虚拟分组撤销策略的云存储访问控制模型

谢丽霞 薄夫宽 赵彬彬
(中国民航大学计算机科学与技术学院 天津 300300)

摘 要 为解决现有云存储访问控制模型用户权限撤销效率低、无法适应大规模用户的问题,在分析基于属性加密的密文策略的基础上提出了一个新的模型,给出了虚拟分组撤销策略,将用户映射到多个虚拟分组中,并重新构建了访问结构。用户权限撤销的范围被限制在一个虚拟分组内,对该虚拟组内的用户重新分发密钥即可实现用户权限撤销,而其它虚拟分组不需要任何变化,从而提高了用户权限撤销的效率。在Hadoop平台下进行了仿真实验,结果表明该模型具有较高的撤销效率。

关键词 云存储,访问控制,虚拟分组,用户权限撤销
中图法分类号 TP309,TP393.08 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.5.023

Virtual Group Revocation Policy-based Cloud Storage Access Control Model

XIE Li-xia BO Fu-kuan ZHAO Bin-bin
(School of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China)

Abstract To solve the problems that the existing cloud storage access control models have low efficiency of users' privilege revocation and are unable to adapt to a large number of users, this paper proposed a new model on the basis of analysis of cipher-text policy attribute-based encryption. Virtual group revocation policy was given, all users were mapped to multiple virtual groups, and the access structure was rebuilt. The range of users' privilege revocation was limited within a virtual group. By redistributing the users' private key in the certain virtual group where revocation takes place, users' privilege revocation can be achieved without any changes in the other virtual groups. Obviously, this approach greatly improves the efficiency of users' privilege revocation. A simulation experiment was conducted in Apache Hadoop platform, and the experiment results demonstrate that this model has higher efficiency on users' privilege revocation.

Keywords Cloud storage, Access control, Virtual group, Users' privilege revocation

1 引言

随着信息数据的爆发式增长,云存储作为一种服务技术[1]发展迅猛,各大IT厂商纷纷推出了自己的云存储服务。云存储由于具有成本低、无限存储容量、随时提供服务等优点得到广泛应用。尽管云存储具有许多优点,然而用户无法控制云存储中的数据,商业性质的云存储服务提供商又不完全可信,因此云存储的数据安全及隐私保护等问题不容忽视[2]。现有云存储模型在用户权限撤销方面存在不足,撤销效率比较低,无法适应具有大规模用户的云存储访问控制系统。本文正是在这种背景下,提出一种高效的用户权限撤销策略,构建安全、高效的云存储访问控制模型。

2 相关工作

访问控制作为核心安全服务之一,是确保云存储服务安全不可缺少的手段。传统的访问控制是建立在可信服务端的基础上的,为适应云存储服务的要求需要在传统访问控制的基础上引入密码机制。Sahai等人提出基于模糊身份的加密[3],将身份用一组属性描述,成为基于属性加密(Attribute Based Encryption)的雏形。随后,基于属性加密的密钥策略(KP-ABE)被提出[4]。在KP-ABE框架中,密文和一组可描述的属性相关联,访问控制结构包含于用户密钥中。这使得用户能够选择匹配的密文,而密文无法选择特定的用户,这不适合云存储的访问控制。基于属性加密的密文策略(CP-ABE)[5]克服了KP-ABE的缺点。在CP-ABE框架中,密文中包含访问控制结构,用户密钥与一组可描述的属性相关联,这与传统的访问控制机制相似。CP-ABE具有灵活以及抗联合密谋攻击的特点,被认为是最适合云存储访问控制的模型。目前,基于CP-ABE的改进模型成为研究热点,各种基于CP-ABE的模型被相继提出。

文献[6]提出一个实现CP-ABE的新方法,该方法具有高效性、可表达性,在具体的假设条件下具有可证明安全性。通

到稿日期:2015-04-13 返修日期:2015-08-08 本文受国家科技重大专项(2012ZX03002002),国家自然科学基金(60776807,61179045),天津市科技计划重点项目(09JCZDJC16800),中国民航科技基金(MHRD201009,MHRD201205)资助。
谢丽霞(1974-),女,硕士,副教授,主要研究方向为网络与信息安全,E-mail:lxxie@126.com;薄夫宽(1989-),男,硕士生,主要研究方向为网络与信息安全;赵彬彬(1990-),男,硕士生,主要研究方向为网络与信息安全。

