-encryption is used to load most of the re-encryption to the cloud server when revocation occurs, which can minimize the computation cost for the data owner (DO). The splited secret key components are generated and distributed by the DO and attribute authorities (AA) respectively without using their global identifier (GID), which can prevent authorities collusion attack. Finally, theoretical analysis has been provided to prove that the new scheme is secure and has high performance on revocation.

Keywords Multi-authority, MA-ABE, Data access control, Access tree

1 引言

随着云存储业务的快速普及,数据在云端的安全性问题已经成为阻碍其业务进一步发展的瓶颈,特别是如iCloud等云存储服务器(Cloud Service Provider,CSP)爆出文件泄露事故之后,云存储安全问题更成为了公众关注的焦点。目前,对云数据进行安全保护主要采用传统的数据加密方法,然而这些方法需要数据属主(Data Owner,DO)针对每个用户的数据进行加密,不仅加密操作运算量较大,而且难以实现数据的共享,不适合于云存储中海量数据加密和用户间数据安全共享的需求。为此,基于属性加密(ABE)的方法被认为是解决云存储中数据安全和共享的有效加密机制[1]。

基于属性的加密算法(Attribute-Based Encryption,ABE)是由Sahai和Waters等人提出的一种通过用户属性控制数据加解密的方法。ABE方法分为Ciphertext-Policy ABE(CP-ABE)[2-3]和Key-Policy ABE(KP-ABE)[4]。在CP-ABE中,密钥与属性集相关,密文与访问结构树相关,若属性集满足该访问结构树,则用户可以解密。在KP-ABE中,密钥与访问结构树相关,密文与属性集相关。针对云存储中密文更新频繁、用户数量大的特点,CP-ABE方案更适合于云存储中的数据访问控制。因此,基于CP-ABE密文访问控制方法已经得到了广泛应用[5-7]。

现有的大多数CP-ABE方案通常仅适用于单授权机构环境下的密文访问控制。然而,随着云存储业务的快速发展,大量云服务提供商(即授权机构)并存,因此如何在多授权机构环境下实现云数据的访问控制和共享已经成为云计算发展所必须解决的问题。目前,对多授权机构环境下的密文访问控制研究主要有Chase提出的多授权机构ABE方案[8],其通过中央授权机(Central Authority,CA)将用户身份标识(Global Identity,GID)和各属性授权机构(Atrribute Authority,AA)生成的私钥结合在一起。然而,由于CA拥有系统的主密钥和用户私钥,其一旦被攻破,将造成数据的泄露。为解决CA造成的安全隐患,Lin等人[9]采用无CA的密钥分发和联合的零秘密共享技术,但此技术最多只能防止m个用户的联合攻击。Lewko和Waters等人提出了一种新的基于多授权机构的CP-ABE方案[10],此方案仅在初始化阶段采用CA,

到稿日期:2015-11-18 返修日期:2016-04-01 本文受国家自然科学基金(61402160),湖南省高校创新平台开放基金(14K023)资助。
李谢华(1977—),女,博士后,助理教授,硕士生导师,主要研究方向为网络安全,E-mail:beverly@hnu.edu.cn;周茂仁(1990—),男,硕士生,主要研究方向为云安全技术;刘 婷(1990—),女,硕士生,主要研究方向为信息安全。

由 CA 为授权机构分发公共参数并根据用户的请求验证 AA，此后 CA 将不再参与任何运算。这些方法对多授权主体环境下的安全接入和数据共享进行了深入细致的研究，但是对于用户的权限更新和撤销却缺乏有效的手段。在该领域，Yu 等人[11]通过代理服务器为非撤销用户更新用户密钥和密文，虽然能够实现用户的立即撤销，但每次权限撤销都会引起大量属性私钥和密文的更新，因此这种方法并不适用于云存储环境。Yang 等人[12]提出了通过更新系统属性列表中的属性版本号，生成代理重加密密钥，更新密文和用户私钥，但每次进行用户撤销时，非撤销用户的密钥都要更新。Li 等人[13]提出了一种基于 KP-ABE 的多授权机构的 ABE 方案，但该方案在云存储环境下的访问控制效率较低。Yang Kan 等人[14-15]提出了一种基于属性的多授权机构访问控制方案，当需要进行用户撤销时，AA 更新版本号以及公钥，并将更新的公钥发送给 DO 和非撤销用户，但是非撤销用户的密钥更新代价比较大。Hur[7]通过属性加密密钥来更新密文和用户私钥，但数据服务管理者需要知道每个用户的属性集才能生成属性加密密钥，这并不适用于云存储环境。Eissa[16]利用 KP-ABE 方案，当用户撤销时，系统不需要进行重加密以及重新生成系统公共参数和用户的私钥。但是用户撤销后，在访问结构树中添加非属性。如果属性集中有满足非属性的属性，用户不能进行解密，该方案的密钥策略比较复杂。

