灵活访问且模糊可搜索的 EHR 云服务系统

闫 铭[1] 张应辉[1,2,3] 郑 东[1,2] 吕柳迪[1] 苏昊楠[1]
(西安邮电大学无线网络安全技术国家工程实验室 西安 710121)[1]
(卫士通摩石实验室 北京 100070)[2] (密码科学技术国家重点实验室 北京 100878)[3]

摘 要 在电子健康记录系统(E-Healthcare Record Systems,EHRS)中,一些方案利用密钥策略 ABE(KP-ABE)来保护隐私。由用户指定一个访问策略,密文只有与访问策略相匹配时才能被解密。现有的 KP-ABE 要求在生成密钥期间必须先确定访问策略,这在 EHRS 中是不可行的,因为有时访问策略在密钥生成后才被决定。基于 KP-ABE,提出一种灵活访问且模糊可搜索的 EHR 云服务系统。该系统不仅实现了基于关键字容错的云端密文搜索,而且允许用户重新定义访问策略并为之生成密钥,因此一个精确的访问策略将不再是必需的。最后,证明了该方案的安全性。

关键词 电子健康记录,属性加密,访问控制,模糊搜索,关键字容错
中图法分类号 TP309 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.10.032

Flexibly Accessed and Vaguely Searchable EHR Cloud Service System

YAN Ming[1] ZHANG Ying-hui[1,2,3] ZHENG Dong[1,2] LV Liu-di[1] SU Hao-nan[1]
(National Engineering Laboratory for Wireless Security,Xi'an University of Posts and Telecommunications,Xi'an 710121,China)[1]
(Westone Cryptologic Research Center,Beijing 100070,China)[2] (State Key Laboratory of Cryptology,Beijing 100878,China)[3]

Abstract In e-healthcare record systems (EHRS),some schemes exploit key-policy ABE (KP-ABE)to protect privacy. An access policy is specified by the user,and the ciphertexts can be decrypted only when they match users' access plicy. The existing KP-ABE requires that the access policies should be confirmed first during key generation,which is not always practicable in EHRS,because the policies are sometimes confirmed after key generation. Based on KP-ABE,this paper proposed a flexibly accessed and vaguely searchable EHR cloud service system. This system not only fulfills the cloud ciphertext search based on keyword fault-tolerant technique,but also allows users to redefine their access policies and generates keys for the redefined ones,hence,a precise policy is no longer necessary. Finally,the scheme was proved to be secure.

Keywords E-healthcare record,Attribute-based encryption,Access control,Vaguely searchable,Keyword fault tolerant

1 引言

基于属性的加密[1-3](Attribute-Based Encryption,ABE)通过将密钥和密文关联到属性集和访问结构实现了对数据的加密及访问控制。在实现 ABE 时要求用户指定一个确定的访问策略,这在实际应用中往往是不可行的。例如,在 EHRS[4]的应用中,医生为 Alice 做诊断,将她的健康记录加密后上传至云服务器,由 Alice 制定访问策略,当医生 Bob 满足访问条件而访问 Alice 的健康记录时发现她可能患有心脏病,则需要一位心脏病专家来为 Alice 做诊断,该心脏病专家就必须被允许在云服务器上访问 Alice 的健康记录。

事实上,访问策略是必须能够被动态更改的。例如,在上述案例中,医生 Bob 发现 Alice 可能患有心脏病后,需要重新定义访问策略(加入新属性,例如心脏病专家)以允许心脏病专家访问 Alice 的健康记录。若采用以上方式,则在生成密钥期间将不再需要一个精确的访问策略,因为之后可以根据实际需要动态更改访问策略。

现有的一些 ABE 方案也支持这种代理授权功能,在文献[5-7]的 CP-ABE 方案中,若用户想要代理授权生成密钥,那么用来生成密钥的集合必须是原始集合的子集,文献[8-10]的 KP-ABE 中也提出了一种代理授权机制,然而它们对访问策略的授权具有过强的局限性,不能够应用于电子医疗场景。若要在上述应用中实现一种合适的代理授权机制,就必须在秘密分享方案下进行。在大多数的 KP-ABE 方案[8-10]中,秘密分享方案[11]都被用于在密钥生成时分享一个秘密和在解密期间重建秘密。在生成密钥时,一个访问策略中的每一个属性都需要与一个秘密分享相关联。如果在目标访问策略中添加了新的属性,那么用户将不能为这个访问策略授权生成

到稿日期:2017-09-11 返修日期:2017-12-05 本文受国家自然科学基金项目(61472472,61402366),陕西省自然科学基础研究计划项目(2015JQ6236,2013JZ020)资助。

闫 铭(1991-),男,硕士生,主要研究方向为网络安全与云存储;张应辉(1985-),男,博士,副教授,主要研究方向为公钥密码学、云存储安全、无线网络安全;郑 东(1964-),男,博士,教授,主要研究方向为基于编码的密码学、云存储安全;吕柳迪(1992-),女,硕士,主要研究方向为网络与信息安全;苏昊楠(1991-),女,硕士生,主要研究方向为网络与信息安全。

密钥,因为在不知道秘密的情况下将不能为新的属性关联一个秘密分享,这就是 KP-ABE 方案访问策略授权具有局限性的原因。

本文提出了一种灵活授权访问且模糊可搜索的云服务系统,该系统基于 KP-ABE 方案,实现了高效的模糊关键字搜索[12-14]功能以及动态的代理授权机制,允许访问策略被重新定义并由用户代理作为授权中心为新的访问策略授权生成密钥。与现有的方案相比,所提方案减弱了对新访问策略授权的局限性。

2　问题描述

2.1　系统模型

首先给出属性向量的概念:将属性按高低层次[15-16]划分,属性向量通过从较高层到较低层选择单一属性得到。将属性域放在一个矩阵中,例如医院名称("医院 A"和"医院 B")、职称(教授)或工龄这类属性放在第一层,医生专业("心脏病专家"和"肠胃病专家")这类属性放在下一层,当医生的第一层属性成功匹配后,可以重新按照专业属性定义访问策略。

