Journal of Computer Applications ISSN 1001-9081 2015-06-10
计算机应用,2015,35(6):1573 - 1579 CODEN JYIIDU http://www.joca.cn

文章编号:1001-9081(2015)06-1573-07 doi:10.11772/j.issn.1001-9081.2015.06.1573

基于密文策略属性加密体制的匿名云存储隐私保护方案

徐 潜*,谭成翔

(同济大学 电子与信息工程学院,上海 201804)

(* 通信作者电子邮箱 1062842783@qq.com)

摘 要: 针对云存储中数据机密性问题,为解决密钥泄漏与属性撤销问题,从数据的机密性存储以及访问的不可区分性两个方面设计了基于密文策略属性加密体制(CP_ABE)的匿名云存储隐私保护方案。提出了关于密钥泄漏的前向安全的不可逆密钥更新算法;在层次化用户组以及改进的 Subset-Difference 算法基础上,利用云端数据重加密算法实现属性的细粒度撤销;基于同态加密算法实现 k 匿名 l 多样性数据请求,隐藏用户潜在兴趣,并在数据应答中插入数据的二次加密,满足关于密钥泄漏的后向安全。在标准安全模型下,基于 l 阶双线性 Diffie-Hellman(判定性 l-BDHE)假设给出所提出方案的选择性安全证明,并分别从计算开销、密钥长度以及安全性等方面验证了方案的性能优势。

关键词: 密文策略基于属性加密体制;可证明安全;重加密;密钥泄漏;属性撤销

中图分类号: TP309 **文献标志码:** A

Anonymous privacy-preserving scheme for cloud storage based on CP_ABE

XU Qian*, TAN Chengxiang

(*College of Electrical and Information Engineering, Tongji University, Shanghai* 201804, *China*)

Abstract: In order to solve the confidentiality issues such as key exposure and attribute revocation of data stored in cloud server, an advanced anonymous privacy-preserving scheme based on Ciphertext-Policy Attributed-Based Encryption (CP_ABE) was proposed by considering confidentiality of data storage and indistinguishability of access. First, the scheme constructed a forward-secure irreversible key-update algorithm to solve key exposure. On the basis of the classified user-group and the advanced Subset-Difference algorithm, fine-grained attribute revocation was implemented with the help of cloud data re-encryption algorithm. The potential interests of user would be concealed when k-anonymity l-diversity data request was introduced based on the homomorphic encryption algorithm. The backward-security of key exposure was realized on the basis of secondary encryption inserted in data response. Under the l-Bilinear Diffie-Hellman Exponent Problem (l-BDHE) assumption, selective security of the proposed scheme was proved in the standard model. The performance advantage of the proposed scheme was demonstrated respectively in terms of efficiency, key length and security.

Key words: Ciphertext-Policy Attributed-Based Encryption (CP_ABE); provable security; re-encryption; key exposure; attribute revocation

0 引言

云存储作为云计算的延伸和发展,其最大特点是存储即服务。由于用户将数据上传到云服务器的同时失去了对数据的绝对控制权,因此如何在保证用户隐私和数据安全的同时尽可能地提高服务质量已经成为安全云存储的关键问题。

云存储中关心的数据的机密性问题包含两个方面,首先是数据存储的机密性,即对于云服务器的不可见性,这一部分可由层次化的加密算法实现。基于属性的加密算法(Attribute-Based Encryption, ABE)是由基于身份的加密算法(Identity-Based Encryption, IBE)[1]发展而来。由于 ABE 算法通过访问结构关联密文与用户,提高了系统的访问效率,放宽了对服务器与访问存储器的安全限制,因此被广泛应用在云存储的访问控制中。其次是数据访问的不可区分性,由于云服务器存在的"诚实但好奇"的特性,即诚实地执行用户的要求,但存在窥探用户数据隐私的可能性。即使数据在服务器中以密文形式保存,服务商也可以在统计用户对密文请求次数的基础上建立用户与特定密文的关系,挖掘潜在的用户兴趣。

1 相关研究

Sahai 等[1]在 2005 年提出了属性加密算法 ABE,只有满足数据属主定义的属性集合的用户才可以对密文进行解密。之后 Goyal 等[2]基于 ABE 的概念,将访问结构与密文或者密钥关联,把 ABE 划分为基于密文的 ABE(Ciphertext-Policy ABE, CP_ABE)和基于密钥的 ABE(Key-Policy ABE, KP_ABE)。

文献[3]将 CP_ABE 应用到云存储中,提出了细粒度访问控制和确认删除云存储方案,简称 FADE 方案,但是无法抵御密钥泄漏以及合谋攻击。文献[4]提出的基于属性的云存储控制方法(Attribute-Based Access Control for Cloud Storage, AB-ACCS),通过私钥属性与密文属性的匹配关系确定访问控制能力,但无法对用户属性单独撤销,粒度过粗。文献[5]在文献[4]的基础上通过代理重加密的方法灵活地控制属性撤销,实现动态的访问控制策略,但是依然无法解决密钥泄漏带

收稿日期:2014-12-24;修回日期:2015-03-09。

作者简介:徐潜(1986 -),男,黑龙江哈尔滨人,博士研究生,主要研究方向:移动网络隐私保护、安全云存储; 谭成翔(1965 -),男,湖北红安人,教授,博士生导师,主要研究方向:网络安全、分布式计算。

来的安全隐患。魏江宏等[6]利用分层的身份加密的思想，通过离散化私钥生命周期的方法实现了前向安全的CP_ABE方案，但是没有考虑后向安全性，且缺乏对属性撤销的支持。王鹏翩等[7]采用合数阶双线性群双系统加密的方法实现细粒度的基于CP_ABE的访问控制，但是公钥长度与用户数量线性相关，这在云存储环境中容易造成公钥长度过长的问题。此外，Yu[8]、Hur[9]、Attrapadung[10]等也都提出了改进的CP_ABE方案，但是均存在密钥过长以及计算复杂度高等问题，并且在属性撤销方面粒度过粗，大多基于用户身份的属性撤销，使得加密方案在云存储环境中的应用受到限制。

本文从数据的机密性存储以及访问的不可区分性两个方面设计基于CP_ABE的匿名云存储的隐私保护方案(Anonymous Privacy-Preserving scheme for cloud storage based on CP_ABE，APPCP_ABE)，着重解决密钥泄漏以及属性撤销问题。主要研究内容如下：

1) 在魏江宏[6]的基础上改进了CP_ABE加密算法，通过维护离散化时间序列二叉树实现高效、不可逆的密钥更新方案，以满足密钥泄漏的前向安全性。

2) 层次化用户与属性的关系，提出用户组和用户组二叉树的概念，通过设计改进的Subset-Difference(Advanced Subset Difference，Adv-Subset-Difference)算法实现属性级别的细粒度的属性撤销；并证明了提出的撤销算法满足前向安全与后向安全性。

3) 基于同态加密算法并利用时间周期的二元序列的唯一性，使得用户向云端提交的数据请求具有k匿名l多样性，且不需引入额外的可信机构，在保证访问的不可区分性的同时抵御拒绝服务(Deny of Service，DoS)与重放攻击。通过在数据应答中插入密文的二次代理重加密，实现方案关于密钥泄漏的后向安全性。

4) 基于l-双线性Diffie-Hellman(Bilinear Diffie-Hellman Exponent，BDHE)假设，在标准安全模型上证明了提出方案的选择安全性。

2 预备知识

定义1 访问结构。假定在参与方集合$P=\{P_1,P_2,\cdots,P_n\}$上共享了一个秘密，定义P的一个非空子集，若N能恢复P上共享的秘密，则称N为授权子集，否则为非授权子集。所有授权子集构成的集簇，称为该秘密的访问结构。称Γ为单调的，如果$A\in\Gamma$，$A\subseteq B\subseteq P$则有$B\in\Gamma$。