综上所述，如何在云存储中解决基于多授权机构并存的属性撤销问题成为本文的重点研究内容。针对此，本文提出了一种多授权机构环境下的有效权限撤销方案。通过密钥分割和代理重加密，分别实现特定用户权限和属性权限的细粒度撤销，并且能够抵御授权机构间以及用户和授权机构的联合攻击。最后，实验验证了本文方案的高效性。

2　系统模型

表 1 列出了文中的符号变量及其代表的含义。

表 1　符号及其意义

符号	含义
AA	属性授权机构统称
AA_j	第 j 个属性授权机构
Sig	所有 AA 属性签名
PK	系统公钥
VK_j	每个 AA_j 生成的认证密钥
SK_j	每个 AA_j 生成的私钥组件
k_x	访问结构树中内部节点的门限值
VK	每个 AA_j 生成的认证密钥的集合

2.1　系统模型和框架

基于多授权机构 ABE 的跨域数据访问控制方案系统模型如图 1 所示，系统模型由以下 5 部分组成：数据属主(DO)、用户、授权机构(AA)、认证机构(CA)、云服务器。

数据属主负责根据安全策略定义访问结构树，并用访问结构树对数据进行加密，然后将密文传递给云端服务器。此外，DO 将生成部分私钥组件，用于防止 AA 间的联合攻击，并通过安全信道将私钥组件发送给用户。用户可以自由获取服务器上的密文文件，但当且仅当其拥有的属性满足密文访问结构树时才能够解密密文，当某个用户撤销时，云端需要对密文进行重加密，并进行部分密钥组件的重新分发。AA 作为属性生成和授权机构，负责向授权用户和数据属主分发属性以及生成部分属性私钥组件。CA 是一个可信的中央认证机构，在认证初始阶段负责生成一系列的公共参数，所有想成为属性授权机构的实体都会生成一个与公共参数相关联的认证密钥(VK)。云服务器是不可信的存储介质，主要用于存储用户数据。用户的属性集由于某种原因需要变动，当属性撤销时，AA 会发送一个关于属性撤销的消息请求，然后云服务器重新加密密文。

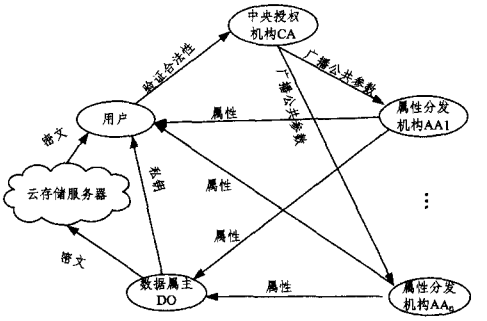


图 1　系统模型

3　基于 MA-ABE 的跨域数据访问控制

3.1　访问结构树定义

本文采用树形访问控制结构，设 Γ 是一棵代表访问控制结构(即安全策略)的树，树的每一个非叶子节点代表一个由其孩子节点和逻辑运算符组成的门限关系函数。如果 num_x 表示节点 x 的孩子节点数目，k_x 表示门限，当 $k_x=1$ 时，阈值门表示或门；当 $k_x=num_x$ 时，阈值门表示与门。每个叶子节点表示一个属性并且其 $k_x=1$。

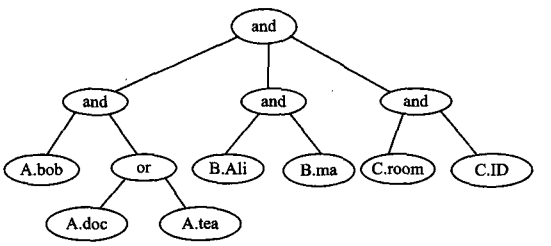


图 2　访问结构树

本方案所构造的访问结构树有以下几个特征：

1)访问结构树是一棵 n 叉树，其根节点必须是“and”节点，代表多个授权机构联合参与某个文件的授权。如果根节点是“or”节点，则表示为单授权机构。

2)根节点下的每棵子树代表各授权机构的访问结构树，每个授权机构的访问控制策略在各子树中定义。为了方便描述，要求每棵子树下定义的属性都由同一授权机构分发。

3)叶子节点代表各个授权机构所分配的属性，每个属性的第一个字节用于标识分配该属性的授权机构。

3.2　用户的属性获取

系统初始化阶段由 CA 生成一系列公共参数，所有想成为 AA 的实体就会生成一个与公共参数相关联的 VK。用户从不同的 AA 中获取属性私钥。假设访问结构树如图 2 所示，用户可以从拥有 VK1 的授权机构 A 中获取属性 A. tea 以及拥有 VK2 的属性授权机构 B 中获取属性 B. ma。具体过程如下。

