基于属性代理重加密技术与可容错机制相结合的数据检索方案

刘新宇[1] 李 浪[1] 肖斌斌[2]
(衡阳师范学院计算机科学与技术学院 湖南 衡阳 421002)[1]
(暨南大学信息科学技术学院 广州 510632)[2]

摘 要 针对云服务器上用户信息的隐私问题,提出一种基于属性代理重加密技术与容错机制相结合的方案。该方案将用户存储的数据分为文件和文件的安全索引,将其分别进行加密后存储在不同的云服务器上。首先,利用倒排序结构构造文件的安全索引,并使用模糊提取器对关键字进行预处理,用户可以通过该安全索引进行容错的多关键字搜索;其次,设置访问控制树对解密密钥重加密,实现权限管理,即实现数据在云端的有效共享;最后,通过 Complex Triple Diffle-Hellman 难题证明该方案生成的系统主密钥是安全的,因此该方案在云环境下也是安全的。与已有的方案的对比分析表明,该方案可减少密钥重加密、解密等的计算量,同时通过加入容错处理机制提高了数据检索的效率。

关键词 重加密技术,容错机制,多关键字,访问控制树,模糊提取器
中图法分类号 TP309 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.07.028

Attribute-based Proxy Re-encryption Technology and Fault-tolerant Mechanism Based Data Retrieval Scheme

LIU Xin-yu[1] LI Lang[1] XIAO Bing-bing[2]
(College of Computer Science and Technology, Hengyang Normal University, Hengyang, Hunan 421002, China)[1]
(College of Information Science and Technology, Jinan University, Guangzhou 510632, China)[2]

Abstract Aiming at the privacy of user information stored in the cloud server problem, a scheme based on property broker re-encryption and fault-tolerant mechanism was proposed. This scheme mainly divides the data stored by users into files and the security index of files, encrypts them separately and then stores them on different cloud servers. Firstly, the security index of file is constructed by using the inverted structure and the keywords are preprocessed by using the fuzzy extractor, so the users can search multi-keywords with fault tolerance through the security index. Secondly, the access control tree is used for re-encryption of decryption key to realize right management, namely, the effective sharing of data in cloud. Finally, the scheme is proved to be secure in cloud environment through Complex Triple Diffie-Hellman problem, proving that the system master key generated by this scheme is secure. Compared with the existing schemes, it is shown that the scheme can reduce the computational complexity of key re-encryption and decryption, and the fault-tolerant mechanism improves the efficiency of data retrieval.

Keywords Re-encryption technology, Fault-tolerant mechanism, Multiple keywords, Access control tree, Fuzzy extractors

1 引言

云技术在电子医疗应用方面的高速发展给用户带来了巨大的便利。为有效管理和利用海量的电子病历数据,医疗机构将耗费巨大的人力、物力以及财力。虽然云计算与云存储的出现在一定程度上为医疗机构带来了很大便利,但其仍然面临着海量的数据存储、数据隐私安全以及数据共享问题,因此设计一种安全高效的数据检索方案一直是国内外学者研究的热点。文献[1-2]提出了数据拥有者利用属性加密[3-4]来实现数据共享的策略,该策略给合法用户派发合适的密钥,只有被授权用户才可以访问被加密的数据。但是,基于属性的加密方案不支持关键字搜索功能,即使授权用户的属性满足指定的访问控制策略,该授权用户也必须在下载所有密文后通过解密来获得明文,这样会增加系统负担。文献[5-7]提出的多关键字检索方案虽然提高了检索效率,但不支持多关键字的容错检索。文献[8-12]提出的基于密文检索的方案都假定

到稿日期:2017-02-08 返修日期:2017-04-10 本文受国家自然科学基金资助项目(61572174),湖南省教育厅资助科研项目(15A029)基金资助。

刘新宇(1990-),男,硕士,主要研究方向为密码学与信息安全,E-mail:x.y.liu@foxmail.com;李 浪(1971-),男,博士,教授,CCF高级会员,主要研究方向为密码学与信息安全,E-mail:lilang911@126.com;肖斌斌(1990-),男,硕士,主要研究方向为密码学与信息安全,E-mail:954743352@qq.com(通信作者)。

云服务器是可信的，即云服务器不会脱离原先设定的安全协议，且此类方案不支持多关键字的容错检索，因此不能有效地应用于电子医疗环境中。

为了解决上述问题，依托云服务器在电子医疗环境中对个人健康记录(PHR)[13]的使用情况，提出一种基于属性代理重加密技术与容错机制相结合的多关键字检索方案。该方案首先允许被授权用户通过关键字搜索感兴趣的密文；然后利用属性代理重加密技术让数据拥有者能够有效地将密文共享给其他用户，同时利用模糊提取器构造容错机制来对关键字的索引进行预处理，用户可利用该索引进行多关键字的容错检索，从而提高检索的效率。文中通过 Complex Triple Diffle-Hellman 难题证明了该方案生成的系统主密钥是安全的，从而保证云服务器环境下该方案是安全的；同时，将所提方案与同类方案[14-16]进行对比，分析结果表明所提方案不仅降低了计算量，而且利用模糊提取器构造的容错机制对关键字索引进行预处理时可提高检索效率。

