一种用于云存储的密文策略属性基加密方案

刘 帆, 杨 明
(解放军理工大学 指挥自动化学院 计算机系, 南京 210007)

摘 要: 云存储的应用环境中存在缺乏细粒度访问控制、密钥管理难度大、难以抵御合谋攻击等问题,为此提出了一种新的用于云存储的密文策略属性基加密(ciphertext-policy attribute-based encryption, CP-ABE)方案。通过引入由数据属主独立控制的许可属性,构建不同属性域的 CP-ABE 方案,能够防止云存储系统特权用户的内部攻击,使数据属主能完全控制其他共享用户对其云数据的访问。实验结果表明,该方案在提供安全性的同时能极大地提高用户属性撤销的效率。最后,对该方案进行了安全分析,并证明了该方案在 DBDH 假设下是 CPA,安全的。

关键词: 云存储; 访问控制; 属性基加密; 存储安全
中图分类号: TP393 **文献标志码**: A **文章编号**: 1001-3695(2012)04-1452-05
doi:10.3969/j.issn.1001-3695.2012.04.070

Ciphertext policy attribute based encryption scheme for cloud storage

LIU Fan, YANG Ming
(*Dept. of Computer, Institute of Command Automation, PLA University of Science & Technology, Nanjing* 210007, *China*)

Abstract: There exists some problems as the lack of fine-grained access control, low efficiency of key management and the vulnerability against collusion attack in the cloud storage environment, where there's a large number of users and files. This paper proposed a new CP-ABE scheme applied in cloud storage. It introduced the permission attributes managed by the data owners, and constructed different attribute field to resolve the security problem such as the inner attacks and data control. Several experiments show that revocation method for user attributes performs wih high efficiency while providing security. In addition, it proves the scheme is secure against the chosen-plaintext attack (CPA) under the decision bilinear diffie hellman(DBDH) assumption.

Key words: cloud storage; access control; attribute-based encryption; storage security

0 引言

云存储是近年来从云计算概念衍生和发展起来的一种数据外包存储服务技术,其依靠低成本、易于使用的接口和高可扩展性的商业优势得到了业内的广泛关注。然而,在使用便利的同时,云存储也引起了用户对于安全性的广泛担忧。据调查,超过 70% 的用户仍然不愿意将关键数据存储于自身控制域之外[1]。另外,Google Docs、The Linkup 等多家著名云服务商都曾先后出现过各种安全问题,并导致了严重的后果[2],安全隐患已成为云存储大规模普及的重要阻碍。

访问控制是实现用户数据机密性和进行隐私保护的重要手段,目前有很多云存储服务提供商(cloud storage provider, CSP)都提供简单的访问控制功能,这些安全机制完全依赖于 CSP 服务器端,但不能确保 CSP 在复杂的网络环境和多变的商业利益之中能够始终保证安全机制的有效性。云存储的外包存储服务模式导致了特权用户的存在,后者具有非授权访问用户数据的能力,易导致数据信息和隐私泄露等内部攻击问题。这样,云存储服务器端用户的敏感数据在遭受网络攻击威胁的同时,还需应对由非可信云服务器引起的数据非授权访问、信息或隐私泄露等问题。传统的加密手段虽然能够保护数据隐私,但增加了系统对用户细粒度访问控制的难度。

为避免特权用户非法访问用户的敏感数据,同时又能够实现在云存储环境中的细粒度访问控制,一种方法是引入以属性基加密(ABE)[3]为代表的加密访问控制技术。ABE 最早由 Sahai 等人在 IBE[4,5]的基础上提出的,能够对共享数据进行细粒度控制且降低了私钥存储和分发的工作量。但是基本的 ABE 无法支持灵活的访问控制策略。后来,Bethencourt 等人[6]提出了适用于访问控制类应用[7]的密文策略的属性基加密 CP-ABE 机制。最初的 CP-ABE[6]虽然属性操作灵活,但其安全性证明仅基于一般的群假设,且没有涉及属性撤销。之后 Cheung 等人[8]首次基于 DBDH 假设构建了 CPA 安全的 CP-ABE 机制,但该方案只能采用与门操作属性。为了能在 CPA 安全的假设下实现策略灵活的 CP-ABE 机制,Goyal 等人[9]和 Liang 等人[10]采用有界树结构,Lbraimi 等人[11]采用一般的访问树结构,消除了界限条件的约束[7]。

构建 ABE 的一个难点是其撤销方案的设计。其困难的原因主要有以下几点:a)ABE 机制本身的复杂性,密文和用户密钥与属性集相关联,密文中的访问策略由数据属主制定,而属性密钥由非可信的授权中心管理;b)用户组与属性集为多对多关系,撤销属性时会涉及其他众多用户,而撤销用户时会涉及用户使用的所有属性;c)撤销对象复杂,包括用户撤销、用

收稿日期: 2011-09-09; **修回日期**: 2011-10-15

作者简介: 刘帆(1985-),男,四川成都人,助理工程师,硕士研究生,主要研究方向为信息安全、存储安全(liufan1945@sina.com);杨明(1968-),男,教授,博士,主要研究方向为信息安全、群组通信.

