基于 CP-ABE 的利益冲突数据集的访问控制

陈 成 努尔买买提·黑力力
(新疆大学数学与系统科学学院 乌鲁木齐 830046)

摘 要 云存储允许数据拥有者将数据储存在云端,以便为用户提供数据共享服务。然而,同一个数据拥有者储存的不同数据之间可能会出现利益冲突。鉴于此,文中提出针对利益冲突数据集的基于密文策略属性基加密(CP-ABE)的访问控制方案。在该方案中,数据拥有者将虚拟属性用"与"门嵌入访问树中以得到修正的访问树,并在修正的访问树下对利益冲突数据集中的各个数据加密,从而避免了一个用户访问利益冲突数据集中的部分或全部数据而导致的错误、欺骗或风险。最后,从理论上对所提方案的效率和安全性进行了分析,分析结果表明了其是高效且安全的。

关键词 访问控制,利益冲突数据集,中国墙策略,CP-ABE
中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.11.022

CP-ABE Based Access Control of Data Set with Conflict of Interest

CHEN Cheng Nurmamat HELIL
(College of Mathematics and System Science, Xinjiang University, Urumqi 830046, China)

Abstract Cloud storage allows data owners to store their encrypted data in the cloud, so as to provide data sharing services for users. However, there might exist a conflict of interest among different data stored by the same data owner. In this regard, this paper proposed a ciphertext-policy attribute-based encryption (CP-ABE) based access control scheme for the data set with conflict of interest. In this scheme, the data owner embeds a virtual attribute into the access tree with the "AND" gate to get the modified access tree, and encrypts the data in the data set with conflict of interest under the modified access tree, thus avoiding errors, cheats or risks caused by an individual user's access to some or all data in the data set with conflict of interest. Finally, the efficiency and security of this scheme were analyzed. The analytical results suggest the proposed scheme is efficient and secure.

Keywords Access control, Data set with conflict of interest, Chinese wall policy, CP-ABE

1 引言

云存储是云计算的重要服务[1],它允许数据拥有者在云中托管其数据,为用户提供数据共享服务。在云存储服务中,数据拥有者不直接与用户进行交互来提供数据。属性基加密 ABE(Attribute-Based Encryption)[2]被认为是最适合应用于云存储系统中的访问控制方案之一。ABE 有两类应用模式,分别为密钥策略属性基加密 KP-ABE(Key-Policy Attribute-Based Encryption)[3-6]和密文策略属性基加密 CP-ABE(Ciphertext-Policy Attribute-Based Encryption)[4-7]。在 CP-ABE 中,数据拥有者将访问结构嵌入密文中。CP-ABE 比 KP-ABE 更适合应用在云存储系统中,文献[11-21]就 CP-ABE 中的属性撤销问题进行了研究。

数据拥有者在云上储存的数据集中可能存在利益冲突的数据,同一个用户先后访问利益冲突数据集中的部分或全部数据可能会给数据拥有者造成损失。中国墙策略(Chinese Wall Policy)[8]将一些可能产生利益冲突的数据划分成不同的数据集,强制任一主体至多只能访问一个数据集中的数据,但访问哪个数据集不受强制规则的限定。访问控制限制很大程度上可以降低访问所带来的损失或风险[9]。因此,需要考虑云上储存数据之间的利益冲突关系。Helil 等[10]考虑云存储上共享数据之间的潜在关系,并提出了一种基于 CP-ABE 的具有隐藏访问策略和隐藏限制策略特性的访问控制方案,用于实现敏感数据集的访问控制限制。本文受文献[10]的启发,提出利益冲突数据集的 CP-ABE 访问控制方案,通过用户属性撤销实施利益冲突数据集限制,增强了数据访问的安全性,使数据拥有者可以更加灵活、有效地控制其分享在云上的加密数据。

2 利益冲突数据集限制

例如,数据拥有者可能将两份具有利益冲突的数据 $M_{Company\_A}$ 和 $M_{Company\_B}$ 储存在云端,数据 $M_{Company\_A}$ 和 $M_{Company\_B}$ 分别属于公司 $Company\_A$ 和 $Company\_B$。如果某个用户被授权访问 $M_{Company\_A}$ 和 $M_{Company\_B}$,那么其访问结果可能会给数

到稿日期:2017-10-12 返修日期:2018-01-21 本文受国家自然科学基金(61562085,11261057,11461069),新疆维吾尔自治区人力资源和社会保障厅留学人员科技活动项目资助。

陈 成(1992-),男,硕士生,主要研究方向为访问控制,E-mail:1576660664@qq.com;努尔买买提·黑力力(1976-),男,博士,教授,主要研究方向为网络信息安全、访问控制,E-mail:nur924@sina.com(通信作者)。

据拥有者或 *Company_A* 和 *Company_B* 公司带来损失或风险。为了避免发生这种情况，任何用户对这两份数据的访问都应加以限制，采用必要的隔离措施。为此，首先提出利益冲突数据集和利益冲突数据集限制的概念。

定义 1(利益冲突数据集) 如果一个用户连续地访问数据 $M\in\{M_{1,1},\cdots,M_{1,n_1}\}$ 和 $M'\in\{M_{2,1},\cdots,M_{2,n_2}\}$ 后产生利益冲突，则称 $CDS=(\{M_{1,1},\cdots,M_{1,n_1}\},\{M_{2,1},\cdots,M_{2,n_2}\})$ 是一个利益冲突数据集。