2　基础知识

2.1　双线性映射

设 G_1 和 G_2 是循环群，其素数阶 p 和 g 是 G_1 的生成元，对于映射 $e:G_1\times G_1\rightarrow G_2$，假设在群 G_1 和群 G_2 上的离散对数问题都是困难性问题，若满足以下 3 个条件，则称此映射为双线性映射。

1)双线性：对于任何的 $g\in G_1$，$h\in G_1$，a、$b\in Z_p$，均能使 $e(g^a,h^b)=e(g,h)^{ab}$ 成立。

2)不可退化性：对于任意的 x，$y\in G_1$，使其满足 $e(x,y)\neq 1_{G_2}$，其中 1_{G_2} 是群 G_2 上的一个单元。

3)可计算性：对于所有的 x，$y\in G_1$，存在一个高效的算法可计算 $e(x,y)$。

2.2　CTDH 复杂性假设

定义 1(Complex Triple Diffle-Hellman(CTDH)问题)　存在一个双线性映射 $e:G\times G\rightarrow G_T$，其中 G 是素数阶为 p 的循环群，g 是该循环群的一个生成元，随机数 n，a，b，c，d，$R\in Z_p$。输入元组$\langle g,n,g_b=g^b,g_c=g^c,g_d=g^d,g_1=g^{\frac{c}{b}},g_2=g^{bc},g_3=g^{ac},g_4=g^{abc-Rc},g_5=g^{c(R+nd)},g_6=g^{\frac{c(R+nd)}{b}}\rangle$，输出 g^{abc}。

定义 2(CTDH 假设)　如果不存在概率多项式时间敌手能够以不可忽略的优势在输入为$\langle g,n,g_b=g^b,g_c=g^c,g_d=g^d,g_1=g^{\frac{c}{b}},g_2=g^{bc},g_3=g^{ac},g_4=g^{abc-Rc},g_5=g^{c(R+nd)},g_6=g^{\frac{c(R+nd)}{b}}\rangle$的情况下输出 g^{abc}，则在群 G 中存在 CTDH 假设。

2.3　访问控制树

文中方案所采用的访问策略是通过访问控制树 $ATree$ 实现的。该树的每个非叶子节点代表一个门限，每一个叶子节点描述一个属性。该访问控制树与文献[13]中所定义的访问控制树类似。对于给定的访问控制树，num_x 表示节点 x 的孩子节点的个数，k_x 是 x 的阈值，且 $0\leqslant k\leqslant num_x$。如果至少有 k_x 个孩子节点被赋值为真，那么该节点将被赋值为真。特别地，当 $k_x=1$ 时，该节点便成为了 OR 门；当 $k_x=num_x$ 时，该节点便成为了 AND 门。

如果用户属性集合 S 满足访问控制树 $ATree$ 或者节点 x，则定义 $ATree(S)=1$ 或 $x(S)=1$。$ATree(S)$是通过以下递归算法计算得出的。如果 x 是叶子节点，当且仅当 $att(x)\in S$ 时，$x(S)=1$；如果 x 是非叶子节点，当节点 x 至少有 k_x 个孩子节点返回 1 时，$x(S)=1$。对于访问控制树 $ATree$ 的根节点 R_p，只有 $R_p(S)=1$ 时，$ATree(S)=1$。

2.4　模糊提取器

模糊提取器由安全概略和强提取器两部分组成。首先给出安全概略的定义：安全概略由一对随机过程$[SS,REC]$定义，SS 是概略过程，REC 是恢复过程。

定义 3(模糊提取[17])　一个(k,ε)的提取器是函数 $EXT:\{0,1\}^n\times\{0,1\}^d\rightarrow\{0,1\}^m$，任意定义在$\{0,1\}^n$ 上具有最小熵 k 的 X 分布 $EXT(X;U_d)$，也就是函数的输出 ε 接近 $\{0,1\}^m$ 的均匀分布 U_m。直观上，随机提取器借助一个真正的随机种子将最小熵从随机变量 X 中提取出来。

定义 4(模糊提取器[17])　一个(M,m,m',n)的安全概略是一对随机过程$[SS,REC]$。这对随机过程具有如下性质：

1)概略过程。SS 的输入为 M 上的一个元素 w，输出为 $s\in\{0,1\}^*$。

2)恢复过程。REC 输入的元素有 2 个，$w'\in M$，$s\in\{0,1\}^*$。如果 w 和 w'的距离小于 t，那么 $Rec(w',ss(w))=w$。

3)安全概略。对任意 M 上的分布 W，只要 W 上的最小熵大于 m，安全概略就可以保证其输出位串 s 后 W 的条件最小熵大于或等于 m'，即 $H\infty(W/SS)\geqslant m'$。有了安全概略的概念之后，即可定义模糊提取器。

定义 5(模糊提取器[17])　一个(M,n,l,t,ε)的模糊提取器是满足以下特性的一对随机过程(GEN,REP)。模糊提取器的结构如图 1 所示。

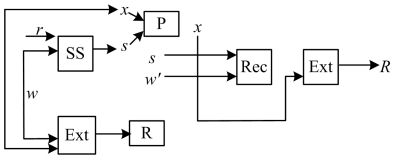

图 1　模糊提取器的结构

Fig. 1　Structure of fuzzy extractor

产生过程：GEN 输入 $w\in M$，输出 $s\in\{0,1\}^l$ 上的一个随机串 R 和$\{0,1\}^*$的一个帮助串 p。

重生过程：REP 输入 $w'\in M$ 和 $p\in\{0,1\}^*$。如果 $dis(w,w')\leqslant t$，$GEN(w)=(R,p)$，那么 $REP(w',p)=w$。