户属性撤销和系统属性撤销。这些原因使得以往很多 ABE 方案不是无法实现撤销的要求就是虽然实现了撤销但在大型系统中可扩性较差。

第一个 ABE 密钥撤销方案由 Pirretti 等人[12]提出,但用户需要频繁地与授权中心交互以维护密钥。Bethencourt 等人[6]通过对密钥期限进行性能改进减少了通信方之间的通信负载,但密钥只能在到达期限时进行间接撤销,无法直接撤销。后来 Ostrovsky 等人[13]通过加入标签实现了用户的直接撤销。Lbraimi 等人[14]通过设计一个半可信第三方仲裁者实现了用户属性的直接撤销。这种引入仲裁者的机制[14,15]无须更新合法用户的密钥,并且减少了授权中心的工作量,但是必须保证仲裁者诚实在线。而 Yu 等人[16]基于代理重加密技术引入了半可信代理,但其在性能上有重大缺陷且代理必须保持在线。

ABE 在提供数据安全性的同时进行细粒度的访问控制,直接将 ABE 技术应用于云存储系统并不能很好地解决一些诸如效率和可扩展性的问题。例如:a)云存储系统涉及大量的用户和数据,加密、用户组和密钥的管理等机制都需要较高的可扩展性,直接应用 ABE 会导致效率不高的问题;b)属性/用户属性的撤销设计难度较大、实现起来效率不高;c)特权用户仍可获取用户敏感数据的内容,数据属主仍未达到对共享数据访问的完全控制。

为了解决这些问题,本文提出了引入由数据属主独立控制许可属性,构建不同属性域的 ABE 方案——ABE-MACCS(attribute-based encryption with multiple access control for cloud storage)。为实现数据属主对共享用户的直接撤销,将密文的访问结构由两部分复合构成不同的属性域,将属性划分为普通属性和许可属性两类。普通属性由授权中心负责管理,而许可属性由数据属主负责管理。用户只有在其持有的属性同时满足密文中的访问结构和许可属性的条件后才能成功解密数据。

1　预备知识

1.1　双线性配对

双线性配对是设计 ABE 加密方案时非常关键的工具之一。选取两个阶为一个大素数 p 的群 G_0 和 G_T,定义一个可有效计算的双线性映射 $e:G_0\times G_0\rightarrow G_T$,该映射必须满足:

a)双线性。一个映射 $e:G_0\times G_0\rightarrow G_T$ 具有双线性,当 $e(g^a,h^b)=e(g,h)^{ab}$,对于所有的 $g,h\in G_0$ 和所有的 $a,b\in Z_p$。

b)非退化性。存在 $g\in G_0$,使得 $e(g,g)\neq 1$。

其中,$e(*,*)$为对称操作,即 $e(g^a,h^b)=e(g^b,h^a)$。

1.2　复杂性假设

定义 1　双线性 Diffie-Hellman(BDH)参数生成器。若一个以安全参数 $k(k>0)$为输入的随机算法 Γ,在以 k 的多项式时间内运行,输出关于两个群 G_0 和 G_T 的描述、它们共同的素数阶 p 以及一个可有效计算的双线性映射 $e:G_0\times G_0\rightarrow G_T$,则称这个算法为一个 BDH 参数生成器。

定义 2　判定性双线性 Diffie-Hellman(DBDH)问题。令 G_0、G_T 和 e 为定义 1 中参数生产器 Γ 的输出,再令 g 为群 G_0 的生成元。DBDH 问题定义如下:给定$\langle g,g^a,g^b,g^c,A\rangle$(其中,随机元素 $a,b,c\in Z_p^*$,$Z\in G_T$),判断等式 $e(g,g)^{abc}=A$ 是否成立。

1.3　CP-ABE

CP-ABE 加密方案主要包括四个算法[6]:a)初始化算法 Setup,生成主密钥 MK 和公开参数 PK;b)密钥生成算法 KeyGen(MK,S),使用 MK、用户属性集 U 生成用户私钥 SK;c)加密算法 Encrypt(PK,M,T),使用 PK、访问结构 T 加密明文 M,输出密文 CT;d)解密算法 Decrypt(CT,SK),使用私钥 SK 解密密文 CT 得到明文 M。

1.4　IND-CPA

定义 3　CP-ABE 加密方案的选择明文攻击安全 IND-CPA。若任何 IND-CPA 攻击者在下面游戏中的优势 $\mathrm{Adv}_A(k)$ 都是可忽略的,那么称该 CP-ABE 方案是选择明文安全的。

