一种改进的动态多用户前向安全可搜索加密方案

王泽贤 汪学明*
(贵州大学计算机科学与技术学院 贵州 贵阳 550025)

摘 要 最近 Wang 等[14]提出了一种在多用户环境下满足前向安全的动态可搜索加密方案。然而该方案利用双线性对实现多用户的访问控制搜索效率低下,在此基础上通过构造两个静态哈希表相互映射构建矩阵索引,并用伪随机函数和 Hash 函数代替双线性对生成密钥提高方案的搜索效率。通过形式化安全证明,该方案满足前向安全。

关键词 可搜索加密 云计算 前向安全 多用户

中图分类号 TP309 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2024.03.046

AN IMPROVED DYNAMIC MULTI-USER FORWARD SECURE SEARCHABLE ENCRYPTION SCHEME

Wang Zexian Wang Xueming*
(*College of Computer Science and Technology*, *Guizhou University*, *Guiyang 550025*, *Guizhou*, *China*)

Abstract Recently, reference[14] proposed a dynamic searchable encryption scheme with forward security in multi-user environment. However, the scheme uses bilinear pairing to achieve multi-user access control, and the search efficiency is low. On this basis, this paper constructed two static Hash tables to map each other to construct matrix index, and used pseudo-random function and Hash function to replace bilinear pair to generate key to improve the search efficiency of the scheme. Through formal security proof, it is proved that the scheme satisfies forward security.

Keywords Searchable encryption Cloud computing Forward security Multi-user

0 引 言

云计算服务使个人和组织能够将他们的数据外包到公共云服务器上,这样云客户就可以减轻个人数据存储和管理的负担。然而,在享受云计算提供的稳定和方便的服务同时。也存在着外包一些有价值的数据时产生数据泄露等安全问题。为了保护数据的安全性,一般的方法是在外包之前对数据进行加密,但这种方法限制了对密文数据进行直接操作(搜索、更新)。

为了解决上述问题提出可搜索加密技术(Searchable Encryption,SE)。在 2000 年 Song 等[1]首次提出基于对称密码学的对称可搜索加密方案。但该方案采用关键词线性扫描匹配的方式搜索效率低下。为提高可搜索加密方案的效率国内外学者展开研究并取得丰硕成果[2-3]。随着互联网的发展数据共享和动态更新成为当前用户的重点需求。支持动态更新的可搜索加密方案成为了研究的热点。然而,最近的研究表明目前的动态 SE 方案仍存在许多不足[4-5]。Zhang 等[6]等提出了在密文文档动态更新时如果通过向云服务器注入几个文件可以实现查询数据的恢复,从而泄露用户的信息。为了解决这个问题国内外学者提出了前向安全 SE 方案。它能够抵抗文件注入式的攻击[7-10]。这些 SE 方案通过引入计数器实现对陷门的有效更新从而避免在更新时的数据泄露。

目前大部分的动态对称可搜索加密方案只能实现单用户的搜索。出于数据共享的需要 Wang 等[14]提出了一种多用户场景下满足前向安全的动态对称可搜索加密方案,该方案通过引入半诚实可信的中间服务器存储关键词的状态信息,每当用户搜索关键词时,需把

收稿日期:2020-11-17。国家自然科学基金项目(61163049);贵州省自然科学基金资助项目(黔科合 J 字[2014]7641)。
王泽贤,硕士生,主研领域:计算机网络与信息安全。**汪学明**,教授。

搜索请求上传到中间服务器进行加工生成搜索陷门从而解决了多用户环境下的前向安全问题。然而该方案中,未对用户的合法身份进行验证,存在恶意用户和服务器串通的安全威胁。随后卢冰洁等[9]虽然采用一种新的索引结构对 Wang 等[14]做出改进,但是只提高文档更新时删除的效率,在加密、搜索、添加更新方面效率仍然低下。

针对以上问题本文提出一种改进的动态多用户对称可搜索加密方案(IDMSE)。首先本文采用伪随机函数和哈希函数代替双线性对为每个用户都生成各自的密钥。然后构建文档/关键词两个静态哈希表。并通过两个哈希表相互交叉映射构建方案的索引。接着引入可信的第三方服务器完成授权用户的身份验证,并根据关键词状态信息为合法用户生成最终的陷门保证前向安全性。