定义2 双线性映射。设G_1和G_2是两个p阶循环群，其中p为大素数。设g为G_1的生成元，双线性映射$e:G_1\times G_1\rightarrow G_2$满足如下条件：

1) 双线性性：对任意的$x,y\in G_1$，$a,b\in\mathbf{Z}_p$，有$e(x^a,y^b)=e(x,y)^{ab}$。

2) 非退化性：$e(g,g)\neq 1_{G_2}$。

3) 可计算性：对任意$x,y\in G_1$，存在一个有效的多项式时间算法计算$e(x,y)$。

定义3 线性秘密共享(Linear Secret Sharing Scheme，LSSS)。一个定义在实体集P上的线性秘密共享方案Π是指：

1) 所有实体的共享组成$\mathbf{Z}_p$上的一个向量。

2) 存在一个$l\times n$的共享生成矩阵$\boldsymbol{M}$和一个从$\{1,2,\cdots,l\}$到P的单射ρ，随机选取向量$\boldsymbol{v}=(s,v_2,\cdots,v_n)\in\mathbf{Z}_p$，其中$s$是要共享的秘密，则$\boldsymbol{Mv}^{\mathrm{T}}$就是利用$\Pi$得到的关于$s$的$l$个共享组成的向量，其中共享$(\boldsymbol{Mv}^{\mathrm{T}})_i$属于实体$\rho(i)$，表示为$\lambda_i=(\boldsymbol{Mv}^{\mathrm{T}})_i$。

按照上述方法构造的LSSS具有线性可重构性：假设Π是一个针对访问结构Λ的LSSS，对授权用户集$S\in\Lambda$，定义$I=\{i:\rho(i)\in S\}\subseteq\{1,2,\cdots,l\}$，存在向量$\boldsymbol{w}=\{w_i\in\mathbf{Z}_p\}_{i\in I}$使得$\sum_{i\in I}w_i\boldsymbol{M}_i=(1,0,\cdots,0)$，其中$\boldsymbol{M}_i$为$\boldsymbol{M}$的第$i$行，从而得到：$\sum_{i\in I}w_i\boldsymbol{M}_i\boldsymbol{v}^{\mathrm{T}}=\sum_{i\in I}w_i\lambda_i=s$。对于非授权用户集，存在向量$\boldsymbol{w}=\{w_i\in\mathbf{Z}_p\}$，使得$\boldsymbol{w}(1,0,\cdots,0)^{\mathrm{T}}=-1$且对$\forall i\in I$有$\boldsymbol{wM}_i^{\mathrm{T}}=0$。

定义4 判定性l-BDHE假设：假定G_1和G_2是两个阶为大素数p的循环群，$e:G_1\times G_1\rightarrow G_2$是双线性映射，$g$是$G_1$的生成元，随机数$a,s\in\mathbf{Z}_p$，设$1\leqslant k\leqslant 2l$，输入为向量$\boldsymbol{y}=(g,g^s,g_1,g_2,\cdots,g_l,g_{l+2},\cdots,g_{2l})$，其中$g_k=g^{ak}$；则$(G_1,G_2)$上的判定性$l$-BDHE假设是指：给定实例$(G_1,G_2,p,e,\boldsymbol{y},T')$，以$\boldsymbol{y}$为输入的算法$B$能在概率多项式时间内判断$G_2$群内元素$T'=e(g^s,g_{l+1})$或者$T'$为$G_2$内随机元素；且若有$T'=e(g^s,g_{l+1})$则算法$B$输出0，否则输出1。定义$B$解决判定性$l$-BDHE假设的优势如下：$Adv_B=\left|[B(\boldsymbol{y},T'=e(g^s,g_{l+1}))=0]-\Pr[B(\boldsymbol{y},T'=\theta)=1]\right|$。

若Adv_B是可忽略的，则称实例$(G_1,G_2,p,e,\boldsymbol{y},T')$满足$(G_1,G_2)$上的判定性$l$-BDHE假设。

3 形式化定义与核心思路

3.1 APPCP_ABE的形式化定义

定义5 用户组。设密文CT的访问结构Γ对应的属性集合为U，$u_i\in U$，设用户群中，拥有属性u_i的用户构成属性u_i的用户组$UsrGroup_i$，U与所对应的用户组的集合$\{UsrGroup\}$构成二部图。

选择性安全的云存储隐私保护方案由以下九个多项式时间算法组成：

1) 系统初始化Setup：根据预定义的G_1和G_2以及T等参数，输出系统公共参数pub以及系统主密钥msk。

2) 用户向云服务器发送注册申请Register并生成二元序列bin。

3) 密钥生成KeyGen：根据系统主密钥msk以及公共参数pub和用户属性集S等，输出时间周期t_0时用户私钥Usk_S^{t0}，同时生成用于同态加密的公私密钥对(epk,esk)。

4) 密钥更新KeyUpdate：根据公共参数pub、当前时间周期用户私钥$Usk_S^{t_y}$、下一时间周期t_n，生成$Usk_S^{t_n}$。

5) 加密算法Encrypt：根据公共参数pub、明文m、时间周期t_n以及访问结构Γ，生成密文CT。

6) 代理重加密ReEncrypt：根据访问结构Γ、密文CT以及Γ中每个属性对应的用户组，生成重加密密文CT_R。

7) 数据包请求DataRequest：根据用户对云端数据包请求的聚类，设计k匿名l差异性的数据包请求，并于数据应答中插入数据的二次加密。

8) 解密算法Decrypt：根据公共参数pub、用户属性集S以及对应的时间周期t_v、密钥$Usk_S^{t_v}$，输出明文m。

9) 属性更新KeyProUpdate：撤销属性相关的密文CT_R，更新的属性集S'，生成新的密文CT'_R。

3.2 APPCP_ABE的核心思路

常规的基于CP_ABE安全方案分为四个算法：系统初始

化、密钥生成、加密和解密。本文提出 APPCP_ABE 模型在常规 CP_ABE 算法基础上侧重于解决密钥泄漏和属性撤销问题,通过密文组件和两次代理加密将两个目标相关联。

为解决魏江宏等[6]模型中密钥泄漏后向安全问题,引入客户端的注册过程 Register,保证不同的用户在云端拥有不同的随机 ID 序列。在密钥生成和更新算法中基于 ID 序列的模 2 加性,实现随机 ID 与密钥的绑定。数据请求 DataRequest 算法除了实现 k 匿名 l 多样性的数据包外,还包含了密文的二次加密,从而使不同用户基于相同密钥更新得到不同的新密钥,而只有合法用户才可以解密云端的密文,实现了密钥泄漏的后向安全性。

属性撤销方面,文献[7]的方法无法满足前向安全;而文献[8-10]的撤销算法是基于用户身份的,粒度过粗;文献[11-12]的撤销算法是基于更新时间的,粒度粗且无法实时变更属性。APPCP_ABE 方案的代理重加密算法 ReEncrypt 实现了用户级别细粒度的属性控制。属性更新 KeyProUpdate 通过改进的 Adv-Subset-Difference 算法,利用云端计算能力在线性时间内完成密文和用户私密信息更新,实现前后向安全的属性撤销。

3.3　APPCP_ABE 的标准安全模型

通过一个攻击游戏来定义 APPCP_ABE 方案的标准安全模型[6]。

1) Init: 敌手选择并公布挑战的访问结构 Γ^*,挑战的时间周期 t_c^*。

2) Setup: 生成 pub 以及 msk,并将 pub 发送给敌手 A,同时保存 msk。

3) Query1: 敌手进行多项式次数的关于属性集合 S、时间周期 t_v 的私钥询问。其中 S 和 t_v 均不满足访问结构 Γ^* 和挑战时间周期 t_c^*。挑战者运行 KeyGen 算法计算私钥 $Usk_S^{t_v}$。