本系统主要包括以下 4 个实体:数据拥有者、数据使用者、公共云服务器以及私有云服务器。公共云服务器负责存储所有加密后的 EHR 数据以及处理私有云服务器发送过来的搜索符号请求,进而为数据使用者提供最初的搜索结果。数据拥有者指病人,数据使用者指医生,假设病人 Alice 的 EHR 数据由属性集 S={医院 A,心脏病专家,教授,工龄≥3}加密并上传至公共云服务器,同时关键字集合以及文件标识符 FID 被上传至私有云服务器。Alice 指定一个访问策略 A ={医院 A,教授,工龄≥5}并为匹配该访问策略的医生生成密钥,医生通过关键字搜索快速得到相应的密文并解密以为 Alice 做诊断。医生发现 Alice 可能患有心脏病,此时医生重新定义了访问策略 A 为 A' ={{医院 A,心脏病专家},教授,工龄≥7}并为 A' 授权生成密钥。由于与 Alice 的 EHR 数据相关的集合 S 能够满足访问结构 A',因此有访问结构 A' 的医生是能够解密的。

A' 中的一对属性{医院 A,心脏病专家}被看作一个属性向量,因此在 A 中添加一个新的属性"心脏病专家"是可行的,这减弱了代理授权的局限性。本文系统的结构如图 1 所示。

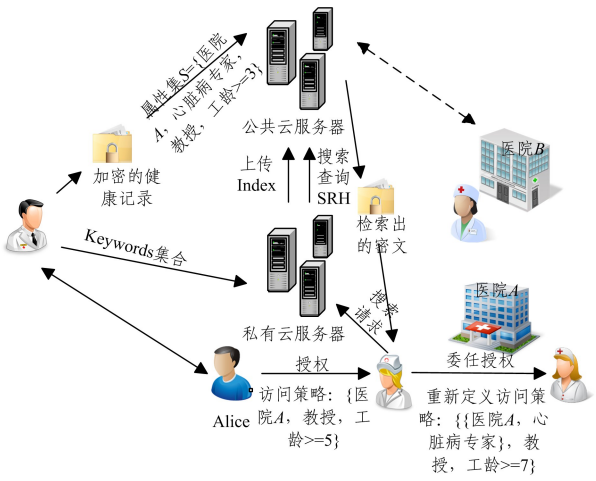

图 1　系统模型

Fig. 1　System model

2.2　访问策略重新定义的模型

将属性域放在一个 L 行 D 列的矩阵 U 中,$U=(u_{i,j})_{L\times D}=(U_1,\cdots,U_i,\cdots,U_L)^{\mathrm{T}}$,其中 U_i 表示 U 的第 i 行且包含 D 个属性。M^{T} 表示矩阵 M 的转置,在矩阵中可能包含一些空的属性,用一个特殊的字符"$\emptyset$"表示空属性。

由属性矩阵引出属性向量的概念,定义一个深度为 k ($1\leqslant k\leqslant L$)的属性向量 $\boldsymbol{u}=(u_1,u_2,\cdots,u_k)$,其中对于每一个从 1 到 k 的 i,都有 $u_i\in U_i$,这表明每一个属性向量是通过由第一层到第 k 层取样得到的。注意,每一个属性 u_i 实际上有两个下标(i,j),这表示它在属性矩阵中的位置,只是这里简化了第二个下标 j 的概念。接着,定义一个属性向量的集合,$S=\{\boldsymbol{u}\}$表示一个深度为 k 的属性向量的集合,且$|S|$表示集合的基数。

对于一个长度为 i 的属性向量 $\boldsymbol{u}'$和一个深度为 k 的属性向量 $\boldsymbol{u}$,如果 $\boldsymbol{u}=(\boldsymbol{u}',u_{i+1},u_{i+2},\cdots,u_k)$($1\leqslant i\leqslant k\leqslant L$),那么称 $\boldsymbol{u}'$是 $\boldsymbol{u}$ 的前缀。

定义 A 长度为 k 的属性向量上的访问结构,A 是所有深度为 k 的属性向量集合的非空子集的集合,如果一个集合 S 满足 $S\in A$,则 S 称为授权集合且满足访问结构 A 。

本文中一个与访问结构 A 相关联的密钥可以解密一个由属性向量集合 S 加密的密文,当且仅当 $S\in A$;一个与访问结构 A'相关联的密钥可以为一个访问结构 A 授权生成密钥,这要求一个集合 $S'\in A'$中的每一个属性向量必须是集合 $S\in A$ 中的一个属性向量的前缀,且 A 中包含的所有属性向量在 A'中有前缀。可以看到,在代理授权添加新的属性时,新的属性是被串联在 A'的现有属性向量的末尾,而不是被当作新的单独的属性分配到新的秘密共享。

2.3　安全模型

本系统定义了一个安全模型对抗选择明文攻击,事实上恶意用户能够获取到系统公钥,他们能够通过询问其他用户的密钥进行合谋攻击。为了捕获这些攻击,定义一个敌手能够访问系统的公共密钥,以可忽略的优势来区分两个消息的密文,敌手并不能得到能够解密挑战密文的密钥。本系统的安全性是通过挑战者 $\mathscr{C}$ 和敌手 $\mathscr{A}$ 之间的一个游戏定义的,具体如下。

(1)初始化:挑战者 $\mathscr{C}$ 执行 setup 算法,并将系统公钥 PK 发送给 $\mathscr{A}$。

(2)阶段 1:$\mathscr{A}$ 有序地向 $\mathscr{C}$ 发出了一系列的询问 $Q_1,\cdots,Q_{q_1}$,其中 $1\leqslant i\leqslant q_1$,$Q_i$ 为以下 3 种类型之一。

Creat(A):$\mathscr{A}$ 指定一个访问结构 A,$\mathscr{C}$ 通过调用密钥生成算法为该访问结构生成密钥,并将其放置在一个集合 $\mathscr{K}$ 中,初始化该集合为空。$\mathscr{C}$ 只是将该密钥的一个参考给予 $\mathscr{A}$,而并非该密钥本身。

Delegate(A,A'):$\mathscr{A}$ 指定一个在集合 $\mathscr{K}$ 中与 A'相关的密钥 SK'以及一个访问结构 A 。如果允许调用代理授权算法,$\mathscr{C}$ 将为 A 生成一个密钥 SK,并且将 SK 添加到集合 $\mathscr{K}$ 中,同样只是给 $\mathscr{A}$ 一个该密钥的参考,而并非其本身。

Reveal(A):$\mathscr{A}$ 指定一个在集合 $\mathscr{K}$ 中的密钥,$\mathscr{C}$ 将该密钥发送给攻击者并将其从集合 $\mathscr{K}$ 中移除。

(3)挑战阶段:$\mathscr{A}$ 声明两个等长的消息 M_0 和 M_1,以及一组具有附加限制的属性向量的集合 S^*,对于任何访问结构 A

的任何泄漏密钥 SK，有 $S^* \notin \mathcal{A}$；对于任何访问结构 $\mathcal{A}'$ 的密钥 SK'，可通过一个泄漏的密钥委任授权 $S^* \notin \mathcal{A}'$。然后，挑战者 $\mathcal{C}$ 随机抛一枚硬币得 $b \in \{0,1\}$，并且以集合 S^* 加密消息 M_b 得出密文 CT^* 发送给攻击者 $\mathcal{A}$。

