软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2019,30(8):2362−2374 [doi: 10.13328/j.cnki.jos.005763]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于倒排索引的可验证混淆关键字密文检索方案*

杜瑞忠[1,2], 李明月[1,2], 田俊峰[1,2], 吴万青[1,2]

[1](河北大学 网络空间安全与计算机学院,河北 保定 071002)

[2](河北省高可信信息系统重点实验室(河北大学),河北 保定 071002)

通信作者: 李明月, E-mail: 15630424277@163.com

摘 要: 随着云计算的发展,以密文检索为核心技术的安全搜索问题日益成为国内外研究的热点.为了提高密文检索方案的安全性,提出了基于倒排索引的可验证混淆关键字密文检索方案.首先,在构建陷门时插入混淆关键字抵抗恶意云服务器的关键字攻击,同时引入数据缓存区,利用 Pailliar 加密技术对包含混淆关键字搜索结果进行盲计算,过滤掉包含目标关键字以外的密文数据,减少通信开销;其次,利用双线性映射生成标签验证搜索结果,并对方案在正确性、安全性和可靠性这 3 个方面进行了验证.在真实数据集上进行反复实验,理论分析和实验结果表明,该方案在保证检索效率的同时,比现有的密文检索方案有效地提高了密文检索的安全性.

关键词: 密文检索;可验证;混淆关键字;数据缓存区;双线性

中图法分类号: TP309

中文引用格式: 杜瑞忠,李明月,田俊峰,吴万青.基于倒排索引的可验证混淆关键字密文检索方案.软件学报,2019,30(8):2362−2374. http://www.jos.org.cn/1000-9825/5763.htm

英文引用格式: Du RZ, Li MY, Tian JF, Wu WQ. Verifiable obfuscated keyword ciphertext retrieval scheme based on inverted index. Ruan Jian Xue Bao/Journal of Software, 2019,30(8):2362−2374 (in Chinese). http://www.jos.org.cn/1000-9825/5763.htm

Verifiable Obfuscated Keyword Ciphertext Retrieval Scheme Based on Inverted Index

DU Rui-Zhong[1,2], LI Ming-Yue[1,2], TIAN Jun-Feng[1,2], WU Wan-Qing[1,2]

[1](School of Cyberspace Security and Computer, Hebei University, Baoding 071002, China)

[2](Key Laboratory on High Trusted Information System of Hebei Province (Hebei University), Baoding 071002, China)

Abstract: With the development of cloud computing, the issue of secure search with ciphertext retrieval as the key technology has become a hot topic at worldwide. In order to improve the security of the ciphertext retrieval scheme, a verifiable ciphertext retrieval scheme is designed based on inverted index. First, insert the confusion keywords when building trapdoorstoagainst a malicious cloud server's keywords attack. At the same time, the data cache area is introduced, which utilize the Pailliar encryption technology to blindly calculate the search results containing the obfuscated keywords, and the ciphertext data other than the target keyword is filtered out to reduce the communication overhead. Secondly, take advantage of bilinear maps generate tags to verify search results, and verify the scheme in terms of correctness, security and reliability. Repeated experiments on real data sets, theoretical analysis and experimental results show that the proposed scheme can improve the security of ciphertext retrieval compared with the existing ciphertext retrieval schemes while ensuring retrieval efficiency.

* 基金项目: 国家自然科学基金(61572170, 61170254); 河北省自然科学基金(F2018201153, F2019201290); 河北省高等学校科学技术研究基金(ZD2016043)

Foundation item: National Natural Science Foundation of China (61572170, 61170254), Natural Science Foundation of Hebei Province (F2018201153, F2019201290); Science and Technology Research Project of Colleges and Universities of Hebei Province (ZD2016043)

本文由“面向自主安全可控的可信计算”专题特约编辑张焕国教授推荐.

收稿时间: 2018-05-27; 修改时间: 2018-09-21; 采用时间: 2018-12-13; jos 在线出版时间: 2019-3-28

CNKI 网络优先出版: 2019-03-29 09:16:37, http://kns.cnki.net/kcms/detail/11.2560.TP.20190329.0915.014.html

Key words: ciphertext retrieval; verifiable; confusing keyword; data buffers; bilinear

在计算机技术和互联网应用迅猛发展的推动下,用户对数据访问的需求日益增多,信息对储存容量的要求日益提高.云计算使用户可以按需地享受高质量服务和无处不在的网络访问[1],但是用户将数据外包给云服务器使数据脱离了物理控制,随之带来了数据隐私泄露的问题.

为了保障用户的数据隐私安全,通常在外包之前对数据进行加密,这样就限制了外包数据的可用性,使得广泛使用的基于关键字的明文信息检索技术不能直接应用于加密数据.为了解决上述问题,Song 等人[2]首次提出了基于密文扫描思想的可搜索加密方案.可搜索加密方案使用户能够将加密的数据存储到云中,并通过密文域执行关键字搜索,有选择地从云中检索感兴趣的文档,为用户节省了大量开销.此后,很多研究人员致力于通过加密数据进行安全关键字搜索.Curtmola 等人[3]提出了实现最优搜索时间的 2 种方案(SSE-1 和 SSE-2),其中,SSE-1 方案对于选择关键字攻击是安全的,SSE-2 对于自适应选择关键字攻击是安全的.但这些早期的方案只是用于单关键字的查询,在功能方面非常简单.为了实现不同的搜索功能,Ibrahim 等人[4]基于信息检索系统和密码学方法,提出了一种安全的关键字可搜索加密方案,并利用密码学原语 PPM 来提高方案的安全性.Chen 等人[5]应用稀疏矩阵的方法实现了大规模方程的安全外包.但以上方案都基于理想的假设,即云服务器“好奇但诚实”.在实际应用中,云服务器可能是恶意的,会删除长期不用的加密文件节省内存空间,或者伪造搜索结果来欺骗用户.为了从加密数据中获得有效的数据检索,需要对搜索的结果进行验证以保证其正确性.Sun 等人[6]首次提出了对加密数据进行安全排序的关键字搜索.此后,为了验证搜索结果正确性和完整性,Chen 等人[7]设计了最小哈希子树的结构,但该方法只适合索引树的结构;Wan 等人[8]设计了一个可信的隐私保护关键字搜索方案 VPSearch,该方案通过对同态 MAC 技术进行改进,实现了多关键字查询方案的隐私保护,但由于检索时需要搜索整个数据库,该方案对于大型数据库效率是非常低的;Liu 等人[9]提出一个动态可验证的排序 SSE 方案以保护云环境中的大数据安全,但是每次返回包含关键词的 top-K 个文件,该方案存在排序泄露等问题;Zhang 等人[10]提出了基于威慑的可验证关键字排名检索方案,在整个验证过程中,云服务器不清楚有哪些数据所有者,并允许数据用户根据喜好来控制验证的通信成本,但该方案依旧存在排序泄露等问题;Jiang 等人[11]提出了可验证的关键字排名密文检索方案,为每个关键字生成二进制向量,并使用 MAC 来检查返回密文的真实性,但由于使用 MAC 技术验证以及涉及大量文件向量间的的内积运算,带来了相对较高的计算开销.总之,现有的方案在安全性和效率方面有待提高,而文献[8,9]提出的可验证密文检索方案在效率和安全方面存在的问题具有代表性,因此在隐私保护度以及查询效率两个方面与其进行了对比实验.

为了获得更好的隐私保证并有效地对密文进行检索,本文设计了基于倒排索引的可验证混淆关键字密文检索方案(a verifiable obfuscated keyword ciphertext retrieval scheme based on inverted index,简称 VOKCRSII),利用安全的倒排索引结构实现次线性搜索,通过插入混淆关键字的技术来抵抗关键字攻击.主要工作如下.