4) Challenge: 敌手选择两个等长的密文 m_0、m_1,挑战者抛硬币并从 m_0、m_1 中等概率选择明文 m_θ 进行加密,将密文返回给敌手。

5) Query2: 敌手继续进行多项式时间的私钥提问,过程与 Query1 相同。

6) Guess: 敌手输出对 θ 的猜测 θ',如果 $\theta = \theta'$,称敌手赢得游戏。

定义 6　如果多项式时间的敌手 A 赢得上述游戏的攻击优势 Adv_A 是可忽略的,则称隐私保护方案关于标准安全模型是选择性安全的。

4　APPCP_ABE 的实施

4.1　方案基本定理

定义 7　完全二叉树 τ 深度为 l,共 2^l 个叶子节点分别对应 2^l 个时间周期 $t_{0^l}, t_{0^{l-1}1}, \cdots, t_{1^l}$。对于任意节点 v,若其深度为 k,则有长度为 k 的序列 $b_v \in \{0,1\}^k$ 对应 v,表示根节点 τ 到 v 的路径,其中 0 和 1 代表当前节点是父节点的左子节点或右子节点 $R(v)$,b_v 也唯一对应一个节点 v。设 $Path_v$ 表示根节点 τ 到叶子 v 的路径,则有集合 V_v 且 $V_v = \{R(v_i) \mid v_i \in Path_v, R(v_i) \notin Path_v\} \cup \{v\}$。

定理 1　设时间周期 t_i、t_j,如果有 $t_i \leqslant t_j$,则对任意节点 $v \in V_j$,存在节点 $v' \in V_i$ 且 $b_j = b_i \| b^*$,其中 $b^* \in \{0,1\}^{0 \leqslant k \leqslant l}$。

证明　令 w 为 $Path_i$、$Path_j$ 的第一个交点,对任意的 $v \in V_j$,若有 $|b_v| < |b_w|$,必有 $v \in V_i$,令 $b^* = 0$ 得证;否则,显然有 $R(w) \in Path_j$,且 $R(w) \notin Path_i$,故 $R(w) \in V_i$。又因为 $R(w) \in Path_j$,因此 $b_j = R(w) \| b^*$,从而令 $v' = R(w) \in V_i$,定理 1 得证。

定理 2　设 χ 为离散 Gaussian 误差分布,标准差为 σ。给定有限域 $F_{p^2} = \mathbf{Z}_p[x]/(x^2+1) = GF(p^2)$,设随机数 $psk_i \in \chi$ 且随机数 $a_{i1} \in F_{p^2}$,设 $q \in F_{p^2}^*$ 为本原元。设 $a_{i0} = -(a_{i1} psk_i + q)$,令公钥为 $epk_i = (a_{i0}, a_{i1})$,私钥为 $esk_i = psk_i$。加密函数为 $E_{epk_i}(m)$: 选定样本 $u \in \chi$,则密文为 $E_{epk_i}(m) = (c_0, c_1)$,其中: $c_0 = a_{i0}uq + m$,$c_1 = a_{i1}uq$。则:

1) 解密函数: $D_{esk_i}(C) = m = c_0 + c_1 psk_i \bmod q$。

2) 满足同态性:

$E_{epk}(m_0) + E_{epk}(m_1) = E_{epk}(m_0 + m_1)$

$E_{epk}(m_0) \times E_{epk}(m_1) = E_{epk}(m_0 * m_1)$

4.2　方案的具体实现

4.2.1　系统初始化 Setup

定义系统时间周期总数 $T = 2^l$ 以及安全参数 κ,阶为大素数 p 的循环群 G_1 和 G_2,双线性映射 $e: G_1 \times G_1 \to G_2$。定义 Hash 函数 H 为: $\{0,1\}^* \to G_1$,选取随机数 $a, \alpha, \beta, q \in \mathbf{Z}_p^*$,$Z = e(g,g)^\alpha$,给定离散 Gaussian 误差分布 χ 且标准差为 σ,定义向量 $\boldsymbol{u} = (u_1, u_2, \cdots, u_l) \in G_1^l$。定义伪随机序列生成器 RG: $\{0,1\}^l \to \{0,1\}^{3l}$,定义 RG_{L} 为取随机生成序列左 1/3 部分,RG_{M} 为中间 1/3,RG_{R} 为右边 1/3 部分,RG_K 为取指定二元序列的左边 K 位。

系统主密钥 $msk = (\beta, g^\alpha)$,系统公共参数 $pub = (G_1, G_2, e, p, q, g, g^\beta, g^{\frac{1}{\beta}}, T, \boldsymbol{u})$。

4.2.2　用户向云服务器发送注册申请 Register

云服务器取种子 $bin \in \mathbf{Z}_2^l$ 并计算 $bin_i = RG_{\mathrm{L}}(bin) \oplus RG_{\mathrm{R}}(bin)$,且 $bin_i \in \mathbf{Z}_2^l$ 作为用户 usr_i 的 ID 并私密保存,$|bin_i| = l$。

4.2.3　密钥生成 KeyGen

已知系统公共参数 pub、主密钥 msk、用户属性集合 S 以及时间周期开始时刻 t_0,选择随机数 $t \in \mathbf{Z}_p$,对任意节点 $v \in V_0$,随机选择 $r_v \in \mathbf{Z}_p$。用户私钥为 $Usk_S^{t_0} = \{\{K_v \mid v \in V_0\}, \forall i \in I, D_i, D'_i, epk, esk\}$,其中 $I = \{i: \rho(i) \in S\} \subseteq \{1,2,\cdots,l\}$。各个部分计算如下:

$K_v = (d_0, d_1, d_1^*, \cdots, d_l^*)$,其中:

$$d_0 = g^{\frac{(\alpha+at)}{\beta}} \left(\prod_{k=1}^{|b_v|} u_k^{b_{vk}[k]} \right)^{r_v} \left(\prod_{k=1}^{|bin_i|} u_k^{bin_{ik}[k]} \right)^{r_v}$$

$$d_1 = g^{\beta r_v}, d_k^* = u_k^{r_v}; 1 \leqslant k \leqslant l$$

对 $\forall i \in I, D_i = g^t, D'_i = H(i)^t$,$epk$、$esk$ 按定理 2 所述生成。

4.2.4　密钥更新 KeyUpdate

已知系统公共参数 pub,时间周期由 t_v 跃迁至 t_w,可由私钥 $Usk_S^{t_v}$ 生成私钥 $Usk_S^{t_w}$ 如下: 用户保持 D_i、D'_i 与 epk、esk 不变。

用户向云端发起密钥更新请求,云端根据用户的二元序列 bin_i,将 bin_i 作为种子代入 RG_{M} 中并计算 $bin'_i = RG_{\mathrm{M}}(bin_i)$,再计算 $bin''_i = bin_i \oplus bin'_i$,将 bin''_i 作为新用户 ID。由定理 1,对任意节点 $w \in V_w$,存在节点 $v \in V_v$ 使得 $b_w = b_v \| b^*$。生成随机数 $r_w \in \mathbf{Z}_p$,有:

$$d'_0 = d_0\Big(\prod_{k=|b_v|+1}^{|b_w|} d_k^{*\,b_w[k]}\Big)\Big(\prod_{k=1}^{|b_w|} u_k^{b_w[k]}\Big)^{r_w}\cdot \Big(\prod_{k=1}^{|bin'_i|} d_k^{*\,bin'_i[k]}\Big)\Big(\prod_{k=1}^{|bin''_i|} u_k^{bin''_i[k]}\Big)^{r_w} = g^{\frac{(\alpha+at)}{\beta}}\Big(\prod_{k=1}^{|b_w|} u_k^{b_w[k]}\Big)^{(r_v+r_w)}\Big(\prod_{k=1}^{|bin''_i|} u_k^{bin''_i[k]}\Big)^{(r_v+r_w)}$$