(4)阶段 2：同阶段 1，$\mathcal{A}$ 有序地向 $\mathcal{C}$ 询问 $Q_{q_1+1},\cdots,Q_q$，任何泄漏密钥的访问结构和任何密钥的访问结构都不能从一个包含 S^* 的泄漏密钥被委任授权。

(5)猜测：攻击者 $\mathcal{A}$ 输出对 $b' \in \{0,1\}$ 的猜想，定义 $\mathcal{A}$ 在该游戏中的优势为 $Adv_{\mathcal{A}}=|\Pr[b=b']-1/2|$。

定义 1 在以上游戏中，如果在所有概率多项式时间内攻击者 $\mathcal{A}$ 的优势可忽略，那么本系统将是足够安全的。

3 预备知识

3.1 访问结构

设 $P=\{P_1,P_2,\cdots,P_n\}$ 是由 n 个参与者组成的集合，$\mathbb{A} \in 2^P$ 是 2^P 的一个非空子集，其中 2^P 表示 P 的所有子集组成的集合，即 $\mathbb{A}$ 是由 P 的若干子集组成的非空集合。如果集合 $\mathbb{A}$ 是单调的，即对任意的由若干个参与者组成的集合 B 和 C，如果 $B \in \mathbb{A}$ 且 $B \in C$，则有 $C \in \mathbb{A}$，那么称 $\mathbb{A}$ 是参与者集合 P 上的一个访问结构。在 $\mathbb{A}$ 中的集合为授权集，不在 $\mathbb{A}$ 中的集合为非授权集。

在以往的 KP-ABE 方案中，参与者指属性；本文中的参与者指属性集合，一个访问结构实际上是属性向量集的一个集合。

3.2 线性秘密共享方案(LSSS)

一个关于参与者集合的秘密共享方案 Π 在 Z_P 上是线性的，则它满足：

(1)所有参与者的分享份额构成 Z_P 上的一个向量。

(2)存在一个 l 行 n 列的矩阵 $\boldsymbol{A}$，称作 Π 的共享生成矩阵，对于所有的 $i=1,\cdots,l$，函数 $\rho(i)$ 表示 $\boldsymbol{A}$ 第 i 行所标记的参与者。设列向量 $\boldsymbol{s}=(s,s_2,\cdots,s_n)$，其中 $s \in Z_P$ 是需要共享的秘密，$s_2,\cdots,s_n \in Z_P$ 是随机选取的，则向量 $\boldsymbol{As}$ 表示 Π 对秘密 s 的 l 个分享份额，$(\boldsymbol{As})_i$ 是第 i 个分享份额，它属于参与者 $\rho(i)$。

由文献[1]可知，LSSS 具有线性重构特性。设 Π 是关于访问结构 $\mathbb{A}$ 的线性秘密共享方案，$S \in \mathbb{A}$ 是一个授权集合，$I=\{i:\rho(i) \in S\} \subseteq \{1,\cdots,l\}$，则可以在多项式时间内找到一组常数 $\omega_i \in Z_P$，如果 $\{\lambda_i\}$ 是对秘密 s 的有效分享，则等式 $\sum_{i \in I}\omega_i\lambda_i=S$ 成立。

3.3 组合阶双线性群

假设 $\mathcal{G}$ 是一个群生成器，ℓ 是一个安全参数，则组合阶双线性群可以被定义为：$N=(p_1p_2p_3,G,G_T,e) \xleftarrow{R} \mathcal{G}(1^\ell)$，其中 p_1,p_2,p_3 是 3 个互不相同的素数，G 和 G_T 是阶为 N 的循环群。高效可计算的映射 $e:G\times G=G_T$ 的性质如下。

(1)双线性：对于所有的 $a,b \in Z_N$，$g,h \in G$，有 $e(g^a,g^b)=e(g,h)^{ab}$。

(2)非退化性：$\exists g \in G$，g 是群 G 的生成元，则 $e(g,g)$ 是 G_T 的生成元。

G_{ij} 表示阶为 p_ip_j 的子群，其中 $i \neq j$；G_1,G_2,G_3 为 G 中阶为 p_1,p_2,p_3 的子群；G_1,G_2,G_3 的正交性的定义如下。

定义 2 对于所有的 $u \in G_i$，$v \in G_j$，有 $e(u,v)=1$，其中 $i \neq j \in \{1,2,3\}$。

3.4 编辑距离

编辑距离是两个字符串之间相似度的测量，比如两个单词 w_1 和 w_2 的编辑距离 $ed(w_1,w_2)$ 是指一个单词变换成另一个单词所执行的最少操作步骤。这其中包含 3 个基本操作。

(1)置换：将一个单词中的字符变换成另一个；

(2)删除：从一个单词中删除一个字符；

(3)插入：将单个字符插入到一个单词中。给定一个关键字 w，如果对于一个确定的整数 d，$ed(w,w')<d$ 满足，则用 $w_{w,d}$ 表示关键字 w' 的集合。

3.5 复杂性假设

假设 1 使 $N=(p_1p_2p_3,G,G_T,e) \xleftarrow{R} \mathcal{G}(1^\ell)$，定义 $g \xleftarrow{R} G_1;X_3 \xleftarrow{R} G_3;D=(G,g,X_3);T_1 \xleftarrow{R} G_1;T_2 \xleftarrow{R} G_{12}$；算法 A 打破假设 1 的优势被定义为：

$$Adv1_A(\ell)=|\Pr[A(D,T_1)=1-\Pr[A(D,T_2)=1]|$$

当且仅当 $Adv1_A$ 在任意 ℓ 的多项式时间算法 A 中可忽略，则假设 1 成立。

假设 2 $N=(p_1p_2p_3,G,G_T,e) \xleftarrow{R} \mathcal{G}(1^\ell)$，定义 $\alpha,s \xleftarrow{R} Z_N;g \xleftarrow{R} G_1;X_2,Y_2,Z_2 \xleftarrow{R} G_2;X_3 \xleftarrow{R} G_2;D=(G,g,g^\alpha X_2,X_3,g^sY_2,Z_2);T_1=e(g,g)^{\alpha s};T_2 \xleftarrow{R} G_T$；算法 A 打破假设 2 的优势被定义为：

$$Adv2_A(\ell)=|\Pr[A(D,T_1)=1-\Pr[A(D,T_2)=1]|$$

当且仅当 $Adv2_A(\ell)$ 在任意 ℓ 的多项式时间算法 A 中可忽略，则假设 2 成立。

4 具体方案

本文基于通配符的方法构造了模糊关键字集合，直接利用通配符表示相同位置上的编辑操作。关键字为 w 且编辑距离为 d 的基于通配符的模糊集合可以表示为 $W_{w,d}=\{w'_{w,d}\}$，其中 $w'_{w,d}$ 表示关键字 w 拥有 d 个通配符的一个模糊关键字。例如，对于关键字 cat 且预先设定编辑距离为 1，它基于通配符的模糊关键字的集合可以构造为 $W_{cat,1}=\{cat,{}^*cat,{}^*at.c^*at,c^*t,cat^*\}$。