模糊提取器提供以下保证：如果定义在 M 上的随机变量 w 的最小熵大于 m，那么敌手便可观察到 p，R 非常接近均匀分布，即如果 $GEN(w)=(R,p)$，那么 $dis((R,p),(U_l,p))\leqslant\varepsilon$。

引理 1(从安全概略中得到模糊提取器[17])　保证(SS,REC)是(M,m,m',n)的安全概略，并且 Ext 是(M,n,l,t,ε)的强提取器，那么(Gen,Rep)是(M,n,l,t,ε)的模糊提取器。其中，Gen 和 Rep 的定义如下。

$Gen(w^*,r,x)$：设 $p=(SS(w,r),x)$，$R=Ext(w,x)$，输出为(R,p)。

$Rep(w',(s,x))$：输入 $w=REP(w',s)$，输出 $R=Ext(w,x)$。

3 基于属性重加密技术与可容错机制相结合的数据检索方案

3.1 系统模型

适用于电子医疗环境下的数据检索系统包含 4 个实体：私有云服务器、公有云服务器、数据所有者以及数据用户。其中，用户是指合法的组织或个人。方案的具体过程如下：

1)为保证外包数据的安全，医疗机构需要加密所属明文文件，并将密文文件上传至公共云服务器。

2)为提供基于密文的检索功能，医疗机构需要将关键字集合上传至私有云服务器。

3)私有云服务器生成安全索引后，将其上传至公有云服务器。

4)用户需要通过关键字检索存储在云端的电子病例数据时，需要将搜索关键字提交给私有云服务器。

5)私有云服务器生成搜索符号，并将此搜索符号提交给公共云服务器。

6)公共云服务器经过重加密过程后，将相应的密文返回给该查询用户。

具体的系统模型如图 2 所示。

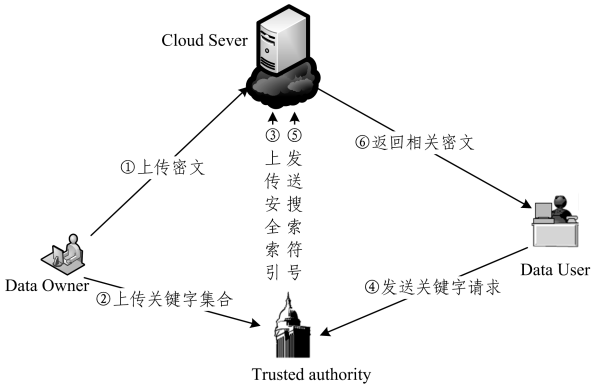

图 2 系统模型

Fig. 2 System model

3.2 形式化定义

在基于属性代理重加密技术与可容错机制相结合的数据检索方案中，数据检索系统主要包括公有云服务端、私有云服务端、数据所有者和数据用户 4 个实体。下面给出该系统设计方案所包含算法的形式化定义。

1)$Setup(1^{\lambda})$：输入安全参数 1^{λ}，算法输出系统参数 λ、主密钥 MK 和 K。

2)$KeyGen(A^{u},MK)$：输入主密钥 MK 和用户属性 A^{u}，密钥生成算法输出用户密钥 $userKey$。

3)$Enc(F,ATree)$：输入明文文件集合 F 和访问控制树 $ATree$，加密算法输出密文 C 以及密文文件 C_F。

4)$KeywordIndex(W,\delta)$：输入关键字集合 W 和文件索引 δ，关键字索引生成算法输出安全索引 γ。

5)$SearchQueryGen(w)$：输入搜索关键字 w，搜索查询生成算法输出搜索符号 τ_Q。

6)$Search(\gamma,\tau_Q)$：输入安全索引 γ 和搜索符号 τ_Q，搜索算法输出对应的密文文件 C_F。

7)$ReKeyGen(userKey,ATree')$：输入用户的属性密钥 $userKey$ 和访问控制树 $ATree'$，重加密密钥生成算法输出重加密密钥 rk。

8)$ReEnc(rk,C)$：输入重加密密钥 rk 和密文 C，重加密算法输出重加密密文 C_{re}。

9)$Dec(C,userKey)$：输入密文 C 和用户属性密钥 $userKey$，解密算法输出解密密钥 K_e，利用该密钥进行解密，从而获得对应明文 F。

3.3 方案的构造

对于文件集合 $F=\{f_1,f_2,\cdots,f_n\}$，为了实现关键字搜索功能，所提方案需要先构造文件安全索引 γ，如图 3 所示。$\delta=\{a_{j,i}\}$ 表示一个 $m\times n$ 的矩阵，用 δ_j 表示 δ 中的第 j 行。如果文件 f_i 中包含关键字 w_j，则 $a_{j,i}=1$，否则 $a_{j,i}=0$。

