doi:10.3969/j.issn.1000-436x.2014.z2.007

云存储系统中支持用户隐私保护的细粒度访问控制

肖敏，王春蕾，周由胜
（重庆邮电大学 计算机科学与技术学院，重庆 400065）

摘 要：从云存储实际需求出发，设计了一个云存储环境下支持用户隐私保护和用户属性撤销的多属性权威的属性加密机制，为了保证系统实现的效率和减轻数据持有者的负担，在属性撤销中，复杂的计算任务都委托给可信第三方或云服务器完成。所提方案在 DBDH 假设下被证明是安全的。
关键词：云存储；属性加密；多授权中心；隐私保护；属性撤销
中图分类号：TP393 **文献标识码**：A **文章编号**：1000-436X(2014)Z2-0042-06

Fine-grained access control scheme with user privacy protection in cloud storage systems

XIAO Min, WANG Chun-lei, ZHOU You-sheng
（College of Computer Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China）

Abstract: Based on the actual demands of cloud storage system, a new multi-authority ABE (MA-ABE) scheme is constructed to support user privacy protection and attribute revocation. For enhancing efficiency and alleviatingowner's computing load， the complex computing works involved in attribute revocation are delegated to a trusted third party orcloud server. The proposed scheme is provably secure under the DBDH assumption.
Key words: cloud storage; attribute based encryption; multi-authority; privacy protection; attribute revocation

1 引言

在云存储系统中，用户可以通过网络实现文件共享。但是在存储一些机密信息的时候，消息通常要以密文的形式存储，传统的一对一加解密模式可扩展性差，复杂的密钥管理会带来大量开销，难以适应外包机密数据的多用户共享的需求，属性加密机制的一对多加密特性和灵活的加密策略是实现云存储中细粒度访问控制的重要方法。

2005 年，Sahai 和 Water[1] 首次引入了属性加密的概念，通过引入 (t,n) 门限访问结构，当且仅当用户的属性集合和密文属性集合的交集中属性的个数满足门限 t 时，用户才能解密密文。Goyal 等[2]在后续的工作中根据访问策略与密钥和密文的关系将属性加密机制分为密钥策略（KP-ABE）和密文策略（CP-ABE），并提出了第一个 KP-ABE 方案。在该方案中，访问结构变成具有表达能力的访问结构树，只有当密文中的属性集合满足用户的属性密钥访问策略时用户才能够解密。Bethencourt 等[3]随后构建了第一个 CP-ABE 方案。与 KP-ABE 相反，访问结构嵌入在密文中，只有用户所拥属性集合满足密文中的访问结构时用户才能进行解密。属性加密机制的一个基本安全需求是抗共谋，即防止用户之间通过联合各自的密钥达到超越自身访问权限的能力。另外，用户密钥和属性撤销也是保障属性密码机制在实践应用

收稿日期：2014-10-29
基金项目：国家社会科学基金资助项目(14CTQ026)；重庆市应用研究项目基金和先进技术资助项目(cstc2014jcyjA40028)；重庆市自然科学基金资助项目(cstc2011jjA40031, cstc2011jjA40042)
Foundation Items: The National Social Science Foundation of China (14CTQ026); The Chongqing Research Program of Application Foundation and Advanced Technology (cstc2014jcyjA40028); Chongqing Natural Science Foundation(cstc2011jjA40031, cstc2011jjA40042)

中安全性的一个重要问题，已有方案的复杂度都与系统中用户数量线性相关。

根据云存储环境对机密数据的细粒度访问控制需求，Yu 等[4]将基本的 KP-ABE 方案应用到云环境中，数据持有者在属性集合下加密消息并将密文发送给云服务器，同时为用户制定访问策略并生成属性密钥。为了减少用户端的负担，将用户属性撤销阶段的主要计算任务委托给云服务器。K Yang 等[5]利用 CP-ABE 加密机制构建了一个云存储数据的细粒度访问控制方案，系统中引入一个属性权威(AA, attribute authority)管理和分发用户属性密钥，同样地把属性撤销阶段的密文更新工作委托给云服务器，大大减轻了数据持有者的负担。

