doi: 10.12068/j.issn.1005-3026.2019.03.004

基于 Paillier 和 PSI 的多关键字可搜索加密方案

周福才, 张宗烨, 王恺璇, 李宇溪
(东北大学 软件学院, 辽宁 沈阳 110169)

摘 要: 围绕多关键字的高效密文搜索和数据安全性保障问题,展开分析与研究,基于同态加密和私有集合交集技术,提出一种面向多关键字的高效的保护搜索模式的可搜索加密方案. 该方案使用随机数填充和 Paillier 同态加密方法构造安全索引和陷门,保护了索引隐私和陷门隐私,进而保护了搜索模式;该方案通过私有集合交集技术进行连接多关键字搜索,搜索中只使用到了乘法和指数运算,与其他方案相比大大提高了效率;安全性和性能分析表明,该方案具有可搜索加密的语义安全性,可以高效地实现对密文的多关键字搜索,且具有良好的计算代价.

关 键 词: 可搜索加密;多关键字;云存储;Paillier 同态加密;私有集合交集
中图分类号: TP 309 **文献标志码**: A **文章编号**: 1005-3026(2019)03-0321-06

Multi-keyword Searchable Encryption Based on Paillier and Private Set Intersection

ZHOU Fu-cai, ZHANG Zong-ye, WANG Kai-xuan, LI Yu-xi
(School of Software, Northeastern University, Shenyang 110169, China. Corresponding author: ZHOU Fu-cai, E-mail: fczhou@mail.neu.edu.cn)

Abstract: Efficient multi-keyword ciphertext search and problems of data security protection were researched. Based on Paillier homomorphic encryption and private set intersection (PSI) technique, a multi-keyword searchable encryption scheme was provided, which can search over encrypted files efficiently and protect the search pattern. First, through random number padding and Paillier homomorphic encryption the index privacy and the trapdoor privacy were protected, and the search pattern was further protected. Then, the scheme supports conjunctive multi-keyword search by using PSI. Because only multiplications and exponentiations were used in searching, the proposed scheme is more efficient than others. Security and performance analysis showed that the scheme has the semantic security of searchable encryption and can perform multi-keyword search efficiently with a good computational cost.

Key words: searchable encryption; multi-keyword; cloud storage; Pailler homomorphic encryption; private set intersection

近年来云存储技术发展迅猛且应用广泛,但同时也带来严重的安全问题. 出于对云存储环境中个人数据隐私性的保护,用户会将数据处理成密文形式再存储到远程的云服务器中,但是这样直接导致无法对加密后的数据进行正常的语义搜索,可搜索加密(searchable encrypion, SE)机制[1]可以更好地解决云存储环境中的密文数据搜索问题.

Song 等[2]首次将可搜索加密机制应用于关键字搜索,通过遍历加密数据库实现了单关键字对称可搜索加密. Dan 等[3]通过双线性映射和陷门置换的方法构建了 2 个 PEKS(public key encryption with keyword search)方案,首次实现了公钥可搜索加密. 随后越来越多的学者研究基于公钥密码学的可搜索加密算法. 基于公钥密码学的密文搜索机制大部分构建于双线性配对之上,

收稿日期: 2018-01-02
基金项目: 国家自然科学基金资助项目(61772127,61872069); 中央高校基本科研业务费专项资金资助项目(N151704002).
作者简介: 周福才(1964-),男,吉林长春人,东北大学教授,博士生导师.

其安全性都是基于离散对数问题、判定性 Diffie-Hellman 问题等不同的困难性假设. 而基于双线性映射构建的公钥可搜索加密方案的效率开销远高于对称可搜索加密方案,致使其在算法和技术上都无法满足现实工作环境的需要.

为了提高搜索效率,Goh 等[4]在可搜索加密方案中引入了安全索引的概念,服务器通过搜索索引来进行查询,将搜索复杂度降低为 $O(n)$(n 为文件的个数). 在后续的研究中,Cao 等[5]、Li 等[6]、Sun 等[7]都是分别设计自己的索引结构来实现基于安全索引的可搜索加密机制,这种使用个人定义索引的主要缺点是每一个方案的索引结构会与其他方案的不匹配,数据拥有者需要针对不同方案生成不同的索引结构,当数据量很大时生成索引的代价较大. 因而,研究者普遍关注一种在文件检索系统中最受欢迎的索引结构——倒排索引. Curtmola 等[8]最先提出一个基于倒排索引的对称可搜索加密方案,该方案较以往方案在效率上有所提升,但还存在两个主要限制:一方面是该方案只支持单关键字搜索,对于现实应用中常用到的多关键字搜索不提供支持;另一方面是该方案在算法设计中并没有考虑到搜索模式和访问模式的保护问题,导致一旦关键字被搜索那么倒排索引的位置和内容都会泄露给云服务器.