(1)Setup(1^ℓ)

整个系统的初始化过程，以安全参数 1^ℓ 作为输入，输出公钥 PK 及主密钥 MSK，建立一个组合阶双线群 $(N=p_1p_2p_3,G,G_T,e) \xleftarrow{R} \mathcal{G}(1^\ell)$，$G_i$ 表示阶为 p_i 的子群，其中 $i=1,2,3$；定义一个哈希函数 $h:\{0,1\}^* \to \{0,1\}^l$；随机选择生成元 $g \in G_1$，$X_3 \in G_3$；随机选择元素 $\alpha \in Z_N$，$v_i,h_j \in G_1$，其中 $i=1,\cdots,D;j=1,\cdots,L$；$PK=(U,N,g,X_3,v_1,\cdots,v_D,h_1,\cdots,h_L,e(g,g)^\alpha)$；$MSK=\alpha$。

(2)Encrypt(M,PK,S)

该步骤由为 Alice 诊断的医生执行，将 Alice 的 EHR 数据 M 根据长度为 k 的属性向量集合 S 加密并上传至公共云服务器，随机选择一个 $s \in Z_N$，计算：

$$C=Me(g,g)^{\alpha s},E=g^s$$

属性向量集合 S 中的每一个属性向量 $\boldsymbol{u}=(u_1,u_2,\cdots,$

u_k),第一个坐标 u_1 实际上有两个下标,记为$(1,j)$,表示 u_1 在属性矩阵中第一行的第 j 个入口,然后从公钥中选择一个与 j 相对应的 v_j,计算:

$$C_j=v_j^s(h_1^{u_1}\cdots h_k^{u_k})^s$$

则密文 $CT=(C,E\langle C_j\rangle_{j=1,\cdots,|S|})$。

(3)KeywordIndexGen(S,FID,W)

当医生将 Alice 的个人健康记录加密上传至云服务器时,执行该算法,生成文件标识符 FID,云服务器获得(S,FID,W)后生成关键字集合 W 的索引。对于每一个 $\omega_i\in W$ 的关键字,云服务器计算其索引$\{I_{\omega',S}=h(S,\omega_i')\}_{\omega_i'\in W_{\omega_{i,d}}}$,得到关键字集合索引为$\{Index_{\omega_i}=\{I_{\omega_i',S}\}_{\omega_i'\in W_{\omega_{i,d}}},FID\}_{\omega_i\in W}$。

本文通过构建一棵多叉树来存储关键字的集合,且该多叉树基于一个有限的符号集合。将每一个关键字的索引(哈希值)分成 l/ℓ 个部分,每一部分由 ℓ 个比特位表示,其中 ℓ 是一些安全参数。例如 $\Delta=\{a_i\}$是预先定义的符号集合,则不同的符号数量为 2^ℓ,一个索引的每一部分都可以被表示成一个特殊的符号。该模糊关键字搜索的过程如下:

1)对于每一个 $\omega_i'\in W_{\omega_{i,d}}$ 以及 $\omega_i\in W$,私有云服务器计算$\{I_{\omega',S}=h(S,\omega_i')\}_{\omega_i'\in W_{\omega_{i,d}}}$,然后将每一个索引分成 $a_{i_1},\cdots,a_{i_{l/\ell}}$,对于 $j\in[1,l/\ell]$,$|a_{i_j}|=2^\ell$。

2)根据预先定义的符号集合 Δ,私有云服务器创建一棵多叉树 T_ω,该多叉树在初始化阶段就已经被创建,且包含了所有模糊关键字 $\omega_i\in W$。如果没有现存的节点等译这个符号,那么这个序列的第一个符号 a_1 将被添加到该树中作为根节点的孩子节点,随后将当前节点移到根节点的孩子节点上,该算法是迭代执行的。当最后一个符号被使用后,对应的标识符 FID 将被作为叶子节点附着在最后一个节点上。

(4)SearchQueryGen(S,ω)

当医生需要在云服务器上搜索包含关键字 ω 的患者健康记录时,需要将(S,ω)发送给私有云服务器,然后生成搜索符号$\{SRH=h(S,\omega'_{\omega,d})\}_{\omega'_{\omega,d}\in W_{\omega,d}}$。

(5)Search($\{SRH\}$)

当公共云服务器获得搜索符号$\{SRH\}$后,公共云服务器将在多叉树 T_ω 中查找,并返回所有附着在叶子节点上的文件标识符集合$\{FID\}$。

(6)KeyGen($PK,MSK,\mathbb{A}$)

该算法由患者 Alice 执行,为拥有访问结构 $\mathbb{A}$ 的医生生成解密密钥。首先为访问结构$\mathbb{A}$ 生成一个 $LSSS(\boldsymbol{A},\rho)$,其中 $\boldsymbol{A}$ 是 l 行 n 列的共享生成矩阵,ρ 是将 $\boldsymbol{A}$ 的每一行映射成一个深度为 k 的属性向量的函数,随机选择 $n-1$ 个元素 $s_2,\cdots,s_n\in Z_N$ 得到一个向量 $\boldsymbol{\alpha}=(\alpha,s_2,\cdots,s_n)$。