过在加密算法中引入线性秘密共享访问矩阵，降低了加密算法和解密算法的时间复杂度，实现了多项式时间内的加密和解密。但是该模型没有考虑用户权限撤销的问题。

文献[7]提出了一个基于 CP-ABE 的云存储访问控制模型，由于引入了两重加密方案，用户可登录后再进行密钥的重新生成且用户不必保持一直在线。但该模型没有属性认证机构，而是由数据归属者进行属性认证，不利于大规模用户以及频繁的访问控制。

文献[8]提出了一种自治的访问方式，通过重新构建访问结构实现用户的撤销。创新点为只需要在原有的访问树中加入一个新的属性便可实现用户权限撤销，方法比较巧妙，简化了用户权限的撤销过程。

文献[9]对用户权限撤销策略进行了改进，引入了密钥更新算法以及版本号的概念。该模型灵活、有效地实现了权限撤销功能，减轻了数据所有者的负担，但认证机构的密钥更新过程的计算量并没有降低。

在云存储访问控制中，一次用户权限的撤销意味着数据所有者要对密文重新进行加密，重新生成用户私钥并分发给每一位用户。由此可见用户权限撤销的计算量巨大，而用户权限撤销是云存储访问控制系统中不可缺少的一部分。本文在现有模型的基础上，提出了基于虚拟分组撤销策略的云存储访问控制模型，重点对用户权限撤销进行改进，给出了虚拟分组撤销策略并重新构建了访问结构，实现了高效的用户权限撤销。

3 模型设计

3.1 模型框架

模型由 5 个部分组成，分别是：数据所有者 DO(Data Owner)、数据用户 DU(Data User)、认证机构 Au(Authority)和云存储 CS(Cloud Storage)。

数据所有者 DO：DO 提供有效的数据，并希望只有特定用户能够访问该数据，其他用户包括云存储提供商均无法获取明文数据。DO 负责确定合法用户的属性集合并制定访问策略。

数据用户 DU：DU 拥有某一属性集合，并希望获取 DO 的共享数据资源。

认证机构 Au：Au 负责用户属性的认证，为合法用户分发密钥。认证机构可以由数据所有者承担也可以由独立的第三方负责，前一种情况适合较小的用户规模，后一种情况可以采用分层的多 Au 结构以适应更大的用户规模。本文提出的权限撤销策略采用独立的认证机构，故适用于大规模用户的访问和绝大部分的云储存服务。

云存储 CS：CS 提供可靠的数据存储和访问服务，是不完全可信的。

虚拟分组撤销策略的云存储访问控制模型框架如图 1 所示，DO 先将数据对称加密，然后通过基于属性的加密算法对对称密钥进行加密，最后将两者合并上传到云存储中。数据所有者将公钥 PK、主密钥 MK 和一组属性集合交给认证中心，认证中心对用户的属性进行认证，为合法用户生成并分发私有密钥 SK。合法用户从云存储中得到密文，并通过私钥 SK 解密得到对称密钥，再利用对称密钥解密对称密文，最终得到数据明文。

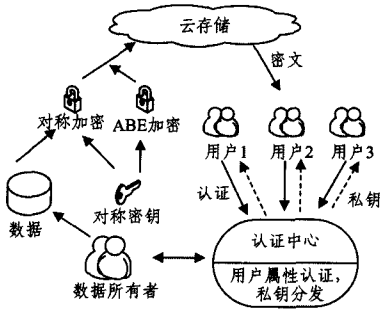

图 1 虚拟分组撤销策略的云存储访问控制模型框架

3.2 算法设计

根据文献[5]，设 G_1 为以素数 p 为阶的双线性群，g 为 G_1 的一个生成元。定义双线性映射 $e:G_1\times G_1\rightarrow G_2$，定义拉格朗日算子为：

$$\Delta i,s(x)=\prod_{j\in s,j\neq i}\frac{x-j}{i-j} \tag{1}$$

其中，$i\in Z_p$，s 中的元素也均属于 Z_p。

引入 Hash 函数 $H:\{0,1\}^*\rightarrow G_1$，将所有属性用二进制字符描述并映射到一个随机的群元素中。根据文献[5]中的模型算法，设计了本模型的 4 个算法，分别为：*Setup*()、*Encrypt*()、*KeyGen*()和 *Decrypt*()。