注意 bin 上为模 2 加运算。

$$d'_1 = d_1 g^{\beta r_w} = g^{\beta(r_v+r_w)}$$

$$d^{*\prime}_k = d_k^* u_k^{r_w} = u_k^{r_v+r_w};1 \leqslant k \leqslant l$$

云服务器将密钥组件: $\forall v \in V_w, K_v = (d'_0, d'_1, d^{*\prime}_1, \cdots, d^{*\prime}_l)$ 发送给用户。

4.2.5 加密算法 Encrypt

已知系统公共参数 pub,访问结构 $(\boldsymbol{M},\rho)$,其中 $\boldsymbol{M}$ 为 $\ell \times n$ 实矩阵,待加密消息为 m,时间周期标识为 t_v,选择随机数 s 并生成向量 $\boldsymbol{v} = (s, v_2, \cdots, v_n) \in \mathbf{Z}_p$,得线性共享 $\lambda_i = (\boldsymbol{M}\boldsymbol{v}^{\mathrm{T}})_i$。计算 $C = mZ^s = me(g,g)^{\alpha s}, C_1 = (g^{\beta})^s = g^{\beta s}, C_2 = \Big(\prod_{k=1}^{l} u_k^{b_v[k]}\Big)^s$。

对 $1 \leqslant i \leqslant \ell$,$C_i = g^{a\lambda_i} H(i)^{-r_i}, C'_i = g^{r_i}$,随机数 $r_i \in \mathbf{Z}_p$,密文为 $CT = \{C, C_1, C_2, C_i, C'_i \mid 1 \leqslant i \leqslant \ell\}$。

4.2.6 代理重加密 ReEncrypt

已知访问结构 Γ,相应的密文 CT,假设 Γ 对应的属性集合为 $S = \{u_i \mid u_i \in S\}$,任取 u_i 对应的用户组为 $UsrGroup_i = \{usr_1, usr_2, \cdots, usr_k\}$,并设用户 $usr_j \in UsrGroup_i$ 且 usr_j 的属性集合为 $S_j = \{u_{j1}, u_{j2}, \cdots, u_{jm}\}$,共 m 个属性。下面分两步执行重加密算法:

1) 对密文访问结构 Γ 对应的属性 u_i,生成随机数 $rand_i \in \mathbf{Z}_p^*$,对密文中 $C'_i = g^{r_i}$ 组件,计算 $C'_i = (g^{r_i})^{rand_i} = g^{r_i rand_i}$ 构成 CT_R。

2) 对 u_i 的用户组 $UsrGroup_i$,生成完全二叉树 T 使得用户为叶子节点。根节点处生成随机二元序列 str,对 T 内所有节点递归计算标值:

若当前节点的标值为 $label$,则左子节点标值为 $RG_L(label)$,右子节点标值为 $RG_R(label)$。

当得到用户叶子节点 usr_u 的标值 $label$ 后,计算 $L_u = RG_M(label)$ 作为 $rand_i$ 对用户 usr_u 的加密密钥。将 $\{(rand_i)_{L_j} \mid usr_j \in UsrGroup_i\}$ 发送给用户,销毁随机序列 str。

用户解密 $rand_i$ 的用户私密信息 I_u 在属性撤销算法 KeyProUpdate 中详细阐述。

4.2.7 数据包请求 DataRequest

假设云端存储数据包数目为 n,用户 usr_i 的数据请求为向量 $\boldsymbol{a}_i = (a_{i1}, a_{i2}, \cdots, a_{in})$,其中 $a_{ij} \in \{0,1\}$ 表示对第 j 个数据包的请求,需要为 1,否则为 0。计算 $E_{epk_i}(\bar{\boldsymbol{a}}_i) = (E_{epk_i}(1-a_{i1}), E_{epk_i}(1-a_{i2}) \cdots, E_{epk_i}(1-a_{in}))$[13],用户 usr_i 广播 $epk_i \parallel E_{epk_i}(\bar{\boldsymbol{a}}_i)$。

对于用户 usr_j,若其也需要发送数据请求,则使用 usr_i 的同态加密公钥 epk_i 生成 $E_{epk_i}(\bar{\boldsymbol{a}}_j)$ 并返回给 usr_i。usr_i 汇总 k 个用户的请求并聚类计算:

$$\prod_{i=1}^{k} E_{epk}(\bar{\boldsymbol{a}}_i) = \Big(\prod_{i=1}^{k} E_{epk}(1-a_{i1}), \prod_{i=1}^{k} E_{epk}(1-a_{i2}) \cdots, \prod_{i=1}^{k} E_{epk}(1-a_{in})\Big)$$

则 l 差异性可由下式计算:

$$l = n - D_{esk}\Big(\sum_{i=1}^{n}\Big(\prod_{j=1}^{k} E_{epk}(1-a_{ji})\Big)\Big)$$

证明 由同态性,上式为:

$$l = n - D_{esk}E_{epk}\Big(\sum_{i=1}^{n}\Big(\prod_{j=1}^{k}(1-a_{ji})\Big)\Big) = \sum_{i=1}^{n}(1-\bar{a}_{1i} \wedge \bar{a}_{2i} \wedge \cdots \wedge \bar{a}_{ki}) = \sum_{i=1}^{n}(a_{1i} \vee a_{2i} \vee \cdots \vee a_{ki})$$

当 k 和 l 大于门限值时,可设用户 usr_i 对数据包 j 的请求为 $C_{ij} = a_{ij} \parallel b_{v_i}$,其中 b_{v_i} 为 usr_i 密钥的时间周期对应的二元序列,对 k 个用户的请求聚集为: $\prod_{i=1}^{k}\Big(\prod_{j=1}^{n} C_{ij}\Big) = C$。为防止攻击者的 DoS 或者重放攻击,加入时间戳 TP,对每个用户 usr_i,设随机数 $r_i \in \mathbf{Z}_p$,作签名为 (S_i, T_i),其中 $S_i = H(C \parallel TP)^{r_i} H(b_{v_i})$,$T_i = g^{r_i}$。令 $S = \prod_{i=1}^{k} S_i, T = \prod_{i=1}^{k} T_i$,数据请求包为: $C \parallel TP \parallel S \parallel T$。

云端收到请求包后,验证:

$$e(S,g) = e(T, H(C \parallel TP))\, e\Big(g, \prod_{i=1}^{k} H(b_{v_i})\Big)$$

若成立,则取出 C_{ij} 并回复 $H(b_{v_i}) \parallel Data_j$ 作为 usr_i 对数据包 j 的请求的响应,其中 $Data$ 为将数据包 CT_R 按 usr_i 对应的二元序列 bin_i 对数据二次重加密后的新的数据包。

取 CT_R 中的 C_2 并计算:

$$C_2 = \Big(\prod_{k=1}^{l} u_k^{b_v[k]}\Big)^s \Big(\prod_{k=1}^{|bin_i|} u_k^{bin_i[k]}\Big)^s。$$

4.2.8 解密算法 Decrypt

已知系统公共参数 pub,密文 CT_R,用户 usr_j 的私钥为 $Usk_S^{t_v}$,如果用户的属性集 S 满足访问结构 $(\boldsymbol{M},\rho)$,则存在向量 $\boldsymbol{w} = \{w_i \in \mathbf{Z}_p\}_{i \in I}$ 使得 $\sum_{i \in I} w_i \lambda_i = s$。