非利益冲突数据集中的数据称为普通数据。

定义 2(利益冲突数据集限制) 设 CDS 是一个利益冲突数据集，如果用户一旦访问了 $\{M_{i,1},\cdots,M_{i,n_i}\}(i\in\{1,2\})$ 中的任一数据，该用户就被禁止访问 $\{M_{j,1},\cdots,M_{j,n_j}\}(j\in\{1,2\},j\neq i)$ 中的任一数据，则称 $c(CDS)=c(\{M_{1,1},\cdots,M_{1,n_1}\},\{M_{2,1},\cdots,M_{2,n_2}\})$ 是一个利益冲突数据集限制。

为了实现利益冲突数据集限制，本文使用附加的虚拟属性。虚拟属性仅用于处理利益冲突数据集，没有特殊的意义。为了简洁，文中若没有特殊说明，就用限制来代替利益冲突数据集限制。

定义 3(虚拟属性) 虚拟属性是用于实现利益冲突数据集限制的专用属性。规定系统启动时所有用户均拥有系统中的全部虚拟属性。

3 知识背景

访问结构仍然是单调访问结构，访问结构的定义请参见文献[7]。

定义 4(双线性对) G_0 和 G_1 是阶为素数 p 的乘法循环群，g 是 G_0 的生成元，映射 $e:G_0\times G_0\rightarrow G_1$ 是一个双线性对，如果 e 满足以下 3 条性质。

1)双线性：$\forall P,Q\in G_0$，$\forall a,b\in Z_p^*$，有 $e(P^a,Q^b)=e(P,Q)^{ab}$；

2)非退化性：$e(g,g)\neq 1$；

3)可计算性：$\forall P,Q\in G_0$，存在有效算法计算 $e(P,Q)$。

双线性 Diffie-Hellman(简称 BDH)假设如下：BDH 问题是给定 G_0 中的 A,B,g^c 以及 G_0 的生成元 g，计算 $e(A,B)^c$。如果 $\Pr[\psi(\phi,G_0,G_1,A,B,g^c)=e(A,B)^c]\geqslant\varepsilon(k)$，其中 k 是安全参数(位长度为 ϕ)，则称算法 ψ 在 $\langle\phi,G_0,G_1,e\rangle$ 上解决 BDH 问题有优势 $\varepsilon(k)$。如果每个多项式时间算法在 $\langle\phi,G_0,G_1,e\rangle$ 上解决 BDH 问题的优势 $\varepsilon(k)$ 是可忽略的，则称 $\langle\phi,G_0,G_1,e\rangle$ 满足 BDH 假设。

4 系统结构和安全性要求

本节描述数据分享系统的结构和安全性要求的定义。数据分享系统结构如图 1 所示。

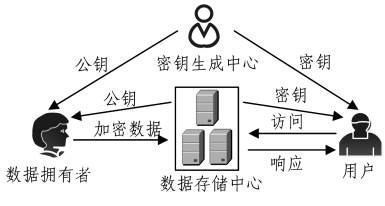

图 1 数据分享系统的结构

Fig. 1 Architecture of data sharing system

4.1 系统描述

1)密钥生成中心。其为整个系统生成公共参数和秘密参数，负责发布、撤消和更新用户的属性密钥。假设它是半诚实的，即它会诚实地执行系统分配的任务，但希望尽可能多地了解加密的数据。

2)数据存储中心。其提供数据共享服务，负责控制外部用户访问加密数据，并提供相应的内容服务。数据存储中心与密钥生成中心共同生成用户个性化密钥。对于每一个属性，数据存储中心为有效用户发布和撤销属性用户组密钥，这些属性用户组密钥用于强制实施细粒度的访问控制。假设数据存储中心也是半诚实的。

3)数据拥有者，即拥有数据的用户。其将自己的数据上传到数据存储中心，以便于共享。数据拥有者负责制定访问策略，并在上传数据前根据访问策略、密钥生成中心生成的公钥和数据存储中心生成的公钥加密其数据。

4)用户，即访问数据的实体。密钥生成中心和数据存储中心根据用户拥有的属性为其生成用户密钥，如果用户拥有的属性满足加密数据中的访问策略，则其能够解密密文并获得数据。

由于密钥生成中心和数据存储中心都是半诚实的，因此它们应该被阻止访问数据明文；同时，它们应该仍能为用户生成并发布密钥。为了满足这种矛盾要求，密钥生成中心和数据存储中心使用各自的主密钥参与 2PC 安全协议[13]，并且在密钥发布阶段向用户发布独立的密钥分量。

4.2 安全性要求

1)数据机密性。防止不满足访问策略的用户访问数据明文；同时，应该防止密钥生成中心和数据存储中心访问数据明文。

2)防合谋。防合谋是 ABE 系统中所需的最重要的安全性质之一。用户均不能通过组合他们的属性来解密密文，但其中至少有个用户能单独解密的除外。

3)向后安全性和向前安全性。向后安全性是指阻止任何拥有属性(满足访问策略)的用户访问在其拥有该属性之前分发的先前加密数据的明文；向前安全性是指阻止撤销给定属性的任何用户访问撤销该属性之后分发的后续加密数据的明文，但其拥有的其他有效属性仍满足访问策略的除外。