1 系统模型和安全性定义

1.1 系统模型

本文系统模型由四个部分组成,如图 1 所示。

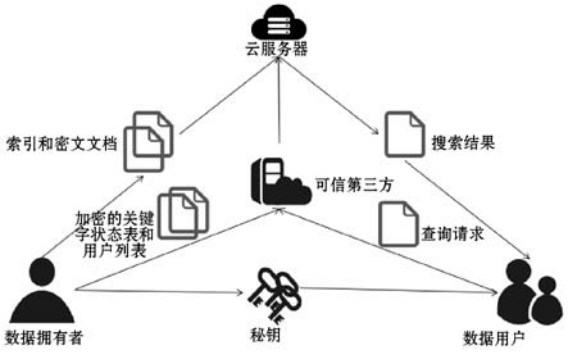

图 1 系统模型

数据拥有者负责构建关键词字典、索引、关键词状态表、数据用户列表和文档加密密钥;并把索引和明文文档加密后上传到云服务器;把用户列表和关键词状态表加密后上传到可信第三方服务器。

云服务器是诚实好奇的,正确执行操作的同时可能记录相关信息;主要负责存储密文文档和相应索引。当收到搜索陷门后,匹配并返回相关搜索结果。

可信第三方服务器是诚实可靠的,负责验证数据用户的身份,并根据关键词状态信息为合法用户生成最终搜索陷门。

数据用户是诚实的,可以根据个人密钥向云服务器查询文档。

1.2 安全性定义

IDMSE 方案的安全性主要有如下两个概率游戏验证,其中:A 表示敌手;S 表示模拟器。

$Real_A(K)$ 挑战者通过 $KeyGen(1^k)$ 产生系统参数 p 和密钥对 S_K,然后敌手 A 得到明文文档和索引并从挑战者处通过得到密文文档、索引 $Enc(S_K,F,I)\rightarrow(C,I_c)$。敌手 A 在多项式时间内采用自适应查询 $q\in\{w,f\}$(w 表示关键词,f 表示文档),如果 $q=w$,敌手从挑战者处得到搜索陷门:$Trapdoor(q_w)\rightarrow(T)$ 如果 $q=f_j$ 且操作为 o_p 时,敌手从挑战者处得到更新令牌 $UpdToken(S_K,o_p,f_j)\rightarrow T_f$ 敌手最终输出一个字符。

$Ideal_{A,S}(k)$ 给定泄露函数 $\ell_1(F,I)$,S 产生 (C^*,I_c^*) 给 A,敌手 A 在多项式时间内采用自适应查询 $q\in\{w,f\}$。对于每一次搜索模拟器 S 都产生相应的泄露函数 $\ell_2(F,I,w,t)$,如果 $q^*=w$,S 产生 T^*,如果 $q=f_j$ 且操作为 o_p 时,模拟器 S 产生新的更新令牌 T_f^*,敌手最终输出一个字符。

定义 1 我们说 IDMSE 方案对于任意多项式时间内的敌手 A 都能够抵抗动态选择关键词猜测攻击,那么一定存在模拟器使得如下等式成立:

$$|\mathrm{pr}[Real_A(k)=1]-\mathrm{pr}[Ideal_{A,s}(k)=1]|\leqslant \mathrm{neg}(k)$$

定义 2 泄露函数(ℓ_1,ℓ_2,ℓ_3)。

(1) $\ell_1(F,I)$ 以文档和索引作为输入,ℓ_1 输出关键词的个数 n、文档数 m、每个文档的标识以及文档 f_j 包含关键词的个数。具体形式如下:$\ell_1(F,I)=(n,m,id_1,id_2,\cdots,id_m,|f_1|,|f_2|,\cdots,|f_m|)$

(2) $\ell_2(F,I,w,t)$ 以文档、索引、在时间节点 t 的搜索关键词 w 作为输入,ℓ_2 输出搜索模式和访问模式 $s(I,q,t)$ 和 $a(F,I,w,t)$ 及用户授权列表 U,具体形式为:$\ell_2(F,I,w,t)=(s(I,q,t),a(F,I,w,t),U)$。

(3) $\ell_3(f,o_p)$ 以文档和动态更新操作为输入,ℓ_3 输出文档标识,文档 f_j 包含的关键词个数以及更新操作 o_p。具体形式为:$\ell_3(f,o_p)=(id_i,\cdots,id_j,|f_i|,\cdots,|f_j|,o_p)$。