用户使用 I_u 解密 $rand_i$ 进而恢复 $C'_i = (C''_i)^{\frac{1}{rand_i}}$,因而可以如下计算解密系数 Δ:

$$\Delta = \frac{e(C_1, d_0)}{e(C_2, d_1)\prod_{i \in I}[e(D_i, C_i)\, e(D'_i, C'_i)]^{w_i}} = \Big[e(g,g)^{s(at+\alpha)}\, e\Big(g^{\beta s}, \Big(\prod_{k=1}^{l} u_k^{b_v[k]}\Big)^{r_v}\Big(\prod_{k=1}^{|bin_j|} u_k^{bin_j[k]}\Big)^{r_v}\Big)\Big] \Big/ \Big\{e\Big(g^{\beta s}, \Big(\prod_{k=1}^{l} u_k^{b_v[k]}\Big)^{r_v}\Big(\prod_{k=1}^{|bin_j|} u_k^{bin_j[k]}\Big)^{r_v}\Big) \prod_{i \in I}[e(g^t, g^{a\lambda_i} H(i)^{-r_i})\, e(g^{r_i}, H(i)^t)]^{w_i}\Big\} = e(g,g)^{\alpha s}$$

从而明文 $m = C/\Delta$。

4.2.9 属性撤销 KeyProUpdate

属性撤销将导致密文的用户组中的用户撤销。设系统公共参数 pub,密文 CT。

1) 对用户组发生改变的密文属性重新生成随机数 $rand_1$,若原来的用户组随机数为 $rand_0$,令 $rand = rand_0 rand_1$。假设发生属性撤销的用户为 usr_j,云服务器首先生成撤销前的用户组对应的完全二叉树 T,令 N 为撤销前的用户组中用户集合,R 为撤销的用户集合。构造划分 $U = N \backslash R = \cup S_{i,j}$,若 T_k 为以 v_k 为根的子树,则有 $S_{i,j} \cap S_{s,t} = \text{null}$ 且 $S_{i,j} = \{v \mid v \in T_i \wedge v \notin T_j\}$。文献[14]中,通过维护撤销用户集 R 的 Steiner 树 ST(R) 构建划分算法 Subset-Difference。具体方式是: 迭代找到 ST(R) 树中出度为 1 的最长链 $\{v_1, v_2, \cdots, v_l\}$,每得到

一条这样的链，即得到一个子集划分 $S_{1,l}$。显然 Subset-Difference 算法运行时间最坏为 $O((N-R)\log N)$。本文改进了文献[14]的划分算法，将最坏时间复杂度约束在线性时间内，如算法 1 所示:

算法 1　Adv-Subset-Difference。

输入　T, N, R;

输出　$\{S_{i,j}\}$。

1) 令 R 表示相邻用户组成的集簇，$R=\{R_1, R_2, \cdots, R_x\}$，则 R 将 N 划分为 $x+1$ 个区间$\{N_0, N_1, \cdots, N_{x+1}\}$。

2) 对每个 $N_y(0 \leqslant y \leqslant x-1)$ 执行 3) ~ 5)。

3) 若 $N_y=\{v_y\}$ 且 v_y 父节点为 v_i，兄弟为 v_j，则 $S_{i,j}=\{v_y\}$；否则执行 4)。

4) 设 N_y 最后一个叶子节点为 v_y，第一个叶子节点为 v_x，若 v_x 与 v_y 最近的公共节点 v 对应的完全子树 T_v 不与 $R_z(1 \leqslant z \leqslant x)$ 相交，则 v_x 到 v_y 之间的叶子属于同一个 S，设 v 的父节点为 v_i，令 v_j 为 v 的兄弟，则 $S_{i,j}=S_{i,j} \cup \{v_x, v_{x+1}, \cdots, v_y\}$。

5) 若 T_v 与某个 R_z 相交，则从 v_y 出发向前直到找到 v_z 使得 v_x 和 v_z 满足公共节点 v 的子树 T_v 不包含 $R_z(1 \leqslant z \leqslant x)$，将 $\{v_x, v_{x+1}, \cdots, v_y\}$ 拆分为$\{v_x, v_{x+1}, \cdots, v_z\}$、$\{v_z, v_{z+1}, \cdots, v_y\}$，分别代入 3) 中递归。

根据文献[14]，撤销用户数为 R 时至多产生 $1.38R$ 个划分。算法 1 运行时间上限为 $O(N-R)$。按重加密算法 ReEncrypt 中的同样方式计算各个节点的 $label$，并计算 $L_{i,j}=RG_{\mathrm{M}}(label(v_j))$ 为 $S_{i,j}$ 的加密密钥，将加密后的 $rand$: $\{(rand)_{L_{i,j}} \mid \forall S_{i,j}\}$ 发送给 $N \backslash R$ 中的用户 usr。

由于加密后的 $rand$ 以广播形式发送，为避免 usr_i 获得 usr_j 的解密密钥 L，设计 usr 的私密信息 I_u 如下:

设 $path_u$ 为用户组二叉树中 usr 的叶节点到树根的路径。设 V_u 为 $path_u$ 的邻接点的集合:

$$V_u=\{v \mid v \notin path_u \wedge \exists e=(v, v_x) \wedge v_x \in path_u\}$$

则 $I_u=\{label(v) \mid v \in V_u\}$。显然 $|I_u|=O(\log N)$。

用户可以在 $O(\log N)$ 时间内找到所属的 $S_{i,j}$，再根据 I_u 计算 $L_{i,j}$ 并解密 $rand$。具体的，从用户节点 v_u 开始向上遍历 $path_u$，若节点 $v_i \in path_u$ 恰好是某个 $S_{i,j}$ 的根节点，则用户节点 $v_u \in S_{i,j}$ 且只属于此 $S_{i,j}$。由于 v_i 在 $path_u$ 上，因此 $S_{i,j}$ 的 $v_j \in V_u$，即 $label(v_j) \in I_u$，则 $L_{i,j}=RG_{\mathrm{M}}(label(v_j))$。

2) 云端重加密: 生成随机数 $s' \in \mathbf{Z}_p$，更新密文为 CT'_R:

$$C'=Ce(g,g)^{\alpha s'}, C'_1=C_1 g^{\beta s'}=g^{\beta(s+s')}, C'_2=C_2\left(\prod_{k=1}^{l} u_k^{b_v[k]}\right)^{s'};$$

对 $1 \leqslant i \leqslant l$ 有:

$$C''_i=(C'_i)^{rand_i}=g^{r_i(rand_i rand_{i0})}$$

$$C_i^{\alpha}=C_i g^{\alpha s' M_{i1}}=g^{a\lambda_i+\alpha s' M_{i1}} H(i)^{-r_i}=$$

$$g^{a\left(\sum_{j=2}^{n} M_{ij} v_j+s M_{i1}\right)+\alpha s' M_{i1}} H(i)^{-r_i}=$$

$$g^{a\left(\sum_{j=2}^{n} M_{ij} v_j+M_{i1}(s+s')\right)} H(i)^{-r_i}$$

其中 M_{ij} 为矩阵 M 的第 i 行、第 j 列元素。因此共享秘密 $s+s'$ 保持线性共享性质: 若重加密前存在向量 $\boldsymbol{w}=\{w_i \in \mathbf{Z}_p\}_{i \in I}$ 且 $\sum_{i \in I} w_i \lambda_i=s$，重加密后，相同的 $\boldsymbol{w}$ 依然满足 $\sum_{i \in I} w_i \lambda_i=s+s'$。