4)利益冲突数据集限制。任何用户访问利益冲突数据集中的数据均不能违反利益冲突数据集限制。

5 利益冲突数据集的 CP-ABE 访问控制方案

5.1 访问树和方案组件

访问树和满足访问树的详细描述请参见文献[7]。用于利益冲突数据集的 CP-ABE 访问控制方案的组件如下：$U=\{u_1,\cdots,u_n\}$ 为系统中的全体用户；$L=\{\lambda_1,\cdots,\lambda_p\}\cup\{\lambda_{\pi_1},\cdots,\lambda_{\pi_m}\}$ 为系统中全体描述性属性，其中 $\{\lambda_1,\cdots,\lambda_p\}$ 为普通属性，$\{\lambda_{\pi_1},\cdots,\lambda_{\pi_m}\}$ 为虚拟属性，$\lambda_{\pi_i}\neq\lambda_{\pi_j}(i\neq j)$ 且 $\{\lambda_1,\cdots,\lambda_p\}\cap\{\lambda_{\pi_1},\cdots,\lambda_{\pi_m}\}=\emptyset$；$ID_t$ 为用户 u_t 的身份；G_y 为拥有属性 λ_y 的一组用户，称为属性用户组；$\mathcal{G}=\{G_1,\cdots,G_p\}\cup\{G_{\pi_1},\cdots,G_{\pi_m}\}$ 为全体属性用户组构成的集合；K_{λ_y} 为被属性用户组 G_y 中用户共享的属性用户组密钥。

5.2 方案结构

设 G_0 是阶为素数 p 的双线性群，g 是 G_0 的一个生成元，

安全参数 k 用于确定群的大小。定义双线性对 $e:G_0\times G_0\rightarrow G_1$;同时定义拉格朗日系数 $\Delta_{i,S}(x)=\prod_{j\in S,j\neq i}\frac{x-j}{i-j}$,$i\in Z_p^*$,$S$ 是属于 Z_p^* 的成员集。此外,我们还定义了哈希函数 $H:\{0,1\}^*\rightarrow G_0$ 和 $H_1:G_1\rightarrow Z_p^*$。

5.2.1　系统设置

系统初始化器根据安全参数选择一个以 g 为生成元的阶为素数 p 的双线性群 G_0,并选取哈希函数 $H:\{0,1\}^*\rightarrow G_0$ 和 $H_1:G_1\rightarrow Z_p^*$。公共参数 $param$ 由(G_0,g,H,H_1)组成。

密钥生成中心选取一个随机数 $\beta\in Z_p^*$,计算 $h=g^\beta$,主公钥和主密钥分别为 $PK_K=h$ 和 $MK_K=\beta$。

数据存储中心选取一个随机数 $\alpha\in Z_p^*$,主公钥和主密钥分别为 $PK_D=e(g,g)^\alpha$ 和 $MK_D=g^\alpha$。同时,数据存储中心选取另一个随机数 $\gamma\in Z_p^*$,并将 $PK_D^{\text{agree}}=g^\gamma$ 作为另外一个主公钥,γ 则保密。

5.2.2　密钥生成

当密钥生成中心验证一个用户 u_t 是可信的时,它为用户选取一个随机数 $r_t\in Z_p^*$,该随机数对用户而言是个性化的且是唯一的秘密,并且当用户 u_t 进行属性添加或撤销时 r_t 是不变的。假设 u_t 拥有一组属性 Λ,密钥生成中心对每一个属性 $\lambda_j\in\Lambda$ 随机选取 $r_j\in Z_p^*$;然后,密钥生成中心和数据存储中心通过文献[13]中的 2PC 安全协议共同生成如下用户密钥:

$$SK_{u_t}=(D=g^{(\alpha+r_t)/\beta},\forall\lambda_j\in\Lambda:D_j=g^{r_t}\cdot H(\lambda_j)^{r_j},D_j'=g^{r_j})$$

同时,数据存储中心为用户输出另外一个值 $SK_{u_t}^{\text{agree}}=H(ID_t)^\gamma=Q_t^\gamma$,该值将用于属性用户组密钥的分发。

5.2.3　数据加密

在数据拥有者将其数据 M 上传至数据存储中心用于共享之前,其会在访问树 T 下加密数据 M。如果 $M\in CDS$,通过“与”门将虚拟属性 λ_{π_i} 嵌入 T 中形成修正的访问树 T',并在 T' 下加密 M,如图 2 所示。数据拥有者通知密钥生成中心其所有的利益冲突数据集和相应的虚拟属性。

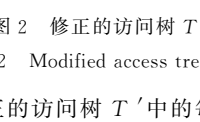

图 2　修正的访问树 T'

Fig. 2　Modified access tree T'

加密算法为修正的访问树 T'中的每一个节点 x 选择一个多项式 q_x,如何选择这些多项式将在文献[7]中详细描述。设 Y 是 T'中叶子节点的集合,算法计算密文如下:

$$CT=(T',\widetilde{C}=Me(g,g)^{as},C=h^s,\forall y\in Y:C_y=g^{q_y(0)},C_y'=H(\lambda_y)^{q_y(0)})$$