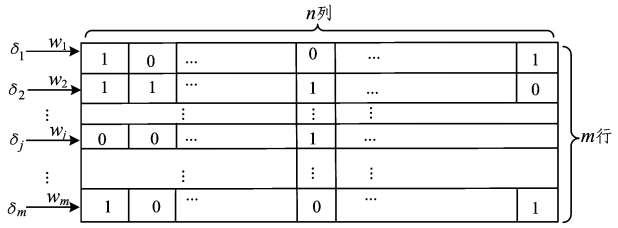

图 3 索引矩阵

Fig. 3 Index matrix

1)$Setup(1^{\lambda})$：系统的初始化阶段。利用一个素数阶为 p 的循环群，生成一个双线性对映射 $e:G\times G\rightarrow G_T$。在 Z_p 中随机选择元素 $k,y,z,t_i(1\leqslant i\leqslant 3n)$，在循环群 G 中随机选择两个生成元 g 和 h。对于每个 $1\leqslant i\leqslant 3n$，使得 $Y=e(g,h)^{y}$，$T_i=g^{t_i}$。与此同时，私有云服务器生成一个密钥集合 $K=(k_1,k_2)$，其中 k_1 和 k_2 是伪随机函数 prf_k 的密钥，则系统公共参数 $pub=\langle e,g^{z},h,Y^{kz},\{T_i,\frac{t_i}{k\cdot z}\}_{1\leqslant i\leqslant 3n}\rangle$，主密钥 $MK=\langle k,y,z,\{t_i\}_{1\leqslant i\leqslant 3n}\rangle$。

2)$KeyGen(A^{u},MK)$：用户的属性密钥生成阶段。A^{u} 表示用户的属性集合。随机从 Z_p 中选择 $r_1,r_2,\cdots,r_n$，使得 $r=r_1+r_2+\cdots+r_n$。接着计算 $\hat{D}=(h^{y-r})^{k}$，同时对于每个 $i\in N$ $(N=\{1,2,,\cdots,n\})$，有 $D_{i,1}=h^{r_i}$。输出用户的密钥 $userKey=\langle A^{u},(D_{i,1})_{i\in N},\hat{D},k,z\rangle$。

3)$Enc(F,ATree)$：数据加密阶段。数据拥有者利用现有加密算法以及对称密钥 K_e、加密文件 F 生成密文文件 C_F。$ATree$ 表示一个访问控制树，为了加密该密钥 K_e，该算法选择一个随机数 $s\in Z_p$，同时计算 $\widetilde{C}=K_e\cdot Y^{skz}$，$\hat{C}=g^{sz}$，$\breve{C}=h^{skz}$，对于 $i\in N$，如果 $+d_i$ 出现在 $ATree$ 上，则 $C_i=T_i^{s}$；如果 $-d_i$ 出现在 $ATree$ 上，则 $C_i=T_{i+n}^{s}$，否则 $C_i=T_{2n+i}^{s}$。最后输出 $C=\langle ATree,\widetilde{C},\hat{C},\breve{C},(C_i)_{i\in N}\rangle$ 以及密文文件 C_F，数据拥有者将 C 和 C_F 上传至公共服务器。

4)$KeywordIndex(W,\delta)$：安全索引生成阶段。$W=\{w_1,$

$w_2,\cdots,w_m\}$表示文件中所有不同关键字的集合，δ 表示已定义的 $m\times n$ 文件索引矩阵。私有云服务器按照如下过程构建安全索引 $\gamma=\{((R_{\varphi(j)},P_{\varphi(j)}),\delta'_{\varphi(j)})_{1\leqslant i\leqslant m}\}$。

①对于每个关键字 $w_j\in W$，计算 $R_{\varphi(j)}=prf_{k_1}(w_j)$，$t_j=w_j\oplus K_1$；

②δ_j'等于 $\delta_j\oplus prf_{k_2}(w_j)$的前 n 比特位；

③运用 φ 在$\{1,2,\cdots,m\}$上的随机排列，最后生成此安全索引 γ 并将其上传至公有云服务器。

5）*SearchQueryGen*(w)：搜索符号生成阶段。w 为用户输入的搜索关键字。私有云服务器从数据使用者处获得搜索关键字 w 后调用此算法生成搜索符号 $\tau_Q=\{R_b',\beta_{j_b}=[prf_{k_2}(w_{j_b})]_{j=1,\cdots,n}\}$，其中 β_j 表示 $prf_{k_2}(w)$的前 n 比特位，私有云服务器将此搜索符号 τ_Q 上传至公共云服务器。

6）*Search*(γ,τ_Q)：基于搜索符号 τ_Q 的密文搜索阶段。此算法被公有服务器调用，根据私有云服务器提交的搜索符号 $\tau_Q=\{R'_b,\beta_{j_b}=[prf_{k_2}(w_{j_b})]_{1,\cdots,n}\}$，公有云服务器识别安全索引 γ 中的每一个元组$\{(R_{\varphi(j)},P_{\varphi(j)}),\delta'_{\varphi(j)}\}$。如果存在 $R_b'=R_{\varphi(j)}$，那么对于 $1\leqslant b\leqslant t$，公有云服务器便通过 $\delta_j=\beta_j\oplus\delta_j'$恢复出 δ_j，则云服务器通过集合$\{\delta_{j_1},\delta_{j_2},\cdots,\delta_{j_n}\}$的交集获得包含有关键字 W 的加密文件集合。