只依赖一个可信机构来负责用户管理和密钥分发的方案，在实际应用中可能由于该机构工作量大且容易受到攻击而成为系统瓶颈。并且，在大规模分布式应用系统中会涉及多个不同的机构来管理不同的用户，每个用户也可能在多个机构中具有不同的身份属性。Chase[6]首次给出了多 AA 的基于门限结构的属性加密方案，该系统由一个中心权威(CA，central authority)和多个分管不同属性集合的 AA 组成，CA 为 AA 和用户分发全局密钥和系统密钥，用户从每一个 AA 处申请其管理的属性密钥。该方案中 CA 掌管着系统私钥可以生成任意用户的解密密钥，所以 CA 必须是一个完全可信的机构。为防止用户间共谋，系统还为每个用户生成唯一的全局身份标识（GIS, global identifier）。考虑到 CA 可能带来的安全隐患，Chase 随后提出了一个去 CA 的多属性权威加密方案[7]，但是运用 GID 抗用户共谋的同时却带来了用户隐私泄漏的风险，该方案通过构建一个基于两方安全计算的匿名密钥分发协议[8]，实现了密钥分发阶段的用户 GID 的保密性。考虑到在云环境下的应用，K.Yang 等提出了一个比较完备的云环境下基于多属性权威的数据访问控制方案[9,10]，但是该方案无论在密钥分发阶段还是属性撤销环节都没有考虑到用户隐私保护。

本文给出了一个支持用户隐私保护和属性撤销的多属性权威属性加密方案及其在云存储中的应用，使得在大规模云存储系统中，实现安全、可扩展的细粒度数据访问控制的同时保护用户的隐私。该方案在保证安全性的同时将复杂的计算任务委托给可信第三方或云服务器完成，大大减轻了用户负担。

2　预备知识

2.1　判定双线性 Diffie-Hellman(DBDH, decisional Bilinear diffie-hellman)假设[1]

q 是一个素数，$\mathbb{G}$ 是 q 阶循环群，g 是 $\mathbb{G}$ 的一个生成元，随机选取 Z_q 中的 4 个元素 a、b、c、z，并给定 2 个四元组 $A=g^a, B=g^b, C=g^c$，$Z=e(g, g)^{abc}$ 和 $A=g^a, B=g^b, C=g^c, Z=e(g,g)^z$，如果不存在攻击者能够在多项式时间内以不可忽略的优势对上述 2 个四元组进行区分，则该假设成立。

2.2　匿名密钥分发协议

如表 1 所示，用户拥有秘密信息 $u\in\mathbb{Z}_q$，AA 拥有秘密信息 $\alpha,\beta,\gamma\in\mathbb{Z}_q$，选择阶为大素数 q 的群 $\mathbb{G}$，记 l,h 为 $\mathbb{G}$ 的 2 个元素，双方进行交互运算在不泄露双方秘密信息的基础上输出 $(h^{\alpha}l^{1/\beta+u})^{\gamma}$ 给用户。在每一步中，PoK 代表在计算中所用到的秘密信息的知识证明。详细的解释见文献[7]。

表 1　匿名密钥分发协议

User u		Attribute Authority
$\rho_1\in Z_q$	$\xleftrightarrow{2\ PC}$	$x:=(\beta+u)\rho_1, \tau\in Z_q$
$\rho_2\in Z_q$	$\xleftarrow{X_1, X_2 PoK\alpha\tau x}$	$X_1:=l^{\tau x}, X_2=h^{\alpha\tau}$
$Y:=(X_1^{\rho_1}X_2)^{\rho_2}$	$\xrightarrow{Y, PoK\rho_2}$	
$D:=Z^{1/\rho_2}$	$\xleftarrow{Z, PoK(\tau\gamma)}$	$Z:=Y^{\gamma/\tau}$

3　模型和方案概述

3.1　系统模型

云数据访问控制下的多授权中心属性加密系统包括 5 部分：一个可信的第三方机构(比如 CA)、属性权威、数据持有者(data owner)、用户(user)和云服务器(cloud server)。

CA 负责用户注册和身份管理，为每个用户分配全局唯一的身份标识 GID，并且 GID 是用户的私密信息。假定 CA 具有足够的计算能力，在密钥分发阶段，能够代替用户与每一个 AA 交互执行匿名密钥分发协议。

AA 之间相互独立，每个 AA 负责管理相应范围内的属性集。在系统初始化阶段，AA 生成自己的属性权威密钥、公钥和属性公钥，并联合共同生成一个系统公钥。属性权威密钥用于生成用户解密密钥。在属性撤销阶段，AA 生成并分发密文更新

密钥和密钥更新密钥。

数据持有者利用系统公钥和属性公钥生成密文，然后发送密文到云服务器。

用户将 GID 作为自己的私钥，并委托 CA 通过匿名密钥分发协议与 AA 进行交互获得与 GID 相关的密钥组件，实现密钥分发时用户身份信息的保密和减轻用户的负担。用户定义自己的伪随机函数用于生成面向不同 AA 时对应的假名，此假名通过 CA 认证后传递给 AA，使得用户能够通过发送假名给对应的 AA 获取解密密钥。另外，用户负责数据的解密和密钥的更新。