对于每一个从 1 到 l 的 i,计算 $\lambda_i=A_i\boldsymbol{\alpha}$,其中 A_i 表示 $\boldsymbol{A}$ 的第 i 行向量;使 $\boldsymbol{u}=(u_1,u_2,\cdots,u_k)$为 ρ 映射第 i 行而成的属性向量;假定 $\boldsymbol{u}$ 的第一个坐标 u_1 是属性矩阵中第一行的第 j 个入口,从公钥中选择一个与 j 相对应的 v_j,然后随机选择元素,$r_i\in Z_N$,$R_{i,0},R_{i,1},R_{i,2},R_{i,k+1},\cdots,R_{i,L}\in G_3$,计算:$K_{i,0}=g^{\lambda_i}v_j^{r_i}R_{i,0}$;$K_{i,1}=g^{r_i}R_{i,1}$;$K_{i,2}=(h_1^{u_1}\cdots h_k^{u_k})^{r_i}R_{i,2}$;$K_{i,k+1}=h_{k+1}^{r_i}R_{i,k+1},\cdots,K_{i,L}=h_L^{r_i}R_{i,L}$。则解密密钥为:$SK=\langle K_{i,0},K_{i,1},K_{i,2},K_{i,k+1},\cdots,K_{i,L}\rangle_{i=1,\cdots,l}$。

(7)Delegate($PK,SK',\mathbb{A}$)

该算法由医生执行,通过自己的访问结构$\mathbb{A}'$的密钥 SK' 为访问结构$\mathbb{A}$ 生成密钥 SK,其中 $SK'=\langle K'_{i',0},K'_{i',1},K'_{i',2},K'_{i',k+1},\cdots,K'_{i',L}\rangle_{i'=1,\cdots l'}$。$\mathbb{A}'$是 l' 个深度为 k 的属性向量上的访问结构,$\mathbb{A}$ 是 l 个深度为 $k+1$ 的属性向量上的访问结构,如果$\mathbb{A}$ 和$\mathbb{A}'$满足授权条件,则算法过程如下:对于$\mathbb{A}$ 中包含的每一个 $\boldsymbol{u}$,在$\mathbb{A}'$中找到它的前缀 $\boldsymbol{u}'$,如 $\boldsymbol{u}=(\boldsymbol{u}',u_{k+1})$。假设 $\boldsymbol{u}'$与$\mathbb{A}'$的共享生成矩阵的第 i'行相关,则对于每一个从 1 到 l 的 i,随机选择元素 $\gamma_i,\delta_i\in Z_N$,$R_{i,0},R_{i,1},R_{i,2},R_{i,k+2},\cdots,R_{i,L}\in G_3$,然后从 SK'提取 $\boldsymbol{u}'$的密钥组件 $K'_{i',0},K'_{i',1},K'_{i',2},K'_{i',k+1},\cdots,K'_{i',L}$,计算 $\boldsymbol{u}$ 的密钥组件:$K_{i,0}=(K'_{i',0})^{\gamma_i}v_j^{\delta_i}R_{i,0}$;$K_{i,1}=(K'_{i',1})^{\gamma_i}g^{\delta_i}R_{i,1}$;$K_{i,2}=(K'_{i',2})^{\gamma_i}(K_{i',k=1})^{\gamma_i u_{k+1}}(h_1^{u_1},\cdots,h_{k+1}^{u_{k+1}})^{\delta_i}R_{i,2}$;$K_{i,k+2}=(K'_{i',k+2})^{\gamma_i}h_{k+2}^{\delta_i}R_{i,k+2}$;$\cdots$;$K_{i,L}=(K'_{i',L})^{\gamma_i}h_L^{\delta_i}R_{i,L}$。

隐式地设 $r_i=\gamma_i r'_{i'}+\delta_i$,其中 $r'_{i'}$是用于为 $\boldsymbol{u}'$创建密钥组件的随机指数。由于 δ_i 是随机产生的,因此 r_i 是一个随机值,最后输出 $SK=\langle K_{i,0},K_{i,1},K_{i,2},K_{i,k+1},\cdots,K_{i,L}\rangle_{i=1,\cdots,l}$。

(8)Decrypt(CT,SK,PK)

该算法由满足访问策略的医生执行,通过关键字搜索得到需要的密文 CT,输入自己在长度为 k 的属性向量上的访问结构 $\mathbb{A}$ 的密钥 $SK=\langle K_{i,0},K_{i,1},K_{i,2},K_{i,k+1},\cdots,K_{i,L}\rangle_{i=1,\cdots,l}$和由长度为 k 的属性向量集合 S 加密的密文 $CT=(C,E,\langle C_j\rangle_{j=1,\cdots,|S|})$,恢复出 Alice 的密文数据 M。

若 $S\in\mathbb{A}$,计算常量$\{\omega_i\in Z_N\}_{\rho(i)\in S}$,例如 $\sum_{\rho(i)\in S}\omega_i A_i=(1,0,\cdots,0)$。将 $\rho(i)$作为 S 中的第 j 个属性向量,计算:$M'=\prod_{\rho(i)\in S}\left(\frac{e(E,K_{i,0})\cdot e(E,K_{i,2})}{e(C_{j,0},K_{i,1})}\right)^{\omega_i}$,输出密文 $M=C/M'$。

在代理授权生成密钥阶段,当从访问结构$\mathbb{A}'$的密钥授权生成访问结构$\mathbb{A}$ 的密钥时,访问结构$\mathbb{A}$ 的 $LSSS(\boldsymbol{A},\rho)$就已经同时生成,共享生成矩阵 $\boldsymbol{A}$ 中的每一行 $A_i=A'_{i'}\gamma_i$,其中 $A'_{i'}$是共享生成矩阵 $\boldsymbol{A}'$的第 i 行,函数 ρ 将第 i 行映射为一个属性向量 $\boldsymbol{u}$。$\lambda_i=\gamma_i\lambda'_{i'}=\gamma_i A'_{i'}\boldsymbol{\alpha}=A_i\boldsymbol{\alpha}$ 是 $\boldsymbol{u}$ 的一个共享,$\lambda'_{i'}$是 $\boldsymbol{u}'$的一个共享。通过计算可得:

$$M'=\prod_{\rho(i)\in S}\left(\frac{e(g^s,g^{\lambda_i})\cdot e(g^s,v_j^{r_i})\cdot e(g^s,(h_1^{u_1}\cdots h_k^{u_k})^{r_i})}{e(v_j^s,g^{r_i})\cdot e((h_1^{u_1}\cdots h_k^{u_k})^s,g^{r_i})}\right)^{\omega_i}=e(g,g)^{s\sum_{\rho(i)\in S^s}\omega_i A_i\boldsymbol{\alpha}}=e(g,g)^{s\boldsymbol{\alpha}}$$

本方案的主要算法复杂度分析包括 Key generation,Key delegation,Encryption 和 Decryption 4 个算法的复杂度。方案是建立在双线性群 G 和 G_T 之上的,最复杂的计算是在子群 G_1 上进行的。因此,通过在 G_1 上群的基本运算、双线性映射和指数运算来评估计算时间 t_p 和 t_e 的消耗。与 t_p 和 t_e 所消耗的时间相比,乘法运算的消耗可以忽略不计。