通过一个挑战者 C 和攻击者 A 之间的攻击游戏来定义:

a)初始化。攻击者 A 选取一个挑战的访问结构 W,将其交给挑战者。

b)建立阶段。挑战者运行 Setup 算法,生产公钥 PK,将其交给攻击者 A。

c)查找阶段 1。对某个属性集所对应的属性密钥抽取查询,挑战者 C 运行属性密钥生成算法后,将生产的用户私钥交给攻击者 A。其中属性集不能满足访问结构 W。

d)挑战。攻击者 A 向挑战者 C 提交两个等长度的明文 M_1 和 M_2,以及一个挑战属性集。唯一的限制是:在查找阶段,攻击者 A 不曾查询过挑战属性集的属性密钥。挑战者 C 随机抽取一个比特 $u\in\{0,1\}$并利用 W 对其进行加密,生成密文 M_u 并发送给挑战者。

e)查找阶段 2。与查找阶段 1 一致。

f)猜测阶段。在这一阶段,攻击者 A 可以提出更多像查找阶段所提及的两类查询,不过不能对挑战属性密钥进行属性密钥抽取查询。

g)输出。最后,攻击者 A 输出一个猜测 $u'\in\{0,1\}$,若 $u'=u$,称其赢得了游戏。上述游戏中的攻击者 A 被称为选择明文(IND-CPA)攻击者。攻击者 A 在上述针对 CP-ABE 方案的攻击游戏中的优势定义为安全参数 k 的一个函数:$\mathrm{Adv}_A(k)=|\Pr[u'=u]-1/2|$。

2　ABE-MACCS 方案

本文方案主要应用于以云存储为代表的大规模分布式文件存储系统。在该场景中,数据属主通过将数据上传至文件系统中的存储服务器让其他共享用户进行存取访问,访问的规则由数据属主来制定。文件系统中的授权中心对共享用户的访问权限进行授权管理,访问策略由存储服务器具体执行。

2.1　方案模型

本文的方案模型主要描述了系统的整体结构,包括系统中各个模块的主要功能以及方案的基本安全假定。

2.1.1　系统模型

本文系统以云存储系统为模型,如图 1 所示。系统由以下几部分组成:数据属主、数量庞大的数据共享用户、云服务器簇群和授权中心。数据属主将数据加密后上传至云数据服务器,密文中新加入了由数据属主控制的包含用户许可数据的访问结构,并与传统的 CP-ABE 访问结构复合构成,用户只有同时满足两种结构才能解密密文。这使得数据属主能够控制别人访问自己的共享数据,也可以对共享用户执行直接撤销操作。授权中心根据共享用户提供的属性生成相应的属性私钥并分配给用户,共享用户为了获取数据内容,必须从云服务器上下载并解密数据。同时,授权中心也负责用户和属性的间接操

作。本文设定数据属主和共享用户并不会一直在线,云服务器会一直在线并提供数据存取服务,授权中心也会一直在线并负责认证和撤销等管理工作。数据属主在上传数据的同时也可制定共享数据的访问策略。

2.1.2 安全模型

假设云服务器在通常情况下会遵循本文设计的访问协议,但是因为云服务器提供的是不可信的外包服务,所以共享数据会面临来自云服务器的内部攻击。并且出于获利的考虑,云服务器可能会与一些恶意用户共谋合作。假定数据属主/共享用户与云服务器之间的通信在一些安全协议的保护下是安全的,协议的每一方都采用一对公私钥进行加密通信。

2.2 复合访问结构

访问结构 T 用“与”操作将传统的表示用户普通属性的结构树 T_A 与一种新型的表示用户 ID 的许可属性的结构树 T_{ID} 相连,如图 2 所示。左子树 T_A 是传统 CP-ABE 方案中的访问结构,其包含的普通属性集由授权中心负责管理。右子树 T_{ID} 包含由数据属主负责管理的属性集。树 T 的根节点是一个 “与” 门,只有当用户持有的属性同时符合 T_A 和 T_{ID} 才能正确解密密文。