1) 在生成陷门时引入混淆关键字,防止云服务器根据关键词的搜索频率推断出包含该关键词文件的价值,从而进行恶意攻击.
2) 引进数据缓存区,过滤返回搜索结果中包含混淆关键字的密文和验证数据,减少通信开销.
3) 引入双线性映射验证返回结果,并对恶意服务器模型中返回结果的正确性、安全性和可靠性进行了验证.
4) 在真实数据集上进行反复实验,性能分析和实验结果表明,VOKCRSII 方案在保证检索效率的同时,有效地提高了密文检索的安全性.

1 背景介绍

1.1 系统模型

本文的系统模型如图 1 所示,将云服务按功能不同分为 3 个实体——数据拥有者、云服务器和数据使用者.

1) 数据拥有者:数据拥有者要处理原始数据、建立倒排索引以及加密和上传数据与索引.此外,数据拥有

者要与数据使用者分享解密密钥,并授予数据使用者查询和验证的权利.

2) 数据使用者:数据使用者是授权用户,将查询的内容生成陷门 TD 发送给云服务器,并要求其返回前 K 个文档以及验证证据.收到搜索结果后,数据使用者执行验证算法:若验证算法返回 0,即返回结果错误;若验证算法返回 1,使用共享密钥对文档进行解密.

3) 云服务器:云服务器存储数据拥有者上传的加密数据与索引,当接到数据使用者合法的查询请求时,根据查询算法利用倒排索引进行计算,返回最相关的前 K 个密文文档以及验证证据.

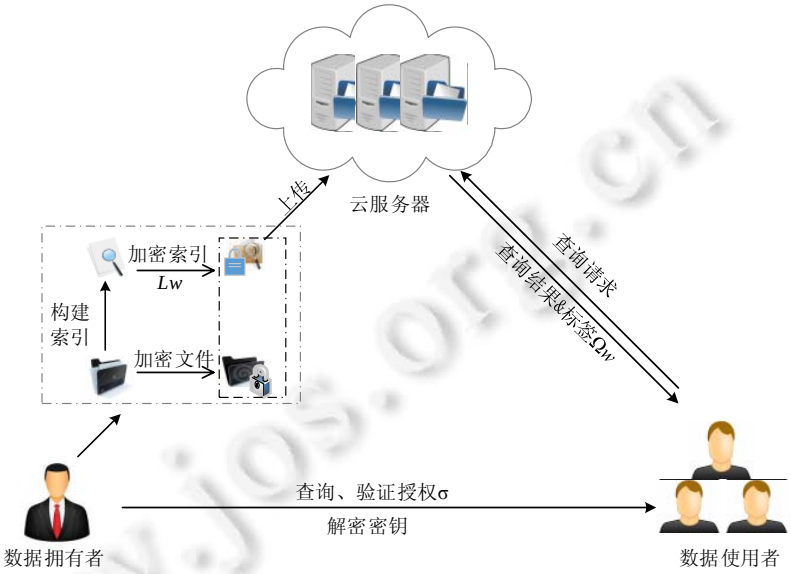


Fig.1 Architecture of ciphertext retrieval

图 1 密文检索的系统架构

1.2 安全模型

为了规范研究范围,本文假设云服务器是非完全可信的,即

1) 为了节省存储空间,云服务器可能删除部分加密文件或索引.

2) 为了节省计算或下载带宽,云服务器可能不执行数据使用者的查询请求,伪造搜索结果来欺骗用户[12].

3) 云服务器存在好奇心,可能会尝试分析其存储的以及消息流中的数据,推断甚至识别某些关键词.

1.3 主要符号表示

本文使用的主要符号的说明如下.

- D:明文集合 $D=\{D_1,\ldots,D_m\}$.
- C:密文集合 $C=\{C_1,\ldots,C_m\}$.
- W:字典集合 $W=\{w_1,\ldots,w_m\}$.
- $\#w$:包含关键字 w 的文件数量.
- m:文件数量.
- n:字典数量.
- τ:混淆参数.
- $I=\{T_s,A_s\}$:排名倒排索引结构.
- L_w:索引标记 $L_w=\{L_{w_1},\ldots,L_{w_n}\}$.
- TD:搜索陷门.

- Ω_w:云服务器生成的验证标签.
- K:用户指定返回文件数量.
- $D_{i,j}$:包含关键词 w_i 的第 j 个文件.
- $C_{w,K}$:关键字 w 的 top-K 搜索密文集合.
- $ID(w,K)$:包含关键字 w 的 top-K 文件标识符.

1.4 安全定义

定义 1(正确性). 设 T 是一个可验证的密文检索方案,如果 T 满足$\forall PK,SK\leftarrow Setup(1^{\lambda}),\forall D_i\subseteq D(1\leqslant i\leqslant n),\forall w\subseteq W$,有 $(search(TD,K,I)=(C_{w,K},\Pi_w)\wedge FliterConfu(C_{w,K},\Omega_w)\rightarrow C_{w_s,K},\Omega_{w_s}\wedge Verify(PK,SK,C_{w_s,K},\Pi_{w_s})=1\wedge DecFile(C_{w_s,K})=D_{w_s,K})=1$,则该方案是正确的.其中,$C_{w,K},\Pi_w$ 是包含混淆关键字的搜索结果和验证标签,$C_{w_s,K}$ 和 Π_{w_s} 是真正要搜索关键字的搜索结果和验证标签.

定义 2(自适应性选择关键字攻击安全). 设 T 是一个可验证的密文检索方案,ρ为攻击者,s 是模拟器,$Leak_1$ 和 $Leak_2$ 为泄漏算法.概率实验的 $Real_{\rho}^{T}(\lambda)$ 和 $Ideal_{\rho,S}^{T}(T)$ 满足:

$Real_{\rho}^{T}(\lambda)$ 由ρ来实现.挑战者通过运行 $Setup(1^{\lambda})\rightarrow PK,SK$ 来生成密钥,ρ选择一个文件集合 D 发送给挑战者,挑战者运行 $EncIndex(SK,D,W)\rightarrow I$ 和 $EncFile(D,SK)\rightarrow C$ 并将(I,C)给ρ.ρ发出多项式的自适应查询 q,对于每个查询 q,ρ接收挑战者运行 $SrcToken(w,PK,SK)\rightarrow TD$ 得到的陷门.其中,$EncIndex$ 是加密索引的算法,$EncFile$ 是加密文档的加密算法,$SrcToken$ 是生成陷门的算法.

$Ideal_{\rho,S}^{T}(T)$ 由ρ和 s 来实现.ρ选择一个文件集合 D,根据 $Leak_1(D)$,s 输出(I,C),并发送给ρ.ρ发出多项式的自适应查询 q,对于每个查询 q,s 根据 $Leak_2(D,w)$返回相应的陷门给ρ.

如果对于多项式时间的ρ,都存在多项式时间的 s,使 $\Pr[Ind_{\rho}^{T}(\lambda)=1]\leqslant\frac{1}{2}+negl(\lambda)$ 成立,其中,$negl(\lambda)$是可以忽略的,则 T 满足自适应选择关键字攻击安全.

定义 3(可靠性). 设 T 是一个可验证的密文检索方案,ρ为攻击者.满足以下条件.