表 1 列出了各个主要算法的时间消耗。其中,L 表示系统的最大深度;l 为与密钥相关联的属性向量的个数;l'为与代理授权密钥相关联的属性向量的个数;k 为与密文相关联的属性向量或用户代理授权生成密钥时属性向量的深度;l^*是在解密时满足访问策略的一个集合的属性向量的个数。可以看出,Key generation 算法的时间消耗随着 L 和 l 的乘积呈线性增长,而与用户的深度无关;Key delegation 算法的时间消耗与代理授权者的属性向量深度有关,并且随着深度的增

加而减小；加密的时间随着集合 S 的基数和其中属性向量深度的乘积呈线性增长。与现有的方案相比，该方案中密文短的原因在于密文与集合 S 的基数呈线性关系，这使得解密时间的消耗在匹配属性向量的数量上是线性的，并且与深度无关。在现有方案中，生成密文组件 C_j 时，需要引入 j 个随机数，而该方案生成 C_j 时并不需要随机数的参与，从而缩短了加密时间以及密文组件的长度。最新的 KP-ABE[1,8,16] 方案与本文方案中的这些特征相比，并没有实现灵活的密钥授权。

表 1 算法时间消耗

Table 1 Time consuming of algorithms

算法	计算复杂度
Key generation	$(L+3)\cdot l\cdot t_e$
Key delegation	$(2L-k+5)\cdot l'\cdot t_e$
Encryption	$((k+2)\|S\|+2)\cdot t_e$
Decryption	$3l^*\cdot t_p$

5 安全性分析

本方案足够安全，在任何多项式时间内，攻击者如果没有密钥，都不能从已加密的消息中获得有用的信息。通过证明 3.5 节中的两个假设成立来说明其安全性。

在证明过程中，利用双重系统加密。双重系统加密是由 Waters[17] 提出的一种方法，该方法非常适用于证明 ABE 方案的完全安全性。按照其要求，需要构建半功能密钥和密文。半功能密钥能够用来解密正常密文，半功能密文可被正常密钥解密。然而，半功能密钥不能解密半功能密文，大多数采用双重系统的加密，都未通过模拟器尝试解密挑战密文来测试挑战密钥的性质。为了避免这种悖论，通过在挑战密钥和挑战密文中设置随机值来确保输入挑战密钥解密总是成功的，还需要证明在敌手的立场上这些随机值都是均匀分布的，并不能得到密钥去解密密文。

接下来需要定义一系列游戏，通过证明在攻击者的立场具有两两不可区分性来证明系统的安全性。第一个游戏是 $Game_{\text{real}}$，此为普通的游戏，它表示如定义 1 所定义的游戏。第二个游戏是 $Game_{\text{real}'}$，除了攻击者 $\mathcal{A}$ 没有向挑战者 $\mathcal{C}$ 请求代理授权密钥外，其余与 $Game_{\text{real}}$ 相似。第三个游戏是 $Game_0$，在该游戏中所有的密钥都正常，挑战密文为半功能密文；设 $\mathcal{A}$ 的密钥查询次数为 q，对于所有的 $v=1,\cdots,q$，定义 $Game_v$，挑战密文为半功能密文，而前 v 次密钥查询返回的是半功能密钥，剩下的查询返回正常密钥。当 $v=q$ 时，在 $Game_q$ 中，所有的密钥都为半功能。最后一个游戏是 $Game_{\text{final}}$，此游戏中所有的密钥都是半功能的，挑战密文则是对一个随机消息加密得到的一个半功能密文。我们将证明这些游戏在假设 1 和假设 2 下是不可区分的。半功能密文和密钥的生成方式如下。

半功能密文：设 g_2 为 G_2 的生成元，初次调用加密算法生成正常的密文为 $(\overline{C},\overline{E},\langle\overline{C_{j^*}}\rangle_{j^*=1,\cdots,|S^*|})$，随机选择 $c\in Z_N$；对所有的 $j^*=1,\cdots,|S^*|$，随机选择 $\varphi_{j^*}\in Z_N$，则生成的半功能密文为 $C=\overline{C},E=\overline{E}g_2^c,C_{j^*}=\overline{C_{j^*}}g_2^{\varphi_{j^*}}$。

半功能密钥：初次调用密钥生成算法生成正常密钥 $\langle\overline{K}_{i,0},\overline{K}_{i,1},\overline{K}_{i,2},\overline{K}_{i,k+1},\cdots,\overline{K}_{i,L}\rangle_{i=1,\cdots,l}$。接着，对共享矩阵 $\boldsymbol{A}$ 的第 i 行随机选择 $f_i\in Z_N$；随机选择元素 $\xi_1,\xi_2,\cdots,\xi_D,\eta_1,\eta_2,\cdots,\eta_L\in Z_N$ 和随机的向量 $\vartheta\in Z_N^n$，则半功能密钥如下：$K_{i,0}=\overline{K}_{i,0}g_2^{A_i\vartheta+f_i\xi_x}$；$K_{i,1}=\overline{K}_{i,1}g_2^{f_i}$；$K_{i,2}=\overline{K}_{i,2}g_2^{f_i\sum_{i=1}^{k}u_j\eta_j}$；$K_{i,k+1}=\overline{K}_{(i,k+1)}g_2^{f_i\eta_{k+1}}$；$\cdots$；$K_{i,L}=\overline{K}_{(i,L)}g_2^{f_i\eta_L}$

最后通过 2.3 节定义的安全模型证明引理 1－引理 3，来说明各个游戏之间的不可区分性，即攻击者在以上定义的游戏中的优势可以忽略。

引理 1 对于任意攻击者 $\mathcal{A}$，$Game_{\text{real}}Adv_{\mathcal{A}}=Game_{\text{real}'}Adv_{\mathcal{A}}$。

证明：在本文所构造的方案中，密钥生成算法生成的密钥和委任授权算法生成的密钥都是均匀分布的，因此在攻击者的立场，这两种密钥是没有区别的。

引理 2 如果 $\mathcal{A}$ 能够以优势 ε 区分 $Game_0$ 和 $Game_{\text{real}'}$，那么我们可以创建算法 $\mathcal{B}$ 以优势 ε 打破假设 1。

证明：通过假设 1 的元组 (g,X_3,T)，构建一个算法 $\mathcal{B}$ 来模拟 $Game_{\text{real}'}$ 或 $Game_0$ 与 $\mathcal{A}$ 的相互作用。