云服务器负责存储和维护数据持有者的数据并处理用户的数据访问请求。在属性撤销阶段，云服务器负责密文的更新，以减轻数据持有者的负担。

3.2 方案

该方案包括 5 个基本算法。

1) 系统初始化

①CA 初始化 $CASetup(\lambda)\rightarrow(GID)$：输入安全参数 $\lambda\in\mathbb{N}$，为系统中的每个用户生成 $u=GID$，该 u 作为用户的私钥。

② 初始化 $AASetup()\rightarrow(d_k,APK_k,ASK_k,v_{i_k},PK_{i_k},Y)$：输入安全参数 $\lambda\in\mathbb{N}$，每个 $AA_k,k\in\{1,\cdots,N\}$ 输出：门限值 $\{d_k\}$、公钥 APK_k 和私钥集合 $ASK_k=\{t_k,x_k,\varphi_k,\omega_k,t_{k,1},\cdots,t_{k,n_k}\}$（$n_k$ 表示 AA_k 中属性的个数）、每个属性 $i_k,i\in\{1,\cdots,n_k\}$ 对应的版本号 v_{i_k} 和公钥 PK_{i_k}。另外，所有的 AA 利用它们的秘钥联合生成系统公钥 Y。

2) 密钥生成

① 生成用户秘钥组件 $USKeyGen(u,ASK_k)\rightarrow D_u$：执行匿名密钥分发协议，$CA$ 和 AA_k 分别输入 u 和 ASK_k 中的 $t_k,x_k,\varphi_k,\omega_k$，输出用户密钥组件 D_u。

②生成用户属性秘钥 $AKeyGen(a_{u,k},ASK_k)\rightarrow SK_{u,k}$：用户通过假名 $a_{u,k}$ 向 AA_k 请求相关属性密钥 $SK_{u,k}$。

3) 数据加密 $Enc(\{A_k,k\in\{1,\cdots,N\}\},m,PK_{i_k})\rightarrow CT$：$A_k$ 表示消息中与 AA_k 相关的属性集合，数据持有者对消息 m 在密文属性集合 $\{A_k,k\in\{1,\cdots,N\}\}$ 及其对应的属性公钥 PK_{i_k} 下加密生成密文 CT

4) 数据解密 $Dce(CT,SK_{u,k},D_u)\rightarrow m$：用户利用密钥 $SK_{u,k}$ 和 D_u 解密密文 CT，得到消息 m。

5) 属性撤销

① 生成更新秘钥 $UPkeyGen(ASK_k,\widetilde{i_k},v_{\widetilde{i_k}})\rightarrow(KUK_{u,\widetilde{i_k}},CUK_{\widetilde{i_k}})$：更新密钥由被撤销的属性 $\widetilde{i_k}$ 对应的 AA_k 生成。输入 ASK_k 中的 φ_k、ω_k，被撤销的属性 $\widetilde{i_k}$ 和当前属性的版本号 $v_{\widetilde{i_k}}$，输出用户的密钥更新密钥 $KUK_{u,\widetilde{i_k}}$ 给用户，密文更新密钥 $CUK_{\widetilde{i_k}}$ 给云服务器。

②秘钥更新 $SKUpdate\left(SK_{u,k},KUK_{u,\widetilde{i_k}}\right)\rightarrow\widetilde{SK}_{u,k}$：密钥更新由未被撤销用户自己完成，输入用户的 $SK_{u,k}$ 和 $KUK_{u,\widetilde{i_k}}$，就得到了更新后的用户解密密钥 $\widetilde{SK}_{u,k}$。

③密文更新 $CTUpdate\left(CT,CUK_{\widetilde{i_k}}\right)\rightarrow\widetilde{CT}$：密文更新由云服务器完成，输入与撤销属性 $\widetilde{i_k}$ 相关的密文和 $CUK_{\widetilde{i_k}}$，输出新的密文 $\widetilde{CT}$。

4 安全模型

通过攻击者 $\mathcal{A}$ 和挑战者 $\mathcal{B}$ 之间的交互式游戏来定义本方案的安全模型。和基于身份加密[1]的选择性安全模型相似，允许攻击者询问足够多的不能够用来解密的属性私钥和密钥更新密钥。游戏分五个阶段：

初始化 $\mathcal{A}$ 发送以下信息给 $\mathcal{B}$：挑战密文对应的属性列表 $A_c=\{A_1^C,\cdots,A_N^C\}$，每个 AA_k 对应一组属性集 A_k^C，并给出一组腐败的 AA 集合（最多有 $k-2$ 个）。然后，$\mathcal{B}$ 生成系统公钥，所有 AA 的公钥和私钥，并且把所有公钥和腐败 AA 的私钥发送给 $\mathcal{A}$。

密钥请求 1 $\mathcal{A}$ 可以进行足够多的满足以下条件的密钥请求：对于每一个被请求的 GID，至少有一个诚实的 AA_k 给出的密钥要少于阈值 d_k；攻击者不可以针对同一个 GID 对某个 AA 进行重复密钥请求。