$Forge_{\rho}^{T}(K)$:由ρ来实现.对于查询陷门 TD,ρ伪造一个虚假的结果集 $C_{w,K}^{*}$ 和对应的证据 Ω_w^{*},其中,

$C_{w,K}^{*}\neq C_{w,K},\Omega_w^{*}\neq\Omega_w,Search(TD,I,K)\rightarrow C_{w,K},GenProof(TD,PK,L_{w_i},C_{w_i,K})\rightarrow\Omega_w,FliterConfu(C_{w,K},\Omega_w)\rightarrow C_{w_s,K},\Omega_{w_s}$.

若 $Verify(PK,SK,C_{w_s,K}^{*},\Omega_{w_s}^{*})=1$ 可能性可以忽略,则 T 是可靠的,即 $\Pr[Fore_{\rho}^{T}(\lambda)=1]\leqslant negl(\lambda)$.其中,$negl(\lambda)$是可以忽略的.此外,$GenProof$ 生成标签的算法,$FliterConfu$ 是过滤算法.

1.5 双线性映射

双线性映射[13]:设 P 为λ比特的素数,Z_p 为有限域,G,G_T 是阶为素数 P 的循环群,g,g_T 分别为 G,G_T 对应的一个生成元.可定义一个双线性映射:e:$G\times G\rightarrow G_T$ 满足以下性质.

1) 双线性性:对于所有的 $a,b\in Z_p$,有 $e(g^a,g^b)=e(g,g)^{ab}$.
2) 非退化性:$e(g,g)\neq 1$.
3) 可计算性:对于任意元素 $g,h\in G$,可有效地计算 $e(g,h)$.

设 $X=\{x_1,...,x_l\}$,$Y=\{y_1,...,y_l\}$是两个 l 维的向量,则双线性映射的计算如下:

$$e(g_1^{X},g_2^{Y^T})=e(g_1^{x_1},g_2^{y_1})\cdot e(g_1^{x_2},g_2^{y_2})\cdot...\cdot e(g_1^{x_l},g_2^{y_l})=e(g_1,g_2)^{x_1y_1+x_2y_2+...+x_ly_l} \tag{1}$$

2 基于倒排索引的可验证混淆关键字密文检索方案

2.1 安全倒排索引创建

由于倒排索引搜索时间是次线性的[14],本文采取倒排索引实现密文的安全搜索,排序的倒排索引结构 $I=\{T_s,A_s\}$,如图 2 所示.

- 搜索数组A_s是一个长度为$M=\left(\sum_{i=1}^{m}\#w_i\right)$的数组,其中,$\#w_i$是指包含关键字$w_i$的文件数量.$A_s[i]$表示存储在位置 i 的值,对于关键词 $w_i\in W$,列表 A_{w_i} 被随机存储在搜索数组 A_s 中.列表 A_{w_i} 由$\#w_i$ 个节点($N_{i,1},\ldots,N_{i,\#w}$)组成,其中,$N_{i,j}=\langle w_i,id_j,RScore,addrs(N_{i,j+1})\rangle$,$id_i(j)\in ID(w)$是包含关键字 w_i 的 rank-j 文件的标识符,$addrs(N_{i,j+1})$是 L_w 的 rank-(j+1)个节点在搜索数组 A_s 中的地址.最后一个节点 $N_{i,\#w}=\langle w_i,id_{\#w},RScore,NULL\rangle$;
- 搜索表 T_s 是一个大小为 n 的字典, A_{w_i} 的头指针存储在搜索表 T_s 中.其中,F 和 P 分别为加密关键字和指针的伪随机函数.

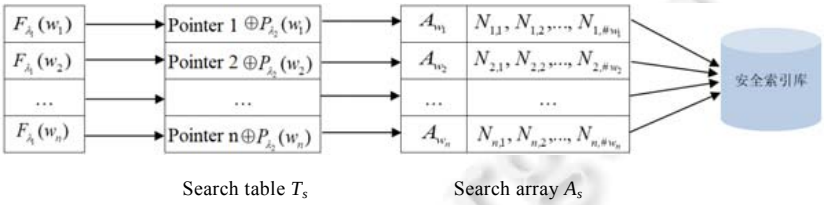


Fig.2 Security inverted index structure

图 2 安全倒排索引结构

2.2 混淆关键字

当数据使用者想要搜索密文时,可以通过关键字陷门来实现.但是,云服务器会根据经常被搜索的关键词推断出与其相关的文件数据非常具有价值,从而选择性地攻击这些文件;另一方面,云服务器还可能将长时间未得到搜索的数据恶意删除[15].为了防止云服务器的恶意攻击,构造如下检索请求.

1) 数据用户通过插入混淆关键字扩大搜索的关键字陷门.假设数据使用者想要检索包含 w_s 的加密文件,为了不让云服务器知道关键字集,可以在集合中添加其他 x 个关键词.
2) 数据使用者为每一个关键字增加一个特殊标志位 τ,然后用 Paillier[7]加密附加的 τ 来区分混淆关键字与用户真实检索关键字.
3) 假设数据使用者想要检索包含 w_s 的文件,并引入 x 个混淆关键字,则将 $\{\langle w_s,E(PK,\tau)\rangle,\langle w_{s+1},E(PK,\tau)\rangle,\ldots,\langle w_{s+x},E(PK,\tau)\rangle\}$ 上传到云服务器.

2.3 数据缓冲区

1) 映射过程

在接收到数据使用者的搜索请求后,云服务器根据倒排索引初步得到搜索结果,其中有包含混淆关键字的密文与验证证据,直接发送给数据使用者会增加云服务器和数据用户之间的通信开销.引入数据缓存区模块,先将搜索结果按照算法 1 映射到数据缓存区.

算法 1. 数据缓存区算法.

输入:搜索结果 θ.

输出:数据缓存区 DB.

1. The cloud initializes DB with x entries, each entry with initial value 1
2. **for** $i\in[s,s+x]$ **do**
3. Locates w_i data θ_i
4. Compute $d=E(PK,\tau)^{\theta_i}$
5. **for** j in $range(0,k)$ **do**
6. $DB_1[h_i(j)]=DB_1[h_i(j)]\cdot d$
7. **end for**

8. **end for**

9. **return** *DB*

其中,x 是数据使用者插入的混淆关键字个数,$E(PK,\tau)$中的τ是混淆参数,τ=1 表示数据使用者需要查询此关键字,τ=0 表示该关键字为混淆关键字.如图 3 中①所示,云服务器首先初始化验证数据缓冲区,并将结果映射到具有 k 散列函数的验证数据缓冲区,每个散列函数的输出属于[0,x].

2) 过滤过程

将云服务器初步的查询结果通过 Paillier 加密的同态性质直接映射到数据缓冲区后,云服务器进一步对密文和验证证据进行盲计算,过程如图 3 中②所示,在数据缓存区中过滤掉含有混淆关键字的密文和验证证据,减少通信开销.此外,从云服务器的角度来看,处理了 x+1 个关键词的验证数据和密文,无法得到实际返回的数据,提高了密文检索的安全性.

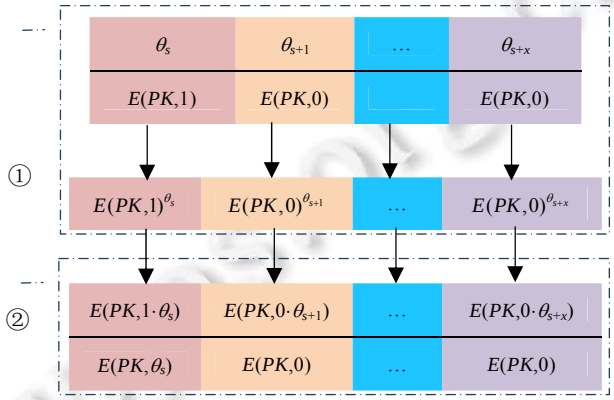


Fig.3 Data buffer

图 3 数据缓存区

3) 解密恢复数据

数据使用者收到云服务器返回的搜索结果后,对其进行解密.图 4 显示了解密结果,数据使用者可以从数据缓存区的第 1 个节点恢复θ_1.由于数据使用者可以预先计算没有发生冲突的条目,而不是解密整个数据缓冲区,可以提高解密效率.