7）*ReKeyGen*$(userKey,ATree')$：重密钥生成阶段。$ATree'$表示一个访问控制树，$userKey$ 表示一个有效的密钥，$userKey=\langle A^u,(D_{i,1})_{i\in N},\hat{D},k,z\rangle$。随机选择 $d\in Z_p$，使得 $\zeta=g^d$，$\hat{D}'=\hat{D}$。对于 $i\in N$，有 $D'_{i,1}=D_{i,1}\cdot h^d$，$C'$是 ζ 在访问控制树 $ATree'$下的密文。该阶段最后输出重加密密钥 $rk=[A^u.ATree',(D'_{i,1})_{i\in N},\hat{D},k\cdot z,C']$。

8）*ReEnc*(rk,C)：重加密阶段。rk 表示一个有效的加密密钥，并且 $rk=\langle A^u.ATree',(D'_{i,1})_{i\in N},\hat{D},k\cdot z,C']$，$C$ 表示一个结构完整的密文，$C=\langle ATree,\widetilde{C},\hat{C},\breve{C},(C_i)_{i\in N}\rangle$。本阶段将会检查 A^u 是否满足访问控制树 $Atree$。如果不满足，则输出⊥；否则，对于 $i\in N$，如果 $+d_i$ 出现在 $Atree$ 上，$T_i=\frac{t_i}{k\cdot z}$；如果 $-d_i$ 出现在 $Atree$ 上，则 $T_i=\frac{t_{n+i}}{k\cdot z}$，否则 $T_i=\frac{t_{2n+i}}{k\cdot z}$。接着计算：

$$T=\frac{1}{\prod\limits_{i\in N}T_i}=\frac{k\cdot z}{\sum\limits_{j\in S}t_i}=\frac{k\cdot z}{t}\tag{1}$$

$$C=\prod_{i\in N}C_i=g^{s\cdot\sum_{j\in S}t_j}=g^{s\cdot t}\tag{2}$$

$$D=\prod_{i\in N}D_i'=h^{d+\sum_{i\in N}r_i}=h^{n\cdot d+r}\tag{3}$$

继续计算 $E=e(C,D^{\mathrm{T}})=e(g,h)^{(n\cdot d+r)(k,s,z)}$，然后计算：

$$\begin{aligned}\overline{C}&=e(\hat{C},\hat{D})\cdot E\\&=e(g^{s\cdot z},h^{(y-r)k})\cdot e(g,h)^{(n\cdot d+r)(k,s,z)}\\&=e(g,h)^{(k\cdot s\cdot z\cdot y)+(n\cdot d\cdot k\cdot s\cdot z)}\end{aligned}$$

最后输出重加密密文 $C_{re}=\langle ATree',\widetilde{C},\overline{C},\breve{C},C'\rangle$。注意，$C_{re}$可以被迭代地重加密。因此，可以得到 $C'_{re}=\langle ATree'',\widetilde{C},\overline{C},\breve{C},C''\rangle$，$C'_{re}$是算法 *ReEnc* 输入 rk'和 C'时得到的结果。密文的长度和重加密的次数呈线性增加。

9）*Dec*$(C,userKey)$：密文的解密阶段。$userKey$ 表示一个有效的用户密钥，$userKey=[A^u,(D_{i,1})_{i\in N},\hat{D},k,z\rangle$。该算法检测 A^u 是否满足 $Atree$。如果不满足，则输出⊥；否则，进行解密操作。若 C 是原始完整的密文且 $C=\langle ATree,\widetilde{C},\hat{C},\breve{C},(C_i)_{i\in N}\rangle$，那么对于 $i\in N$，如果 $+d_i$ 出现在 $Atree$ 上，则 $T_i=\frac{t_i}{k\cdot z}$；如果 $-d_i$ 出现在 $Atree$ 上，则 $T_i=\frac{t_{n+i}}{k\cdot z}$，否则 $T_i=\frac{t_{2n+i}}{k\cdot z}$。该算法首先计算：

$$T=\frac{1}{\prod\limits_{i\in N}T_i}=\frac{k\cdot z}{\sum\limits_{j\in S}t_i}=\frac{k\cdot z}{t}\tag{4}$$

$$C=\prod_{i\in N}C_i=g^{s\cdot\sum_{j\in S}t_j}=g^{s\cdot t}\tag{5}$$

$$D=\prod_{i\in N}D_i'=h^{d+\sum_{i\in N}r_i}=h^{n\cdot d+r}\tag{6}$$

然后计算 $E=e(C,D^{\mathrm{T}})=e(g,h)^{k\cdot r\cdot s\cdot z}$，则输出为$\frac{\widetilde{C}}{e(\hat{C},\hat{D})}=\frac{K_e\cdot e(g,h)^{k\cdot s\cdot y\cdot z}}{e(g^{s\cdot z},h^{(yr)\cdot k})\cdot e(g,h)^{k\cdot r\cdot s\cdot z}}=K_e$，从而获得对称密钥 K_e，再进行解密操作。