首先用户分别从两个 AA 获取关于属性 A. tea 的 Sig(1)以及属性 B. ma 的 Sig(2),其中 Sig(1)利用 VK1 生成,Sig(2)利用 VK2 生成。然后用户将获得的签名以及 VK1 和 VK2 提交给 CA,经过 CA 认证后,授权机构的合法性得以验证。CA 为用户验证 AA 合法性之后便不再参与任何操作,因此系统的安全性与 CA 的关联性不大,避免了由于使用 CA 所引入的安全隐患。最后,合法的 AA 向用户分发属性以及属性私钥。

3.3 访问控制方案

MA-ABE 跨域数据安全访问控制方案包括以下 4 个基本算法:Setup,Encrypt,KeyGen,Decrypt。

(1)Setup:选择阶为素数 p 的乘法群 G,g 是 G 的生成元,构造双线性映射 $e:G\times G\rightarrow G_T$,随机选取 $\alpha,\eta\in Z_p$,生成公钥 $PK=(g,G,g^{\eta},e(g,g)^{\alpha})$,$MSK=(g^{\alpha},\eta)$。

(2)Encrypt(PK,M,Γ):DO 利用系统的公钥和访问结构树对消息 M 进行加密。首先 DO 根据每个 AA 分发的属性制定一棵访问结构树,随机选取 $s,\eta,\rho\in Z_p$,使得访问结构树根节点的值为 $q_r(0)=s$。采用自上而下的方式为访问结构树中的每个叶子节点分配私钥值$\frac{q_y(0)}{\rho}$,利用每个叶子节点的私钥值进行加密。令 Y 是树 Γ 叶子节点的集合。密文构建如下:

$$CT=(\Gamma,C'=M\cdot e(g,g)^{\alpha s},C=g^{\eta s},\forall y\in Y,C_y=H(att(y))^{q_y(0)/\rho},C_y'=g^{q_y(0)/\rho}) \tag{1}$$

(3)KeyGen(MSK,S):私钥的生成由 DO 和 AA 共同完成。

1) DO 随机选取 $\lambda\in Z_p$,生成私钥组件如下:$D=g^{(\alpha-\lambda)/\eta}$,并将 D 和参数 λp 通过安全信道发送给用户。由于每个用户私钥中的 λp 值不同,因此可以防止用户之间的联合攻击。

2) 每个 AA 随机选取 $r_i\in Z_p$,对于任意的属性 $k\in S_j$,生成相应的属性私钥组件:

$$SK_j=(\forall k\in S_j,V_i=g\cdot H(i)^{r_i},L_i=g^{r_i}) \tag{2}$$

其中,$j=1,\cdots,n$;S_j 代表第 j 个 AA 分发给用户的属性集合。令每个 AA 将 SK_j 通过安全信道发送给用户。

(4)Decrypt(CT,SK):本方案的解密操作分为两部分:CSP 端解密和用户解密。CSP 只负责数据的部分解密,并将解密的结果发送给用户,虽然 CSP 能够得到部分结果,但是仍不能获取最终的明文数据,因为关键参数 λp 只有 DO 和用户知道,由此保证了数据的安全性。具体操作如下:

1)CSP 解密(DTK):用户接收到 DO 和每个 AA 发送的密钥后,将私钥组件 $SK_{j\in(1,\cdots,n)}$ 发送给 CSP 端,CSP 端收到用户发送的私钥组件后,运行解密算法,输入包含访问结构树的密文以及私钥 $SK_{j\in(1,\cdots,n)}$,通过递归算法从下向上进行解密操作,生成解密所需的参数。令 $i=attr(y)$,$attr(y)$ 表示叶子节点 y 相对应的属性值,如果 x 是叶子节点且 $x\in S$,则有:

$$\begin{aligned}\frac{e(V_i,C_x')}{e(L_i,C_x)}&=\frac{e(g\cdot H(i)^{r_i},g^{q_y(0)/\rho})}{e(g^{r_i},H(i)^{q_y(0)/\rho})}\\&=\frac{e(g,g^{q_y(0)/\rho})e(H(i)^{r_i},g^{q_y(0)/\rho})}{e(g^{r_i},H(i)^{q_y(0)/\rho})}\\&=e(g,g)^{\frac{q_y(0)}{\rho}}\end{aligned} \tag{3}$$

如果 x 不是叶子节点,对于节点 x 的所有孩子 z,将解密输出的结果记为 F_z,令 S_x 是任意的大小为 k_x 的孩子节点 z 的集合。如果这样的集合不存在,则节点不满足解密,函数将返回 $\perp$。计算:

$$\begin{aligned}F_x&=\prod_{z\in S_x}F_z^{\Delta_{i,S_x'(0)}},\text{where}\begin{cases}i=index(z)\\S_x'=(index(z):z\in S_x)\end{cases}\\&=\prod_{z\in S_x}(e(g,g)^{q_z(0)/\rho})^{\Delta_{i,S_x'(0)}}\\&=\prod_{z\in S_x}(e(g,g)^{q_{parent(z)}(index(0))/\rho})^{\Delta_{i,S_x'(0)}}\\&=\prod_{z\in S_x}(e(g,g)^{q_x(i)/\rho})^{\Delta_{i,S_x'(0)}}\\&=e(g,g)^{q_x(0)/\rho}\end{aligned} \tag{4}$$