从上述内容看出,目前的密文搜索方案大多只支持单关键字搜索,无法保护搜索模式且效率低下. 而实际应用场景中,用户通常需要进行高效的连接多关键字搜索并且希望搜索隐私得到更好的保护. 故针对现有公钥可搜索加密方案效率低下、表达能力弱且安全性不佳的问题,本文基于同态加密和私有集合交集技术,提出一种面向多关键字的可搜索加密方案(multi-keyword searchable encryption,MKSE). 方案以提高多关键字密文搜索效率为前提,以保护密文搜索的搜索模式和搜索内容为目标,围绕多关键字的精确匹配和数据安全性保障问题进行方案设计,具有重要的理论价值和应用前景.

1 预备知识

1.1 Paillier 同态加密

全同态加密[9]方案允许在不解密的情况下对密文数据进行任意的计算. 即给定消息(m_1,…,m_n)的密文,全同态加密方案能在不解密的情况下计算出 $f(m_1,\cdots,m_n)$ 的密文,其中 f 为任意的可计算函数.

Paillier 同态加密方案[10]是一个基于决定性组合剩余类问题假设的公钥加密方案,具有加法同态性以及一次乘法同态性,可表述如下:

1) 加法同态性:给定密文 $E(a_1)$,$E(a_2)$,那么可以通过计算 $E(a_1+a_2)=E(a_1)E(a_2)$ 得到 a_1+a_2 的密文;

2) 一次乘法同态性:给定密文 $E(a_1)$,可计算 $a_1\times a_2$ 的密文为 $E(a_1)^{a_2}$.

1.2 私有集合交集

私有集合交集[11-12](private set intersection,PSI)是对于集合交集的私有匹配协议,允许两个或更多实体来秘密地计算他们数据的交集,并且在计算过程中除了集合交集本身不泄露任何额外信息.

通过采用 Paillier 同态加密方法构建私有集合交集方案,可以保留 Paillier 同态加密方法的加法同态性和一次乘法同态性. 根据这两个属性可以得出一个推论:给定一个 k 阶多项式 P 的系数 $a_0,\cdots,a_k$ 的密文,和一个明文 y,可以直接计算 $P(y)$ 的密文.

应用这个推论,构建客户端 C 和服务器 S 之间的私有集合交集协议,其基础结构如下:首先,C 定义 1 个根为他的输入集合元素的多项式 P,$P(y)=(x_1-y)(x_2-y)\cdots(x_{k_c}-y)=\sum_{u=0}^{k_c}\alpha_u y^u$;之后 C 将上述多项式系数的密文发送给 S. S 对自己的每一个集合元素,都使用加密系统的同态属性来计算该多项式,然后 S 使用一个随机数 r 乘以每一个多项式结果来获得一个中间结果,最后再加上他的输入集合元素的密文,也就是说,S 计算 $\mathrm{Enc}(r\cdot P(y)+y)$. 对于两方输入的交集中的任何一个元素,$\mathrm{Enc}(r\cdot P(y)+y)$ 的结果是相应元素的值,而对于其他不属于两方交集的元素,结果是随机的.

2 MKSE 模型

2.1 MKSE 模型的架构

MKSE 模型包含三个实体:数据拥有者、用户和云服务器. 模型的架构如图 1 所示. 数据拥有者将本地文件集进行处理,并上传到云服务器. 对文件的处理包括将文件内容处理成密文形式,以及为其生成安全索引. 用户通过关键字进行文件搜索,首先用户向数据拥有者提交多个搜索关键字,之后利用数据拥有者返回给他的搜索陷门向云服务器提交搜索请求. 云服务器接收到搜索陷门并

执行搜索操作,返回给用户加密的文件标签集合. 最后,用户将加密的文件标签集合发送给数据拥有者解密,得到搜索关键字对应的文件标签信息.