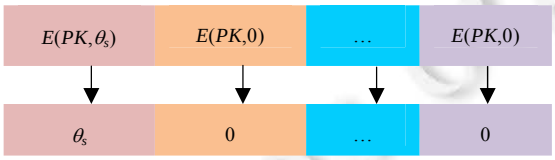


Fig.4 Paillier decryption filter

图 4 Paillier 解密过滤

2.4 方案设计

可验证的关键字排序可搜索加密方案构造如下.

- 初始化阶段

$Setup(1^\lambda)\rightarrow(PK,SK)$:用户运行 $KeyGen(1^\lambda)$产生(e,q,g),然后随机选择 3 个λ位的向量$\lambda_1,\lambda_2,\lambda_3$作为 F,P,H 的随机种子,$F:\{0,1\}^\lambda\times\{0,1\}^*\rightarrow\{0,1\}^\lambda$是一个无冲突散列函数,$P:\{0,1\}^\lambda\times\{0,1\}^*\rightarrow\{0,1\}^*$,$H:\{0,1\}^\lambda\times\{0,1\}^*\rightarrow\{0,1\}^\lambda$,再生成一个$\lambda$维向量 S, $S\xleftarrow{R}\{0,1\}^\lambda$.同时生成 2 个$\lambda\times\lambda$维的可逆矩阵$(M',M'')$.最后得到 $Key=\{PK,SK\}$,其中,$PK=(q,g)$,$SK=(e,\lambda_1,\lambda_2,\lambda_3,S,M'^T,M''^T)$.

- 存储加密阶段

1) *EncIndex*(*SK*,*D*,*W*)→*I*:对于每个关键字 $w_i \in W$,用户执行以下操作.

① 在 A_s 中随机选择#w_i 个位置创建列表 A_{w_i}.对于 $j \in [1,\#w_i]$,将 $N_{i,j}=\{w_i,id_j,RScore,addrs(N_{i,j+1})$加密得到 $A_s[addrs(N_{i,j})]=(N_{i,j} \oplus H_{\lambda_3}(w_i))$.

② 对于 $i \in [1,n]$,利用向量 S 将 $F_{\lambda_1}(w_i)$ 分割成 $F_{\lambda_1}(w_i)'$ 和 $F_{\lambda_1}(w_i)''$.

$$\begin{cases} F_{\lambda_1}(w_i)'=F_{\lambda_1}(w_i)''=F_{\lambda_1}(w_i)(\mathrm{mod}\,q),\ s_j=0 \\ F_{\lambda_1}(w_i)'+F_{\lambda_1}(w_i)''=F_{\lambda_1}(w_i)(\mathrm{mod}\,q),\ s_j=1 \end{cases} \tag{2}$$

加密 $F_{\lambda_1}(w_i)'$ 和 $F_{\lambda_1}(w_i)''$ 这两个向量得索引标记 $L_{w_i}=(L_{w_i,1},L_{w_i,2})=(g^{M'^T F_{\lambda_1}(w_i)'^T}, g^{M''^T F_{\lambda_1}(w_i)''^T})$.

T_s 是存储 A_{w_i} 头指针和索引标记 L_w 的列表,$T_s[F_{\lambda_1}(w)]=addrs(N_1)\|L_w \oplus P_{\lambda_2}(w)$.

最后,输出加密索引 $I=(T_s,A_s)$上传到云服务器.

2) *EncFile*(*D*,*SK*)→*C*:对于文件 $D_i \in D$,数据拥有者运行 *EncFile*(D_i)来生成密文 C_i,得到密文集合 $C=\{C_1, C_2,\ldots,C_m\}$,上传到云服务器.

3) *AccGen*(*PK*,*SK*,*D*,*W*)→σ:对于 $i \in [1,n]$,数据所有者将关键字集合 $W=\{w_1,...,w_n\}$中每个关键词根据 L_w 计算输出签名集合 $\sigma=\{\sigma_{w_1},...,\sigma_{w_n}\}$,其中,$\sigma_{w_i}=L_{w_i}\|\prod_{j=1}^{\#w_i}(id_i(j)+q)$. 数据拥有者赋予数据使用者验证权限,将签名集合$\sigma$发给数据使用者.

- 查询阶段

1) *SrcToken*(*w*,*PK*,*SK*)→*TD*:为了检索包含关键字 w_s 的 top-*K* 文件,同时为了不让云服务器对关键词 w_s 进行猜测,获取用户隐私,用户引入 x 个混淆关键字$\{w_{s+1},w_{s+2},\ldots,w_{s+z}\}$生成搜索陷门 $TD=(\zeta_1,\zeta_2,\zeta_3)$上传到云服务器.对于 $s<i<s+x$,

$$\zeta_1=((F_{\lambda_1}(w_s),E(PK,1)),...,(F_{\lambda_1}(w_{s+x}),E(PK,0))),$$
$$\zeta_2=(P_{\lambda_2}(w_s),P_{\lambda_2}(w_{s+1}),...,P_{\lambda_2}(w_{s+x})),$$
$$\zeta_3=(H_{\lambda_3}(w_s),H_{\lambda_3}(w_{s+1}),...,H_{\lambda_3}(w_{s+x})),$$

其中,$E(PK,1)$表示数据使用者需要查询此关键字,$E(PK,0)$表示该关键字为混淆关键字.

2) *Search*(*TD*,*K*,*I*)→$C_{w,K}$:云服务器接收到 $TD=(\zeta_1,\zeta_2,\zeta_3)$后,对于 $i \in (s,s+x)$,定位 $T_s[F_{\lambda_1}(w_i)]$:如果 $F_{\lambda_1}(w_i)$ 不在 T_s 中,返回 0;否则,对于 $i \in (s,s+x)$,计算 $T_s(F_{\lambda_1}(w_i)) \oplus P_{\lambda_2}(w_i)$ 恢复 A_{w_i} 的头指针与索引标记 L_{w_i},进而通过 $A_s[addr(N_{i,j}) \oplus H_{\lambda_3}(w_i)]$ 恢复 $N_{i,j}$,得到分别包含 $w=\{w_s,w_{s+1},w_{s+2},\ldots,w_{i+x}\}$的密文集合:

$$C_{w,K}=\{C_{w_s,K},C_{w_{s+1},K},...,C_{w_{s+x},K}\}.$$

3) *GenProof*(*TD*,*PK*, L_{w_i} ,$C_{w,K}$)→Ω_w:云服务器通过计算 $\Omega_{w_i}=\prod_{id \notin id(w_i,K)}(id_i(j)+q)$ 得到包含混淆关键字所有的验证标签集合 $\Omega_w=\{\Omega_{w_s},\Omega_{w_{s+1}},...,\Omega_{w_{s+x}}\}$,其中,$i \in (s,s+x)$.之后,将$\Omega_w$连同密文集合 $C_{w,K}$ 映射到数据缓存区.