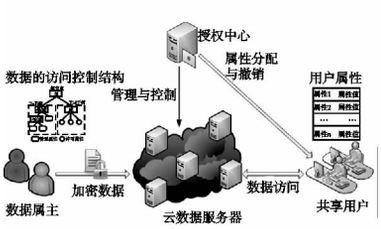

图1 系统模型

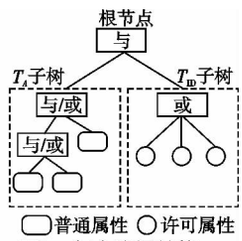

图2 复合访问结构

2.3 加密方案

1)Setup(k) 初始化算法 Setup 的输入为安全参数 k,算法首先随机选择生成元为 g、阶为 p 的双线性群 G_0 和双线性映射 $e:G_0\times G_0\rightarrow G_T$。随后产生全体属性集 $\Omega=\{a_1,a_2,\cdots,a_n\}$ 和全体用户身份集 $\Phi=\{b_1,b_2,\cdots,b_m\}$,以及一些随机元素 $t_1,t_2,\cdots,t_n,t_{n+1},t_{n+2},\cdots,t_{n+m}\in Z_p^*$。算法随机选取参数 $\alpha_1,\beta_1\in Z_p$ 生成第一主私钥 $MK_1=(\beta_1,g^{\alpha_1})$ 和第一主公钥 $PK_1=\{G_0,g,h_1=g^{\beta_1},e(g,g)^{\alpha_1}\}$,并随机选取参数 $\alpha_2,\beta_2\in Z_p$ 生成第二主私钥 $MK_2=(\beta_2,g^{\alpha_2})$ 和第二主公钥 $PK_2=\{G_0,g,h_2=g^{\beta_2},e(g,g)^{\alpha_2}\}$。

2)KenGen1(MK_1,A_u) 密钥生成算法 KeyGen1 将用户 u 的属性集 U 作为输入。算法首先选择随机数 $r_1\in Z_p^*$,计算 $d_1=g^{\alpha_1-r_1}$,再为 U 中的每个属性 $a_j\in U$ 随机选择 $r_j\in Z_p^*$,计算 $d_j=g^{rt_j^{-1}}$。输出用户私钥第一部分:$SK_1=(d_1=g^{\alpha_1-r_1},\forall j\in A_u:d_j=g^{rt_j^{-1}})$。

3)KenGen2(MK_2,t_k) 密钥生成算法 KeyGen2 将用户身份 b_k 作为输入,并输出一个与用户身份对应的密钥 SK_2。算法首先选择随机数 $r_2\in Z_p^*$,计算 $d_2=g^{\alpha_2-r_2}$,再为用户身份 b_k 随机选择 $r_k\in Z_p^*$,计算 $d_k=g^{rt_n^{-1}t_k}$。输出用户私钥第二部分 $SK_2=(d_2=g^{\alpha_2-r_2},d_k=g^{rt_n^{-1}t_k})$。

在执行完算法 KenGen1 和 KenGen2 之后,用户 u 的一个完整私钥 $SK_U=(SK_1,SK_2)$,生成完毕。

4)Encrypt(PK,M,T) 加密算法 Encrypt 依据访问结构 T 加密消息 M,主要分为生成复合访问结构 T 以及计算密文主体。

(1)生成复合访问结构 T 部分。算法首先选取两个随机数 $s_1,s_2\in Z_p$,分别分配为子树 T_A 和 T_{ID} 的根节点,再用递归的方式为 T_A 中每个非叶节点分配共享秘密:

已知该节点值为 s_1,

①若该节点是“与”操作且其子节点未被标记,标记该节点,对其每个子节点分配一个随机值 $s_i(1\leqslant s_i\leqslant p-1)$,将最后一个子节点的值设为 $s_t=s_1-\sum_{i=1}^{t-1}s_i \bmod p$;

②若该节点是“或”操作,标记该节点,并将其所有的子节点的值设为 s_1。

T_{ID} 只有两层结构,将其所有的子节点的值设为 s_2。

对于 T_A 中的每个叶节点 $a_{j,i}\in T_A$,计算 $c_{j,i}=g^{t_j s_i}$,i 为该属性在树中的序号。对于 T_{ID} 中的每个叶节点 $b_{j,i}\in T_{\mathrm{ID}}$,计算 $k_{j,i}=g^{t_{n+j}s_i}$,i 为该属性在树中的序号。

(2)计算密文主体部分。分别计算 $c_1=g^{s1}$,$c_2=g^{s2}$,$\tilde{c}=Me(g,g)^{\alpha_1 s_1}e(g,g)^{\alpha_2 s_2}$。最终输出密文:$ct=(T,\tilde{c},c_1,c_2,\forall a_{j,i}\in T_A:c_{j,i},\forall t_{j,i}\in T_{\mathrm{ID}}:k_{j,i})$。

5)Decrypt($ct,SK_U=(SK_1,SK_2)$) 算法包括两部分,分别通过 T_A 和 T_{ID} 解密部分密文:a)如果用户属性集 U 不满足 T_A,算法返回 $\perp$,否则选取一个满足 T_A 的最小子集 U' 计算 $\prod_{a_j\in U'}e(c_{j,i},d_j)=\prod_{a_j\in U'}e(g^{t_j s_i},g^{rt_j^{-1}})=e(g,g)^{rs_1},e(c_1,d_1)\cdot e(g,g)^{rs_1}=e(g^{s1},g^{\alpha_1})$,并返回中间结果 $c_\Delta=\dfrac{\tilde{c}}{e(g^{s1},g^{\alpha_1})}=Me(g,g)^{\alpha_2 s_2}$;b)如果用户身份 b_k 不满足 T_{ID},算法返回 $\perp$,否则进行计算得到明文:$\dfrac{c_\Delta}{e(k_{j,i},d_k)}=\dfrac{c_\Delta}{e(g^{s2},g^{\alpha_2})}=M$。