之后,数据拥有者将安全地把 CT 发送至数据存储中心。

5.2.4　数据重加密

在分发密文之前,数据存储中心通过使用出现在密文访问树 T'中的属性用户组 $G\subseteq\mathcal{G}$ 的一组成员信息运行重加密算法加密密文 CT。具体算法步骤如下:

1)对所有 $G_y\in G$,选择一个随机数 $K_{\lambda_y}\in Z_p^*$,重加密 CT 如下:

$$CT'=(T',\widetilde{C}=Me(g,g)^{as},C=h^s,\forall y\in Y:C_y=g^{q_y(0)},C_y'=(H(\lambda_y)^{q_y(0)})^{K_{\lambda_y}})$$

2)选择随机数 $\rho,R\in Z_p^*$,对 $\forall u_t\in G$,计算 $x_t=H_1(e(Q_t^\rho,PK_D^{\text{agree}}))$。每个 x_t 都可以在系统设置阶段一次性预先计算出来。

3)对所有 $G_y\in G$,构造多项式函数 $f^y(x)=\prod_{i=1}^{m}(x-x_i)=\sum_{i=0}^{m}a_ix^i\pmod p$,其中 $G_y=\{u_1,\cdots,u_m\}$;指数函数$\{P_0,\cdots,P_m\}=\{g^{a_0},\cdots,g^{a_m}\}$,$m$ 代表属性用户组 G_y 中用户的个数。

4)构造 $Hdr_y=\{K_{\lambda_y}\cdot P_0^R,P_1^R,\cdots,P_m^R\}$,并生成标头信息:$Hdr=(g^\rho,\forall y\in Y:Hdr_y)$。

通过标头信息 Hdr 将属性用户组密钥分发给有效用户。数据存储中心以(Hdr,CT')响应用户的数据访问请求。

5.2.5　数据解密

数据解密阶段包含以下两个步骤。

1)属性用户组密钥解密:用户 u_t 从数据存储中心收到(Hdr,CT')时,首先通过 Hdr 从他拥有的一组属性 Λ 中解密出所有属性用户组密钥。例如,用户 u_t 拥有属性 λ_j,则他能从 Hdr_j 中解密出属性用户组密钥 K_{λ_j}。首先,计算 $x_t=H_1(e(g^\rho,SK_{u_t}^{\text{agree}}))$;然后,计算 $K_{\lambda_j}\cdot P_0^R\cdot\prod_{i=1}^{m}(P_i^R)^{x_t^i}=K_{\lambda_j}\cdot g^{Rf^j(x_t)}=K_{\lambda_j}$。用户将解密出的所有属性用户组密钥用于更新其密钥,具体更新公式如下:

$$SK_{u_t}=(D=g^{(\alpha+r_t)/\beta},\forall\lambda_j\in\Lambda:D_j=g^{r_t}\cdot H(\lambda_j)^{r_j},D_j'=(g^{r_j})^{1/K_{\lambda_j}})$$

2)数据解密:解密算法是一个递归过程,用户使用其更新的密钥解密密文 CT'。定义递归算法 DecryptNode(CT',SK,x),并以密文 CT'、与一组属性 Λ 相关联的密钥 SK 和访问树 T'中的一个节点 x 为输入;输出群 G_1 中的一个元素或$\perp$。

不失一般性,假设用户 u_t 执行解密算法。如果 x 为一个叶子节点,则定义如下:如果 $\lambda_x\in\Lambda$ 以及 $u_t\in G_x$,计算:

$$\begin{aligned}&\text{DecryptNode}(CT',SK_{u_t},x)\\&=\frac{e(D_x,C_x)}{e(D_x',C_x')}=\frac{e(g^{r_t}\cdot H(\lambda_x)^{r_x},g^{q_x(0)})}{e((g^{r_x})^{1/K_{\lambda_x}},(H(\lambda_x)^{q_x(0)})^{K_{\lambda_x}})}\\&=e(g,g)^{r_tq_x(0)}\end{aligned}$$

如果 $u_t\notin G_x$,则 u_t 不能计算出 $e(g,g)^{r_tq_x(0)}$,因为 SK_{u_t}中 D_x'的指数无法包含 C_x'的指数 K_{λ_x} 的逆。因此,当 $u_t\notin G_x$ 或 $\lambda_x\notin\Lambda$ 时,定义 DecryptNode(CT',SK,x)$=\perp$。

如果 x 为一个非叶子节点,考虑递归的情况,算法 DecryptNode(CT',SK,x)的操作如下:对于节点 x 的所有子节点 z,令 $F_z=$DecryptNode(CT',SK,z)。假设存在一个随机的大小为 k_x 的节点集合 S_x,集合中的节点都是 x 的子节点且满足 $F_z\neq\perp$。若没有这样的集合存在,则节点 x 不满足算法,即 $F_x=\perp$;否则,计算:

$$\begin{aligned}F_x&=\prod_{z\in S_x}F_z^{\Delta_{i,S_x'}(0)}(i=index(z),S_x'=\{index(z):z\in S_x\})\\&=\prod_{z\in S_x}(e(g,g)^{r_t\cdot q_z(0)})^{\Delta_{i,S_x'}(0)}\\&=\prod_{z\in S_x}(e(g,g)^{r_t\cdot q_{parent(z)}(index(z))})^{\Delta_{i,S_x'}(0)}\\&=\prod_{z\in S_x}(e(g,g)^{r_t\cdot q_x(i)})^{\Delta_{i,S_x'}(0)}\\&=e(g,g)^{r_t\cdot q_x(0)}\end{aligned}$$