1)*Setup*：DO 选取以素数 p 为阶、以 g 为生成元的双线性群 G_1，随机选取 $\alpha,\beta\in Z_p$ 作为指数，则公钥 PK 和主密钥 MK 分别为：

$$\begin{aligned}&PK=(G_1,g,h=g^{\beta},e(g,g)^{\alpha})\\&MK=(\beta,g^{\alpha})\end{aligned} \tag{2}$$

2)*Encrypt*(PK,M,A_T)：DO 首先生成访问树 A_T，然后利用 A_T 对消息 M 进行加密。该算法从根节点开始，采取自顶向下的方式，为 A_T 中的每一个节点(包括叶子节点)x 选取一个多项式 q_x。令节点 x 处多项式 q_x 的度为：$d_x=k_x-1$。从根节点开始选取随机数 $y\in Z_p$，令 $q_r(0)=y$，对于其它节点 x，令：

$$q_x(0)=q_{parent(x)}(index(x)) \tag{3}$$

随机选择其它 d_x 个节点完成 q_x 的定义。加密得密文 CT：

$$\begin{aligned}CT=(A_T,\tilde{C}=Me(g,g)^{\alpha y},C=h^{y},\forall i:C_i=g^{q_i(0)},\\C_i^*=H(att(i))^{q_i(0)})\end{aligned} \tag{4}$$

3)*KeyGen*(MK,Att)：Att 为用户的属性集合。随机选取 $s\in Z_p$，并且为每一个属性 j 随机选取 s_j，用户私钥 D 为：

$$D=(DK=g^{\frac{\alpha+s}{\beta}},\forall j\in Att:D_j=g^{s}\cdot H(j)^{sj},D_j^*=g^{sj}) \tag{5}$$

4)*Decrypt*(CT,D)：DU 用自己的私钥对密文解密。

如果节点 x 为叶节点，令 $k=att(x)$，若 $k\in Att$，则执行解密节点函数：

$$\begin{aligned}&DecryptNode(CT,D,X)=\frac{e(D_K,C_X)}{e(D_K^*,C_X^*)}\\&=\frac{e(g^{s}\cdot H(k)^{s_k},h^{q_x(0)})}{e(g^{s_k},H(k)^{q_x(0)})}=e(g,g)^{sq_x(0)}\end{aligned} \tag{6}$$

如果 $k\notin Att$，则得到不正确的输出。如果 x 不是叶节点，则以如下方式计算 *DecryptNode*(CT,D,X)：输入 x 的所有孩子节点 z 计算解密节点函数，输出结果设为 F_z。若 $F_z=\perp$，

则 $DecryptNode(CT,D,X)$ 的输出为 $\perp$，否则计算：$F_x=\prod_{z\in T_x} F_z^{\Delta_{i,T_x^*}(0)}$，其中 $i=index(z)$，$T_x^*=\{index(z):z\in T_x\}$，$T_x$ 表示子节点 z 中 k_x 的大小满足 $F_z\neq\perp$ 的子节点集合。

完成了解密节点函数的定义后，给出解密算法。对根节点 r 执行算法 $DecryptNode(CT,D,r)$，输出结果设为 A，即 $A=DecryptNode(CT,D,r)=e(g,g)^{sq_r(0)}=e(g,g)^{ys}$。从而得到明文 M：

$$M=\frac{\tilde{C}}{e(C,DK)/A}=\frac{\tilde{C}}{e(C,DK)/e(g,g)^{ys}} \tag{7}$$

4 虚拟分组撤销策略

本文在现有模型[9-11]的基础上，对用户撤销策略进行改进并给出了虚拟分组撤销策略。当对某一用户进行权限撤销时，只需对密文数据重新加密并对一小部分用户进行密钥更新即可实现用户权限撤销过程，而无需对所有用户都进行密钥更新，实现了高效的用户权限撤销。

4.1 虚拟分组撤销策略

现有的用户权限撤销策略在进行一次撤销时需要对全体用户的密钥进行更新，为了缩小密钥更新的范围，就需要将用户虚拟地划分为多个组，指定其隶属的组以限制和确定其范围。撤销某一用户时，先确定其分组，对该分组的用户重新审核，为合法用户指定新的分组并重新生成密钥，而无需对全体用户的密钥进行更新。需要指出的是用户分组并不影响访问控制的细粒度，因为分组是虚拟的，即分组可以看作是用户的一个动态变化的特殊属性。另外，由于虚拟分组是用户的一个动态变化且实时更新的属性，因此虚拟分组不会相互制约，也不会影响权限的授权。