2.4 撤销方案

ABE-MACCS 方案的撤销工作主要包括用户撤销、用户属性撤销以及系统属性撤销[7]三个操作。用户撤销由数据属主直接执行,该操作只涉及被撤销用户,对其余合法用户没有影响。用户属性撤销由授权中心周期性间接执行,主要撤销对象用户持有的部分属性,不影响其他合法用户持有的相同属性。系统属性撤销由数据属主或者授权中心直接执行,所有用户手中的属性通过系统周期性地更新密钥来移除。

加密数据包含的复合访问结构将访问策略分成了 T_A 和 T_{ID} 两部分,用户每次更新密文时可以选择更新 T_A 或者 T_{ID} 而不需要更新全部的访问策略,提高了密文更新的效率。同时,数据属主可以直接撤销用户和系统属性。

2.4.1 用户撤销

用户撤销由数据属主直接执行,使其能够直接控制自己的共享数据的访问。UserRevoke 算法将第二主私钥 MK_2 和撤销用户 b_k 作为输入。执行撤销时,将被撤销用户的身份 b_k 所对应的许可属性在 T_{ID} 中去掉,重新生成新的策略结构 $T_{\mathrm{ID}}'=T_{\mathrm{ID}}-\{b_k\}$。由于算法并没有更改 T_{ID} 根节点 s_2 的值,所以无须重新计算密文主体。

```
UsrRevoke(MK2, bk)
{ renew(ct): TID' = TID − {bk};
//更新密文中的复合访问结构的 TID 子树
}
```

2.4.2 用户属性撤销

用户属性撤销由算法执行,算法的输入参数为被撤销的用户对象集合 U,以及被撤销的属性 a_k。由于撤销用户属性不可避免地会重新生成新的用户私钥 SK_{t_i} 并分发给用户组中所有的合法用户,为系统带来不小的计算负担。本文采用一种懒撤销机制,密钥的更新工作由定时器 timer( ) 触发并按周期执行。

定时器 timer( )触发以后,系统会执行 LazyRevoke 算法:首先更新主公钥 PK_1 和 PK_2,以及密文主体部分 $\tilde{c}=Me(g,g)^{\alpha_1 s_1}e(g,g)^{\alpha_2 s_2}$ 和复合访问结构的 T_A 子树:$T_A'=T_A-\{a_k\}$;接着更新合法用户组 $\Phi'=\Phi-\{U\}$ 并为组 Φ' 中的每个用户生成新的属性密钥 SK_{a_i} 并发送。

```
UsrAttriRevoke(U,a_k)
{ if(timer(T) triggered)
   LazyRevoke( );//执行间接撤销 }
LazyRevoke( )
{ renew(PK_1,PK_2);//更新主公钥
renew(ct):
ct=(T,c~,c_1,c_2,∀a_{j,i}∈T_A:c_{j,i},∀t_{j,i}∈T_ID:k_{j,i});//更新密文
Φ'=Φ-{U}//更新合法用户组
renew(SK_{a_i}):a_i∈Φ';//更新相关用户私钥
send(SK_{a_i})//发送用户私钥 }
```

2.4.3　系统属性撤销

系统属性撤销的过程与用户撤销类似,由数据属主或授权中心直接执行。GlobalAttriRevoke 算法将第一主私钥 MK_1 和被撤销的系统属性 a_k 作为输入。执行撤销时,只需更新密文中的复合访问结构的子树 T_A 而不用对其余合法用户发布新私钥。执行撤销时,算法 GlobalAttriRevoke( )将被撤销属性在 T_A 中去掉,重新生成新的策略结构 $T_A'=T_A-\{a_k\}$,由于算法并没有更改 T_A 的根节点 s_1 的值,所以无须重新计算密文主体。

```
GlobalAttriRevoke(MK_1,a_k)
{ renew(ct):T_A'=T_A-{a_k};
//更新密文中的复合访问结构的 T_A 子树}
```

相对于以往的 ABE 撤销机制,本文方案的优势如下:a)可以让数据属主直接进行用户的高效撤销,能够完全控制其他共享用户对自己数据的访问;b)在撤销用户和系统属性时,授权中心不需要对其余合法用户发布新私钥,提高了撤销用户的效率;c)对用户属性撤销则是利用懒撤销(lazy revocation)的方式来减缓新用户私钥的分发操作对系统性能的影响。