解密算法仅调用 DecryptNode 在访问树 T'根节点 R 的

函数值。如果用户 u_t 拥有的一组属性 Λ 满足访问树 T'，并且对所有 $\lambda_i \in \Lambda$ 的用户在属性用户组 G_i 中具有有效身份时，令 $A=\mathrm{DecryptNode}(CT', SK, R)=e(g,g)^{r_t s}$。算法解密密文如下：

$$\begin{aligned}\widetilde{C}/(e(C,D)/A)&=\widetilde{C}/(e(h^s, g^{(\alpha+r_t)/\beta})/e(g,g)^{r_t s})\\&=\widetilde{C}/(e(g^{\beta s}, g^{(\alpha+r_t)/\beta})/e(g,g)^{r_t s})\\&=Me(g,g)^{\alpha s}/e(g,g)^{\alpha s}\\&=M\end{aligned}$$

5.3 密钥更新

为了实施利益冲突数据集限制，当用户访问了某个利益冲突数据集 CDS 中的任意数据时，系统将撤销该用户拥有的对应于 CDS 的虚拟属性 λ_{π_i}。为了方便描述虚拟属性的撤销过程，本文定义一些符号，记 $AR(u,CDS)$ 表示 CDS 中用户 u 已访问过的和不受 $c(CDS)$ 影响的数据的集合，记 $FR(u,CDS)$ 表示 CDS 中用户 u 被禁止访问的数据的集合，记 $AU_k(CDS)(k\in\{1,2\})$ 表示已访问 CDS 中 $\{M_{k,1},\cdots,M_{k,n_k}\}$ 的数据的用户集。以下给出系统的工作流程，即实施本文限制的过程。

1)用户 u 开始访问 CDS 中的数据。u 向数据存储中心发出一个访问请求，不妨假设其第一个访问请求为 $M_1\in\{M_{1,1},\cdots,M_{1,n_1}\}$。如果 u 的有效属性满足密文 CT_1' 中的访问树，则 u 能成功解密 CT_1' 并访问 M_1；否则，解密失败。

2)数据存储中心通过 $\mathcal{G}$ 比对 M_1 中嵌入的访问树和属性用户组中 u 的有效成员信息，判断 u 是否有能力解密数据 M_1。如果 u 有能力解密，则数据存储中心对 $AR(u,CDS)$ 进行更新，如 $AR(u,CDS)=\{M_{1,1},\cdots,M_{1,n_1}\}$；否则 $AR(u,CDS)=\varnothing$。

3)一旦用户 u 成功地访问了 CDS 中的数据，密钥生成中心将撤销 u 的虚拟属性 λ_{π_i}。数据存储中心生成 $FR(u,CDS)=\{M_{2,1},\cdots,M_{2,n_2}\}$，同时更新 G_{π_i} 和 $AU_2(CDS)$ 中的用户成员关系。数据存储中心更新 λ_{π_i} 的属性用户组密钥，其不需要对与 λ_{π_i} 相关联的所有密文重加密，而仅需对 $FR(u,CDS)$ 中数据对应的密文重加密。以 $M_2\in\{M_{2,1},\cdots,M_{2,n_2}\}\in FR(u,CDS)$ 为例，此时，数据存储中心选择随机数 $s_1'\in Z_p^*$ 和 $K'_{\lambda_{\pi_i}}$ $(K'_{\lambda_{\pi_i}}\neq K_{\lambda_{\pi_i}})$；之后，数据存储中心使用公钥对 CT_2' 进行重加密，具体计算式如下：

$$CT_2'=(T_2', \widetilde{C}=M_2 e(g,g)^{\alpha(s_1+s_1')}, C=h^{s_1+s_1'}, C_{\pi_i}=g^{q_{\pi_i}(0)+s_1'}, C'_{\pi_i}=(H(\lambda_{\pi_i})^{q_{\pi_i}(0)+s_1'})^{K'_{\lambda_{\pi_i}}}, \forall y\in Y_2\setminus\{\lambda_{\pi_i}\}: C_y=g^{q_y(0)+s_1'}, C_y'=(H(\lambda_y)^{q_y(0)+s_1'})^{K_{\lambda_y}})$$