虚拟分组撤销策略如图 2 所示，当有新增用户时，系统为其分配一个有效的虚拟分组号并记录该组号，每个虚拟分组内的用户数量尽量均衡。当需要撤销某一用户时，首先获取被撤销用户的虚拟分组号，然后重新认证该虚拟分组的所有用户，除被撤销的用户无法通过认证外，其他用户均可成功通过认证并得到新的虚拟分组号。最后更新访问结构中的虚拟分组号属性。由于被撤销的用户无法通过认证而继续保持原来的虚拟分组号，而原来的虚拟分组号无法满足更新后的访问结构，由此实现了该用户权限的撤销。

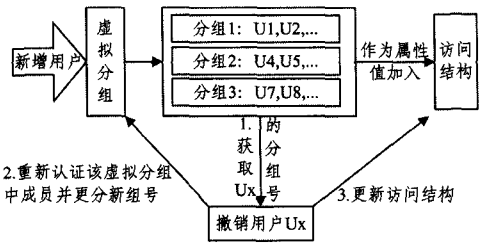

图 2 虚拟分组撤销策略图

文献[11]给出了访问结构(Access Structure)的定义。设 $\{p_1,p_2,\cdots,p_m\}$ 为 m 个参与者组成的集合，设 $\boldsymbol{T}\in\{p_1,p_2,\cdots,p_m\}$。如果 $\boldsymbol{T}$ 为单调的访问结构，那么 $\boldsymbol{T}$ 必须满足：对任意 $\boldsymbol{G}\in\boldsymbol{T}$ 且 $\boldsymbol{G}\subseteq\boldsymbol{H}$，则 $\boldsymbol{H}\in\boldsymbol{T}$。

访问结构通过定义授权访问集合和非授权访问集合，对访问策略进行了逻辑化的描述。本文采用访问树来描述访问结构，其中的叶子节点代表属性值，非叶节点代表阈值。设 k_x 为节点的阈值，num_x 为节点 x 的孩子节点个数，显然有 $0<k_x\leqslant num_x$，则当 $k_x=1$ 时，表示逻辑“或”关系；当 $k_x=num_x$ 时，表示逻辑“与”关系。

图 3 为撤销发生后访问树的更新过程。访问树中不仅包含了用户的属性关系，也包含了虚拟分组的属性关系。图 3 中将用户划分为 3 个虚拟分组，因而访问树中存在 3 个虚拟分组号属性，用户私钥中只要有其中的一个虚拟分组号就能够满足访问树。图 3 中对虚拟分组号为 VG2 的分组中的某一用户进行撤销，该分组的其他用户经认证后重新分配虚拟分组号 VG4，而被撤销的用户无法更新虚拟分组号，不能满足更新后的访问树。

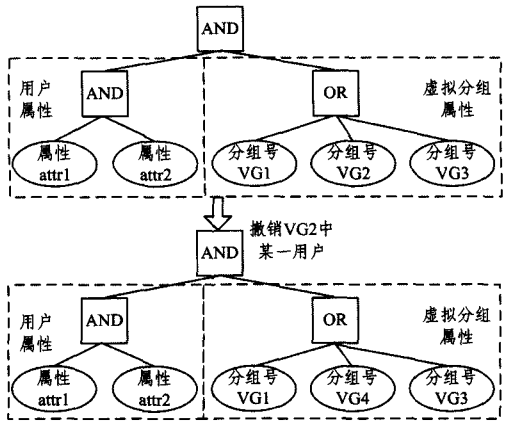

图 3 访问树更新过程

4.2 撤销策略正确性证明

设有 n 个数据用户分别为 $\{DU_1,DU_2,\cdots,DU_n\}$，需要进行权限撤销的用户为 DU_i，$DU_i\in\{DU_1,DU_2,\cdots,DU_n\}$，用 $VG(u_x)$ 函数表示用户的虚拟分组号。

当 DO 执行撤销用户 DU_i 时，首先得到该用户的虚拟分组号 $VG(u_i)$，并重新构造访问树 A_T，将 A_T 中的 $VG(u_i)$ 属性用 $VG(u_i)+1$ 代替。这样凡是虚拟分组号为 $VG(u_i)$ 的用户均无法通过 A_T。令 U_h 表示在 $\{DU_1,DU_2,\cdots,DU_n\}$ 中除 DU_i 外虚拟分组号为 $VG(u_i)$ 的用户集合，U_m 表示 $\{DU_1,DU_2,\cdots,DU_n\}$ 中虚拟分组号不为 $VG(u_i)$ 的用户集合。集合 U_h 中的用户通过认证重新获得了包含虚拟分组号为 $VG(u_i)+1$ 的私钥。