3　ABE-MACCS 方案的安全性证明

定理 1　如果 DBDH 假设在 (G_0,G_T) 上成立,则 ABE-MACCS 方案在标准模型下是 CPA,安全的。

证明　采用反证法,先假设存在一个攻击者 A 在多项式时间内能够以不可忽略的优势 ε 赢得游戏。构造一个模拟者 S,只需证明它可以利用攻击者 A 的能力以不可忽略的概率 $\varepsilon/2$ 区分 DBDH 元组 D_{bdh} 和随机元组 D_{rand},即 DBDH 假设不成立即可。过程如下:

挑战者首先生成系统公钥参数,包括生成元为 g 的群 G_0 和 G_T、一个有效映射 e 和随机值 $a,b,c,\theta\in Z_p^*$。挑战者执行一个公平的抛币协议,得到随机值 $u\in\{0,1\}$ 并对模拟者 S 保密。若 $u=1$,挑战者设置 $(g,A,B,C,Z)\in D_{\text{bdh}}$,否则设 $(g,A,B,C,Z)\in D_{\text{rand}}$。

1)初始化　攻击者选取挑战的访问结构 T^*,将其交给模拟者 S。

2)建立阶段　模拟者 S 选择一个随机值 $x'\in Z_p^*$,令 $\alpha_1=ab+x'$,$e(g,g)^{\alpha_1}=e(g,g)^{ab}e(g,g)^{x'}$。对于属性集中的所有元素 $a_j\in\Omega(1\leqslant j\leqslant n)$,随机选择 $k_j\in Z_p^*$,若 $a_j\in T^*$,则设置 $t_j=g^{k_j}$。然后,模拟者 S 将公共参数传给攻击者 A。

3)查找阶段 1　攻击者 A 通过向模拟者提交属性集 $w_j=\{a_j\mid a_j\in\Omega\cap a_j\notin T^*\}$ 进行私钥查询。模拟者 S 随机选取 $r_1'\in\mathbf{Z}_p^*$ 并令 $d_1=g^{x'-r_1'b}$,即 $r_1=ab+r_1'b$,由 $\alpha_1=ab+x'$ 可得 $d_1=g^{\alpha_1-(ab+r_1'b)}$。对 w_j 中每个元素 $a_l\in w_j$ 构建 $d_j=g^{(ab+r_1'b)k_j/b}$,最后将生成的私钥交给攻击者 A。

4)挑战阶段　攻击者 A 向模拟者 S 提交两个等长度的明文 M_1 和 M_2 以及一个挑战属性集。唯一的限制是在查找阶段,攻击者 A 不曾查询过挑战属性集的属性密钥。模拟者随机抽取一个比特 $b\in\{0,1\}$ 并对明文 M_b 进行加密:$c_1=g^c$;$\tilde{c}=M_b e(g,g)^{\alpha c}e(g,g)^{\alpha_2 s_2}=M_b e(g,g)^{(ab+x')c}e(g,g)^{\alpha_2 s_2}=M_b e(g,g)^{abc}e(g^c,g^{x'})e(g,g)^{\alpha_2 s_2}=M_b Ze(g^c,g^{x'})e(g,g)^{\alpha_2 s_2}$。对于访问结构树 T_A,设置根节点为 g^c,加密的其他过程同 2.3 节描述的加密算法。模拟者加密后将密文传给攻击者 A。如果 $(g,A,B,C,Z)\in D_{\text{bdh}}$ 且选择 $s_1=c,\alpha_2=x'$,由于 Z 是群 G_T 中的一个随机元素,那么模拟者 S 进行了一次完美的模拟。

5)查找阶段 2　与查找阶段 1 一致。

6)猜测阶段　攻击者 A 输出一个猜测 $b'\in\{0,1\}$,此时模拟者根据攻击者猜测的不同结果也作出相应的猜测,若攻击者给出了正确的猜测 $b'=b$,则模拟者 S 在与挑战者的游戏中输出猜测 $u'=1$,并指出给它的元组是从 D_{bdh} 选择的;若攻击者猜测错误 $b'\neq b$,则模拟者 S 输出 $u'=0$,并指出给它的元组是从 D_{rand} 选择的。下面计算模拟者 S 在与挑战者 C 之间的游戏中成功的概率。

当 $u=1$ 时,即 $(g,A,B,C,Z)\in D_{\text{bdh}}$,密文是一个合法的随机的关于消息 M_b 的密文,模拟者 S 进行了一次完美的模拟。通过定义,此时攻击者 S 具有不可忽略的优势 ε 来猜测正确的 b',其成功的概率为 $Pr[b'=b\mid(g,A,B,C,Z)\in D_{\text{bdh}}]=\frac{1}{2}+\varepsilon$。