算法调用生成访问结构树的拉格朗日插值函数,如果属性集合 S 满足访问结构树 Γ,则 $DTK=e(g,g)^{\frac{q_R(0)}{\rho}}=e(g,g)^{\frac{s}{\rho}}$。CSP 计算出 DTK 后并将其发送给合法用户。

2)用户解密:用户接收到 CSP 发送的 DTK 后,利用 DO 发送的私钥再次进行解密,计算:

$$\begin{aligned}(e(D,C)\cdot(DTK)^{\lambda\rho})&=e(g^{(\alpha-\lambda)/\eta},g^{\eta s})\cdot e(g,g)^{\lambda\rho s/\rho}\\&=e(g,g)^{\alpha s}\end{aligned} \tag{5}$$

最后得到:

$$M_0=\frac{C'}{e(g,g)^{\alpha s}} \tag{6}$$

4 基于 MA-ABE 的撤销方案

在本方案中,撤销方案包含两种情况:用户撤销和属性撤销。本方案中的用户撤销是指对特定权限用户的撤销,可以是一个或者多个用户。用户被撤消后,并不影响云服务系统中其他用户的访问控制权限。属性撤销是指某个属性撤销后,该属性将失去对某个文件的访问控制权限。即在原来的访问结构树中,该属性被删除,数据属主需要重新定义新的访问结构树。例如,在图 2 中 A. doc 被撤销,则包含 A. doc 的用户无法访问资源,但包含 A. tea 的用户仍然可以访问,这就需要将更新后的解密参数重新发送给含有 A. tea 的用户。

4.1 用户撤销

在云存储系统的现实应用中,如果用户购买的服务已结束或者用户有恶意行为,DO 可以撤销任何访问公有资源的用户。被撤销的用户将不能再访问此公有资源,因此 DO 需要重加密数据以防止非法用户获取明文信息。用户撤销后需要改变用于解密的公钥和私钥以及密文。整个用户撤销过程在保证安全的前提下进行,CA 只需要更新公共参数,DO 根据更新的参数重新加密密文,能够实现有效的用户撤销,整个撤销过程由 DO、用户、CSP 和 CA 共同完成。

图 3 中用户 u_1 被撤销的具体过程如下。

(1)当用户 u_1 不再拥有某个文件的访问权限时,DO 需要撤销 u_1,首先 DO 会向 CA 发送关于用户 u_1 撤销的信息。

(2)CA 接收到 DO 发送的消息后,更新公共参数 PK 和 MK,生成新的参数 α_1,$e(g,g)^{\alpha_1}$ 以及主密钥 (g^{α_1},η),并公布参数 $e(g,g)^{\alpha_1}$,将 (g^{α_1},η) 单独发送给 DO,保存 α_1。

(3)CA 公布更新的参数后,DO 根据新的参数更新密文和私钥组件,利用 $e(g,g)^{\alpha_1-\alpha}$ 对数据文件重新加密,生成新的

密文$\widetilde{C'}=Me(g,g)^{\alpha_1 s}$，将生成的新密文发送至云服务器端存储。同时利用 g^{α_1} 生成 $g^{\alpha_1-\alpha}$，并将 $g^{\alpha_1-\alpha}$ 通过安全信道发送给未被撤销的用户。

(4)CSP 一旦收到 DO 发送的更新后的密文，就立刻更新存储的密文信息。

(5)未撤销用户接收到 DO 发送的 $g^{\alpha_1-\alpha}$ 后，立刻更新自己的私钥组件，具体更新操作如下：$g^{\alpha_1-\alpha}\cdot D=g^{(\alpha_1-\lambda)/\eta}$，如果未被撤销的用户需要继续访问该数据文件，首先需要从云服务器端下载更新后的密文数据，然后利用更新后的私钥以及 AA 之前发送的 DTK 进行解密。具体的解密过程如下：

$$(e(D,C)\cdot(DTK)^{\lambda\rho})=e(g^{(\alpha_1-\lambda)/\eta},g^{\eta s})\cdot e(g,g)^{\lambda\alpha s/\rho}$$
$$=e(g,g)^{\alpha_1 s} \tag{7}$$