下面对用户 $\forall DU_h\in U_h$，$\forall DU_m\in U_m$，以及 DU_i 做解密算法分析。

1)用户 DU_i 执行解密算法，对 A_T 的每一个节点执行节点解密函数 $DecryptNode()$，当 A_T 的叶节点为虚拟分组号属性时，由于 $VG(u_i)\notin A_T$，则 $DecryptNode(CT,SK,VG(u_i))$ 得到不正确的输出而最终无法对密文 CT 解密。

2)用户 DU_h 执行解密算法，对 A_T 的每一个节点执行节点解密函数 $DecryptNode()$，由于 DU_h 的属性值只有虚拟分组号发生了变化，且 $VG(u_i)=VG(u_i)+1\in A_T$，根据式(6)和式(7)得到正确的明文 M。

3)用户 DU_m 执行解密算法，对 A_T 的每一个节点执行节点解密函数 $DecryptNode()$，由于 DU_m 的虚拟分组号 $VG(u_m)\neq VG(u_i)$，且 $VG(u_m)\in A_T$，根据式(6)和式(7)能够得到正确的明文 M。

综上，被撤销的用户 DU_i 无法正确地解密密文，而与

DU_i 同一虚拟分组的其他用户能够得到正确明文，其他虚拟分组的用户也能够得到正确明文，因此虚拟分组撤销策略是正确的。

4.3 撤销效率分析

设系统中用户数量为 N_u，虚拟分组数量为 N_g，每个虚拟分组中平均用户数量为 N_i，其中 $N_i=[N_u/N_g]$，符号$[x]$表示对实数 x 取整。设在没有虚拟分组属性的情况下，每更新一个用户私钥的时间代价为 t。设每增加一个虚拟分组生成用户私钥的时间代价增加值为 Δt。

令撤销一个用户的时间代价为 T，由于撤销发生时只需要更新一组虚拟分组的用户私钥(不包括被撤销用户)即可，则 T 为

$$T=t\cdot(N_i-1)+\Delta t\cdot N_g$$
$$=t\cdot([N_u/N_g]-1)+\Delta t\cdot N_g \quad (8)$$

现有模型大多采用与文献[9]相同的撤销策略，设文献[9]中用户撤销时间代价为 T'，通过分析得

$$T'=t\cdot(N_u-1) \quad (9)$$

由式(8)知，撤销时间代价 T 与虚拟分组数量 N_g 有关，N_g 取值过大或过小都不能使 T 取得最小值。令函数$T(N_g)=t\cdot([N_u/N_g]-1)+\Delta t\cdot N_g$，根据不等式定理，$T(N_g)\geqslant 2\sqrt{N_u\cdot\Delta t\cdot t}-t=T_{\min}$，当且仅当

$$N_g=\sqrt{\frac{t\cdot N_u}{\Delta t}} \quad (10)$$

函数 $T(N_g)$取得最小值，即用户撤销的时间代价最小。$T_{\min}$ 要远远小于 T'，即本模型的用户撤销时间代价较低。

5 实验与结果

5.1 实验环境与仿真系统实现

实验平台为一台式计算机，主频为 2.5GHz，内存为 4GB。在 Ubuntu12.04 操作系统下，搭建 Hadoop 平台，利用 Hadoop 中的文件系统 HDFS 模拟云存储平台。

实验中使用了由萨勒诺大学 GAS 实验室用 Java 语言编写的 jPBC 库[12]。本实验在 Eclipse 编程环境下用 Java 语言编程实现了一个仿真系统。仿真系统功能框图如图 4 所示。加密模块将输出的密文上传至 Hadoop 文件系统的 HDFS 中，并为用户提供访问路径。用户管理模块设置系统的用户数量、属性种类，由私钥生成模块负责生成用户私钥。用户撤销模块负责完成用户撤销，可以设置需要撤销的用户，并记录撤销时间消耗。虚拟分组管理模块负责整个系统的虚拟分组号管理，解密模块根据用户私钥从文件系统 HDFS 中解密密文。