当 $u=0$ 时,即 $(g,A,B,C,Z)\in D_{\text{bdh}}$,攻击者 S 没有得到任何关于消息 M_b 的信息,没有优势来猜测正确的 b',其成功的概率为 $Pr[b'\neq b\mid(g,A,B,C,Z)\in D_{\text{rand}}]=\frac{1}{2}$。那么,模拟者 S 在与挑战者 C 之间的游戏中成功的概率为

$$\text{Adv}_S=\frac{1}{2}(Pr[b'=b\mid(g,A,B,C,Z)\in D_{\text{bdh}}]+Pr[b'\neq b\mid(g,A,B,C,Z)\in D_{\text{rand}}])-\frac{1}{2}=\frac{1}{2}\times(\frac{1}{2}+\varepsilon+\frac{1}{2})-\frac{1}{2}=\frac{\varepsilon}{2}$$

4　实验及分析

4.1　复杂度分析

分析 Setup、KenGen、Encrypt、Decrypt、UsrRevoke、UsrAttriRevoke 和 GlobalAttriRevoke 算法的复杂度。Setup 算法包括执行 BDH 参数生成器主公钥和主私钥的生成,其复杂度为:$O((n)G_0)$;KenGen 算法的复杂度与用户持有的属性数量有关:$O((2^{|w|+1})G_0)$;Encrypt 算法的复杂度与访问结构中的属性数量有关:$O((2^{|T|})G_0)$;Decrypt 算法的复杂度与满足访问结构中的最小属性集大小有关:$O((|w'|)G_0)$;UsrRevoke 算法的复杂度与访问结构中 T_{ID} 的属性数量有关:$O((2^{|T_{\text{ID}}|})G_0)$;UsrAttriRevoke 算法等同于 KenGen 算法和 Encrypt 算法的复杂度之和,即 $O((2^{|T|})G_0)+O((2^{|w|+1})G_0)$;GlobalAttriRevoke

算法与 UsrRevoke 算法类似,其复杂度与访问结构中 T_A 的属性数量有关,即 $O((2^{|T_A|})G_0)$。算法的分析结果如表 1 所示。

表 1 方案的计算复杂度

算法	复杂度
Setup	$O((n)G_0)$
KenGen	$O((2^{\|w\|+1})G_0)$
Encrypt	$O((2^{\|T\|})G_0)$
Decrypt	$O((\|w'\|)G_0)$
UsrRevoke	$O((2^{\|T_{ID}\|})G_0)$
UsrAttriRevoke	$O((2^{\|T\|})G_0)+O((2^{\|w\|+1})G_0)$
GlobalAttriRevoke	$O((2^{\|T_A\|})G_0)$

4.2 实验环境

通过实验将 ABE-MACCS 方案与文献[11]描述的普通 CP-ABE 访问控制方案进行综合对比分析。实验设备为 Intel Core2 Duo 2.80 GHz,2 GB 内存,操作系统为 Linux Fedora 13,内核版本 2.6.33,对基本 CP-ABE 的工具包[17]进行改进后实现 ABE-MACCS 算法。

4.3 实验结果

实验主要对用户属性密钥的生成、加密算法、解密算法以及系统属性撤销的时间代价进行了分析。图 3 展示了基本 CP-ABE 与 ABE-MACCS 的私钥生成时间代价对比。图 4 展示了基本 CP-ABE 与 ABE-MACCS 加密数据时间代价对比,其中文件大小均为 10 MB。

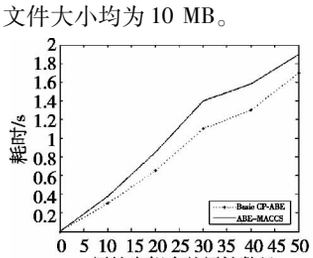

图3 私钥生成耗时与私钥中属性数目

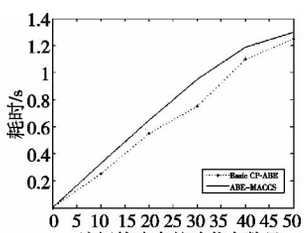

图4 加密耗时与访问策略规模

图 5 展示了基本 CP-ABE 与 ABE-MACCS 解密数据时间代价对比。图 6 展示了在属性集 $\Omega=\{a_1,a_2,\cdots,a_n\}$ 中随机选取的一个属性 a_i 进行撤销操作时,基本 CP-ABE 与 ABE-MACCS 方案的时间代价对比。

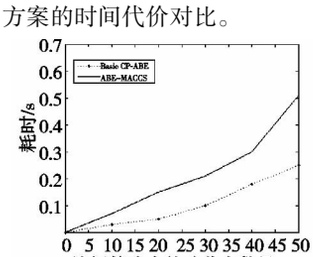

图5 解密耗时与访问策略规模

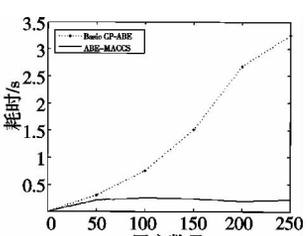

图6 撤销耗时与用户数目

4.4 实验分析

实验结果表明,相对于基本 CP-ABE,ABE-MACCS 方案在执行私钥生成、加密和解密操作的耗时有所增加,但仍在可接受范围内。在密钥撤销方面,ABE-MACCS 由于不必进行合法用户密钥的更新,极大地降低了撤销的时间代价。