其中，Y_2 是加密 M_2 的修正的访问树 T_2' 中叶子节点的集合。

数据存储中心为新的属性用户组 G_{π_i} 和 $AU_2(CDS)$ 中的用户生成一个新的多项式函数 $f^{\pi_i}(x)$，通过 $K'_{\lambda_{\pi_i}}$ 计算新的标头信息 $Hdr=(g^{\rho}, Hdr_{\pi_i}, \forall y\in Y_2\setminus\{\lambda_{\pi_i}\}: Hdr_y)$。当任一用户向数据存储中心发出 M_2 的访问请求时，数据存储中心将使用在更新的密钥 $K'_{\lambda_{\pi_i}}$ 下重加密的 CT_2' 进行回应，即给他提供 (Hdr, CT_2')。此时，用户 u 访问数据 M_2 时 λ_{π_i} 已不再有效，即他不能对 CT_2' 成功解密，但他仍然能访问已经访问过的数据 M_1，即使他的虚拟属性 λ_{π_i} 已被撤销。而对于 $\forall u'\in G_{\pi_i}\cup AU_2(CDS)$，用户 u' 能通过新的标头信息 Hdr 更新其密钥，即将其密钥 $SK_{u'}$ 中的组件 $D'_{\pi_i}=(g^{r_{\pi_i}})^{1/K_{\lambda_{\pi_i}}}$ 更新为 $D'_{\pi_i}=((g^{r_{\pi_i}})^{1/K_{\lambda_{\pi_i}}})^{K_{\lambda_{\pi_i}}/K'_{\lambda_{\pi_i}}}=(g^{r_{\pi_i}})^{1/K'_{\lambda_{\pi_i}}}$，此时，用户 u' 可以按照5.2.5节中的数据解密算法对加密的 CT_2' 进行解密。

6 效率分析

本节在理论方面分析和比较本文方案与之前已提出的CP-ABE方案(即文献[7]的方案、文献[11]的方案和文献[13]的方案)的效率。

6.1 密钥托管和撤销

表1列出了对比方案和本文方案的撤销粒度和密钥托管情况。本文方案中的密钥更新可以立即进行，而文献[7]的方案只能定时更新。此外，由于本文方案是对每个属性而不是整个系统实现细粒度的用户撤销，因此本文方案中即使用户在服务中丢弃了某些属性，只要他拥有的其他属性还满足访问策略，他仍然可以访问加密数据。本文方案利用2PC安全协议的无代理密钥发布协议解决密钥托管问题。

表1 撤销粒度和密钥托管的比较

Table 1 Comparison of revocation and key escrow

方案	撤销粒度	密钥托管
文献[7]的方案	定时属性撤销	是
文献[11]的方案	即时属性撤销	是
文献[13]的方案	即时属性撤销	否
本文方案	即时属性撤销	否

6.2 效率比较

各方案的理论效率比较结果如表2所列，其中所用符号的说明如表3所列。

表2 通信开销与储存开销

Table 2 Communication cost and storage cost

方案	密文长度	系统公钥长度	密钥长度
文献[7]的方案	$(2t+1)L_0+L_1$	L_0+L_1	$(2k+1)L_0$
文献[11]的方案	$(w+1)L_0+L_1$	$(3w+1)L_0+L_1$	$(2w+1)L_0+L_k$
文献[13]的方案	$(2t+1)L_0+L_1$	L_0+L_1	$(2k+2)L_0$
本文方案	$(2t+1)L_0+L_1$	L_0+L_1	$(2k+2)L_0$

表3 符号说明

Table 3 Notations

符号	符号说明
L_0	G_0 中元素的长度
L_1	G_1 中元素的长度
L_k	与用户密钥相关联的一组属性的大小
t	与密文相关联的属性的个数
k	与用户密钥相关联的属性的个数
w	全体属性的大小

表2中对本文方案中的用户密钥长度、密文长度和系统公钥长度与文献[7]、文献[11]及文献[13]中的方案进行了比较。数据拥有者将密文上传至数据存储中心，用户从数据存储中心取得密文，因此密文长度反映了系统中的通信开销；同时，每个用户需要存储各自的密钥，因此密钥长度也反映了用户的储存开销。密钥生成中心和数据存储中心共同生成系统公钥，它储存在本地或是用户需要时向密钥生成中心和数据存储中心申请获取，分别对应系统储存和通信开销。由表2可以看出，本文方案的用户密钥较短，利于储存，且密文长度较短，利于通信传输。限制是访问控制中最高级别的策略，本文方案考虑计算复杂性效率，但更加注重实施利益冲突数据

集限制,即任何用户访问任一利益冲突数据集中的数据均不能产生利益冲突,避免给系统带来损失。

7　安全性分析

本节从数据机密性、防合谋、向后安全性和向前安全性以及利益冲突数据集限制等方面对本文提出的方案做简单的分析。

7.1　数据机密性

当一个用户拥有的一组属性不满足密文中嵌入的访问树时,他在解密过程中就不能恢复出 $e(g,g)^{rs}$,其中 r 对每个用户而言都是随机且唯一的。

7.2　防合谋

在密文策略属性基加密中,分享的秘密嵌入在密文而不是用户的密钥中。为了解密密文,若攻击者们必须恢复出 $e(g,g)^{\alpha s}$,则攻击者(假设攻击者没有属性 λ_y)必须使密文中的 C_y 部分与另一个拥有属性 λ_y 的合谋用户密钥中的 D_y 部分相匹配。然而,D_y 被随机数 r 盲化,对每个用户而言,r 都是随机选取的,而且不同的用户对应的 r 互不相同。因此,攻击者不能恢复出 $e(g,g)^{\alpha s}$。