初始化：算法 $\mathcal{B}$ 随机选择 $\alpha\in Z_N$，对所有的 $i=1,\cdots,D$ 和 $j=1,\cdots,L$，它随机选择 $\overline{\xi}_i,\overline{\eta}_j\in Z_N$，并计算 $v_i=g^{\overline{\xi}_i}$，$h_j=g^{\overline{\eta}_j}$，将公共密钥 PK 发送给 $\mathcal{A}$：

$$PK=(U,N,g,v_1,\cdots,v_D,h_1,\cdots,h_L,e(g,g)^{\alpha})$$

密钥生成阶段 1、阶段 2：注意 $\mathcal{B}$ 得到了主密钥 $MSK=\alpha$，因此在阶段 1 和阶段 2 中，$\mathcal{B}$ 能够执行密钥生成算法生成正常的密钥。

挑战：$\mathcal{A}$ 给 $\mathcal{B}$ 提供两个长度相等的消息 M_0,M_1 和属性向量的集合 $S^*=\{\boldsymbol{u}\}$，$\mathcal{B}$ 使用已给出的元组中的 T 按照如下方式生成半功能或正常密文。

$\mathcal{B}$ 随机抛一枚硬币得 $b\in\{0,1\}$，则半功能密文 CT 为：$C=M_be(g,T)^{\alpha},E=T,C_{j^*}=T^{\overline{\xi}_x}T^{(\overline{\eta}_1u_1+,\cdots+\overline{\eta}_ku_k)}$；如果设 $T=g^sg_2^c$，则隐式地设：$\varphi_{i^*}=c(\overline{\xi}_x+\sum_{j=1}^{k}u_j\overline{\eta}_j)$。

但是根据中国剩余定理，在 $\varphi_{i^*}\bmod p_2$ 与 $(\overline{\xi}_x,\overline{\eta}_j\bmod p_2)$ 之间没有不必要的相关性，因此密文的 G_1 部分与 G_2 部分是不相关的。

猜测：如果 $T\in G_{12}$，则 CT 为正确分布的半功能密文，因此，当前游戏为 $Game_0$；如果 $T\in G_1$，通过隐式设置 $T=g^s$，则 CT 为正确分布的正常密文，因此，当前游戏为 $Game_{\text{real}'}$；当 $\mathcal{A}$ 输出 $b'=b$ 时，$\mathcal{B}$ 输出 0，因此，在元组 (g,X_3,T) 下，打破假设 1，$\mathcal{B}$ 的优势为：

$$|\Pr[\mathcal{B}(g,X_3,T\in G_{12})=0]-\Pr[\mathcal{B}(g,X_3,T\in G_1)=0]|=|Game_0Adv_{\mathcal{A}}-Game_{\text{real}'}Adv_{\mathcal{A}}|=\varepsilon$$

其中，$Game_0Adv_{\mathcal{A}}$ 为 $\mathcal{A}$ 在游戏 $Game_0$ 中的优势，$Game_{\text{real}'}Adv_{\mathcal{A}}$ 为 $\mathcal{A}$ 在游戏 $Game_{\text{real}'}$ 中的优势。

引理 3 如果 $\mathcal{A}$ 能够以优势 ε 区分 $Game_q$ 和 $Game_{\text{final}}$，那么我们可以创建算法 $\mathcal{B}$ 以优势 ε 打破假设 2。

证明：通过假设 2 的元组 $(g,g^{\alpha}X_2,X_3,g^sY_2,Z_2,T)$，构建一个算法 $\mathcal{B}$ 来模拟 $Game_q$ 或 $Game_{\text{final}}$ 与 $\mathcal{A}$ 的相互作用。

初始化：对所有的 $i=1,\cdots,D$ 和 $j=1,\cdots,L$，$\mathcal{B}$ 随机选择 $\overline{\xi}_i,\overline{\eta}_j\in Z_N$，并计算 $v_i=g^{\overline{\xi}_i}$，$h_j=g^{\overline{\eta}_j}$，将公共密钥 PK 发送给 $\mathcal{A}$：

$$PK=(U,N,g,v_1,\cdots,v_D,h_1,\cdots,h_L,e(g,g^{\alpha}X_2))$$

$\mathcal{B}$ 并不知道主密钥 α。

密钥生成阶段 1 和阶段 2：为了模拟 $\boldsymbol{A}$ 的半功能密钥，$\mathcal{B}$

先为$\mathbb{A}$生成$LSSS(\boldsymbol{A},\rho)$。选择两个向量:1)$\phi$,它的第一个坐标设为 1,剩余的 $n-1$ 个坐标在 Z_N 中随机取值获得;2)ψ,它的第一个坐标设为 0,剩余的 $n-1$ 个坐标在 Z_N 中随机取值获得。这将隐式地设 $\alpha=\alpha\phi+\psi$。

对 $\boldsymbol{A}$ 的第 i 行 A_i,$\mathcal{B}$ 随机选择 $r_i,\overline{f}_i\in Z_N$;$R_{i,0},R_{i,1},R_{i,2},R_{i,k+1},\cdots,R_{i,L}\in G_3$;$\mathcal{B}$ 随机选择 $\xi_1,\cdots,\xi_D,\eta_1,\cdots,\eta_L$,按照如下方式生成密钥:$K_{i,0}=g^{A_i\psi}(g^{\alpha}X_2)^{A_i\phi}v_j^{r_i}Z_2^{\overline{f}_i\xi_j}R_{i,0}$;$K_{i,1}=g^{r_i}Z_2^{\overline{f}_i}R_{i,1}$;$K_{i,2}=(h_1^{u_1}\cdots h_k^{u_k})^{r_i}Z_2^{\overline{f}_i\sum_{j=1}^{k}u_j\eta_j}R_{i,2}$;$K_{i,k+1}=h_{k+1}^{r_i}Z_2^{\overline{f}_i\eta_{k+1}}R_{i,k+1}$;$\cdots$;$K_{i,L}=h_L^{r_i}Z_2^{\overline{f}_i\eta_L}R_{i,L}$。

通过设 $X_2=g_2^{c_2}$,$Z_2=g_2^{d_2}$,隐式地设 $\vartheta=c_2\phi$,$f_i=d_2\overline{f}_i$。需要注意,在 G_2 中这些值都是被共享的,通过 f_i 适当地随机分配。因此,在 $\mathcal{A}$ 的立场,这些半功能密钥都是均匀分布的。