前向安全:对于新添加的文档,旧的陷门无法泄露相关信息。如果一个动态可搜索加密方案泄露可以写成 ℓ_1、ℓ_2、ℓ_3,则该方案是前向安全的。

2 基础知识

2.1 哈希函数和伪随机函数

本文使用哈希函数形式如下:$G\rightarrow\{0,1\}^k$ 使用的伪随机函数形式如下:

$$A:\{0,1\}^k\times\{w_1,w_2,\cdots,w_n\}\rightarrow Z_q$$

$$B:\{0,1\}^k\times\{0,1\}^*\rightarrow\{0,1\}^k$$

其中 n 表示当前文档中不重复关键词 w 的个数。

2.2　索引结构

本文采用两个静态哈希链表交叉映射构建 $m \times n$ 阶矩阵的索引结构,m 表示文档数,n 表示关键词字典中关键词数。$I(i,j)=(w_i,f_j)$ 表示在文档 f_j 中是否存在关键词 w_i,存在值为 1 否则为 0。行由静态关键词链表 α_x 产生对于 $w_i, i\in\{1,2,\cdots,n\}$ 横坐标 x 的值为 $x=\alpha_x(\lambda x),\lambda x=H(h^{\sigma}),\sigma=A_{k1}(w_x)$,对于矩阵的 f_j, $j\in\{1,2,\cdots,m\}$ 纵坐标 y 用文档静态链表 α_y 生成 $y=\alpha_y(\lambda y),\lambda y=B_{k2}(f_y)$。本文索引结构如图 2 所示。

文档	关键词			
	$\alpha_x(\lambda w_9)$	$\alpha_x(\lambda w_{36})$	…	$\alpha_x(\lambda w_{22})$
$\alpha_y(\lambda f_{31})$	0	1	…	0
$\alpha_y(\lambda f_{17})$	1	0	…	1
…	…	…	…	…
$\alpha_y(\lambda f_{22})$	1	1	…	1

图 2　索引结构

2.3　关键字状态表

为记录动态更新时关键词的状态构建如图 3 所示的关键词状态表 D_w。$st(w_i)$ 表示关键词 w_i 当前的陷门。c_i 表示一个计数器。

w_1	w_2	…	w_n
$st(w_1)$	$st(w_2)$	…	$st(w_m)$
c_1	c_2	…	c_m

图 3　关键词状态表

前向安全性是动态可搜索加密方案的基本要求。因此,结构设计的第一步是满足前向安全性。通过构建关键词状态表存储每个关键字的当前状态,以便生成搜索令牌。当添加关键字/文档对时,对状态进行更新。本文根据哈希函数 H 的单向散列的特性构建搜索令牌 $st(w)=H(kt_w)$。具体形式如下:

$$H^c(kt_w)=\begin{cases}H(kt_w) & c=1\\ H(H^{c-1}(kt_w)) & c\geqslant 2\end{cases}$$

对于给定关键字 w 和计数器 $c(c\leftarrow c_{\text{len}})$,通过关键词状态表可以得出搜索令牌 $H^c(kt_w)$,通过如上公式迭代计算可以得出关键词 w 过去状态的所有搜索令牌。由于哈希函数单向散列的特点,无法得知之后的搜索令牌,从而保证了方案的前向安全。

3　本文方案

3.1　方案定义

(1) $KeyGen(1^{\lambda})\rightarrow(p,S_K)$ 通过输入安全参数 λ 产生系统参数 $p=(A,B,h,H)$ 和密钥 $S_K=(k_1,k_2,k_3,sk_1,a,d)$。

(2) $Setup(S_K,F,I,)\rightarrow(C,I_c,D_w)$ 数据拥有者构建明文索引结构和关键词状态表,然后使用密钥对 S_K 对明文 F 和索引 I 进行加密并上传到云服务器。同时用密钥对 S_K 把关键词状态表 D_w 加密后上传到可信第三方服务器。

(3) $Adduser(u_i,S_K)\rightarrow U$ 数据拥有者采用唯一标识 u_i 和密钥对 S_K 为用户生成各自的加解密密钥(a_{ui},d_{ui}),以及辅助密钥(a_{si},d_{si})。然后把(a_{si},d_{si},u_i)上传到云服务器和可信第三方服务器。并用密钥对 S_K 加密用户列表完成更新。