7.3　向后安全性和向前安全性

鉴于普通数据不受限制的影响,本文遵循文献[13]的方案处理普通数据以保证向后安全性和向前安全性。本文主要分析利益冲突数据集中数据的向后安全性和向前安全性。

不妨以 $CDS=(\{M_{1,1},\cdots,M_{1,n_1}\},\{M_{2,1},\cdots,M_{2,n_2}\})$ 为例。如果用户 u 被撤销了虚拟属性 λ_{π_i},则对 u 而言,CDS 中的数据分为 $AR(u,CDS)$ 和 $FR(u,CDS)$ 两个部分。对于 $FR(u,CDS)$ 中的数据而言,用户不能从标头信息 Hdr 中解密出更新的属性用户组密钥 $K'_{\lambda_{\pi_i}}$。因此,用户不能解密 $FR(u,CDS)$ 中任何数据对应的密文,即对 $FR(u,CDS)$ 中的数据满足向后安全性和向前安全性。但如果用户拥有的普通属性满足 $AR(u,CDS)$ 中数据嵌入的原访问树。则用户能解密 $AR(u,CDS)$ 中数据对应的密文。

7.4　利益冲突数据集限制

当用户访问了某个利益冲突数据集中的数据时,系统撤销用户默认拥有的对应于该数据集限制的虚拟属性。此时,对用户而言,该利益冲突数据集中的数据分为两个部分,用户仅能访问其中一部分的数据,如果用户拥有的普通属性满足该数据对应的原访问树。即用户对利益冲突数据集的访问不会违反本文限制。

结束语　本文首先介绍了利益冲突数据集的概念,并定义了虚拟属性用于实施利益冲突数据集限制。通过"与"门将虚拟属性嵌入访问树中得到修正的访问树,在修正的访问树下对利益冲突数据集中的数据进行加密。当用户访问了利益冲突数据集中的数据后,系统将撤销其相应的虚拟属性。当用户继续发出访问利益冲突数据集中被禁止访问的数据的请求时,数据存储中心将使用在更新的属性用户组密钥下重加密的密文进行回应。此时,用户在解密密文时已失去相应的虚拟属性,即不能成功解密;但他仍可访问利益冲突数据集中已访问过的数据,即使其相应的虚拟属性已被撤销。因此,在保证向后安全性和向前安全性的基础上,本文提出的方案实现了利益冲突数据集限制,从而降低了访问可能带来的损失、欺骗或风险。

参 考 文 献

[1] MELL P,GRANCE T. The NIST definition of cloud computing [J]. Communications of the ACM,2011,53(6):50.

[2] SAHAI A,WATERS B. Fuzzy identity-based encryption[C]// International Conference on Theory and Applications of Cryptographic Techniques. Springer-Verlag,2005:457-473.

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

[4] OSTROVSKY R,SAHAI A,WATERS B. Attribute-based encryption with non-monotonic access structures[C]// ACM Conference on Computer & Communications Security. 2007:195-203.

[5] ATTRAPADUNG N,IMAI H. Conjunctive broadcast and attribute-based encryption[C]// Third International Conference. DBLP,2009:248-265.

[6] ATTRAPADUNG N,IMAI H. Attribute-based encryption supporting direct/indirect revocation modes[C]// Ima International Conference on Cryptography and Coding. Springer-Verlag,2009:278-300.

[7] BETHENCOURT J,SAHAI A,WATERS B. Ciphertext-policy attribute-based encryption[C]// IEEE Symposium on Security and Privacy. IEEE Computer Society,2007:321-334.

[8] BREWER D F C,NASH M J. The chinese wall security policy [C]// IEEE Symposium on Security and Privacy. IEEE Xplore,1989:206-214.

[9] BARACALDO N,JOSHI J. A trust-and-risk aware RBAC framework:tackling insider threat[C]// Proceedings of the 17th ACM Symposium on Access Control Models and Technologies. ACM,2012:167-176.

[10] HELIL N,RAHMAN K. CP-ABE access control scheme for sensitive data set constraint with hidden access policy and constraint policy[J]. Security & Communication Networks,2017,2017(6):1-13.

[11] YU S,WANG C,REN K,et al. Attribute based data sharing with attribute revocation[C]// ACM Symposium on Information,Computer and Communications Security. ACM,2010:261-270.

[12] YANG K,JIA X,REN K. Attribute-based fine-grained access control with efficient revocation in cloud storage systems[C]// ACM Sigsac Symposium on Information,Computer and Communications Security. ACM,2013:523-528.

[13] HUR J. Improving security and efficiency in attribute-based data sharing[J]. IEEE Transactions on Knowledge & Data Engineering,2013,25(10):2271-2282.

[14] ZU L,LIU Z,LI J. New ciphertext-policy attribute-based encryption with efficient revocation[C]// IEEE International Conference on Computer and Information Technology. IEEE,2014:281-287.

[15] WANG P P,FENG D G,ZHANG L W. CP-ABE scheme supporting fully fine-grained attribute revocation[J]. Journal of Software,2012,23(10):2805-2816. (in Chinese)
王鹏翩,冯登国,张立武. 一种支持完全细粒度属性撤销的 CP-

ABE 方案[J]. 软件学报,2012,23(10):2805-2816.

[16] 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.

[17] FENG D G,CHEN C. Research on attribute-based cryptography [J]. Journal of Cryptologic Research,2014,1(1):1-12. (in Chinese)
冯登国,陈成. 属性密码学研究[J]. 密码学报,2014,1(1):1-12.

[18] YAN X X,MENG H. Ciphertext policy attribute-based encryption schemesupporting direct revocation[J]. Journal on Communications,2016,37(5):44-50. (in Chinese)
闫玺玺,孟慧. 支持直接撤销的密文策略属性基加密方案[J]. 通信学报,2016,37(5):44-50.