挑战 $\mathcal{A}$ 向 $\mathcal{B}$ 提供 2 个长度相等的消息 M_0 和 M_1，以及密文所对应的属性集合 A_c。$\mathcal{B}$ 随机选取其中一个设为 M_θ 并生成挑战密文 C 给 $\mathcal{A}$。

密钥请求 2 与密钥请求 1 相同。

猜测 $\mathcal{A}$ 给出对 θ 的猜测值 θ'。如果 $\theta=\theta'$，攻击者成功。

定义 1 游戏中 $\mathcal{A}$ 的优势定义为 $\Pr[\theta=\theta']-\frac{1}{2}$。

定义 2 如果在以上游戏中，所有多项式时间

的攻击者 $\mathcal{A}$ 最多具有可忽略的优势，则称本文的方案在选择属性攻击下是安全的。

定义 3　当单个用户不能解密数据时，如果没有多项式时间的攻击者 $\mathcal{A}$ 能够联合这些用户的密钥解密数据，则称本文的方案是抗用户合谋攻击的。

定义 4　当所有 AA 不能联合追踪用户得到完整的用户属性集合，则称本文的方案是抗AA合谋攻击的。

5　具体方案

5.1　系统初始化

设 $\mathbb{G}$ 和 $\mathbb{G}_T$ 是阶为素数 q 的循环群，$e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$ 为双线性映射，g 为 $\mathbb{G}$ 的生成元。

5.1.1　*CA* 初始化

当有新用户加入系统时，*CA* 首先对该用户进行认证，如果用户合法，就为该用户生成一个全局身份 $u=GID\in\mathbb{Z}_q$ 作为用户私钥。用户 u 定义伪随机函数 $PRF_u(\cdot)$ [11]并生成面向 AA_k 的假名 $a_{u,k}$，由 *CA* 认证后传递给 AA_k。

5.1.2　*AA* 初始化

设 S_{A_k} 表示 AA_k 管理的所有属性的集合，S_{A_k} 中属性个数为 n_k。AA_k 选取随机数 $t_k,x_k,\varphi_k,\omega_k,t_{k,1},\cdots,t_{k,n_k}\in\mathbb{Z}_q$ 作为 AA_k 的属性权威私钥 $ASK_k=(t_k,x_k,\varphi_k,\omega_k,t_{k,1},\cdots,t_{k,n_k})$，并生成属性权威公钥 $APK_k=(g^{\omega_k},g^{x_k})$。对于每个属性 $i_k\in S_{A_k}$，AA_k 选择随机数 v_{i_k} 作为属性 i_k 的版本号并生成属性公钥：

$$PK_{i_k}=(T_{k,i}=g^{t_{k,i}},g^{t_{k,i}\omega_k v_{i_k}})$$

AA_k 将 $Y_k=e(g,g)^{t_k}$ 发送给其他 AA，各个 AA 独自计算得到系统公钥

$$Y=\prod Y_k=e(g,g)^{\sum_k t_k}$$

AA_k 和 AA_j 通过两方密钥交换协议，获取共享密钥 $s_{kj}=s_{jk}\in\mathbb{Z}_q$，然后共同定义一个伪随机函数 $PRF_{kj}(h)=g^{x_k x_j/(s_{kj}+h)}$，设 $y_k=g^{x_k}$，则该式可以通过 $y_k^{x_j/(s_{kj}+h)}$ 或 $y_j^{x_k/(s_{kj}+h)}$ 由 AA_j 或者 AA_k 分别计算得到。

5.2　密钥分发

5.2.1　生成用户秘钥组件

为了减少用户的负担，通过 *CA* 的用户管理权限，委托 *CA* 与各个 AA_k 分别执行匿名密钥分发协议，其中 $l=y_j^{x_k},h=g,\alpha_k=\delta_{kj}R_{kj},\beta_k=s_{kj},\gamma_k=\delta_{kj}$，$R_{kj}\in\mathbb{Z}_q$ 是由 AA_k 和 AA_j 共同选取的随机数，则得到 $D_{kj}=(h^{\alpha_k}l^{1/\beta_k+u})^{\gamma_k}=(g^{\delta_{kj}R_{kj}}y_j^{x_k/(s_{kj}+u)})^{\delta_{kj}}=g^{R_{kj}}(PRF_{kj}(u))^{\delta_{kj}}$。如果 $k>j$，$\delta_{kj}=1$，*CA* 得到 $D_{kj}=g^{R_{kj}}PRF_{kj}(u)$；否则 $\delta_{kj}=-1$，*CA* 得到 $D_{kj}=g^{R_{kj}}/PRF_{kj}(u)$。