代入到解密系数 Δ 中可得 $\Delta=e(g,g)^{\alpha(s+s')}$，明文 $m=C'/\Delta$。

5　APPCP_ABE 的安全分析

5.1　安全性证明

本节改进了魏江宏等[6]的标准安全模型，使之适用于 APPCP_ABE 方案的安全性验证。

定理 3　若存在概率多项式时间敌手 A 以优势 $Adv_A=\varepsilon$ 赢得第 3 章定义的安全游戏，则存在概率多项式时间算法 B 以优势 $Adv_B=\varepsilon/T$ 解决判定性 l-BDHE 假设。

证明　定义安全模型中挑战者 C 为概率多项式时间算法 B，通过构造半功能密文，利用 l-BDHE 假设证明半功能密文和随机密文的不可区分，由于敌手解密随机密文的优势是可忽略的，从而敌手攻破 APPCP_ABE 方案的优势是可忽略的。具体如下:

1) Init: 敌手 A 选择挑战的访问结构 Γ^*，挑战的时间周期 t_c^*。给定(G_1, G_2) 上判定性 l-BDHE 假设实例$(G_1, G_2, p, e, \boldsymbol{y}, T')$，其中: $\boldsymbol{y}=(g, g^s, g^a, g^{a^2}, \cdots, g^{a^l}, g^{a^{l+2}}, \cdots, g^{a^{2l}})$。设 $g_k=g^{a^k}$，访问结构 $\Gamma^*=(\boldsymbol{M}, \rho)$，其中 $\boldsymbol{M}$ 为 $l' \times n$ 实矩阵。

2) Setup: B 选择随机数 $\alpha' \in \mathbf{Z}_p$，选择随机数 $\beta \in \mathbf{Z}_p^*$。设 $Z=e(g,g)^{\alpha'+a^{l+1}}$，随机数 $\delta_1, \delta_2, \cdots, \delta_l \in \mathbf{Z}_p$，$u_k=g^{\delta_k} g_{l-k+1}^{-1}$，生成向量 $\boldsymbol{u}=(u_1, u_2, \cdots, u_l) \in G_1^l$。对 Hash 函数的询问 $H(i)$，生成随机数 $t_i \in \mathbf{Z}_p$，返回 g^{t_i} 作答。伪随机序列生成器 RG 生成敌手二元序列 bin。令 $\alpha=\alpha'+a^{l+1}$，系统公共参数 $pub=(G_1, G_2, e, p, q, g, g^{\beta}, g^{1/\beta}, T, \boldsymbol{u})$，系统主密钥 $msk=(\beta, g^{\alpha^{l+1}+\alpha'})$。

3) Query1: 设敌手 A 属性集合为 S，时间周期为 t_v。算法 B 基于 l-BDHE 假设实例给出敌手密钥。

Case1: 若 S 不满足访问结构 $\Gamma^*=(\boldsymbol{M}, \rho)$，则由 LSSS 定义，存在向量 $\boldsymbol{w}=\{w_i \in \mathbf{Z}_p\}$，使得 $\boldsymbol{w}(1,0,\cdots,0)^{\mathrm{T}}=-1$，即 $w_1=-1$，且 $\boldsymbol{w}\boldsymbol{M}_i^{\mathrm{T}}=0(i \in I)$，其中 I 的定义与定义 3 中相同。生成随机数 $t' \in \mathbf{Z}_p$，设 $t=t'+\sum_{1}^{n} w_j a^{l-j+1}$，对任意节点 $v \in V_v$，随机选择 $r_v \in \mathbf{Z}_p$，算法 B 生成密钥 $Usk_S^{t_v}=\{\{K_v \mid v \in V_v\}, \forall i \in S, D_i, D'_i\}$，其中: $K_v=(d_0, d_1, d_1^*, \cdots, d_l^*)$，且

$$d_0=g^{\frac{\alpha+at}{\beta}}\left(\prod_{k=1}^{|b_v|} u_k^{b_v[k]}\right)^{r_v}\left(\prod_{k=1}^{|bin|} u_k^{bin[k]}\right)^{r_v}=g^{\frac{\alpha'}{\beta}} g_1^{\frac{t'}{\beta}}\left(\prod_{j=2}^{n} g_{l-j+1}^{\frac{w_j}{\beta}}\right)\cdot \left(g_{k=1}^{\sum_{k=1}^{|b_v|}\delta_k b_v[k]} \prod_{k=1}^{|b_v|} g_{l-k+1}^{-b_v[k]}\right)^{r_v}\left(g_{k=1}^{\sum_{k=1}^{|bin|}\delta_k bin[k]} \prod_{k=1}^{|bin|} g_{l-k+1}^{-bin[k]}\right)^{r_v}$$

$$d_1=g^{\beta r_v}, d_k^*=u_k^{r_v};\ 1 \leqslant k \leqslant l$$

对 $\forall i \in S$

$$D_i=g^t=g^{t'+\sum_{j=1}^{n} w_j a^{l-j+1}}=g^{t'} \prod_{j=1}^{n} g_{l-j+1}^{w_j}$$

$$D'_i=H(i)^t=\left(g^{t'} \prod_{j=1}^{n} g_{l-j+1}^{w_j}\right)^{t_i}$$

算法 B 将私钥 $Usk_S^{t_v}$ 返回给 A。

Case2: 若 S 满足访问结构 $\Gamma^*=(\boldsymbol{M}, \rho)$，但时间周期 $t_v \neq t_c^*$，不妨设 $t_v > t_c^*$，令 h 为使得 $b_v[h]+bin[h] \neq 0$ 的最小正整数，由定理 1 中 $b_v=b_c \| b^*$ 知 h 一定存在且 $h \leqslant |b_v|$。随机生成 $t \in \mathbf{Z}_p$，对任意节点 $v \in V_v$，随机选择 $r'_v \in \mathbf{Z}_p$。设 $r_v=r'_v+a^h/(\beta(bin[h]+b_v[h]))$，则:

$$d'_0=g^{\frac{\alpha+at}{\beta}}\left(\prod_{k=1}^{h} u_k^{b_v[k]}\right)^{r_v}\left(\prod_{k=1}^{h} u_k^{bin[k]}\right)^{r_v}=$$

$$g^{\frac{\alpha'}{\beta}} g_1^{\frac{t}{\beta}} g^{\frac{a^{l+1}}{\beta}} g^{r'_v \sum_{k=1}^{h}\delta_k(b_v[k]+bin[k])} g_h^{\frac{\sum_{k=1}^{h}\delta_k(b_v[k]+bin[k])}{\beta(b_v[h]+bin[h])}}\cdot \left(\prod_{k=1}^{h} g_{l-k+1}^{-(b_v[k]+bin[k])}\right)^{r'_v}$$

$$\left(\prod_{k=1}^{h-1} g_{l-k+1+h}^{-(b_v[k]+bin[k])}\right)^{\frac{1}{\beta(b_v[k]+bin[k])}} g^{\frac{-a^{l+1}}{\beta}} =$$

$$g^{\frac{\alpha'}{\beta}} g^{\frac{1}{\beta} r'_v \sum_{k=1}^{h} \delta_k(b_v[k]+bin[k])} g_h^{\frac{\sum_{k=1}^{h}\delta_k(b_v[k]+bin[k])}{\beta(b_v[k]+bin[k])}} \cdot$$

$$\left(\prod_{k=1}^{h} g_{l-k+1}^{-(b_v[k]+bin[k])}\right)^{r'_v}$$

$$\left(\prod_{k=1}^{h-1} g_{l-k+1+h}^{-(b_v[k]+bin[k])}\right)^{\frac{1}{\beta(b_v[k]+bin[k])}}$$