4) *FilterConfus*($C_{w,K}$,Ω_w)→ $C_{w_s,K}$,Ω_{w_s} :云服务器返回包含混淆关键字所有的验证数据会导致云和数据用户之间的通信成本很高,如公式(3)所示.

$$\begin{cases} E(PK,1)^{\theta}=E(PK,1\cdot\theta)=E(PK,\theta) \\ E(PK,0)^{\theta}=E(PK,0\cdot\theta)=E(PK,0) \end{cases} \tag{3}$$

其中,$\theta=\{C_{w,K},\Omega_w\}$.过滤后得到的 $C_{w_s,K}$ 和 Ω_{w_s} 是密态的,云服务器无法识别,保障了用户数据的安全性.

通过 Paillier 加密的同态性质直接将验证数据映射到数据缓冲区,使云端对搜索结果进行盲计算,对验证数据Ω_w与密文 $C_{w,K}$进行过滤,得到数据使用者要检索的关键词 w_s 对应的密文集合和验证标签.

- 验证解密阶段

1) *Verify*(*PK*,*SK*, $C_{w_s,K}$,Ω_{w_s})→{0,1}:根据数据缓存区得到的密文与标签,数据使用者进行验证.

① 令 $\beta=F_{\lambda_1}(w_s)$, 利用 S 向量将β分裂成两个向量β'和β''.

$$\begin{cases}\beta'=\beta''=\beta \pmod q, & s_i=1\\ \beta'+\beta''=\beta \pmod q, & s_i=0\end{cases} \tag{4}$$

加密β'和β''得到$VK_{w_s}=(VK_{w_s,1},VK_{w_s,2},VK_{w_s,3})=(\beta'M'^{-1^T},\beta''M''^{-1^T},g^{|\beta|^2})$.

② 数据使用者利用公式(5)验证返回的是否为包含关键字 w_s 的密文集:

$$\boldsymbol{L}_{w_s,1}^{\boldsymbol{VK}_{w_s,1}}\cdot\boldsymbol{L}_{w_s,2}^{\boldsymbol{VK}_{w_s,2}}\overset{?}{=}\boldsymbol{VK}_{w_s,3} \tag{5}$$

验证失败返回 0;否则,再用公式(6)验证云服务器返回的是否为 top-K 文件:

$$L_{w_s}\|\prod\nolimits_{i\in id(w_i,K)}(id_i(j)+q)\cdot\Omega_{w_s}\overset{?}{=}\sigma_{w_s} \tag{6}$$

2) $DecFile(C_{w_s,K},SK)\to D_{w_s,K}$:若通过验证,则利用私钥解密得到明文集 $D_{w_s,K}$;若没有通过验证,则返回 0.

3 方案的安全性验证

定理 1. VOKCRSII 方案是正确的.

证明:根据陷门 $TD=(\zeta_1,\zeta_2,\zeta_3)$,云服务器可以定位关键字集 $w=\{w_s,w_{s+1},w_{s+2},\ldots,w_{i+x}\}$在表 T_s 中的位置,并通过计算 $T_s[F_{\lambda_1}(w)]\oplus P_{\lambda_2}(w)$ 来解密列表 A_w 在数组 A_s 的相应地址,从而得密文集合 $C_{w,K}=\{C_{w_s,K},C_{w_{s+1},K},\ldots,C_{w_{s+x},K}\}$.此外,云服务器运算 $\Omega_{w_i}=L_{w_i}\|\prod_{j=1}^{\#w_i}(id_i(j)+q)$ 得到标签集合 $\Omega_w=\{\Omega_{w_s},\Omega_{w_{s+1}},\ldots,\Omega_{w_{s+x}}\}$,再通过 Paillier 加密的同态性质直接将包含混淆关键字的验证数据和密文映射到验证数据缓冲区,对搜索结果进行盲计算,筛选得到 $C_{w_s,K}$ 和 Ω_{w_s} 之后发送给数据使用者.接收到来自云服务器的搜索结果 $C_{w_s,K}$ 和证据 Ω_{w_s} 后,数据使用者首先检查 $\boldsymbol{L}_{w_s,1}^{\boldsymbol{VK}_{w_s,1}}\cdot\boldsymbol{L}_{w_s,2}^{\boldsymbol{VK}_{w_s,2}}\overset{?}{=}\boldsymbol{VK}_{w_s,3}$,运算过程如公式(7)所示.

$$\boldsymbol{L}_{w,1}^{\boldsymbol{VK}_{w_s,1}}\cdot\boldsymbol{L}_{w,2}^{\boldsymbol{VK}_{w_s,2}}=(g^{M'^TF_{\lambda_1}(w_i)'^T})^{t_i'M'^{-1^T}}\cdot(g^{M''^TF_{\lambda_1}(w_i)''^T})^{t_i''M''^{-1^T}}=g_i^{t_i'(M'M'^{-1})^TF_{\lambda_1}(w_i)'^T}\cdot g_i^{t_i''(M''M''^{-1})^TF_{\lambda_1}(w_i)''^T}=\boldsymbol{VK}_{w_s,3} \tag{7}$$

然后检验 $L_{w_i}\|\prod_{j\in id(w_i,K)}(id_i(j)+q)\cdot\Omega_{w_s}\overset{?}{=}\sigma_{w_s}$, 如果云服务器没有恶意行为,则验证通过.

此外, $L_{w_i}\|\prod_{i\in id(w_i,K)}(id_i(j)+q)\cdot\Omega_{w_s}=\sigma_{w_s}$ 保证了 L_{w_i} 和相应加密索引的绑定和不可伪造性.最后,数据使用者解密文档得到排序搜索结果.因此,方案是正确的.

在验证 VOKCRSII 满足自适应选择关键字攻击之前,对泄漏函数进行形式化描述[16],泄漏函数 $Leak_1$ 定义为 $Leak_1(D)=(\#D,m,\#D_i,id(D_i))$.它将文档集合 D 作为输入,输出文档集的大小$\#D$,文档数量 m,每个文档的大小$\#D_i$和文件标识符 $id(D_i)$.泄漏函数 $Leak_2$ 定义为 $Leak_2(D,w)=(AP(w),TD)$,将文档集合和查询关键字 w 作为输入,并输出关键字 w 的访问模式和陷门.其中,$AP(w)=(id(D_1),\ldots,id(D_{\#w_i}))$. □

定理 2. VOKCRSII 方案满足自适应性选择关键字攻击安全.

证明:令$\lambda\in N$ 为安全参数,ρ为攻击者,s 模拟器,需要证明:对于多项式时间的ρ, $\Pr[Ind_\rho^T(\lambda)=1]\leqslant\frac{1}{2}+negl(\lambda)$.

模拟器自适应地生成模拟加密索引 $I'=(T_s',\{A_i'\mid i=1,\ldots,n\})$,模拟密文序列 C'和模拟陷门 TD'的过程如下.