[19] ZHANG K,MA J F,LI H,et al. Multi-authority attribute-based encryption with efficient revocation[J]. Journal on Communications,2017,38(3):83-91. (in Chinese)
张凯,马建峰,李辉,等. 支持高效撤销的多机构属性加密方案[J]. 通信学报,2017,38(3):83-91.

[20] SHAN Z Y,SUN Y F. A study of security attributes immediatere vocation in secure OS[J]. Journal of Computer Research and Development,2002,39(12):1680-1688. (in Chinese)
单智勇,孙玉芳. 安全操作系统安全属性即时撤消研究[J]. 计算机研究与发展,2002,39(12):1680-1688.

[21] FANG L,YIN L H,GUO Y C,et al. A survey of key technologies in attribute-based access control scheme[J]. Chinese Journal of Computers,2017,40(7):1680-1698. (in Chinese)
房梁,殷丽华,郭云川,等. 基于属性的访问控制关键技术研究综述[J]. 计算机学报,2017,40(7):1680-1698.

(上接第 137 页)

开销,实验结果如图 8 所示。从图 8 中可看出,批量校验有效降低了 TPA 端的审计开销,且本文方案的 TPA 批量审计效率优于 Wang 等的方案,与性能分析的析结果一致。

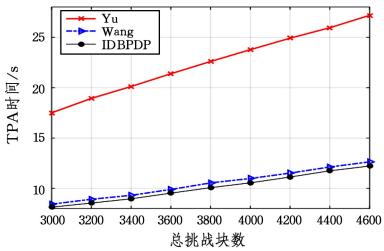

图 8 总挑战块数增加时 TPAverify 的计算开销对比

Fig. 8 Comparison of computational cost for TPAverify under increased total number of challenged blocks

以上实验表明,本文方案是高效且可行的。

结束语 本文提出了一个多服务器环境下支持完美隐私保护的批处理 PDP 方案,并对方案的正确性和安全性进行了证明。最后,将本文方案与在多服务器环境下简单拓展的 Yu 等的方案和 Wang 等的方案进行了理论分析对比与实验对比。实验结果表明,本文方案是高效且可行的。

参 考 文 献

[1] ATENIESE G,BURNS R,CURTMOLA R,et al. Provable data possession at untrusted stores[C]// ACM Conference on Computer and Communications Security. ACM,2007:598-609.

[2] ATENIESE G,PIETRO R D,MANCINI L V,et al. Scalable and efficient provable data possession[C]// Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks. ACM,2008:1-10.

[3] WANG Q,WANG C,LI J,et al. Enabling public verifiability and data dynamics for storage security in cloud computing[C]// European Conference on Research in Computer Security. Springer-Verlag,2009:355-370.

[4] ZHANG J,TANG W,MAO J. Efficient public verification proof of retrievability scheme in cloud[M]. Kluwer Academic Publishers,2014.

[5] YU Y,NI J,MAN H A,et al. Comments on a Public Auditing Mechanism for Shared Cloud Data Service[J]. IEEE Transactions on Services Computing,2015,8(6):998-999.

[6] YU Y,LI Y,NI J,et al. Comments on "Public Integrity Auditing for Dynamic Data Sharing With Multiuser Modification" [J]. IEEE Transactions on Information Forensics & Security,2016,11(3):658-659.

[7] YU Y,XUE L,MAN H A,et al. Cloud data integrity checking with an identity-based auditing mechanism from RSA[J]. Future Generation Computer Systems,2016,62(C):85-91.

[8] WANG C,WANG Q,REN K,et al. Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing[C]// Infocom,2010 Proceedings IEEE. IEEE,2010:1-9.

[9] HAO Z,ZHONG S,YU N. A Privacy-Preserving Remote Data Integrity Checking Protocol with Data Dynamics and Public Verifiability[J]. IEEE Transactions on Knowledge & Data Engineering,2011,23(9):1432-1437.

[10] YU Y,AU M H,MU Y,et al. Enhanced privacy of a remote data integrity-checking protocol for secure cloud storage[J]. International Journal of Information Security,2015,14(4):307-318.

[11] YU Y,MAN H A A,ATENIESE G,et al. Identity-based Remote Data Integrity Checking with Perfect Data Privacy Preserving for Cloud Storage[J]. IEEE Transactions on Information Forensics & Security,2017,PP(99):1-1.

[12] ZHU Y,HU H,AHN G J,et al. Cooperative Provable Data Possession for Integrity Verification in Multicloud Storage [J]. IEEE Transactions on Parallel & Distributed Systems,2012,23(12):2231-2244.

[13] HE K,HUANG C,WANG J,et al. An efficient public batch auditing protocol for data security in multi-cloud storage[C]// 2013 8th ChinaGrid Annual Conference. IEEE,2013:51-56.

[14] WANG H. Identity-Based Distributed Provable Data Possession in Multicloud Storage[J]. IEEE Transactions on Services Computing,2015,8(2):328-340.

[15] ATENIESE G,BURNS R,CURTMOLA R,et al. Remote data checking using provable data possession[J]. Acm Transactions on Information & System Security,2011,14(1):1-34.