CA 再将用户 u 对应的所有 D_{kj} 相乘得到 $D_u=\prod_{(k,j)\in\{1,\cdots,N\}\times\{1,\cdots,N\}\backslash\{k\}}D_{kj}=g^{R_u}$ 并发送给用户 u，其中 $R_u=\sum_{(k,j)\in\{1,\cdots,N\}\times\{1,\cdots,N\}\backslash\{k\}}R_{kj}$。

5.2.2　生成用户属性密钥

1) 设 A_k^u 表示用户 u 所拥有的由 AA_k 管理的属性集合。用户 u 用假名 $a_{u,k}$ 向 AA_k 请求属性集合 A_k^u 中每个属性 i_k 对应的密钥。

2) AA_k 随机生成一个 d_k-1 阶多项式 $p_k(\cdot)$，满足 $p_k(0)=t_k-\sum_{j\in\{1,\cdots,N\}\backslash\{k\}}R_{kj}$，然后生成用户属性秘钥：$SK_{u,k}=(L_{u,k}=g^{a_{u,k}\varphi_k+\omega_k},K_{u,i_k}=g^{p_k(i_k)/t_{k,i_k}}g^{(a_{u,k}\omega_k\varphi_k+\omega_k^2)v_{i_k}},\forall i_k\in A_k^u),k\in\{1,\cdots,N\}$。

5.3　加密

设 A_k^C 表示密文中包含的由 AA_k 管理的属性集合。输入消息 m、系统公钥 Y、密文属性集 $S_c=\{i_k^c\},k\in\{1,\cdots,N\}$ 和属性公钥 $PK_{i_k^c}(i_k^C\in S_c)$，然后选取随机数 $s\in\mathbb{Z}_q$，计算出密文 CT：

$$CT=(C_0=mY^s=me(g,g)^{s\sum_k t_k},C_1=g^s,C_2=g^{st_{k,i_k^C}},C_3=g^{-st_{k,i_k^C}\omega_k v_{i_k^C}},i_k^c\in A_k^c,\forall k\in\{1,\cdots,N\})$$

5.4　解密

当 u 的属性集合满足每个 AA_k 的门限 d_k 时，才能正确解密密文。

对于 $AA_k,k\in\{1,\cdots,N\}$，任何 d_k 个属性 $i_k\in A_k^c\cap A_k^u$，计算

$$e(C_2,K_{u,i_k})e(C_3,L_{u,k})=e(g^{st_{k,i_k}},g^{p_k(i_k)/t_{k,i_k}}g^{(a_{u,k}\omega_k\varphi_k+\omega_k^2)v_{i_k}})e(g^{-st_{k,i_k}\omega_k v_{i_k}},g^{a_{u,k}\varphi_k+\omega_k})=e(g,g)^{sp_k(i_k)}$$

再对所有的 $e(g,g)^{sp_k(i_k)}(i_k\in A_k^c\cap A_k^u)$ 进行多项式插值运算，得到 $P_k=e(g,g)^{sp_k(0)}=e(g,g)^{s(t_k-\sum_{j\neq k}R_{kj})}$。然后，所有 $P_k,k\in\{1,\cdots,N\}$ 相乘得到 $P=\prod_{k=1}^N P_k=e(g,g)^{s(\sum t_k-R_u)}$，最后解密得到 $m=\dfrac{C_0}{P\cdot e(D_u,C_1)}$。

5.5　用户属性撤销

当撤销用户 u 的属性 $\widetilde{i_k}$ 时，需要更新其他未被撤销用户的属性秘钥来防止被撤销的用户越权访问（前向安全），还要更新密文来保证新加入的用

户只要拥有足够的属性仍然能够解密旧的密文（后向安全）。属性撤销分为 3 个阶段。

5.5.1 AA_k 生成更新密钥

AA_k 为属性 $\widetilde{i_k}$ 生成一个新的属性版本号 $v'_{\widetilde{i_k}}$，然后计算出用户的密钥更新密钥 $KUK_{u,\widetilde{i_k}}=g^{(v'_{\widetilde{i_k}}-v_{\widetilde{i_k}})(a_{u,k}\omega_k\varphi_k+\alpha_k^2)}$ 和密文更新密钥 $CUK_{\widetilde{i_k}}=(v'_{\widetilde{i_k}}-v_{\widetilde{i_k}})\omega_k$。

5.5.2 未撤销用户更新密钥

AA_k 把 $KUK_{u,\widetilde{i_k}}$ 发送给拥有撤销的属性 $\widetilde{i_k}$ 的未撤销用户，然后用户使用 $KUK_{u,\widetilde{i_k}}$ 对他的属性密钥部分进行更新。更新后用户属性密钥变为

$$\widetilde{SK}_{u,k}=(L'_{u,k}=L_{u,k},K'_{u,\widetilde{i_k}}=K_{u,\widetilde{i_k}}\cdot KUK_{u,\widetilde{i_k}},\forall i_k\in A_k^u,i_k\neq\widetilde{i_k},K'_{u,i_k}=K_{u,i_k}),k\in\{1,\cdots,N\}$$