最后计算得到：

$$M_0=C'/e(g,g)^{\alpha_1 s} \tag{8}$$

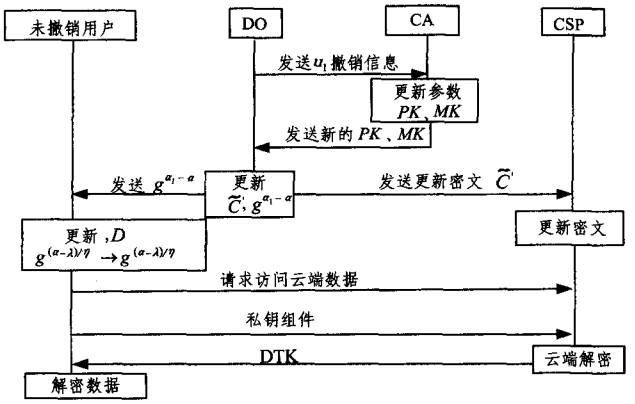


图 3　用户撤销流程图

本方案中，用户撤消的具体更新操作如图 4 所示。

/ * 撤销用户 * /

CA：

$\alpha\rightarrow\alpha_1$，$e(g,g)^{\alpha}\rightarrow e(g,g)^{\alpha_1-\alpha}$

MK'：$(g^{\alpha_1-\alpha},\eta)$

CA 公布 $e(g,g)^{\alpha_1}$，保留 α_1，并将 MK 发送给 DO。

DO：

一旦接收到 CA 公布的 $e(g,g)^{\alpha_1}$，DO 将更新密文并更新私钥组件。

/ * 私钥组件更新 * /

$D_1=g^{(\alpha_1-\lambda)/\eta}$

/ * 密文 * /

$\widetilde{C'}=Me(g,g)^{\alpha_1 s}$

图 4　用户撤销

4.2　属性撤销

属性撤销是云存储中一项非常具有挑战性的任务，由于撤销操作会引起大量用户的属性密钥和用于加密数据的公钥更新。传统的解决方法是数据属主需要在属性撤销后为每个仍拥有解密权限的合法用户生成新的密钥，这样大大增加了数据属主的计算代价并且十分浪费资源。

本文采用代理重加密技术，提出一种更为有效灵活的属性撤销方案，DO 负责生成代理重加密密钥(Proxy Re-Encryption，PRE)，并将 PRE 发送给 CSP，将大部分的重加密计算代价转移给 CSP，能够实现系统属性级的权限撤销。不失一般性，假设 AA_k 管理的属性 i_k 被撤销，假设属性 i_k 是访问数据时必不可少的一个属性，撤销后，认为即使用户拥有其他所有属性但仍不满足访问结构树的要求，那么拥有属性 i_k 的用户群将不能继续访问数据。

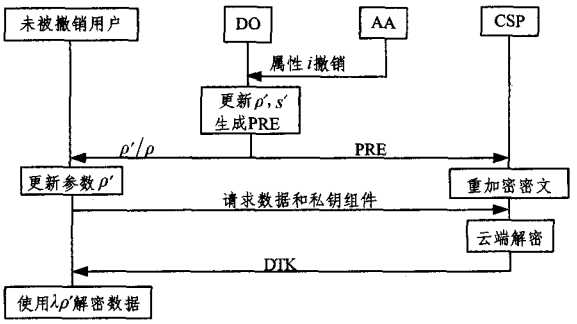


图 5　属性撤销流程图

图 5 中具体的属性撤销操作如下：

(1)首先，AA_k 通知 DO 撤销属性 i_k。

(2)在接收到 AA_k 发送撤销的属性 i_k 的消息后，DO 首先更新访问结构树，重新选取随机数 s'，$\rho'\in Z_p$，参数 ρ' 用于抵抗 CA 和 CSP 之间的联合攻击。根据新访问结构树从上向下为每个叶子节点分配私钥值 $q_y'(0)$。计算代理重加密密钥 PRE，并将 PRE 发送给 CSP，同时将解密关键参数发送给未撤销用户。

(3) CSP 接收到 DO 发送的 PRE 后，利用代理重加密技术更新密文，生成新的密文组件 $C_{y,1}$ 和 $C'_{y,1}$，其中 $y\in Y\backslash\{i\}$。CSP 存储更新后的密文。

(4)未被撤销的用户需要访问数据时，首先给 CSP 发送一个访问数据请求，然后将自己拥有的私钥组件发送给 CSP，CSP 根据用户发送的私钥组件计算得出 DTK，并将 DTK 发送给用户。

(5)用户接收到 CSP 发送的 DTK 后，再根据 DO 发送的参数 ρ'/ρ 计算获得明文信息，如果用户的属性集满足密文相关的访问策略，则用户可以解密。具体的撤销操作如图 6 所示。