另一方面，若 C 是一个重加密且形式完整的密文，$C=\langle ATree,\widetilde{C},\hat{C},\breve{C},(C_i)_{i\in N}\rangle$，则该算法使用 $userKey$ 解密 C'，并获得 $\zeta=g^d$，最后输出$\frac{\widetilde{C}\cdot e(\zeta,\breve{C})}{\overline{C}}=\frac{K_e\cdot e(g,h)^{k\cdot s\cdot y\cdot z}}{e(g,h)^{(k\cdot s\cdot y\cdot z)+(n\cdot d\cdot k\cdot s\cdot z)}}=K_e$，从而获得密钥 K_e，进而获得文件明文；如果 C 是多次重加密后的密文，则解密操作与上述过程相似。

4　方案的安全性证明与性能分析

4.1　安全性证明

定理 1　如果 CTDH 假设在群 G 和 G_T 上都存在，那么该方案的主密钥是安全的。

证明：假设模拟器 S 接收一元组$\langle g,n,g_b=g^b,g_c=g^c,g_d=g^d,g_1=g^{\frac{c}{b}},g_2=g^{bc},g_3=g^{ac},g_4=g^{abc-Rc},g_5=g^{c(R+nd)},g_6=g^{\frac{c(R+nd)}{b}}\rangle$和一个挑战索引集合 I^*，为了输出 g^{abc}，该模拟器 S 将进行以下计算。

1）初始化：模拟器 S 从 Z_p 中随机选取 $\alpha_i,\beta_i,\gamma_i$，其中 $i\in N$；同时使得 $Y=e(g,h)^{ab}=e(ga,hb)$，并生成公共密钥。公共密钥的计算过程如下：如果 $i\in I^*$，则 $T_i=g^{\alpha_i}$，$T_{n+i}=g_b^{\beta_i}$，$T_{2n+i}=g^{\gamma_i}$；如果 $i\notin I^*$，则 $T_i=g_b^{\alpha_i}.T_{n+i}=g^{\beta_i}$，$T_{2n+i}=g^{\gamma_i}$。

2）密钥生成预言模型：敌手 A 利用索引 $I_q\subseteq N$ 向密钥生成预言模型发起查询，$I_q\neq I^*$，索引 j 必须满足$(j\in I_q)\wedge(j\notin I^*)$或$(j\notin I_q)\wedge(j\in I^*)$。

首先分析$(j\notin I_q)\wedge(j\in I^*)$的情况。对于每一个 $i\in N$，S 从 Z_p 中随机选取 r_i'，令 $r_i=br_i'i\neq j$，$r_j=ab+br_j'$，$r=\sum\limits_{i=1}^{n}r_i=ab+\sum\limits_{i=1}^{n}r_i'\cdot b$，$\hat{D}=(h^{y-r})^k=hk(-\sum\limits_{i=1}^{n}r_i'\cdot b)$。

①对于 $i\in I_q, i\neq j$：如果 $i\in I^*$，则 $D_{i,1}=D_{i,2}=h^{r_i}=g_2^{r_i'}$；如果 $i\notin I^*$，则 $D_{i,1}=D_{i,2}=h^{r_i}=g_2^{r_i'}$；

②对于 $i\notin I_q, i=j$：如果 $i\in I^*$，则 $D_{i,1}=D_{i,2}=h^{r_i}=g_2^{r_i'}$；如果 $i\notin I^*$，则 $D_{i,1}=D_{i,2}=h^{r_i}=g_2^{r_i'}$；

③对于 $i=j$：$D_{i,1}=D_{i,2}=h^{r_i}=h^{ab+br_i'}$。

最后输出密钥 $userKey=\langle A^u,(D_{i,1})_{i\in N},\hat{D},k,z\rangle$。

3)重加密密钥生成模型：敌手 A 利用索引 $I_q\subseteq N$ 和访问控制树 $Atree$ 向密钥生成预言模型发起查询。如果 $I_q\neq I^*$，该敌手 A 从 $KeyGen(I_q)$ 获得用户密钥 $userKey$，同时生成重加密密钥 $rk=ReKeyGen(userKey,ATree')$。如果 $I_q=I^*$，对于每个 $i\in N$，随机选取 $j\in I^*$ 和 r_i'，$r\in Z_p$，并使 $r_j'=r+R+nd-\sum_{i=1,i\neq j}^{n}r_i'$，$r_i=r_i'-d$。接着计算 $\hat{D},\hat{D}=h^{k(y-\sum_{i=1}^{n}r_i)}=h^{k(y-\sum_{i=1}^{n}r_i'-d)}=h^{k(ab-R-r)}=(g_4h^{-r})^k$。对于 $i\in N$：如果 $i\neq j$，$i\in I^*$，$D_{i,1}'=h^{r_i+d}=h^{r_i'}$；如果 $i=j, i\notin I^*$，$D_{i,1}'=h^{r_i+d}=h^{r_i'}$；如果 $i=j, i\in I^*$，$D_{i,1}'=h^{r_i+d}=h^{r+R+nd-\sum_{i=1,i\neq j}^{n}r_i'}=g_5h^{r-\sum_{i=1,i\neq j}^{n}r_i'}$。

最后输出重加密密钥 $rk=\langle A^u.ATree',(D_{i,1}')_{i\in N},\hat{D},k\cdot z,C'\rangle$。

4)重加密预言模型：敌手 A 利用索引 $I_q\subseteq N$ 和访问控制树 $Atree$ 向密钥生成预言模型发起查询。如果 $I_q\neq I^*$，该敌手 A 从 $ReKeyGen(userKey,ATree')$ 中获取用户密钥 rk，同时生成密文 $C'=ReEnc(rk,C)$，C 是 $g_d\cdot g^t$ 在访问控制树 $Atree$ 下的密文。