(4) $Trapdoor(q_w)\rightarrow(T)$ 用户根据查询关键词 w 和密钥 a_{ui} 产生请求 q_w 并上传到可信第三方服务器,可信第三方服务器结合查询关键词 w 最新的状态信息产生最终的搜索陷门 T 并上传到服务器进行搜索。

(5) $Search(I_c,T_q)\rightarrow(C_w)$ 云服务器在收到陷门后与索引进行匹配并返回搜索的密文文档。

(6) $Dec(C_j,s'_k,d_{si},d_{ui})\rightarrow(f_j)$ 用户使用自己的解密密钥 d_{ui} 和辅助密钥 d_{si} 对返回的密文文档进行解密。

(7) $Update(S_K,o_p,f_j)\rightarrow T_f$ 数据拥有者对更新文档 f_j 和更新操作 o_p 产生添加文档的密文和更新索引,然后上传到云服务器对密文和索引进行更新。最后把新的关键词状态表加密后上传到可信第三方服务器。

3.2　符号定义

F:表示明文文件集合 $F=\{f_1,f_2,\cdots,f_m,\}$。

C:表示对应明文 F 的密文集合 $C=\{c_1,c_2,\cdots,c_n\}$。

W:表示关键词字典 $W=\{w_1,w_2,\cdots,w_n\}$。

q:查询请求。

I_c:加密的索引向量。

T_w:搜索陷门。

3.3　IDMSE 构造

$KeyGen(1^{\lambda})\rightarrow(p,S_K)$ 通过输入安全参数产生系统公共参数 $p=(G,g,q,h=g^a,H,A,B)$ 和密钥对 $S_K=(a,d,k_1,k_2,k_3,s_{k_1})$ 其中:$a\in Z_q^*$,$d\in\mathbf{Z}_q^*$,$s_{k_1}=g^u$,$g^u\leftarrow G$ 并为可信第三服务器初始化用于存储状态信息的空表 D_w。

$Setup(S_K,F,I)\rightarrow(C,I_c,D_w)$ 首先,初始化 $n\times m$ 阶矩阵并置矩阵元素值为 0。从所有明文文档 $F(f_1,f_2,\cdots,f_m)$ 中提取关键字字典 $W(w_1,w_2,\cdots,w_n)$。然后采用 2.2 节方法构建索引矩阵并赋值,具体形式如下:

(1) $a=A_{k1}(w_x),\lambda x=H(h^a),x=\alpha(\lambda x),\lambda y=$

$B_{k2}(f_y)$,$y=\alpha(\lambda y)$。

(2) 只要关键词 w_x 在文档 f_y 中出现,就把矩阵 $\boldsymbol{I}[x,y]$ 赋值为 1,否则为 0。对于赋值为 1 的关键字/文档二元组添加状态值为 $kt_w \leftarrow F(k_3,w_x)\ s_{t_x} \leftarrow H^{\text{clen}}(kt_{w_x})\{H(s_{t_x} \| c_{\text{len}}) \oplus (w_x) \| (f_y)\}$,赋值为 0 的状态值为 $H(s_t \| c_{\text{len}})$。接着对所有明文文档$(f_1 f_2 \cdots f_m)$用密钥 s_{k_1}加密上传到云服务器。然后以列形式对构建的索引进行加密 $I_c[i,*]=Enc(h^{\lambda x},I[i,*])x \in (1,2,\cdots,n)$。最后把加密索引 $\gamma=(I_c,\alpha_x)$和密文文档一起上传到服务器中,把关键字状态表 D_w 加密后上传到可信第三方服务器。数据拥有者保留(α_x,α_y)和关键字状态表 D_w。

$Adduser(u_i,S_K)\rightarrow(U)$当添加新用户时,首先数据拥有者为用户 u 产生唯一的用户标识 u_i,然后为 $a_{ui} \leftarrow Z_q$每个用户产生各自的加密和解密密钥,$d_{ui} \leftarrow Z_q$然后通过对 a_{ui}和 d_{ui}计算产生相应的辅助加解密密钥 $a_{si}=a-a_{ui}$。$d_{si}=d/d_{ui}$最后把用户加解密密钥以三元组的形式(u_i,a_{ui},d_{ui})发送给用户,辅助查询密钥(u_i,a_{si})和辅助解密密钥(u_i,d_{si})分别上传到可信第三方服务器和云服务器。为便于解密,数据拥有者对密钥 S_K 重加密 $s'_{k_1}=s_k^d=g^{\mu d}$并上传服务器。然后可信第三方服务器和云服务器分别对密钥列表进行更新。