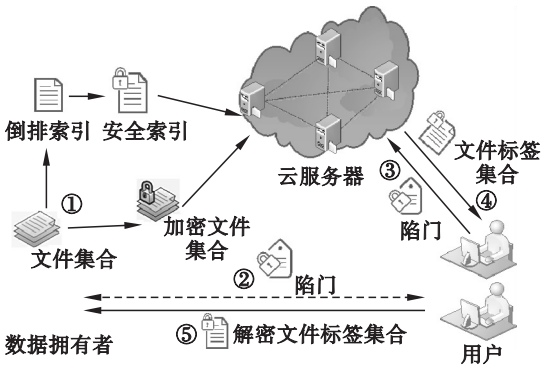

图 1　MKSE 模型的总体架构

Fig. 1　Structure of MKSE model

各个实体的功能介绍如下：

1）数据拥有者是与数据直接接触的一种特殊的可信赖用户. 他负责初始化方案中要使用到的密钥等;他要对自己的文件集合生成安全索引,并上传到云服务器;他接受用户选择的若干个关键字并生成搜索陷门.

2）用户是数据拥有者授权的可信实体. 他选择搜索的关键字并向数据拥有者申请搜索陷门;他使用该陷门向云服务器发起搜索请求;他将搜索结果发送给数据拥有者解密,得到关键字对应的文件标签信息.

3）由于不希望云服务器获得自身敏感数据,因而云服务器只提供存储服务并可以根据用户的请求执行特定的搜索操作.

2.2　形式化定义

MKSE 由 4 个算法组成,形式化定义为: MKSE = (Setup, IndexGen, Trapdoor, Search),具体描述如下：

1）MK←Setup(k)：初始化算法. 输入安全参数 k,输出主密钥 MK. MK 被保存在系统的整个生命周期中,并与被存储的数据相互独立.

2）($\boldsymbol{M}'_{\mathrm{D}}$,$\tilde{\boldsymbol{I}}$)←IndexGen(MK,$\boldsymbol{I}$)：安全索引生成算法. 输入主密钥 MK 和倒排索引 $\boldsymbol{I}$,输出安全索引 $\tilde{\boldsymbol{I}}$ 和字典矩阵 $\boldsymbol{M}'_{\mathrm{D}}$.

3）T←Trapdoor(MK,Q)：陷门生成算法. 输入为主密钥 MK 和搜索请求 Q,输出搜索陷门 T.

4）$P_{\mathrm{R}}(x)$←Search($\tilde{\boldsymbol{I}}$,T)：搜索算法. 输入安全索引 $\tilde{\boldsymbol{I}}$ 和搜索陷门 T,输出一个含有文件标签信息的结果多项式 $P_{\mathrm{R}}(x)$.

2.3　安全性定义

首先,给出 MKSE 方案的正确性定义. 在 MKSE 方案中,对于概率多项式次搜索,算法 Search 返回对应的搜索结果,搜索结果为所有包含查询关键字的文件,则称 MKSE 方案是正确的.

定理 1（正确性）　基于 Paillier 和 PSI 的多关键字可搜索加密方案返回所有包含查询关键字的文件.

本文假设数据拥有者使用块加密法如 AES 来加密文件内容,本文不再对文体本身安全性进行讨论. MKSE 方案应提供的隐私性保护包括：

1）索引隐私. 索引隐私的保护是双重的,首先,索引应该被加密,使得云服务器不能学习到索引的内容. 其次,通过分析安全索引,云服务器无法推论出关于文件的信息,包括是否一个文件包含某些关键字以及是否不同的文件包含一个相同的关键字.

2）陷门隐私. 陷门包含加密形式的查询信息,陷门隐私要求给定一个搜索陷门,云服务器不能从中学习到任何关于用户查询的信息,包括查询的内容,查询的关键字数量以及相同查询是否已经被搜索过.

给定关键字集合 Ω 和安全索引 $\tilde{\boldsymbol{I}}$,任意概率多项式时间(PPT)敌手 A 伪造一系列查询,A 始终不能区分算法 Search 返回的搜索结果. 下面通过安全性实验来形式化描述该方案的安全性.