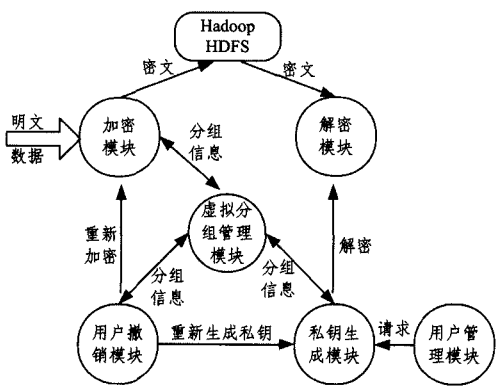

图 4 仿真系统功能框图

5.2 实验过程与结果

在该仿真系统上，进行了用户权限撤销时间与虚拟分组数量的关系以及本模型与现有模型[9]的用户权限撤销时间对比等实验。实验流程如图 5 所示。

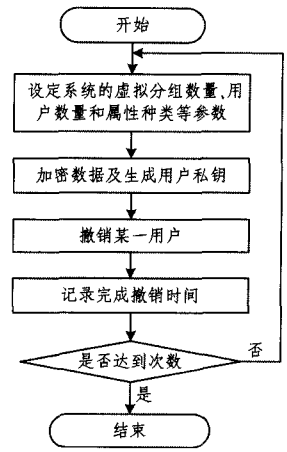

图 5 实验流程

详细实验过程如下：

1)设定系统的虚拟分组数量、用户数量和属性种类的数量等系统参数。

2)将本地的 2.4M 的 PDF 文件采用密钥长度为 128 位的 AES 对称加密得到 T_{aes}，将对称密钥进行基于属性的加密得到 T_k，并将 T_k 追加到 T_{aes} 后作为密文 CT 上载到 HDFS 中。

3)系统为所有合法用户生成并分发私钥，利用私钥可以从 HDFS 下载并解密 CT。

4)系统撤销任意某一用户，撤销过程包括数据的对称加密、对称密钥的加密以及用户的私钥更新。

5)记录完成撤销的时间消耗。根据具体实验类型调整相应参数，重复进行上述流程直到满足实验要求的采集数据量。

图 6 给出了用户权限撤销时间与虚拟分组个数的关系。测量时保持用户数量为 200 个，私钥中属性个数为 10(不包括虚拟分组号属性)。当虚拟分组数量取值为 1 时，撤销时间为 266s，为使图中坐标比例更加协调，图 6 中横坐标以 5 为起点作图。整个曲线是先急剧下降到平稳再缓慢上升，这是因为，当用户权限撤销未采用基于虚拟分组的权限撤销策略时(即相当于虚拟分组为“1”组)，此时权限撤销的效率较低且消耗时间较多；当虚拟分组划分的组数增加时只需对某一组内的用户权限进行更新，故权限撤销的效率会增加，消耗的时间会减少；若存在多个虚拟分组，则需更多的分组号(即：动态属性)描述虚拟分组，那么生成用户私钥的时间将会增加。故随着虚拟分组的增加，消耗的时间开始减少；当虚拟分组的数量增加到某值时，消耗的时间达到最小；之后，随着虚拟分组的增加，消耗的时间又会慢慢地增加。由式(8)也可知，随着虚拟分组的增多，用户私钥更新数量减少，撤销时间相应急剧减少；而随着虚拟分组的增多，用户的虚拟分组属性的数量也不断增加，撤销时间又缓慢增加。由式(10)可知，必然存在特定虚拟分组取值使得撤销时间代价取得最小值，通过实验测量，式(10)中参数 t 约为 1.15s，Δt 约为 0.08s，N_u 为 200 个，代入式(10)计算得出当虚拟分组数量 N_g 取值为 54 时，用户权限撤销时间最短。综上所述，虚拟分组会增加权限撤销的效

率，不恰当的虚拟分组会降低权限撤销的效率，但一定存在一个虚拟分组值使得权限撤销的效率达到最高。

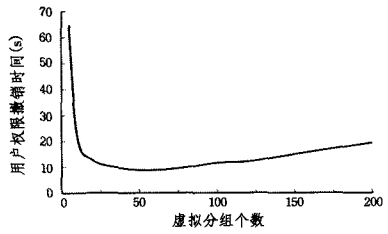

图 6　用户权限撤销时间与虚拟分组个数的关系

为了与现有基于CP-ABE的云存储访问控制模型进行对比，本文也实现了文献[9]提出的模型的仿真系统，两种模型的用户权限撤销时间比较如表 1 所列。在实验中，保持用户属性种类为 10 种，其中虚拟分组数量取值为式(10)中的计算结果。随着用户数量的增加，两种模型的用户权限撤销时间呈增加趋势，但是两种模型的用户权限撤销时间的差距是巨大的。随着用户数量的增多，两种模型的撤销时间差距也不断增大。从表 1 可见：