$d'_1 = g^{\beta r_v} = g^{\beta r'_v} g_h^{\frac{1}{(b_v[k]+bin[k])}}$

$d^*{}'_k = u_k^{r_v}$; $1 \leqslant k \leqslant l$

之后将 K'_v 更新到 K_v 上，选择随机数 $r''_v \in \mathbf{Z}_p$：

$$d_0 = d'_0 \left(\prod_{k=h+1}^{|b_v|} (d^*{}'_k)^{b_v[k]}\right)\left(\prod_{k=1}^{|b_v|} u_k^{b_v[k]}\right)^{r''_v} \cdot$$

$$\left(\prod_{k=h+1}^{|bin|} (d^*{}'_k)\right)^{bin[k]} \left(\prod_{k=1}^{|bin|} u_k^{bin[k]}\right)^{r''_v}$$

$d_1 = d'_1 g^{\beta r''_v}$，$d^*_k = d^*{}'_k u_k^{r''_v}$; $1 \leqslant k \leqslant l$

对 $\forall i \in S$，$D_i = g^t$，$D'_i = H(i)^t = g^{tt_i}$。

算法 B 将私钥 $Usk_S^{t_v}$ 返回给 A。

4) Challenge: 敌手选择两个等长的密文 m_0、m_1，挑战者抛硬币并从 m_0、m_1 中等概率选择明文 m_θ 进行加密，事件 T_Δ 为时间周期与挑战时间周期相符，概率为 $P_T = 1/T$。$\theta \in \{0,1\}$ 平均分布，有 $P_{\theta=1} = P_{\theta=0} = 1/2$。令：

$$C^* = m_\theta T' e(g,g)^{s\alpha'}, C_1^* = g^{\beta s}$$

$$C_2^* = \left(\prod_{k=1}^{l} u_k^{b_v[k]}\right)^s \left(\prod_{k=1}^{|bin|} u_k^{bin[k]}\right)^s$$

对 $1 \leqslant i \leqslant \ell$，作随机数 $r_i \in \mathbf{Z}_p$，设 $C^*{}'_i = g^{r_i}$，对随机数 $s \in \mathbf{Z}_p$，有向量 $\boldsymbol{\eta} = (s, sa, \cdots, sa^{n-1})$，则有 LSSS 的共享分量 $\lambda_i = \sum_{j=1}^{n} (sa^{j-1} M_{ij})$，从而：$C_i^* = g^{a\lambda_i} (g^{t_i})^{-r_i} = \prod_{j=1}^{n} g_j^{sM_{ij}} (g^{t_i})^{-r_i}$。发送密文 $CT^* = \{C^*, C_1^*, C_2^*, C_i^*, C^*{}'_i \mid 1 \leqslant i \leqslant \ell\}$ 到敌手 A。

5) Query2: 敌手继续进行多项式时间的私钥提问，过程与 Query1 相同，但需满足属性集合不是授权子集或者 $t_v > t_c^*$。

6) Guess: 如果有 $T' = e(g^s, g_{l+1})$，则 $C^* = m_\theta e(g,g)^{s(\alpha' + a^{l+1})} = m_\theta Z^s$ 有效，有公式：

$$\Pr[B(\boldsymbol{y}, T' = e(g^s, g_{l+1})) = 0 \mid T_\Delta] = 1/2 + \varepsilon$$

否则，T' 为 G_2 内随机元素，敌手无法获得明文任何信息，敌手猜对 θ 的概率至多是 1/2，即：$\Pr[B(\boldsymbol{y}, T' = \theta) = 1 \mid T_\Delta] = 1/2$。

因此算法 B 的优势为：

$$Adv_B = \left|\Pr[B(\boldsymbol{y}, T' = e(g^s, g_{l+1})) = 0] - \Pr[B(\boldsymbol{y}, T' = \theta) = 1]\right| = \frac{1}{T}(1/2 + \varepsilon - 1/2) = \varepsilon/T$$

由 l-BDHE 假设可知敌手优势 ε 是可忽略的，因此 APPCP_ABE 方案关于标准安全模型是安全的。

5.2 属性撤销的安全性

首先，由 Adv-Subset-Difference 算法，当用户撤去若干属性后，由于用户在用户组二叉树中属于 R，不属于任何划分 $S_{i,j}$，无法由属性撤销前的私密信息推出 $L_{i,j}$，也无法解密新的随机数 $rand$，即无法恢复密文组件 C'_i，从而属性撤销算法具有后向安全性；其次，密文的解密系数更新为 $\Delta = e(g,g)^{\alpha(s+s')}$，若用户的属性集合不是授权子集，则用户无法恢复共享秘密 $s + s'$。即使用户保存了撤销属性之前的解密系数 $\Delta = e(g,g)^{\alpha s}$，由于 s' 对于用户是未知的，用户也无法解密其属性撤销前可以解密的密文，即属性撤销算法具有前向安全性。因此，本文方案关于属性的撤销是安全的。

5.3 密钥泄漏的安全性

对于用户密钥泄漏问题，本文提出的安全模型通过时间周期完全二叉树进行密钥更新，将时间周期与用户私钥绑定，使用户私钥在每个时间周期内均不相同。假设 $t_y > t_y'$，按密钥更新算法 KeyUpdate 将密钥 $Usk_S^{t_y}$ 更新为 $Usk_S^{t_y}$。由于更新过程不可逆，从而使得当 $Usk_S^{t_y}$ 发生泄漏时，之前需使用 $Usk_S^{t_y'}$ 解密的密文依然是安全的（前向安全性）。反之，如果 $Usk_S^{t_y'}$ 发生泄漏，敌手获取了 $Usk_S^{t_y'}$ 并通过 KeyUpdate 将 $Usk_S^{t_y'}$ 更新到 $Usk_S^{t_y}$，然而由于敌手与原用户使用 RG 生成的二元序列 bin_i 不一致，用户与敌手更新到的密钥 $Usk_S^{t_y}$ 中，u_k 的阶不同。对于云端来说，其发送给用户的密文是基于用户的二元序列 bin 重加密过的，根据解密算法 Decrypt，只有密文 C_2 与密钥 d_0 中关于 u_k 的阶相同时才可以解密，即敌手无法通过更新密钥来解密 t_y 时间周期的密文（后向安全性）。本文方案关于密钥泄漏是安全的。

6 APPCP_ABE 的性能分析

本章将 APPCP_ABE 方案与 Sahai 等[1]的方案，魏江宏等[6]的方案以及 Hur 等[9]的方案进行计算损耗、空间占用情况以及安全性的比较。设 e_1 为一次 G_1 指数运算损耗，e_2 为一次 G_2 指数运算损耗；l 为密文访问结构中属性数目，p 为一次双线性运算损耗，$|S|$ 为用户属性集大小，N 为一个用户组平均大小，r 为用户组平均撤销用户数目，G_1 表示群 G_1 中的元素，G_2 表示群 G_2 中的元素。对比结果如表 1 ~ 3 所示。

分析可知，APPCP_ABE 方案较已有方案在属性撤销以及密钥泄漏方面具有更好的安全性，尽管时间复杂度与空间复杂度有所提高，但提高的幅度至多在 $O(\log T)$ 或 $O(\log N)$ 级别，均在可接受范围内。

表 1 时间损耗比较

方案	加密	解密	密钥更新	属性更新
APPCP_ABE	$(3l+1+2\log T)e_1 + e_2$	$(2l+2)p + le_1 + O(\log N)$	$O((\log T)^2 + 1)e_1$	$[2l+1+O(\log T)]e_1 + e_2 + O(N-R)$
文献[1]方案	$(2l+3)e_1 + 2e_2$	$(2l+1)p + le_2$	未考虑	未考虑
文献[6]方案	$(2l+3+\log T)e_1 + e_2$	$(2l+2)p + le_2$	$O((\log T)^2)e_1$	未考虑
文献[9]方案	$(3l+1)e_1 + e_2$	$(3l+2)p + le_2 + O(N\log N)$	$(3l+1)e_1$	$(3l+1)e_1 + (2r-1)O(N\log N)$