Setup：

挑战者 C 创建一个关键字集合 Ω. C 从集合 Ω 对应的文件中挑选出一些子集合构成文件集合 $\boldsymbol{\Sigma}$. 首先挑战者 C 运行 Setup 生成密钥,然后执行 IndexGen 为 $\boldsymbol{\Sigma}$ 生成安全的倒排索引 $\tilde{\boldsymbol{I}}$. 最后,C 将关键字集合 Ω 和安全索引 $\tilde{\boldsymbol{I}}$ 发布给敌手 A.

Queries：

敌手 A 被允许向挑战者 C 请求对于查询 $Q \in \Omega$的陷门 T. A 可以在安全索引上使用 T 执行 Search,从而得到结果 P_{R}.

Challenge：

在某个时刻,敌手 A 选择两个非空查询 V_0, $V_1 \subset \Omega$,其中 $V_0 - V_1 \neq \phi$ 且 $V_1 - V_0 \neq \phi$. 然后将查询请求发送给挑战者 C.

在接收到 V_0 和 V_1 后,C 选择 $b \xleftarrow{R} \{0,1\}$,然后对 V_b 生成陷门 T_b. 允许 A 使用 T_b 生成对 T_b 的搜索结果 P_{R}.

敌手 A 的挑战就是判断 b. 挑战被发布后,敌手 A 可以继续请求陷门.

Response：

敌手 A 输出他关于 b 的猜想 b'.

在这个安全性实验中敌手 A 获胜的优势定义为:$\mathrm{Adv}_{\mathrm{A}} = |\Pr[b = b'] - 1/2|$.

如果没有多项式时间敌手能够以不可忽略的优势赢得上述的安全性实验,则方案是语义安全的.

定理 2(安全性) 如果方案所基于的加法同态加密方法是语义安全的,那么基于 Paillier 和 PSI 的多关键字可搜索加密方案是语义安全的.

3 MKSE 方案

3.1 系统初始化

MK←Setup(k):由数据拥有者执行. 首先,选择 2 个 k - bit 的素数 p 和 q,满足 $\gcd(pq,(p-1)(q-1))=1$. 计算 $n=pq$,并选取 $g \in \mathbf{Z}_{n^2}^*$,计算 $\mu = \left(\frac{g^2 \bmod n^2 - 1}{n}\right)^{-1} \bmod n$ 和 $\lambda = \mathrm{lcm}(p-1, q-1)$,从而得到 Paillier 同态加密算法的密钥对 $\mathrm{sk} = (\lambda, \mu)$,$\mathrm{pk} = (n, g)$. 之后,把私钥 sk、伪随机置换方法 f 和一个可逆矩阵 $\boldsymbol{M}$ 作为主密钥 MK,其中矩阵 $\boldsymbol{M}$ 的度由倒排索引关键字字典 d 的大小决定. 最后将公钥 pk 发布.

3.2 安全索引生成

$(\boldsymbol{M}'_{\mathrm{D}}, \tilde{\boldsymbol{I}}) \leftarrow \mathrm{IndexGen}(\mathrm{MK}, \boldsymbol{I})$:由数据拥有者执行. 本文假设,数据拥有者已经为自己的文件集合 $\boldsymbol{\Sigma}$ 构建好了倒排索引 $\boldsymbol{I} = (I_{W_1}, I_{W_2}, \cdots, I_{W_m})^{\mathrm{T}}$,一个倒排索引包括一个关键字字典 d 和针对每个关键字的文件列表 I_{W_i}. 本文不再对构建倒排索引的过程进行赘述. 安全索引生成算法的主要工作是将倒排索引加密成安全索引,算法的具体过程如下:

1) 为每个关键字 $W_i \in \boldsymbol{\Omega}$ 生成标签 $t_{W_i} = f(W_i)$,同时为每一个文件 $\delta_i \in \boldsymbol{\Sigma}$ 生成标签 $t_{\delta_i} = f(\delta_i)$. 定义关键字标签的集合为 $f(\boldsymbol{\Omega})$,文件标签的集合为 $f(\boldsymbol{\Sigma})$. 假设所有文件列表 I_{W_i} 的最大长度设置为 L,为 I_{W_i} 生成一系列的随机数 $R_i = \{r_j\}$,其中 $r_j \in \mathbf{Z}_n^*$,$r_j \notin f(\boldsymbol{\Omega})$,然后,将 R_i 填充到文件列表中. 为填充后的文件列表 I_{W_i} 生成一个多项式 $P_{W_i}(x) = \prod_{\delta_j \in I_{W_i}} (x - t_{\delta_j}) \prod_{r_j \in R_i} (x - r_j)$.