1) 模拟加密索引 I':为了模拟索引,s 初始化最大长度为$\#w_i$的 A_i',A_i' 的每个条目为 N_j' ($1<j<M$),再用 ζ_3' 加密 N_j',ζ_3' 是由随机函数生成的字符串.s 将 T_s' 设置为具有 n 个条目的查找表.对于 $1<i<n$,生成一个二元组 $(\zeta_1',addr(A_i')\oplus\zeta_2')$,$\zeta_1'$ 和 ζ_2' 都是由随机函数生成的字符串, $addr(A_i')$ 是数组 A_i' 的地址.而在构造索引的过程中, $Real_\rho^T(\lambda)$ 利用伪随机函数 F,P 和 H,ρ在不知道密钥的情况下,不能区分伪随机函数的输出和相同大小的随机字符串,因此,ρ不能区别 I 与 I'.即

$$|\Pr[EncIndex(Key,D,W)\to I]|-|\Pr[Random\to I']|\leqslant negl_1(\lambda).$$

2) 模拟密文序列 C':s 根据泄漏函数 $Leak_1(D)$模拟加密文档 C_j' ($1<j<m$),得到 $C'=\{C_1',C_2',\ldots,C_m'\}$.$\rho$没有密钥可以保证加密文档 C_j 和密文 C_j' 在计算上是不可区分的,即

$$|\Pr[EncFile(Key,D)\to C]|-|\Pr[Random\to C']|\leqslant negl_2(\lambda).$$

3) 模拟陷门 TD':假设插入 x 个混淆关键字,对于 $s<i<s+x$,有:

$$Leak_2(D,w_i)=(AP(w_i),TD),AP(w_i)=(id(D_1),\ldots,id(D_{\#w_i})).$$

s 通过 $Leak_2(D,w_i)$得到 $AP(w_i)$.根据模拟的加密索引有 $T_s'[\zeta_1']=(addr(A_w')\|L_w\oplus\zeta_2')$:

- 如果 $j\neq\#w$,s 计算 $N_{i,j}=(\langle w_i,id_j,RScore,addrs(N_{i,j+1})\rangle\oplus\zeta_3'),A_i'[j]=(N_{1,j},\zeta_3')$;
- 如果 $j=\#w$,s 计算 $N_{1,j}=(\langle w_i,id_j,RScore,addrs(N_{i,j+1})\rangle\|Null\oplus\zeta_3'),A_i'[j]=(N_{i,j},\zeta_3')$.

最后,s 返回陷门 $TD'=\{\zeta_1',\zeta_2',\zeta_3'\}$.由于不具有密钥$\lambda_1$,$\lambda_2$ 和λ_3,可保证 $Real_\rho^T(\lambda)$ 中的 TD 与 $Ideal_{\rho,S}^T(T)$ 中的 TD'的不可区分性.即

$$|\Pr[Keygen(1^\lambda)\to Key]|-|\Pr[Random\to Key']|\leqslant negl_3(\lambda).$$

由于ρ试图通过分析加密索引、密文和密钥来获胜,则

$$\left.\begin{aligned}\Pr(Ind_\rho^T(\lambda)=1)&=\frac{1}{2}+Adv(Adv(\rho(I))+Adv(\rho(C)+\rho(Key)))\\&=\frac{1}{2}+|\Pr[EncIndex(Key,D,W)\to I]|-|\Pr[Random\to I']|+\\&\quad|\Pr[EncFile(Key,D)\to C]|-|\Pr[Random\to C']|+\\&\quad|\Pr[Keygen(1^\lambda)\to Key]|-|\Pr[Random\to Key']|\\&\leqslant\frac{1}{2}+|negl_1(\lambda)+negl_2(\lambda)+negl_3(\lambda)|\end{aligned}\right\}\tag{8}$$

令 $negl(\lambda)=negl_1(\lambda)+negl_2(\lambda)+negl_3(\lambda)$,则 $\Pr[Ind_\rho^T(\lambda)=1]\leqslant\frac{1}{2}+negl(\lambda)$.其中,$Adv(\rho(Key))$是$\rho$区分密钥与随机字符串的优势,$Adv(\rho(I))$是$\rho$区分索引与随机字符串的优势,$Adv(\rho(C))$是$\rho$区分加密文档和真实密文的优势.

综上,对于多项式时间的ρ, $Real_\rho^T(\lambda)$ 与 $Ideal_{\rho,S}^T(T)$ 的输出是不可区分的.VOKCRSII 方案满足自适应性选择关键字攻击安全. □

定理 3. VOKCRSII 方案满足定义 3 中的可靠性.

证明:假设 VOKCRSII 方案不可靠,则对于搜索请求 TD 返回无效搜索结果 $C_{w,k}^*$ 和伪造证据 ${}^*\Omega_w$,使得算法 $Verify(PK,SK,C_{w,k}^*,{}^*\Omega_w)$输出 1.令 $C_{w,k}=\{C_1,C_2,\ldots,C_{\#w}\}$表示正确的搜索结果:首先可能是文档的内容会被修改,即在返回的结果集中存在密文 C_j^*,但 $C_j^*\neq C_j$, 其中,$j\in\{1,\ldots,\#w\}$,ρ伪造证据 ${}^*\Omega_w=\prod_{id\notin id(w_i,K)}(id_i(j)+q)$ 发送给数据使用者;其次是返回的结果集中缺少文档 C_j,ρ伪造证据 ${}^*\Omega_w=\prod_{id=j,id\notin id(w_i,K)}(id_i(j)+q)$ 发送给数据用户.

如果 VOKCRSII 方案不可靠,无效的搜索结果 $C_{w,k}^*$ 和 ${}^*\Omega_w$ 将通过验证算法.然而,由于双线性映射提供了消息不可伪造性,伪造有效证据的可能性可以忽略不计.这与上述假设相悖.因此,VOKCRSII 方案具有可靠性. □

4 性能分析

本文实验原型系统的开发和测试环境是基于 windows 7(64 位)系统,具体硬件配置是 intel(R)Core(TM)i7-6700(3.40GHz)处理器,配备 8GB 内存和网速为 1Gbps 的校园网环境.使用国内云存储提供商阿里云的云存储平台(搭载 centos 7.3 64 位系统,主频为 2.5GHz 的 4 核 CPU,16GB 内存,内网宽带为 0.8Gbps,公网宽带为 100Mbps,系统盘为 40GB 高效云盘)搭建存储系统.实验数据使用 20 431 篇英文文章作为测试数据集,用 Lucene 分词器对纯文本字节流进行分词,过滤掉 29.1%的停用词,进行关键词提取形成关键词数为 7 200 的关键词集合.在实验中,从数据集中选择 m=3012 个文件,不同关键字的数量 n=1000,每个关键字出现在 1 个~44 个文件中.实验构建了不同关键字数量的索引(即 n=100,200,…,1000),使用 VOKCRSII 加密索引,使用 AES 加密数据集,加密算法由 JPBC 库实现.多次执行每个实验以获得平均执行时间.

4.1 功能比较

如表 1 所示,文献[9]和文献[8]的方案都可以实现关键词搜索结果排序与结果可验证的功能.

与上述方案相比,VOKCRSII 方案不仅支持多关键字搜索结果排序与可验证的功能,还支持插入混淆关键字,使方案安全性更高.

Table 1 Function comparison

表 1 功能比较

功能	文献		
	文献[9]	文献[8]	VOKCRSII
结果排序	√	√	√
可验证	√	√	√
混淆关键字	–	–	√

4.2 安全性

实验利用公式(9)检测文献[8]、文献[9]和 VOKCRSII 方案的隐私保护水平.

$$H(D)=-\sum_{i=1}^{m}p(D_i)\log_2 p(D_i) \tag{9}$$

其中,$0<p(D_i)<1$, $\sum_{i=1}^{m}p(D_i)=1$.

$H(D)$越大,隐私泄露可能性就越小,在没有外部条件影响时,该值是一个确定的值[17].

如图 5 所示是文献[8]、文献[9]与 x=2 时的 VOKCRSII 隐私保护度对比.VOKCRSII 赋予用户可验证的权利,且在查询陷门中引进混淆关键字,防止云服务器恶意攻击搜索频率高的数据,或者删除搜索频率低的数据,因此,x=2 时,VOKCRSII 方案的隐私保护度高于文献[9]和文献[8]提出的方案,安全性更高.