(1)当用户数量达到 200 个时，文献[9]模型的撤销时间为 266.89s，本文模型的撤销时间仅为 9.14s；

(2)当用户数量达到 300 个时，文献[9]模型的撤销时间为 372.27s，本文模型的撤销时间仅为 11.07s；

(3)当用户数量达到 400 个时，文献[9]模型的撤销时间为 553.62s，本文模型的撤销时间仅为 14.23s；

(4)当用户数量达到 500 个时，文献[9]模型的撤销时间为 740.73s，本文模型的撤销时间仅为 16.17s。

表 1　用户权限撤销时间比较

用户数量(个)	文献[9]模型撤销时间(s)	本文模型撤销时间(s)
10	11.07	1.57
30	47.23	2.93
50	65.16	3.74
100	105.24	6.23
150	179.19	7.56
200	266.89	9.14
250	309.76	9.78
300	372.27	11.07
350	456.18	12.68
400	553.62	14.23
450	592.76	15.46
500	740.73	16.17

上述实验结果表明，本文模型在用户权限撤销时间上表现出了巨大的优势，其撤销效率远远高于现有模型。

结束语　本文在分析现有基于CP-ABE的云存储访问控制模型的基础上，给出了虚拟分组权限撤销策略。当进行用户权限撤销时，将用户私钥重新生成的范围缩小至一个虚拟分组内，通过重新构建访问树，实现了高效的用户权限撤销。最后通过实验验证了本模型的有效性和撤销的高效性。

随着云存储服务的发展，越来越多的单位或团体不但希望通过云存储平台实现数据的访问控制，而且希望不同单位或团体之间能够实现访问控制策略的融合，即不同云存储访问控制模型之间的访问控制策略的合成。如何将基于属性的访问控制策略逻辑融合方法应用到云存储访问控制模型中，是下一步的研究重点。

参 考 文 献

[1] Wu J, Fu J, Lin Z, et al. A survey on cloud storage [J]. Journal of Computers, 2011, 6(8): 1764-1771

[2] Elavarasi P, Parijatham R. Key updation for the dynamic attributes in cloud computing for competent user retraction [J]. International Journal of Engineering Science and Technology, 2013, 5(06s): 2278-9510

[3] Sahai A, Waters B. Fuzzy identity-based encryption[C]// The 24th Annual International Conference on Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2005: 457-473

[4] Goyal B, Pandey O, Sahai A, et al. Attribute based encryption for fine-grained access control of encrypted data[C]// The 13th ACM Conference on Computer and Communications Security. New York: ACM Press, 2006: 89-98

[5] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption[J]. Journal of Network and Computer Applications, 2010, 33(2): 76-83

[6] Waters B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization [C]// The 14th International Conference on Practice and Theory in Public Key Cryptography. Berlin: Springer, 2011, 6571: 53-70

[7] Zhang R, Chen P. A Dynamic Cryptographic Access Control Scheme in Cloud Storage Services [C]// 2012 8th International Conference on Computing and Networking Technology. Washington D C: IEEE CS Press, 2012: 50-55

[8] Pervez Z, Khattak A M, Lee S, et al. SAPDS: self-healing attribute-based privacy aware data sharing in cloud[J]. The Journal of Supercomputing, 2012, 62(1): 431-460

[9] Yang K, Jia X, Ren K. Attribute-based Fine-Grained Access Control with Efficient Revocation in Cloud Storage Systems[C]// The 8th ACM Symposium on Information, Computer and Communications Security. New York: ACM Press, 2013: 523-528

[10] Yu S, Wang S, Ren K, et al. Attribute based data sharing with attribute revocation[C]// The 5th ACM Symposium on Information, Computer and Communications Security. New York: ACM Press, 2010: 261-270

[11] Su J S, Cao D, Wang X F, et al. Attribute-based encryption schemes[J]. Journal of Software, 2011, 22(6): 1299-1315(in Chinese)
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

[12] GAS lab. Java Pairing Based Cryptography Library[EB/OL]. (2013-12-01). http://gas.dia.unisa.it/projects/jpbc/contact.html