5.5.3 云服务器更新密文

AA_k 把 $CUK_{\widetilde{i_k}}$ 发送给云服务器，然后云服务器使用 $CUK_{\widetilde{i_k}}$ 对密文进行更新。云服务器只对密文中与撤销的属性 $\widetilde{i_k}$ 相关的密文组件进行更新，得到新的密文为

$$\widetilde{CT}=(C'_0=C_0,C'_1=C_1,C'_2=C_2$$

$\forall i_k^c\in A_k^c$，如果 $i_k^c=\widetilde{i_k}$，$C'_3=C_3(C_2)^{CUK_{\widetilde{i_k}}}$；否则，$C'_3=C_3,k\in\{1,\cdots,N\}$）。

6 安全性证明

定理 1 如果匿名秘钥分发协议是安全的，并且 DBDH 假设成立，则本文的方案是选择属性安全的。

证明 假设 CA 已经与所有的 AA 执行了匿名密钥分发协议。协议的安全性已经在文献[7]中进行了证明。所有用户的假名 $a_{u,k}$ 都有 CA 认证后由 AA 存储。

给出 $A=g^a,B=g^b,C=g^c$ 和 $Z=e(g,g)^{abc}$ 或者 $e(g,g)^z$，其中 $z\in\mathbb{Z}_q$。根据匿名密钥分发协议的安全性要求，最多有 $N-2$ 个腐败 AA。允许敌手 A 请求多个用户属性集合的私钥，并且至少有一个诚实的 $AA_{\hat{k}}$ 给出的属性密钥个数少于门限值，即 $A_{\hat{k}}^u\cap A_{\hat{k}}^c<d_{\hat{k}}$。隐式设定 $\sum t_k s=abc$，其中 $s=c$ 和 $\sum t_k=ab$。不失一般性，假定 $\hat{k}=1$。选择 N 个随机数 $r_1,r_2,\cdots,r_N$ 满足 $r_1=r_2+\cdots+r_N$。隐式设定 $t_{\hat{k}}=ab-r_1$，则 $p_{\hat{k}}(0)=ab-r_1-\sum_{j\neq\hat{k}}R_{\hat{k}j}$ 不可计算。对于其他诚实的 AA_k，设 $t_k=r_k$，则 $p_k(0)=r_k-\sum_{j\neq\hat{k}}R_{\hat{k}j}$ 可计算。

1) 初始化。$\mathcal{A}$ 发送以下信息给 $\mathcal{B}$：挑战密文对应的属性列表 $A_c=\{A_1^C,\cdots,A_N^C\}$，每个 AA_k 对应一组属性集 A_k^C，并给出一组腐败的 AA 集合（最多有 $k-2$ 个）。然后由 $\mathcal{B}$ 生成系统公钥 $Y=e(A,B)=e(g,g)^{ab}$，对于诚实的 AA_k，$\mathcal{B}$ 选取随机数 $\beta_{k,i}\in\mathbb{Z}_q$，生成属性公钥 $PK=\{\{T_{k,i_k}=B^{\beta_{k,i_k}}=g^{b\beta_{k,i_k}}\}_{i_k\in A_k^u-A_k^C},\{T_{k,i_k}=g^{\beta_{k,i_k}}\}_{i_k\in A_k^u\cap A_k^C}\}$；对腐败 AA_k，$\mathcal{B}$ 生成私钥 $ASK_k=(t_k=?,x_k,\varphi_k,\omega_k,t_{k,1},\cdots,t_{k,n_k})$。最后，$\mathcal{B}$ 将以上生成的所有公钥和腐败 AA 的私钥发送给 A。

2) 密钥请求 1。对诚实的 AA_k，$\mathcal{A}$ 可以进行足够多的满足条件的密钥请求。

①当 $k\neq\hat{k}$ 时，选取随机数 $z_{k,u}\in\mathbb{Z}_q$，隐式设定 $p_k(0)=r_k-\sum_{j\neq k}R_{kj}=bz_{k,u}$，并选取一个 d_k-1 阶随机多项式 $\rho_k(i)$，满足 $\rho_k(0)=z_{k,u}$，并假定 $p_k(i)=b\rho_k(i)$。若 $i_k\in A_k^u\cap A_k^C$，则 $t_{k,i_k}=\beta_{k,i_k}$，$g^{p_k(i_k)/t_{k,i_k}}=g^{b\rho_k(i_k)/\beta_{k,i_k}}=B^{\rho_k(i_k)/\beta_{k,i_k}}$；若 $i_k\in A_k^u-A_k^C$，则 $t_{k,i_k}=b\beta_{k,i_k}$，$g^{p_k(i_k)/t_{k,i_k}}=g^{b\rho_k(i_k)/b\beta_{k,i_k}}=g^{\rho_k(i_k)/\beta_{k,i_k}}$。