如果 $I_q=I^*$，对于 $i\in N$：若 $+d_i$ 出现在 $Atree$ 中，则 $(D_{i,1}')^{T_i}=h^{\frac{r_i+d}{kza_i}}=h^{\frac{r_i'}{kza_i}}$，$T_i=\frac{t_i}{k\cdot z}$；若 $-d_i$ 出现在 $Atree$ 中，则 $(D_{i,1}')^{T_i}=h^{\frac{r_i+d}{kz\beta_i}}=h^{\frac{r_i'}{kz\beta_i}}$，$T_i=\frac{t_{n+i}}{k\cdot z}$；否则，$(D_{i,1}')^{T_i}=h^{\frac{r_i+d}{kz\gamma_i}}=h^{\frac{r_i'}{kz\gamma_i}}$，$T_i=\frac{t_{2n+i}}{k\cdot z}$。最后，该算法针对 I^* 输出一个用户密钥 $userKey^*$。如果该密钥是一个有效密钥，则 $userKey^*$ 将满足：$e(g^z,\hat{D})\prod_{i\in I^*}e(T_i,D_{i,1}^T)\cdot e(T_{n+1},D_{i,1}^T)=e(g,h)^y$。

5)解密预言模型：加密密钥和重加密密钥直接通过密钥生成预言模型和重加密密钥生成模型正确生成，因此该模型将会输出：$(\hat{D})^z\prod_{i\in I^*}D_{i,1}^{T\cdot\alpha_i}\prod_{i\in I^*}D_{i,1}^{T\cdot\beta_i}=g^{abc}$，从而解决了 CDTH 难题。

4.2 性能分析

大多数基于属性的代理重加密方案在生成密文的过程中都需要进行大量双线性对运算，而双线性对运算相对于其他运算会耗费更多的计算时间。本文方案通过模糊提取器与倒排序结构构建新的安全索引用户，实现了可容错的多关键字检索，并提高了检索的效率。与文献[14-16]提出的方案相比，在加密、搜索、重加密和解密 4 个阶段中，所提方案在双线性对运算的时间开销和指数运算的时间开销方面有一定的优势。其对比结果如表 1 所列。

表 1 方案的效率对比

Table 1 Efficiency comparison of schemes

操作 / 方案	本文方案	文献[14]中的方案	文献[15]中的方案	文献[16]中的方案
加密	T_e	$2T_p+5T_e$	$2T_p+2T_e$	$8T_e$
搜索	$O(n)$	$O(n)$	$O(\log_2 n)$	$O(\log_2 n)$
重加密	$2T_p$	$4T_p+T_e$	T_p+T_e	T_p+2T_e
解密	$2T_p$	$5T_p+T_e$	T_p+T_e	$5T_e$

结束语 文章结合基于属性代理重加密技术与模糊提取器工具，提出一种以电子医疗环境为应用背景的数据检索方案。该方案首先通过访问控制树来设置用户的权限，并通过对密钥的重加密来管理用户的权限；然后利用模糊提取器构造容错机制，并以此容错机制预处理多关键检索索引，提高检索效率。与同类方案进行的对比分析表明，所提方案的检索效率更高。

后续将对检索的返回结果设置一种有效且高效的验证机制，增强方案的实用性。

参 考 文 献

[1] HOHENBERGER S, WATERS B. Attribute-based encryption with fast decryption[M]. Berlin: Springer, 2013: 162-179.

[2] ATTRAPADUNG N, HERRANZ J, LAGUILLAUMIE F, et al. Attribute-based encryption schemes with constant-size ciphertexts[J]. Theoretical Computer Science, 2012, 422(3): 15-38.

[3] KHADER D. Introduction to attribute based searchable encryption[M]// Communications and Multimedia Security. Springer Berlin Heidelberg, 2014: 131-135.

[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 and Distributed Systems, 2013, 24(1): 131-143.

[5] GOLLE P, STADDON J, WATERS B. Secure conjunctive keyword search over encrypted data[C]// Proc of the applied Cryptography and Network Security. Springer Berlin Heidelberg, 2004: 31-45.

[6] CAO N, WANG C, LI M, et al. Privacy-preserving multi-keyword ranked search over encrypted cloud data[J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(1): 222-233.

[7] SUN W, WANG B, CAO N, et al. Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking[C]// Proc of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security. ACM, 2013: 71-82.

[8] CURTMOLA R, GARAY J, KAMARA S, et al. Searchable symmetric encryption: Improved definitions and efficient constructions[J]. Journal of Computer Security, 2011, 19(5): 895-934.

[9] GOLLE P, STADDON J, WATERS B. Secure Conjunctive Keyword Search over Encrypted Data[M]. Berlin: Springer, 2004: 31-45.

(下转第 196 页)

[5] BEZDEK J C. Fuzzy Mathematics in Pattern Classification[D]. Ithaca:Cornell University,1973.

[6] BEZDEK J C. Pattern Recognition with Fuzzy Objective function Algorithms [M]. New York:Plenum Press,1981.