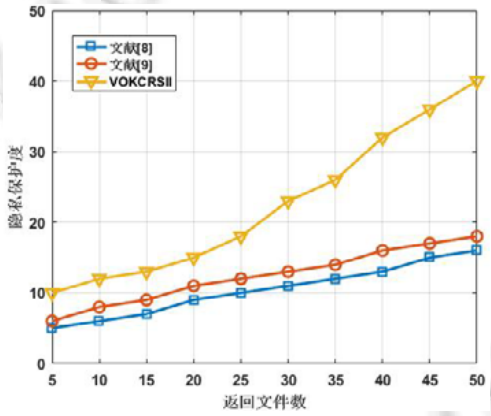


Fig.5 Comparison chart of privacy protection

图 5 隐私保护度比较图

4.3 检索效率

授权用户在进行检索时,总希望快速地得到检索结果[18],本文分别对方案的生成陷门时间、查询时间和验证时间进行实验.由第 4.2 节可知,x=2 时,VOKCRSII 方案的安全性已经高于文献[9]和文献[8]的方案.随着插入混淆关键字个数的增长,安全性会增加,但会带来更多的计算开销.因此选择引入 2 个混淆关键字的 VOKCRSII 方案做对比实验.

1) 数据缓存区

VOKCRSII 比文献[9]方案多消耗的时间主要体现在云服务器将查询数据映射到数据缓存区过滤掉包含混淆关键字和利用 Paillier 解密恢复数据两个方面.表 2 的第 2 行是插入不同数量混淆关键字时映射和过滤消耗的时间,当 x=2 时,时间为 0.160s.表 2 的第 3 行是随着插入混淆关键字数量增加利用 Paillier 解密恢复数据的时间.当 x=2 时,时间为 0.031s.

Table 2 Time of mapping filter and decryption data

表 2 映射过滤和解密时间

混淆关键字个数	1	2	3	4	5	6	7	8	9
映射过滤时间(s)	0.081	0.160	0.252	0.325	0.401	0.474	0.582	0.701	0.823
Paillier解密时间(s)	0.023	0.031	0.034	0.042	0.045	0.043	0.047	0.051	0.051

2) 查询效率

查询过程可以分为陷门生成、查询和验证 3 个部分.

生成陷门时,如表 3 所示,文献[8]的方案用到了大量的内积运算,复杂度是 $O(n^2)$,与关键词集大小有关.文献[9]的方案中,陷门只是由 3 个 PRF 产生的 3 个伪随机位序列组成.构造陷门的复杂度是 $O(\lambda)$.VOKCRSII 虽然在陷门中加入了混淆关键字,但构造陷门的复杂度仍是 $O(\lambda)$.如图 6,VOKCRSII 和文献[9]的构造陷门时间只与随机种子λ相关,而与 n 无关,随着关键词数量的增加时间几乎不变.

Table 3 Comparison of time complexity

表 3 时间复杂度比较

Scheme	Trapdoor	Search	Verify
文献[8]	$O(n^2)$	$O(nm)$	$O(nm)+O(m\log m)$
文献[9]	$O(\lambda)$	$O(\#w)$	$O(\#w+\Sigma\#C_{top\text{-}K})+O(\Sigma\#w)$
VOKCRSII	$O(\lambda)$	$O(xK)$	$O(\#w+\lambda)+O(\#w)$

查询时,文献[8]涉及搜索陷门和每个文档子索引的内积,见表 3,查询时间的复杂度为 $O(nm)$.由于倒排索引搜索的时间成本与包含 w 的文档的数量成线性关系,文献[10]查询时间复杂度为 $O(\#w)$,VOKCRSII 由于要搜索 x 混淆关键字的文件以及将数据映射到数据缓存区,因此查询时间的复杂度为 $O(xK)$.如图 7 所示,由于文献[8]的方案检索时间随着文档数量的增加而增加,时间最长.而文献[9]与 VOKCRSII 只与包含搜索关键字的文件数量相关,相较文献[8]的方案查询时间增长缓慢,其中,VOKCRSII 引入了混淆关键字来提高检索的安全性,需要过滤混淆关键字,VOKCRSII 比文献[9]的方案的所用时间长,但相差不是很多.

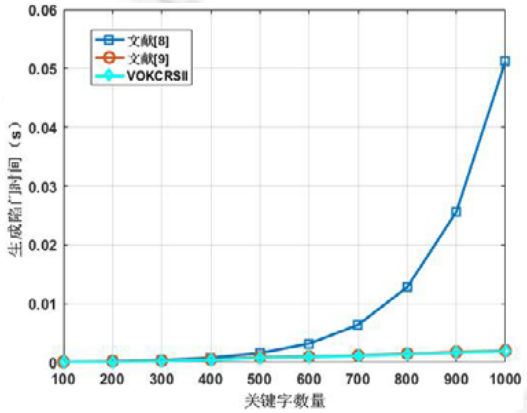


Fig.6 Time of generate trapdoor

图 6 陷门生成时间

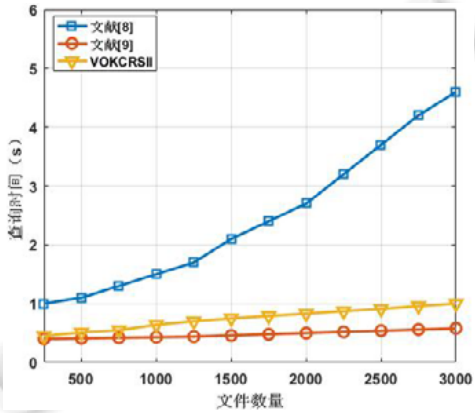
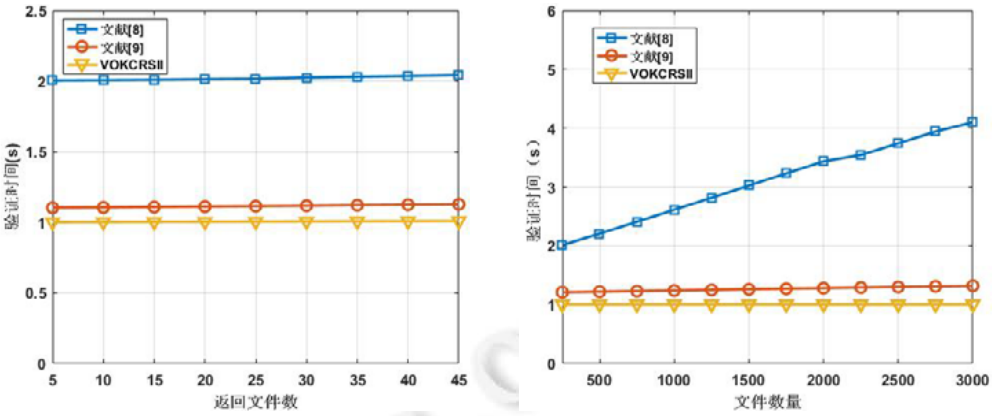


Fig.7 Search time

图 7 查询时间

验证时间包括在云服务器端生成标签的时间与用户验证的时间两部分.如表 3 所示,由于文献[8]要计算文档之间的向量积,在客户端验证搜索结果的复杂度为 $O(nm)$;云服务器端通过 hash 验证树生成标签,时间复杂度为 $O(m\log m)$.文献[9]在云服务器链签名技术生成标签,时间复杂度为 $O(\Sigma\#w)$;收到来自云服务器的返回结果和标签后,客户端利用 MAC 将查询关键字和返回的 top-K 文档的连接作为输入进行验证,复杂度为 $O(\#w+\Sigma\#C_{top\text{-}K})$,其中,$\#w$ 表示查询关键字的长度,$\#C_{top\text{-}K}$ 表示返回的 top-K 文档的总长度.VOKCRSII 在云服务器生成标签时间复杂度为 $O(\#w)$;在客户端数据用户先利用双线性映射的性质确定返回的结果是否是包含关键字 w_s

的文件,再验证来确定返回的结果是否正确,复杂度为 $O(\#w+\lambda)$.如图 8(a)所示,由于文献[8]利用 MAC 来验证,验证时间最长.文献[9]验证的复杂度与 top-K 文档的总长度相关,随着用户要求返回文档数量的增加,检索时间增长.由于 VOKCRSII 引入混淆关键字,映射过滤消耗的时间要随之增加,但 VOKCRSII 验证时不涉及对返回密文的计算,验证时间最短.如图 8(b)所示,当 Top-K=20 时,随着文件集数量变化,文献[9]验证消耗时间呈线性增长.而文献[9]和 VOKCRSII 与包含查询关键字的文档数量有关,验证时间增长缓慢.