/ * 属性撤销 * /

AA：首先 AA_k 撤销属性 i_k，并向整个系统广播撤销消息。

DO：DO 会重新选择私钥值 s'，$\rho'\in Z_p$，生成代理重加密密钥 PRE。

$$PRE=(PRE_0=\frac{M_1}{M}e(g,g)^{\alpha(s'-s)},PRE_1=s'/s,\forall y\in Y\backslash\{i\},$$
$$PRE_{2,y}=\frac{q_y'(0)-q_y(0)}{\rho'})$$

CSP：一旦接收到 DO 发送的 PRE 和私钥组件，CSP 便会更新密文，重加密操作如下：

$$CT'=(T',\widetilde{C'}=\widetilde{C}\cdot PRE_0,C'=(C)^{PRE_1},\forall y\in Y\backslash\{i\},C_{y,1}=$$
$$(C_y)^{PRE_{2,y}},C_{y,2}=(C_y')^{PRE_{2,y}},\forall i,C_{i,1}=C_i,C_{i,2}=C_i')$$

CSP 重新加密密文后，根据用户发送的私钥组件执行部分解密操作，并将计算出的 DTK 发送给用户。具体过程如下：

$$F=\frac{e(V_i,C_{y,1})}{e(V_i,C_{y,2})}=e(g,g)^{q_y'(0)/\rho'},DTK=\prod_{i\in S'}F=e(g,g)^{s'/\rho'}$$

用户：一旦接收到云端发送的 DTK，则进行解密运算。

$$A=e(D,C')=e(g^{(\alpha-\lambda)/\eta},g^{\eta s'})=e(g,g)^{(\alpha-\lambda)s'},M=\frac{\widetilde{C'}}{A\cdot(DTK)^{\lambda\rho'}}$$

图 6　属性撤销

5 安全性分析

5.1 数据机密性

假设加密数据文件的对称加密算法是安全的，那么机密性主要取决于密钥密文的安全性。本文的数据密文是利用改进的 CP-ABE 算法进行加密生成的，而文献[2]已经证明该算法是安全的。如果用户的属性集不能满足访问策略，则不能进行解密，当一个用户从满足访问策略的属性集中撤销时，除非剩余的属性集能够满足访问策略，否则将失去继续访问数据的能力。由于云服务器会为了某种利益而泄露用户数据，因此云服务器是不可信机构。当属性撤销时，云服务器负责对数据进行重加密，云服务器只知道秘密数 s'，并不知道数据属主加密文件所采用的私钥值 s。因此，云服务器也不能解密。另一种攻击来自认证机构 CA，CA 只负责广播公共参数以及验证授权机构的合法性，并不负责属性私钥的生成。因此 CA 也不能解密密文。因此抗服务器的联合攻击和 CA 的攻击也得到了保证。

5.2 抗联合攻击

本文的方案能抵抗用户的联合攻击和授权机构之间的联合攻击。

证明：在本方案中，为了能够解密，攻击者必须得到 $e(g,g)^{\alpha s}$，攻击者将密文组件 C 和私钥组件 D 进行双线性配对得到 $(e(g,g)^{(\alpha-\lambda)/\eta})$，然而计算得到的结果中仍然会有 $e(g,g)^{\lambda s}$。对于非法用户来说，每个用户的属性集不满足访问结构树，无法独立进行解密。当多个非法用户进行联合攻击时，由于都是非法用户，DO 只会将私钥组件 $D=g^{(\alpha-\lambda)/\eta}$ 以及参数 λ 发送给合法用户，他们无法得到可以解密密文的私钥组件 $D=g^{(\alpha-\lambda)/\eta}$ 以及参数 λ，因此即使非法用户通过计算获得 $e(g,g)^{s}$，也无法进行解密。对于加密文件，只要有用户撤销，DO 就会重新加密明文，得到新的密文 $Me(g,g)^{\alpha' s}$、新的私钥组件 $D=g^{(\alpha'-\lambda')/\eta}$ 以及参数 λ'，而已撤销用户只拥有以前的 $D=g^{(\alpha-\lambda)/\eta}$ 和参数 λ，用户通过计算只能得到 $e(g,g)^{s}$ 而不知道 λ'，所以用户不能解密，因此本方案能够抵御用户的联合攻击。对于授权机构来说，每个授权机构只负责生成属性私钥组件，即使多个服务器进行联合攻击，授权机构执行解密操作只能得到 $e(g,g)^{s}$，只有 DO 和合法用户知道参数 λ，所以服务器不能获得 $e(g,g)^{\lambda s}$，因此无法计算解密私钥，从而阻止了授权机构之间的联合攻击。