[7] JIANG Z H,LI T T,MIN W F,et al. Fuzzy c-means clustering based on weights and gene expression programming [J]. Pattern Recognition Letters,2017,90:1-7.

[8] CHEN H P,SHEN X J,LV Y D,et al. A novel automatic fuzzy clustering algorithm based on softpartition and membership information [J]. Neurocomputing,2017,236(SI):104-112.

[9] KRISHNAPURAM R,KELLER J M. A Possibilistic Approach to Clustering [J]. IEEE Transactions on Fuzzy Systems,1993,1(2):98-110.

[10] KRISHNAPURAM R,KELLER J M. The Possibilistic C-Means Algorithm: Insights and Recommendations [J]. IEEE Transactions on Fuzzy Systems,1996,4(3):385-393.

[11] BARNIM,CAPPELLINI V,MECOCCI A. Comments on A Possibilistic Approach to Clustering [J]. IEEE Transactions on Fuzzy Systems,1996,4(3):393-396.

[12] LINGRAS P, WEST C. Interval Set Clustering of Web Users with Rough K-Means [J]. Journal of Intelligent Information Systems,2004,23(1):5-16.

[13] PAWLAK Z. Rough Sets[J]. International Journal of Computer & Information Sciences,1982,11(5):341-356.

[14] MAJI P,PAL S K. RFCM:A Hybrid Clustering Algorithm Using Rough and Fuzzy Sets[J]. Fundamenta Informaticae,2007,80(4):475-496.

[15] MITRA S,BANKA H,PEDRYCZ W. Rough-Fuzzy Collaborative Clustering [J]. IEEE Transactions on Systems Manand Cybernetics Part B Cybernetics,2006,36(4):795-805.

[16] PAUL S,MAJI P. A New Rough-Fuzzy Clustering Algorithm and its Applications [C]// Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012). New Delhi:Springer,2014:1245-1251.

[17] SHI J,LEI Y,ZHOU Y,et al. Enhanced rough-fuzzy c-means algorithm with strict rough setsproperties [J]. Applied Soft Computing,2016,46:827-850.

[18] ZHANG T F,CHEN L,MA F M. A modified rough c-means-clustering algorithm based on hybrid imbalanced measureof distance and density[J]. International Journal of Approximate Reasoning,2014,55(8):1805-1818.

[19] WANG H,ZHOU M. A refined rough k-means clustering with hybrid threshold [C]// International Conference on Rough Sets and Current Trends in Computing. Berlin Heidelberg:Springer,2012:26-35.

[20] MALYSZKO D,STEPANIUK J. Rough Entropy Based k-Means Clustering [C]// Rough Sets, Fuzzy Sets, Data Mining and Granular Computing. Berlin Heidelberg: Springer, 2009: 406-413.

[21] PAL S K,SHANKAR B U,MITRA P. Granular computing, rough entropy and object extraction [J]. Pattern Recognition Letters,2005,26(16):2509-2517.

[22] PETERS G. Rough Clustering Utilizing the Principle of Indifference [J]. Information Sciences,2014,277(2):358-374.

[23] WANG X E,HAN D Q,HAN C Z. Selection method for Parameters of Rough Fuzzy C-Means Clustering Based on Uncertainty Measurement[J]. Journal of Xi'an Jiaotong University,2013,47(6):55-60. (in Chinese)
王学恩,韩德强,韩崇昭. 采用不确定性度量的粗糙模糊 C 均值聚类参数获取方法[J]. 西安交通大学学报,2013,47(6):55-60.

[24] PETERS G. Some refinements of rough k-means clustering [J]. Pattern Recognition,2006,39(8):1481-1491.

[25] BEZDEK J C,PAL N R. Some New Indexes of Cluster Validity [J]. IEEE Transactions on System Manand Cybernetics Part B Cybernetics,1988,28(3):301-315.

(上接第 166 页)

[10] KAMARA S,PAPAMANTHOU C,ROEDER T. Dynamic searchable symmetric encryption[C]// Acm Conference on Computer & Communications Security. ACM,2012:965-976.

[11] KAMARA S,PAPAMANTHOU C. Parallel and Dynamic Searchable Symmetric Encryption[M]// Berlin:Springer,2013:258-274.

[12] CASH D,JARECKI S,JUTLA C,et al. Highly-Scalable Searchable Symmetric Encryption with Support for Boolean Queries [M]. Berlin:Springer,2013:353-373.

[13] AU M H,TSANG P P,SUSILO W,et al. Dynamic universal accumulators for DDH groups and their application to attribute-based anonymous credential systems [M]. Berlin: Springer, 2009:295-308.

[14] SHAO J,CAO Z,LIANG X,et al. Proxy re-encryption with keyword search[J]. Information Sciences,2010,180(13):2576-2587.

[15] LEE S H,LEE I Y. A Study of Practical Proxy Reencryption with a Keyword Search Scheme considering Cloud Storage Structure[J]. The Scientific World Journal, 2014, 2014(2): 1661-1667.

[16] FANG L,SUSILO W,GE C,et al. Chosen-ciphertext secure anonymous conditional proxy re-encryption with keyword search [J]. Theoretical Computer Science,2012,4629(1):39-58.

[17] DODIS Y,OSTROVSKY R,REYZIN L,et al. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data[J]. SIAM Journal on Computing,2008,38(1):97-139.