7 结语

本文在 CP_ABE 加密的基础上，引入用户组二叉树与时间周期序列二叉树，给出了关于密钥泄漏的安全的 CP_ABE 方案；在二次代理重加密算法的基础上设计细粒度的访问控制策略；基于高效的同态加密算法实现了具有 k 匿名 l 多样性

的数据请求; 通过标准安全模型证明了方案的选择安全性。

后续工作的重点将放在: 1) 数据请求 DataRequest 过程中不同用户的同步问题; 2) 用户组二叉树的存储开销优化问题。

表 2　空间占用比较

方案	密钥长度	密文长度
APPCP_ABE	$(\|S\|+O(\log T)^2)G_1+O(\log N)$	$(3+2l)G_1+G_2$
文献[1]方案	$(\|S\|+2)G_1$	$(2+l)G_1+G_2$
文献[6]方案	$(\|S\|+O(\log T)^2)G_1$	$(2+l)G_1+G_2$
文献[9]方案	$(\|S\|+l)G_1+O(\log N)$	$(3+2l\log N)G_1+G_2$

表 3　安全性对比

方案	属性撤销安全性	密钥泄漏安全性
APPCP_ABE	前后向安全	前后向安全
文献[1]方案	未考虑	未考虑
文献[6]方案	未考虑	前向安全
文献[9]方案	前向安全	未考虑

参考文献:

[1] SAHAI A, WATERS B. Fuzzy identity based encryption [C]// Proceedings of 2005 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer-Verlag, 2005: 457 - 473.

[2] GOYAL V, PANDEY O, SAHAI A, *et al*. Attribute-based encryption for fine-grained access control of encrypted data [C]// Proceedings of the 13th ACM Conference on Computer and Communications Security. New York: ACM, 2006: 89 - 98.

[3] TANG Y, LEE P, LIU J, *et al*. Secure overlay cloud storage with access control and assured deletion [J]. IEEE Transactions on Dependable and Secure Computing, 2012, 9(6): 903 - 916.

[4] HONG C, ZHANG M, FENG D. AB-ACCS: a cryptographic access control scheme for cloud storage [J]. Journal of Computer Research and Development, 2010, 47(Z1): 259 - 265. (洪澄, 张敏, 冯登国. AB-ACCS: 一种云存储密文访问控制方法[J]. 计算机研究与发展, 2010, 47(Z1): 259 - 265.)

[5] HONG C, ZHANG M, FENG D. Achieving efficient dynamic cryptographic access control in cloud storage [J]. Journal on Communications, 2011, 32(7): 125 - 132. (洪澄, 张敏, 冯登国. 面向云存储的高效动态密文访问控制方法[J]. 通信学报, 2011, 32(7): 125 - 132.)

[6] WEI J, LIU W, HU X. Forward-secure ciphertext-policy attribute-based encryption scheme [J]. Journal on Communications, 2014, 35(7): 38 - 45. (魏江宏, 刘文芬, 胡学先. 前向安全的密文策略基于属性加密方案[J]. 通信学报, 2014, 35(7): 38 - 45.)

[7] WANG P, FENG D, ZHANG L. CP-ABE scheme supporting fully fine-grained attribute revocation [J]. Journal of Software, 2012, 23(10): 2805 - 2816. (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.)

[8] YU S, WANG C, REN K, *et al*. Achieving secure, scalable and fine-grained data access control in cloud computing [C]// Proceedings of the 2010 INFOCOM. Piscataway: IEEE, 2010: 1 - 9.

[9] 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.

[10] ATTRAPADUNG N, IMAI H. Conjunctive broadcast and attribute-based encryption [C]// Proceedings of the Third International Conference on Pairing-Based Cryptography — Pairing 2009, LNCS 5671. Berlin: Springer, 2009: 248 - 265

[11] WAN Z, LIU J, DENG R H. HASBE: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing [J]. IEEE Transactions on Information Forensics and Security, 2012, 7(2): 743 - 754.

[12] SAHAI A, SEYALIOGLU H, WATERS B, *et al*. Dynamic credentials and ciphertext delegation for attribute-based encryption [C]// Proceedings of the 32nd Annual Cryptology Conference on Advances in Cryptology — CRYPTO 2012, LNCS 7417. Berlin: Springer, 2012: 199 - 217.

[13] LU R, LIN X, SHI Z, *et al*. PLAM: A privacy-preserving framework for local-area mobile social networks [C]// Proceedings of the INFOCOM 2014. Piscataway: IEEE, 2014: 763 - 771.

[14] NAOR D, NAOR M, LOTSPIECH J. Revocation and tracing schemes for stateless receivers [C]// Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology — CRYPTO 2001, LNCS 2139. Berlin: Springer, 2001: 41 - 62.

(上接第 1566 页)

[3] MOSTAFAVI M A, GOLD C. A global kinetic spatial data structure for a marine simulation [J]. International Journal of Geographical Information Science, 2004, 18(3): 211 - 227.

[4] DUTTON G. Modeling locational uncertainty via hierarchical tessellation [M]. Accuracy of Spatial Databases. London: Taylor & Francis, 1989: 125 - 140.

[5] WANG L, ZHAO X, CAO W, *et al*. A GPU-based algorithm for the generation of spherical Voronoi diagram in QTM mode [J]. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2013, XL-4/W2: 45 - 50.

[6] NVIDIA. CUDA C programming guide Version 6.5 [EB/OL]. [2014 - 12 - 01]. http://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf.

[7] COOK S. CUDA programming: a developer's guide to parallel computing with GPUs [M]. SU T, LI D, LI S, *et al*. translated. Beijing: China Machine Press, 2014: 12 - 13. (COOK S. GPU 并行程序设计——GPU 编程指南[M]. 苏统华, 李东, 李松泽, 等译. 北京: 机械工业出版社, 2014: 12 - 13.)

[8] HE Y, YE C, LIU Z, *et al*. Parallel simulation and optimization of CUDA-based real-time huge crowd behavior [J]. Journal of Computer Applications, 2012, 32(9): 2466 - 2469. (贺毅辉, 叶晨, 刘志忠, 等. 基于 CUDA 的大规模群体行为实时仿真并行实现及优化[J]. 计算机应用, 2012, 32(9): 2466 - 2469.)

[9] XU S, ZHANG E. CUDA-based parallel visualization of 3D data [J]. CT Theory and Applications, 2011, 20(1): 47 - 54. (徐赛花, 张二华. 基于 CUDA 的三维数据并行可视化[J]. CT 理论与应用研究, 2011, 20(1): 47 - 54.)

[10] DESCHIZEAUX B, BLANC J-Y. Imaging earth's subsurface using CUDA [C/OL]// GPU Gems. 2007. [2014 - 12 - 01]. http://http.developer.nvidia.com/GPUGems3/gpugems3_ch38.html.

[11] SANDERS J, KANDROT E. CUDA by example: an introduction to general-purpose GPU programming [M]. NIE X, *et al*. translated. Beijing: China Machine Press, 2011: 54 - 55, 78. (SANDERS J, KANDROT E. GPU 高性能编程——CUDA 实战[M]. 聂雪军, 等译. 北京: 机械工业出版社, 2011: 54 - 55, 78.)

[12] ZHAN S, ZHU Y, ZHAO K, *et al*. GPU high performance computation -CUDA [M]. Beijing: China Water & Power Press, 2009: 46 - 47, 141 - 142. (张舒, 褚艳利, 赵开勇, 等. GPU 高性能运算之 CUDA[M]. 北京: 中国水利水电出版社, 2009: 46 - 47, 141 - 142.)