②当 $k=\hat{k}$ 时，不失一般性，假定有 d_k-1 个属性 $i_k\in A_k^u\cap A_k^C$。选取随机数 $w_{k,u}\in\mathbb{Z}_q$，隐式设定 $p_k(0)=ab-r_1-\sum_{j\neq k}R_{kj}=ab-w_{k,u}b$。再选取 d_k-1 个随机数 $\tau_i\in\mathbb{Z}_q$，设 $p_k(i_k)=b\tau_{i_k},i_k\in A_k^u\cap A_k^C$。若 $i_k\in A_k^u\cap A_k^C$，有 $t_{k,i_k}=\beta_{k,i_k}$，则 $g^{p_k(i_k)/t_{k,i_k}}=g^{b\tau_{i_k}/\beta_{k,i_k}}=B^{\tau_{i_k}/\beta_{k,i_k}}$；若 $i_k\in A_k^u-A_k^C$，有 $t_{k,i_k}=b\beta_{k,i_k}$，由于 $p_k(0)=ab-w_{k,u}b$，并且已知 $A_k^u\cap A_k^C$ 中的 d_k-1 个点 $p_k(i_k)=b\tau_{i_k}$，所以多项式 $p_k(\cdot)$ 可以根据拉格朗日插值公式完全确定，可以表示为

$$p_k(i_k)=\Delta_0(i_k)(ab-w_{k,u}b)+\sum\Delta_{jk}(i_k)b\tau_{jk},$$

则

$$\begin{aligned}g^{p_k(i_k)/t_{k,i_k}}&=g^{\frac{\Delta_0(i_k)(ab-w_{k,u}b)+\sum\Delta_{jk}(i_k)b\tau_{jk}}{b\beta_{k,i_k}}}\\&=g^{\frac{a\Delta_0(i_k)}{\beta_{k,i_k}}}g^{\frac{\sum\Delta_{jk}(i_k)\tau_{jk}-w_{k,u}\Delta_0(i_k)}{\beta_{k,i_k}}}\\&=A^{\frac{\Delta_0(i_k)}{\beta_{k,i_k}}}g^{\frac{\sum\Delta_{jk}(i_k)\tau_{jk}-w_{k,u}\Delta_0(i_k)}{\beta_{k,i_k}}}\end{aligned}$$

挑战者 $\mathcal{B}$ 利用 AA_k 的私钥得到属性私钥 $SK_{u,k}=\{L_{u,k},K_{u,i_k}\}$，并发送给 $\mathcal{A}$。

挑战 $\mathcal{A}$ 给 $\mathcal{B}$ 2 个长度相等的消息 M_0 和 M_1。

$\mathcal{B}$ 随机选取 $M_\theta, \theta \in \{0,1\}$ 生成挑战密文 CT 给 $\mathcal{A}$。

$$CT = (C_0 = M_\theta Z, C_1 = C, C_2 = g^{c\beta_{k,i_k^c}} = C^{\beta_{k,i_k^c}}, C_3 = C^{\beta_{k,i}\omega_k v_{i_k^c}}, i_k^c \in A_k^c, \forall k \in \{1,\cdots,N\})$$

密钥请求 2　与密钥请求 1 相同。

猜测　$\mathcal{A}$ 给出对 θ 的猜测值 θ'。如果 $\theta = \theta'$，挑战者 $\mathcal{B}$ 输出 $Z = e(g,g)^{abc}$；否则输出 $Z = e(g,g)^z$。因此，如果在上述游戏中攻击者 $\mathcal{A}$ 具有不可忽略的优势，则意味着挑战者 $\mathcal{B}$ 也能以不可忽略的优势区分 DBDH 中的 2 个四元组 $(A = g^a, B = g^b, C = g^c, Z = e(g,g)^{abc})$ 和 $(A = g^a, B = g^b, C = g^c, Z = e(g,g)^z)$，即 DBDH 假设不成立。

定理 2　本文的方案是抗用户合谋攻击的。

证明　在该方案中，每个用户都分配了一个全局身份 $GID = u$，每个 AA_k 为用户分发的密钥 $SK_{u,k} = (L_{u,k}, K_{u,i_k})$ 都与全局身份 u 相关，对不同的用户即使拥有相同的属性，其所对应的属性密钥也是不一样的。所以任何用户想要联合起来获取别人的密钥来进行解密是不可能的。