$Trapdoor(q_w)\rightarrow(T)$用户 u_i 查询关键词 w 需使用自己的查询加密密钥 a_{ui}以及随机参数 $r \leftarrow Z_q$产生搜索请求 $q_{ui(w)}=(g^{-r}g^{\sigma},h^r g^{-a_{ui}r}g^{a_{ui}\sigma})$上传到可信第三方服务器。可信第三方服务器首先通过计算 $u_i^{a_{ui}\cdot a_{si}}$判断 u_i 是否为授权访问用户。如果是则结合 w 最新的状态信息,形成最终的搜索陷门并上传到云服务器 $T=(g^{-r}g^{\sigma},h^r g^{-a_{ui}r}g^{a_{ui}\sigma},H(s_{t_*} \| c))$。

$Search(EDB,T(w))\rightarrow(f_{id})$云服务器计算 $r_w=g^{-r}g^{\sigma a_{si}} \cdot h^r g^{-a_{ui}r}g^{a_{ui}\sigma}=h^{\sigma}$,$\lambda x=H(r_w)$,$x=\alpha_x(\lambda x)$,$H(s_{t_*} \| c_*)=H(s_{t_x} \| c_x)$,然后计算对于 $I'[i,j]=Dec(r_x,I[i,j])$如果 $I'[i,j]=1$,则为相关密文文档 f_j,返回当前状态 s_{t_x}下所有文档。然后计算判断计数器 c 是否为 c_{len}。若计算器 c 的值不为 c_{len}则计算上一状态值 $H(s_{t_{x-1}} \| c-1)=H(H(s_{t_x} \| c))$,继续执行上述搜索。直至计数器 c 值为 c_{len}。则搜索结束,返回所有相关文档。

$Dec(C_j,s'_k,d_{si},d_{ui})\rightarrow(f_j)$云服务器根据返回用户标识 u_i 得到辅助解密密钥 d_{si}对每个 u_i 计算解密密钥 $s''_k=(s'_k)^{d_{si}^{-1}}=g^{\mu d_{ui}}$后连同密文文档一起返回给用户,用户 u_i 使用自己的解密密钥 $s_k=(s'_k)^{d_{ui}^{-1}}=g^{\mu}$计算最终的文档解密密钥 s_k。最后用户使用文档解密密钥 $f=AES.Dec(s_k,C_j)$得到目标查询文档。

$Update(S_K,o_p,f_j)\rightarrow T_f$动态更新时数据拥有者首先初始化两个大小为 n 的空向量 $\boldsymbol{D}$、$\overline{\boldsymbol{D}}$,集合元素全赋为 0。然后计算 $y=\alpha_y(\lambda y)$,$\lambda y=B_{k2}(f_y)$。如果更新操作 o_p 为添加新文档,则首先提取添加文档 f_j 中存在的关键词集合$(w_1 w_2 \cdots w_j)$然后对于关键词集合$(w_1 w_2 \cdots w_j)$构建更新令牌。如存在关键词 w_x,计算其所在位置,并赋值为 1,$a=A_{k1}(w_x)$,$\lambda x=H(h^{\lambda x})$,$x=\alpha(\lambda_x)$ $D[x]=1$。然后对关键词状态进行更新 $c=c-1$,$s_{t_{x+1}}=H^{c-1}(s_{t_x})$。然后分别对文档和索引加密后上传到云服务器。$c_j \leftarrow Enc(sk,f_j)$,$a=A_{k1}(w_x)$,$\lambda_x=H(h^{\lambda x})$,$x=\alpha(\lambda_x)\overline{\boldsymbol{D}}[x] \leftarrow Enc(h^{\lambda x},\boldsymbol{D}[x])$云服务器收到更新后对矩阵和文档进行扩充 $I[j,*]=\overline{\boldsymbol{D}}^{\mathrm{T}}$得到新的密文集合 C'和索引 γ'。最后把关键词状态表更新后上传到可信第三方服务器。

4 安全性分析

本文方案从文档、索引、搜索陷门、更新令牌四个方面进行安全性分析。根据定义 2 在随机预言机模型下证明方案是安全的。证明过程如下:在与挑战者交互的过程中敌手得到 $R_1=(C,\gamma,f_1,f_2,\cdots,f_m,Q_t,U,\tau_t)$,模拟器根据泄露信息模拟得到 $R_2=(C^*,\gamma^*,f_1,f_2,\cdots,f_m,Q_t^*,U^*,\tau_t^*)$