5.3 前向和后向安全性

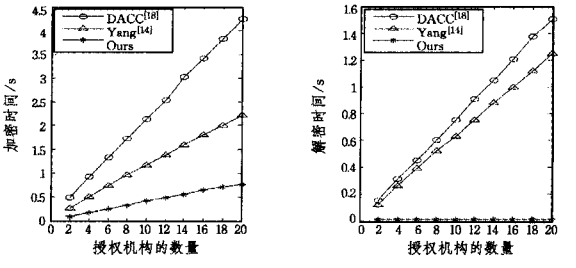
本方案能够实现前向和后向安全性，当新的用户加入系统时，此用户拥有一系列满足访问结构树的属性，首先，DO 选择随机数 $s',\rho'\in Z_p$，生成 PRE，并将 PRE 发送给 CSP，CSP 根据 PRE 更新相应的密文组件，对密文重新加密。即使新用户拥有的属性满足访问结构树，但是新用户在解密过程中只能计算出 $e(g,g)^{\alpha s'}$，并不能得出 $e(g,g)^{\alpha s}$，不能获得以前的明文信息，因此此方案能够实现后向安全性。同时，当某个用户（拥有的属性满足访问策略）需要撤出系统时，此后该用户失去访问权限，CSP 同样会根据 PRE 更新相应的密文组件，对密文重新加密。即使用户能够获得被撤销权限后的密文，但是仍不能解密，因为被撤销的用户只能计算出 $e(g,g)^{\alpha s}$，并不能根据自己拥有的属性计算出 $e(g,g)^{\alpha s'}$，因此共享数据的前向安全性同样能够得到保证。

6 性能分析

6.1 计算代价

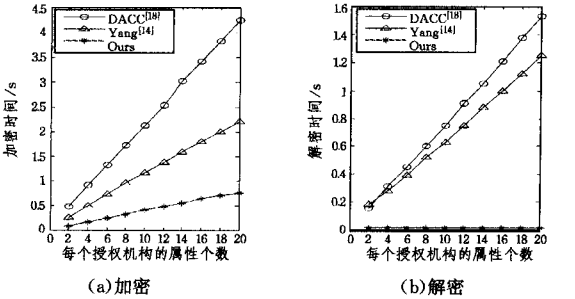
下面将从解密时间与加密时间两个方面对文献[18]、文献[14]和本文的方案进行比较。本文方案采用的实验环境为 Inter(R) Core(TM) i3 CPU 主频 3.19GHz，内存 2GB，VMware Workstation 虚拟机上的 Centos 系统，实验代码是基于 cpabe-0.11 库[19]改进的，使用 PBC 库，其版本为 0.5.12，对称加密使用基于 openssl-1.0.0c 库的 128 位 AES 加密算法。

图 7(a)描述了随着 AA 个数的增加，加密时间的变化，图 7(b)描述了随着 AA 个数的增加，解密时间的变化，其中每个 AA 所拥有的属性为 5 个。图 8(a)描述了加密时间与每个授权机构属性个数的关系，图 8(b)描述了解密时间与每个授权机构的属性个数的关系，其中 AA 的个数也为 5。



(a)加密 (b)解密

图 7 加密时间和解密时间与授权机构个数的关系比较



(a)加密 (b)解密

图 8 加密时间和解密时间与每个授权机构的属性个数的关系比较

从图 9 中可以看出，与文献[18]相比，本文的方案在系统加解密效率方面有了较大的提高。虽然本文方案在重加密方面的性能与文献[14]相比差不多，但解密的性能却有了较大的提高。因此，综合而言，本文提出的方案能够进一步提高多授权机构下数据加解密的效率。

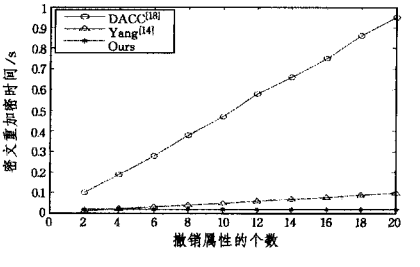


图 9 重加密

6.2 通信代价

本方案主要比较更新密文和密钥的计算代价，当用户撤

销时，文献[18]的 DACC 方案中不需要更新用户的解密密钥，但需要为每个非撤销用户更新密文，特别是未被撤销用户数量较多的情况下，DO 的计算代价较高。在保证数据信息安全的情况下，本方案只需要更新 1 个密文组件和 1 个私钥组件，特别是在云存储环境中撤销多个用户时，本方案的计算效率明显提高。

表 2　用户撤销的计算代价

操作	Rju's DACC[18]	Our shceme
Key Update	N/A	$\|p\|$
Ciphertext Update	$(n_{c,x} \cdot n_{non,x}+1)\|p\|$	$\|p\|$

其中，$n_{non,x}$ 代表合法用户拥有撤销属性 x 的数量，$n_{c,x}$ 代表与撤销属性相关的密文组件数量，$|p|$ 代表 G 和 G_T 元素的点个数。

由表 3 可以看出，本方案属性撤销的主要通信代价在于 DO，虽然文献[12]中 DO 的计算代价比本方案低，但是 AA 的通信代价、密钥更新通信量以及属性更新通信量比较高。