定理 3　本文的方案是抗 AA 合谋攻击的。

证明　在密钥分发阶段，CA 与 AA 间执行的匿名密钥分发协议，使得用户的 GID 对 AA 是保密的。用户请求密钥时，通过一个伪随机函数 $PRF_u(\cdot)$ 生成面向各个 AA_k 的假名 $a_{u,k}$，利用 $a_{u,k}$ 向各个 AA_k 请求属性密钥 $SK_{u,k} = (L_{u,k}, K_{u,i_k})$。对不同的 AA_k，同一个用户 u 所用的假名 $a_{u,k}$ 互不相同，而且 $PRF_u(\cdot)$ 和 u 对 AA_k 都是保密的，所以 AA_k 难以联合追踪用户得到完整的用户属性集合。

定理 4　被撤销属性的用户不能与未被撤销的用户共谋以更新他的密钥。

证明　$KUK_{u,\widetilde{i_k}} = g^{(v'_{\widetilde{i_k}} - v_{\widetilde{i_k}})(a_{u,k}\omega_k\varphi_k + \omega_k^2)}$ 中的 $a_{u,k}$ 与用户身份标识 u 相关，所以被撤销属性的用户不能利用未被撤销用户的 KUK 生成自己的密钥更新密钥。

7　结束语

本文给出了一个实现云存储数据的细粒度访问控制方案。该方案基于多授权中心的属性加密机制，支持用户属性撤销。通过在密钥中引入用户全局身份信息，防止用户共谋，同时通过匿名密钥分发协议和用户假名，防止 AA 间的共谋，实现用户私密身份信息的保护。为减少用户和数据拥有者的负担，密钥生成阶段的主要计算任务委托给可信第三方 CA，属性撤销阶段的密文更新工作委托给了云服务器。最后证明了本方案在 DBDH 假设下是选择属性安全的。

参考文献：

[1] SAHAI A, WATERS B. Fuzzy identity-based encryption[A]. Advances in Cryptology-Eurocrypt[C]. 2005.457-473.

[2] GOYAL V, PANDEY O, SAHAI A, *et al*. Attribute-based encryption for fine-grained access control of encrypted data[A]. ACM Conference on Computerand Communications Security-CSS[C]. 2006. 99-112.

[3] BETHENCOURT J, SAHAI A, WATERS B. Ciphertext-policy attribute-based encryption[A]. IEEE Symposium on Security and Privacy[C]. New York: IEEE Press, 2007. 321-334

[4] YU S, WANG C, REN K, *et al*. Attribute based data sharing with attribute revocation[A]. ACM Symp Information, Computer and Comm Security (ASIACCS '10)[C]. 2010.

[5] YANG K, JIA X, REN K. Attribute-based fine-grained access control with efficient revocation in cloud storage systems[A]. The 8th ACM SIGSAC Symposium on Information, Computer and Communications Security(ASIA CCS'13)[C]. 2013. 523-528.

[6] CHASE M. Multi-authority attribute based encryption[A]. The 4th Theory of Cryptography Conference on Theory of Cryptography (TCC'07)[C]. Springer, 2007. 515-534.

[7] CHASE M, CHOW S. Improving privacy and security in multi-authority attribute-based encryption[A]. The 16th ACM Conference on Computer and Communications Security (CCS'09)[C]. ACM, 2009.121-130.

[8] NAOR M, PINKAS B, REINGOLD O. Distributed pseudo-random functions and KDCs[A]. Advances in Cryptology: EUROCRYPT '99 (J. Stern, ed.), Lecture Notes in Computer Science[C]. 1999.

[9] YANG K, JIA X, REN K, *et al*. DAC-MACS: effective data access control for multi-authority cloud storage systems[A]. IEEE INFOCOM[C]. 2013. 2895-2903.

[10] YANG K, JIA X. Expressive, efficient and revocable data access control for multiauthority cloud storage[J]. IEEE Transactions on Parallel and Distributed Systems, 2014,25:1735-1744.

[11] JARECKI S, LIU X. Efficient oblivious pseudorandom function with applications to adaptive OT and secure computation of set intersection[J]. Theory of Cryptography Lecture Notes in Computer Science, 2009,(5444):577-594.

作者简介：

肖敏（1971-），女，湖北宜昌人，博士，重庆邮电大学副教授，主要研究方向为网络与信息安全管理、云计算数据安全、现代密码理论与应用等。

王春蕾（1991-），女，江苏扬州人，重庆邮电大学硕士生，主要研究方向为云环境下的用户访问控制。

周由胜（1979-），男，湖北利川人，博士，重庆邮电大学讲师，主要研究向为网络安全、云计算数据安全、现代密码理论与应用等。