文档、索引安全:对于给定的泄露函数 $\ell_1(F,I)=(n,m,id_1,id_2,\cdots,id_m,|f_1|,|f_2|,\cdots,|f_m|)$中,模拟器首先通过对称加密算法得到密文 C^*,然后模拟器先构造哈希表再构造索引矩阵并随机为矩阵元素赋值 0 或 1,从而得到密文索引 $\gamma^*=(I_c^*,\alpha_w)$,具体过程如下:首先随机选择密钥 λ_x^*,然后模拟器构建索引矩阵 $\boldsymbol{I}^*$,然后以列的形式对索引进行加密得到密文索引 $I_c^*[\alpha(H(h^{\lambda x})),y]=Enc(h^{\lambda x},I_c^*[\alpha(H(h^{\lambda x})),y])$。接着模拟器根据用户数$|U|$构造用户列表 U^*。最后为构造的用户随机分配用户标识 u_i,再为每个用户生成密钥对 $a_{si}^* \leftarrow Z_q^*$,$d_{si}^* \leftarrow Z_q^*$。文档的安全性是由对称加密保证的,所以在计算上无法区分 C、C^*。对称加密和伪随机函数在计算上保住了 γ、γ^* 的不可区分。在授权用户的构造中,互补密钥是随机产生分发给用户的,因此在计算上也 U、U^* 是不可区分。

陷门安全:对于泄露函数 $\ell_2(F,I,w,t)=(s(I,q,t),a(F,I,w,t),U)$ $R_2=(C^*,\gamma^*,f_1,f_2,\cdots,f_m,Q_t^*,U^*,\tau_t^*)$为模拟器构造的结果。针对不同用户对相同关键词的查询,模拟器作出如下构造:首先对用户列表的构造证明过程如泄露函数 ℓ_1所示,由于辅助密钥是

随机分发给用户的所以在计算上是不可区分的。接着对于 C^*、γ^* 的证明也同上。对于搜索陷门由用户搜索请求和关键词状态信息共同构成 $T_q=(q_{ui},S_w,s_{t_w})$。模拟器通过随机选择用户 u_i 以及他的加密密钥 a_{ui}^* 生成用户搜索请求 $q_{ui}q_{ui}$,$a_{ui}^*=a^*-a_{si}^*$,$r^*\leftarrow Z_q^*$,再结合关键词状态信息 D_w 得到最终的搜索陷门 $T_q^*=(q_{ui*},S_{w*},s_{t_{w*}})$。由于随机选择密钥 λ_x^* 和伪随机函数 $\lambda_x=A_{k1}(w)$ 在计算上不可区分,所以 T_q、T_q^* 在计算上不可区分。因此不同用户在搜索相同关键词时产生的陷门也不可区分。

前向安全:对于泄露函数 $\ell_3(f,o_p)=(id_i,\cdots,id_j\ |f_i|,\cdots,|f_j|,o_p)$ 针对 o_p 操作为添加文档时,模拟器 S 首先初始化两个空集 D^*、$\overline{D}^*$,根据敌手 A 提供的文件更新信息得到关键词集合的更新信息 $(w_1,w_2,\cdots,w_j)^*$。然后随机选择 λ_x^* 密钥构建更新令牌具体操作如下:

$a=A_{k1}(w_x)$,$\lambda_x=H(h^{\lambda x})$,$x=\alpha(\lambda_x)D[x]=1$,$c=c-1$,$s_{t_{x+1}}=H^{c-1}(s_{t_x})$ 得到关键词最新状态信息表 $s_{t_{w*}}$,模拟更新信息和真实更新信息在大小和形式上是一致的,由于 $\lambda_x=A_{k1}(w)$ 在未知 k_3 的情况下无法产生新陷门,敌手 A 无法区分。所以仅知道泄露函数 $\ell_3(f,o_p)$ 就可以模拟更新操作,但模拟出的陷门无法对更新后的文档进行搜索,因此证明了本文方案满足前向安全。