挑战:当 $\mathcal{B}$ 给出两个长度相等的消息 M_0,M_1 和属性向量集合 S^* 时,$\mathcal{B}$ 随机抛一枚硬币得 $b\in\{0,1\}$,则密文为 $C=M_bT$,$E=g^sY_2$,$C_{j^*}=(g^sY_2)^{\overline{\xi}_x}(g^sY_2)^{(\overline{\eta}_1u_1+\cdots+\overline{\eta}_ku_k)}$。

如果设 $Y_2=g_2^c$,则隐式地设:

$$\varphi_{i^*}=c(\overline{\xi}_x+\sum_{j=1}^{k}u_j\overline{\eta}_j)$$

根据中国剩余定理,在 $\varphi_{i^*}\bmod p_2$ 与 $(\overline{\xi}_x,\overline{\eta}_j\bmod p_2)$ 之间没有不必要的相关性。如果 $T=e(g,g)^{\alpha}$,则密文为消息 M_b 的半功能密文。如果 T 是 G_T 中的随机元素,则密文为一个随机消息的半功能加密。

猜测:如果 $T=e(g,g)^{\alpha}$,则当前游戏为 $Game_q$;如果 T 是 G_T 中的随机元素,则当前游戏为 $Game_{\text{final}}$。当 $\mathcal{A}$ 输出 $b'=b$ 时,$\mathcal{B}$ 输出 0,因此在元组$(g,g^{\alpha}X_2,X_3,g^sY_2,Z_2,T)$下,打破假设 2,$\mathcal{B}$ 的优势为:

$$|\Pr[\mathcal{B}(g,g^{\alpha}X_2,X_3,g^sY_2,Z_2,T=e(g,g)^{\alpha})=0]-\Pr[\mathcal{B}(g,g^{\alpha}X_2,X_3,g^sY_2,Z_2,T\overset{R}{\leftarrow}G_T)=0]|=|Game_qAdv_{\mathcal{A}}-Game_{\text{final}}Adv_{\mathcal{A}}|=\varepsilon$$

其中,$Game_qAdv_{\mathcal{A}}$ 为 $\mathcal{A}$ 在游戏 $Game_q$ 中的优势,$Game_{\text{final}}Adv_{\mathcal{A}}$ 为 $\mathcal{A}$ 在游戏 $Game_{\text{final}}$ 中的优势。

密文完全地隐藏了 b,因此 $\mathcal{A}$ 在游戏中的优势是可以忽略的。通过引理 1-引理 3,说明安全游戏 $Game_{\text{real}}$ 与 $Game_{\text{final}}$ 不可区分,$\mathcal{A}$ 在 $Game_{\text{real}}$ 中的优势是可以忽略的。因此,没有敌手可以在多项式时间内攻破系统。

结束语　本文研究的 ABE 算法基于 KP-ABE,是一种新的 ABE 算法,支持代理授权机制,允许用户作为代理动态地重新定义访问策略。该新方案可适用于在电子医疗环境下对病人的健康数据进行隐私保护。在该应用中,若必须优先确定访问策略,将使得在生成密钥时太过僵化,甚至根本不可用,而新方案可以有效地解决此问题。

参 考 文 献

[1] HOHENBERGER S, WATERS B. Attribute-Based Encryption with Fast Decryption [M] // Public-Key Cryptography-PKC 2013. Springer Berlin Heidelberg, 2013: 162-179.

[2] ZHANG Y H, ZHENG D, LI J, et al. Attribute directly-revocable attribute-based encryption with constant ciphertext length [J]. Journal of Cryptologic Research, 2014, 1(5): 465-480. (in Chinese)
张应辉,郑东,李进,等. 密文长度恒定且属性直接可撤销的基于属性的加密[J]. 密码学报,2014,1(5):465-480.

[3] LI S, XU M Z. Attribute-based searchable encryption scheme [J]. Chinese Journal of Computers, 2014, 37(5): 1017-1024. (in Chinese)
李双,徐茂智. 基于属性的可搜索加密方案[J]. 计算机学报,2014,37(5):1017-1024.

[4] LI M, YU S, ZHENG Y, et al. Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption[J]. IEEE Transactions on Parallel & Distributed Systems, 2012, 24(1): 131-143.

[5] GOYAL V, JAIN A, PANDEY O, et al. Bounded Ciphertext Policy Attribute Based Encryption[M] // Automata, Languages and Programming. DBLP, 2008: 579-591.

[6] WATERS B. Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization [C] // Public Key Cryptograpy-PKC 2011. Springer Berlin Heidelberg, 2011: 53-70.

[7] DENG H, WU Q, QIN B, et al. Ciphertext-policy hierarchical attribute-based encryption with short ciphertexts[J]. Information Sciences, 2014, 275(11): 370-384.

[8] GOYAL V, PANDEY O, SAHAI A, et al. Attribute-based encryption for fine-grained access control of encrypted data[C] // Proceedings of ACM CCS. 2006: 89-98.

[9] ROUSELAKIS Y, WATERS B. Practical constructions and new proof methods for large universe attribute-based encryption[C] // ACM Sigsac Conference on Computer & Communications Security. ACM, 2013: 463-474.

[10] LEWKO A, WATERS B. Unbounded HIBE and Attribute-Based Encryption[C] // International Conference on Theory and Applications of Cryptographic Techniques: Advances in Cryptology. Springer-Verlag, 2011: 547-567.

[11] JUNG T, LI X Y, WAN Z, et al. Control Cloud Data Access Privilege and Anonymity With Fully Anonymous Attribute-Based Encryption[J]. IEEE Transactions on Information Forensics & Security, 2014, 10(1): 190-199.

[12] SUN W, WANG B, CAO N, et al. Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking[J]. IEEE Transactions on Parallel & Distributed Systems, 2014, 25(11): 3025-3035.

[13] SUN W, LIU X, LOU W, et al. Catch you if you lie to me: Efficient verifiable conjunctive keyword search over large dynamic encrypted cloud data[C] // IEEE Conference on Proc of the Computer Communications (INFOCOM). IEEE, 2015: 2110-2118.

[14] YANG B, PANG X Q, DU J Q, et al. Effective Error-Tolerant Keyword Search for Secure Cloud Computing[J]. Journal of Computer Science and Technology, 2014, 29(1): 81-89.

[15] WAN Z, LIU J, DENG R H. HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing[M]. New York: IEEE Press, 2012.

[16] LEWKO A, OKAMOTO T, SAHAI A, et al. Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption[M] // Advances in Cryptology-EUROCRYPT 2010. Springer Berlin Heidelberg, 2010: 62-91.

[17] WATERS B. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions[C] // International Cryptology Conference on Advances in Cryptology. Springer-Verlag, 2009: 619-636.