表 3　属性撤销的计算代价

方案	文献[12]	Ours
DO	N/A	$l+2$
AA	$(3+n_{non,x}) \cdot n$	N/A
密钥更新	$n \cdot n_{non,x}$	N/A
属性更新	n	N/A

其中，l 表示与密文相关的属性个数，n 表示被撤销属性的个数。

结束语　本文提出了一种高效的 MA-ABE 方案，该细粒度访问控制具有用户属性级的撤销能力。当用户属性撤销时，云服务器利用代理重加密技术对密文组件进行重加密，大大降低了数据属主的计算代价。本方案采用树访问策略，撤销时的计算复杂度与撤销属性的数量相关，计算效率较高。此外，本方案具有数据安全性及抵抗联合攻击的能力。安全和性能分析表明本方案是安全的，并且权限撤销时效率很高。但是本方案在降低 DO 计算代价的同时会增加云端的计算量，因此如何在降低 DO 的计算量的同时也使云端的计算量降低是今后需要进一步解决的问题。

参 考 文 献

[1] SAHAI A, WATERS B. Fuzzy identity-based encryption[C]// Proc of Advances in Cryptology-EUROCRYPT' 05. Aarhus, Springer Berlin Heidelberg, 2005: 457-473.

[2] BETHENCOURT J, SAHAI A, WATERS B. Ciphertext-policy Attribute-based Encryption[C]// Proc of IEEE Symposium Security and Privacy. Berkeley, CA, IEEE, 2007: 321-334.

[3] WATERS B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization[C]// Proc of PKC'11. Taormina, Italy, Springer Berlin Heidelberg, 2011: 53-70.

[4] GOYAL V, PANDEY O, SAHAI A, et al. Attribute-based encryption for fine-grained access control of encrypted data[C]// Proc of ACM Conference on Computer and Communications Security. Virginia, ACM, 2006: 89-98.

[5] HUR J, 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.

[6] CHEUNG L, NEWPORT C. Provably secure ciphertext policy abe[C]// CCS'07. New York, NY, USA: ACM, 2007: 456-465.

[7] LIANG X H, LU R D, et al. Ciphertext-policy At tribute Based Encryption with Efficient Revocation[R]. Technical Report, University of Waterloo, 2010.

[8] CHASE M. Multi-authority attribute based encryption [C]// Proc of Cryptography Conference on Theory of Cryptography (TCC'07). Amsterdam, Springer Berlin Heidelberg, 2007: 515-534.

[9] LIN H, CAO Z F, LIANG X. Secure threshold multi-authority attribute-based encryption without a central authority[C]// Proc of International Conference on Cryptology. India, Springer Berlin Heidelberg, 2008: 426-436.

[10] LEWKO A, WATERS B. Decentralizing attribute-based encryption[C]// Proc of International Conference on the Theory and Applications of Cryptographic Techniques. Tallinn, Springer Berlin Heidelberg, 2011: 568-588.

[11] YU S C, WANG C, REN K, et al. Achiving secure, scalable, and fine-grained data access control in cloud computing [C]// Proceedings of IEEE INFOCOM 2010. San Diego, CA, 2010.

[12] YANG K, JIA X H, REN K. DAC-MACS: Effective Date Access Control for Multi-Authority Cloud Storage Systems[J]. IEEE Transactions on Information Forensics and Security, IEEE, 2013, 8(11): 1790-1801.

[13] LI J, REN K, ZHU B, et al. Privacy-aware attribute-based encryption with user accountability[M]// Lecture Notes in Computer Science, ISC'09. Springer, vol. 5735, 2009: 347-362.

[14] YANG K, JIA X H. Expressive, Efficient and Revocable Data Access Control for Multi-Authority Cloud Storage[J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(7): 1735-1744.

[15] YANG K, JIA X H. Attribute-based Access Control for Multi-Authority System in Cloud Storage[C]// Proc of International Conference on Distributed Computing Systems (ICDCS). Macau, IEEE, 2012: 536-545.

[16] EISSA T, CHO G H. A Fine Grained Access Control and Flexible Revocation Scheme for Data Security on Public Cloud Storage Service[C]// 2012 International Conference on Cloud Computing Technologies, Applications and Management (ICCCTAM). Dubai, 2012: 27-33.

[17] BENALOH J, LEKCHTER J. Generalized secret sharing and monotone functions[C]// Proc of Crypto'88, Lecture Notes in Computer Science. Springer-Verlag, Berlin, 1989: 213-222.

[18] SUSHMITA R, AMIYA N, IVAN S. DACC: Distributed Access Control in Clouds[C]// Proc of IEEE TrustCom. Changsha, IEEE, 2011: 91-98.

[19] BETHENCOURT J, SAHAI A, WATERS B. The cpabe toolkit [OL]. http://acsc.csl.sri.com/cpabe.