5 结束语

云存储中共享数据的安全隐患一直是研究的热点,而利用以 ABE 为代表的加密访问控制技术是解决该问题的一个有效途径。本文针对共享数据的访问控制提出了一种复合访问结构的 CP-ABE 方案,令数据属主能够控制自己存放在云服务器端的数据访问,并且能够实现对用户属性的高效撤销。最后,对算法进行了性能分析,并证明了该方案在 DBDH 假设下是 CPA,安全的。在下一步的工作中,会重点研究将 CP-ABE 和代理重加密技术(PRE)相结合以实现用户属性密钥的安全分发,以及设计出用户与云服务器之间的安全通信协议和协议的安全性分析等工作。

参考文献:

[1] LARRY D. Cloud computing hasn't gone fortune 500 yet, but it's coming[EB/OL]. (2007-12-18)[2008-03-10]. http//blogs. zdnet. corn/BTL/? p=8199.

[2] CHRISTIAN C, IDIT K, SHRAER A. Trusting the cloud[J]. ACM SIGACT News, 2009, 40(2): 81-86.

[3] SAHAI A, WATES B. Fuzzy identity-based encryption[C]//Advances in Cryptology-EUROCRYPT. Berlin: Springer-Verlag, 2005: 457-473.

[4] SHAMIR A. Identity - based cryptosystems and signature schemes [C]//Advances in Cryptology CRYPTO'84. Berlin: Springer-Verlag, 1984: 47-53.

[5] BONEH D, FRANKLIN M. Identity-based encryption from the weil pairing[C]//Advances in Cryptology-CRYPTO'01. Berlin: Springer-Verlag, 2001: 213-229.

[6] BETHENCOURT J, SAHAI A, WATERS B. Ciphertext-policy attribute-based encryptio[C]//Proc of IEEE Sympoisum on Security and Privacy. Washington DC: IEEE Computer Society, 2007: 321-334.

[7] 苏金树,曹丹,王小峰. 属性基加密机制[J]. 软件学报, 2011, 22(6): 1299-1315.

[8] CHEUNG L, NEWPORT C. Provably secure ciphertext policy ABE [C]//Proc of ACM Conference on Computer and Communications Security. New York: ACM Press, 2007: 456-465.

[9] GOYAL V, JAIN A, PANDEY O, *et al*. Bounded ciphertext policy attribute based encryption[C]//Proc of ICALP. Berlin: Springer-Verlag, 2008: 579-591.

[10] LIANG X H, CAO Z F, LIN H, *et al*. Provably secure and efficient bounded ciphertext policy attribute based encryption[C]//Proc of ASIAN ACM Sympoisum on Information, Computer and Communications Security. New York: ACM Press, 2009: 343-352.

[11] LBRAIMI L, TANG Q, HARTEL P, *et al*. Efficient and provable secure ciphertext-policy attribute-based encryption schemes[C]//Proc of Information Security Practice and Experience. Berlin: Springer-Verlag, 2009: 1-12.

[12] PIRRETTI M, TRAYNOR P, MCDANIEL P, *et al*. Secure attribute-based systems[C]//Proc of ACM Conference on Computer and Communications Security. New York: ACM Press, 2006: 99-112.

[13] OSTROVSKY R, SAHAI A, WATERS B. Attribute-based encryption with non-monotonic access structures[C]//Proc of ACM Conference on Computer and Communications Security. New York: ACM Press, 2007: 195-203.

[14] LBRAIMI L, PETKOVIC M, NIKOVA S, *et al*. Mediated ciphertext-policy attribute-based encryption and its application[C]//Proc of the 10th International Workshop on Information Security Applications-WISA. Berlin: Springer-Verlag, 2009: 309-323.

[15] ATTRAPADUNG N, LMAI H. Attribute-based encryption supporting direct/indirect revocation modes[C]//Proc of Cryptography and Coding. Berlin: Springer-Verlag, 2009: 278-300.

[16] YU S C, WANG C, REN K, *et al*. Attribute based data sharing with attribute revocation[C]//Proc of ASIAN ACM Conference on Computer and Communications Security. New York: ACM Press, 2010: 261-270.

[17] BETHENCOURT J, SAHAI A, WATERS B. The cpabe toolkit, 2006 [EB/OL]. (2006-01-18)[2006-03-10]. Http://acsc. csl. sri. com/cpabe.