因此,R、R^* 在计算上是不可区分的,所以在多项式时间内敌手 A 无法区分 $Real_A(k)$、$Ideal_{A,s}(k)$,如下等式成立:

$$\left|\mathrm{pr}[Real_A(k)=1]-\mathrm{pr}[Ideal_{A,s}(k)=1]\right|\leqslant \mathrm{neg}(k)$$

5　性能分析

本文从加密、搜索、更新三个方面与 Wang 等[14]、卢冰洁等[9]和 Ye 等[15]提出的方案进行对比分析。其中 Ye 等[15]提出的方案不支持动态更新,故更新效率为 0。具体的运算效率如表 2 所示,其中:n 表示关键词集合的个数;m 表示文档数;E 表示一次对称加密操作;e 表示一次指数运算;g 表示一次双线性运算。

表 2　效率对比

方案	加密	搜索	文档添加更新
文献[14]	$2g+3E+2e$	$2ng+2e$	nE
文献[7]	$2g+3E+2e$	$2ng+2e$	nE
文献[15]	$2g+5e$	$6ng+4e$	0
本文	$2E+e$	$nE+e$	nE

由表 1 可得本文方案在加密、搜索两个方面的效率与 Wang 等[14]、卢冰洁等[9]和 Ye 等[15]的方案相比均有提升。

6　结　语

本文提出一种动态更新的多用户可搜索加密方案。采用哈希函数和伪随机函数来代替双线性对实现密钥分发完成多用户的访问控制,通过两个静态哈希表相互映射构造矩阵索引结构,并引入可信第三方服务器保存关键词状态信息保证了方案的前向安全。由于本文索引结构为正向索引结构,搜索效率无法达到倒排索引结构的次线性搜索效率。接下来将继续研究在未来提出更加高效的可搜索加密方案。

参　考　文　献

[1] Song D X, Wagner D, Perrig A. Practical techniques for searches on encrypted data[C]//2000 IEEE Symposium on Security and Privacy,2000: 44 - 55.

[2] Bosch C, Hartel P H, Jonker W, et al. A survey of provably secure searchable encryption[J]. ACM Computing Surveys (CSUR),2014,47(2):1 - 51.

[3] Stefanov E, Papamanthou C, Shi E. Practical dynamic searchable encryption with small leakage[C]//Network and Distributed System Security Symposium,2014.

[4] Ghorbel A, Ghorbel M, Jmaiel M. Privacy in cloud computing environments: A survey and research challenges[J]. The Journal of Supercomputing,2017,73(6):2763 - 2800.

[5] Sun P. Security and privacy protection in cloud computing: Discussions and challenges[J]. Journal of Network and Computer Applications,2020,160:102642.

[6] Zhang Y, Katz J, Papamanthou C. All your queries are belong to us: the power of fifile-injection attacks on searchable encryption[C]//25th USENIX Conference on Security Symposium,2016:707 - 720.

[7] Bost R. $\Sigma o\phi o\varsigma$: Forward secure searchable encryption[C]//2016 ACM SIGSAC Conference on Computer and Communications Security,2016: 1143 - 1154.

[8] Bost R, Minaud B, Ohrimenko O. Forward and backward private searchable encryption from constrained cryptographic primitives[C]//2017 ACM SIGSAC Conference on Computer and Communications Security,2017:1465 - 1482.

[9] 卢冰洁,周俊,曹珍富. 一种增强的多用户前向安全动态对称可搜索加密方案[J]. 计算机研究与发展,2020,57(10):2104 - 2116.

(下转第 349 页)

(4) 执法监察运维子系统。该子系统监控整个系统的运行状态,完成注册用户管理、权限管理、设备绑定与注销、实时在线设备、实时人员动态、系统使用情况分析等监控功能;也能完成系统字典维护、信息管理维护、运行日志分析和崩溃日志分析等功能。

(5) 执法监察共享服务子系统。该子系统能为其他市级平台和其他需要数据调用的平台提供数据共享服务。基于 OAuth2.0 协议规范,通过给所有合法的调用端分配 appid 和 secret 来区分不同的用户并确定其调用权限,返回授权码(authorization code)。然后通过授权码从共享服务子系统交换临时分配的 access token 来验证调用端的身份并作出相应的响应。给授权码和 access token 设置过期时间确保信息不会被滥用,通过该方式实现关键信息的安全和数据的共享服务[15]。

(6) 执法监察数据汇总子系统。该子系统通过对执法监察数据的台账管理,智能分析,生成各类数据汇总结果。可以根据各种数据模板导出相应的数据结果,为数据汇总上交、决策支持等提供服务支撑。