(a) 不同的返回文档数量对应的验证时间 (b) 不同的文档数量对应的验证时间

Fig.8 Verification time

图 8 验证时间

5 总 结

本文提出了一个安全有效的关键字密文检索方案 VOKCRSII.该方案通过混淆关键字隐藏搜索频率,并利用双线性映射生成标签验证搜索结果,提高方案的安全性.同时,对其正确性、安全性和可靠性这 3 个方面进行了验证.通过分析验证,VOKCRSII 满足自适应性选择关键字攻击安全.此外,利用加密标志位区分混淆关键字和真正要检索的关键字,生成陷门上传至云服务器,但根据陷门得到的搜索结果有包含混淆关键字的文件,利用 Paillier 加密算法生成数据缓存区过滤掉多余文件,以减少通信开销.通过建立密文检索实验平台,验证 VOKCRSII 在保证检索效率的同时,提高了密文检索的安全性.但 VOKCRSII 只支持加密文档集的查询,将可搜索加密技术扩展到关系型数据库,是未来要做的工作.

References:

[1] Li JW, Jia CF, Liu ZL, Li J, Li M. Survey on the searchable encryption. Ruan Jian Xue Bao/Journal of Software, 2015, 26(1):109−128 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4700.htm [doi: 10.13328/j.cnki.jos.004700]

[2] Song XD, Wagner D, Perrig A. Practical techniques for searches on encrypted data. In: Proc. of the IEEE Symp. on Security and Privacy. IEEE Press, 2000. 44−55.

[3] Curtmola R, Garay J, Kamara S, Ostrovsky R. Searchable symmetric encryption: Improved definitions and efficient constructions. In: Proc. of the 13th ACM Conf. on Computer and Communications Security. New York: ACM Press, 2006. 79−88.

[4] Ibrahim A, Jin H, Yassin AA, *et al*. Secure rank-ordered search of multi-keyword trapdoor over encrypted cloud data. In: Proc. of the IEEE Asia-Pacific Services Computing Conf. IEEE Computer Society, 2012. 263−270.

[5] Chen XF, Huang XY, Li J, *et al*. New algorithms for secure outsourcing of large-scale systems of linear equations. IEEE Trans. on Information Forensics & Security, 2014,10(1):69−78.

[6] Sun W, Wang B, Cao N, *et al*. Privacy-Preserving multi-keyword text search in the cloud supporting similarity-based ranking. In: Proc. of the ACM Sigsac Symp. on Information, Computer and Communications Security. ACM Press, 2013. 71−82.

[7] Chen C, Zhu X, Shen P, *et al*. An efficient privacy-preserving ranked keywordssearch method. IEEE Trans. on Parallel & Distributed Systems, 2016,27(4):951−963.

[8] Jiang X, Yu J, Yan J, *et al*. Enabling efficient and verifiable multi-keyword ranked search over encrypted cloud data. Information Sciences, 2017,403(3):22−41.

[9] Liu Q, Nie X, Liu X, *et al*. Verifiable ranked search over dynamic encrypted data in cloud computing. In: Proc. of the Int'l Symp. on Quality of Service. IEEE, 2017. 1−6.

[10] Zhang W, Lin Y, Gu Q. Catch you if you misbehave: ranked keyword search results verification in cloud computing. IEEE Trans. on Cloud Computing, 2018,1(6):74−86.

[11] Wan Z, Deng RH. VPSearch: Achieving verifiability for privacy-preserving multi-keywordsearch over encrypted cloud data. IEEE Trans. on Dependable & Secure Computing, 2018,15(6):1083−1095.

[12] Zhang R, Xue R, Yu T, *et al*. PVSAE: A public verifiable searchable encryption service framework for outsourced encrypted data. In: Proc. of the IEEE Int'l Conf. on Web Services. IEEE, 2016. 428−435.

[13] Qiu S. Research on privacy-preserving keyword search and set operations over encrypted data [Ph.D. Thesis]. Beijing: Beijing Jiaotong University, 2017 (in Chinese with English abstract).

[14] Wu ZQ, Li KL, Zheng H. Efficient and scalable architecture forsearchable symmetric encryption. Journal on Communications, 2017,38(8):79−93 (in Chinese with English abstract).

[15] Wang SP, Liu LJ, Zhang YL. Verifiable dictionary-based searchable encryption scheme. Ruan Jian Xue Bao/Journal of Software, 2016,27(05):1301−1308 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4912.htm [doi: 10.13328/j.cnki.jos.004912]

[16] Du MX, Wang Q, He MQ, *et al*. Privacy-Preserving indexing and query processing for secure dynamic cloud storage. IEEE Trans. on Information Forensics and Security, 2018,13(9):2320−2332.

[17] Peng CG, Ding HF, Zhu YJ, Tian YL, Fu ZF. Information entropy models and privacy metrics methods for privacy protection. Ruan Jian Xue Bao/Journal of Software, 2016,27(8):1891−190 (in Chinese with English abstract). http://www.jos. org.cn/1000-9825/5096.htm [doi: 10.13328/j.cnki.jos.005096]

[18] Dong XL, Zhou J, Cao ZF. Research advances on secure searchable encryption. Journal of Computer Research and Development, 2017,54(10):2107−2120 (in Chinese with English abstract).

附中文参考文献:

[1] 李经纬,贾春福,刘哲理,李进,李敏.可搜索加密技术研究综述.软件学报,2015,26(1):109−128. http://www.jos.org.cn/1000-9825/4700.htm [doi: 10.13328/j.cnki.jos.004700]

[13] 邱硕.面向隐私保护的密文数据检索与集合操作的关键技术研究[博士学位论文].北京:北京交通大学,2017.

[14] 吴志强,李肯立,郑蕙.高效可扩展的对称密文检索架构.通信学报,2017,38(8):79−93.

[15] 王尚平,刘利军,张亚玲.可验证的基于词典的可搜索加密方案.软件学报,2016,27(5):1301−1308. http://www.jos.org.cn/1000-9825/4912.htm [doi: 10.13328/j.cnki.jos.004912]

[17] 彭长根,丁红发,朱义杰,田有亮,符祖峰.隐私保护的信息熵模型及其度量方法.软件学报,2016,27(8):1891−1903. http://www.jos.org.cn/1000-9825/5096.htm [doi: 10.13328/j.cnki.jos.005096]

[18] 董晓蕾,周俊,曹珍富.可搜索加密研究进展.计算机研究与发展,2017,54(10):2107−2120.

杜瑞忠(1975－),男,河北献县人,博士,教授,CCF 专业会员,主要研究领域为可信计算,信息安全.

田俊峰(1975－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为分布计算,可信计算,信息安全.

李明月(1993－),女,硕士生,主要研究领域为可信计算,信息安全.

吴万青(1981－),男,博士,讲师,主要研究领域为信息安全,密码学.