(7) 执法监察数据交换子系统。该子系统将异源异构数据转换为执法监察系统的数据,并入库以供执法监察各子系统调用。如:将卫片导入 GIS 库、将各相关网站的举报信息导入违法来源库、将执法依据的法律法规整理入库、将历史矿产资源数据整理入库等。

4 结　语

该系统基本涵盖了自然资源和规划执法监察的现有业务范围,整合了原有执法监察系统,使国土执法监察业务与规划执法监察业务有效地集成和衔接,并实现了数据高效、快速的共享服务。系统上线以来,使用用户将近千人,运行稳定,极大地提高了工作效率和业务流转速度,提升了自然资源和规划机构的信息化水平。该系统对业务的整合是在原有业务的基础上进行的综合与归纳,如何进一步挖掘执法监察业务的需求,优化业务水平是下一个阶段研究的重点。

参 考 文 献

[1] 许传刚,尹丹. 浅谈自然资源执法监察工作新思路[J]. 国土资源,2019(6):56 - 57.

[2] 林坚,骆逸玲,吴佳雨. 自然资源监管运行机制的逻辑分析[J]. 中国土地,2016(3):17 - 19.

[3] 林坚,吴宇翔,吴佳雨,等. 论空间规划体系的构建——兼析空间规划、国土空间用途管制与自然资源监管的关系[J]. 城市规划,2018,42(5):9 - 17.

[4] 后向东. "互联网 + 政务":内涵、形势与任务[J]. 中国行政管理,2016(6):6 - 10.

[5] 张翠肖,王锋. 利用 VPN 技术构建企业内部网[J]. 计算机应用研究,2001(5):52 - 53.

[6] 陈莉莉,胡波,狄颖琪. 以大数据为核心的线网指挥中心建设方案[J]. 城市轨道交通研究,2020,23(1):51 - 54.

[7] 温昱. 软件架构设计[M]. 北京:电子工业出版社,2007:202 - 246.

[8] 李志昌,谢刚生. 采用多级缓存消息数据方法的国土资源移动执法监察系统[J]. 测绘通报,2013(3):102 - 105.

[9] 罗锡文,周志艳,李庆,等. 基于地图匹配的导航定位数据模糊校正算法[J]. 江苏大学学报(自然科学版),2006(5):396 - 400.

[10] 叶忠文,黄鹏,施金金. 基于 WebSocket 的 Web 实时通信系统[J]. 火力与指挥控制,2014,39(S1):181 - 183.

[11] 李文逍,杨小虎. 基于分布式缓存的消息中间件存储模型[J]. 计算机工程,2010,36(13):93 - 95.

[12] 管建和,甘剑峰. 基于 Lucene 全文检索引擎的应用研究与实现[J]. 计算机工程与设计,2007(2):489 - 491.

[13] 李承林. 基于光闸单向传输数据交换技术研究[J]. 激光杂志,2018,39(4):134 - 138.

[14] 周伟杰. 国土一张图移动执法实时巡查系统建设方案[J]. 测绘通报,2015(8):137 - 138.

[15] 沈海波,陈强,陈勇昌. 基于 OAuth 2.0 扩展的客户端认证方案[J]. 计算机工程与设计,2017,38(2):350 - 354.

(上接第 307 页)

[10] 孙越. 前后向安全的云数据密文搜索技术研究[D]. 合肥:安徽大学,2020.

[11] Poh G S, Chin J J, Yau W C, et al. Searchable symmetric encryption: Designs and challenges[C]//ACM Computing Surveys (CSUR),2017,50(3):1 - 37.

[12] Cao L, Wang Y, Dong X, et al. Multiuser access control searchable privacy preserving scheme in cloud storage[J]. International Journal of Communication Systems, 2018, 31(9):e3548.

[13] 张玉磊,文龙,王浩浩,等. 多用户环境下无证书认证可搜索加密方案[J]. 电子与信息学报,2020,42(5):1094 - 1101.

[14] Wang Q, Guo Y, Huang H, et al. Multi-user forward secure dynamic searchable symmetric encryption[C]//International Conference on Network and System Security, 2018: 125 - 140.

[15] Ye F, Dong X, Shen J, et al. A verifiable dynamic multi-user searchable encryption scheme without trusted third parties[C]//2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS),2019.