2) 计算多项式矢量 $\boldsymbol{I} = (P_{W_1}, P_{W_2}, \cdots, P_{W_m})^{\mathrm{T}}$.

3) 用多项式 P_{W_i} 本身的系数 $(a_L, \cdots, a_1, a_0)$ 来表示多项式,采用 Paillier 同态加密算法,使用其公钥 $\mathrm{pk} = (n, g)$ 来加密每一个多项式 P_{W_i} 的系数得到 $(\mathrm{Enc}(a_L), \cdots, \mathrm{Enc}(a_1), \mathrm{Enc}(a_0))$,将加密后的索引表示为 $\tilde{\boldsymbol{I}} = \mathrm{Enc}_{(n,g)}(\boldsymbol{I}')$.

4) 构建一个字典矩阵 $\boldsymbol{M}_{\mathrm{D}}$,形如:

$$\boldsymbol{M}_{\mathrm{D}} = \begin{bmatrix} t_{W_1}^m & t_{W_2}^m & \cdots & t_{W_m}^m \\ t_{W_1}^{m-1} & t_{W_2}^{m-1} & \cdots & t_{W_m}^{m-1} \\ \vdots & \vdots & \vdots & \vdots \\ t_{W_1} & t_{W_2} & \cdots & t_{W_m} \end{bmatrix}.$$

利用可逆矩阵 $\boldsymbol{M}$ 加密 $\boldsymbol{M}_{\mathrm{D}}$,得到 $\boldsymbol{M}'_{\mathrm{D}} = \boldsymbol{M} \cdot \boldsymbol{M}_{\mathrm{D}}$.

最后,将加密的字典矩阵 $\boldsymbol{M}'_{\mathrm{D}}$ 和安全索引多项式矢量 $\tilde{\boldsymbol{I}}$ 发送给云服务器.

3.3 陷门生成

$T \leftarrow \mathrm{Trapdoor}(\mathrm{MK}, Q)$:由数据拥有者执行. 为用户的多关键字搜索请求生成搜索陷门,算法的具体过程如下:

1) 当接收到一个搜索请求 Q,将构建多项式 $P_{\mathrm{Q}}(x) = \prod_{W_i \in \Omega}(x - t_{W_i}) / \prod_{W_i \in Q}(x - t_{W_i})$,为了隐藏陷门长度,数据拥有者向 $P_{\mathrm{Q}}(x)$ 中填充随机元组,得到

$$P'_{\mathrm{Q}}(x) = P_{\mathrm{Q}}(x) \prod_{q+1}^{m} (x - r_j),\ q = |Q|,\ r_j \notin f(\Omega).$$

2) 使用 $P'_{\mathrm{Q}}(x)$ 的系数 $(a_m, a_{m-1}, \cdots, a_1, a_0)$ 来表示 $P'_{\mathrm{Q}}(x)$. 计算 $T[1] = (a_m, a_{m-1}, \cdots, a_1) \cdot \boldsymbol{M}^{-1}$,$T[2] = \mathrm{Enc}_{(n,g)}(a_0)$. 最后,数据拥有者将二元组陷门 $T = (T[1], T[2])$ 返回给用户.

3.4 多关键字搜索

$P_{\mathrm{R}}(x) \leftarrow \mathrm{Search}(\tilde{\boldsymbol{I}}, T)$:由云服务器执行. 云服务器利用用户发送的陷门 $T = (T[1], T[2])$ 在存储的安全索引中进行搜索. 算法的具体过程如下:

1) 接收到陷门后,云服务器首先计算值 $\boldsymbol{V}$:

$$\boldsymbol{V} = T[1] \cdot \boldsymbol{M}'_{\mathrm{D}} = (v_1, v_2, \cdots, v_m).$$

2) 对于每一个 $v_i (i \in [m])$,云服务器计算 $v'_i = \mathrm{Enc}_{(n,g)}(v_i) +_h T[2]$. 其中 $+_h$ 是 Paillier 加密算法的同态加法. 然后所有的值构成一个矢量:$\boldsymbol{V}' = (v'_1, v'_2, \cdots, v'_m)$.

3) 计算 $P_{\mathrm{R}}(x) = \boldsymbol{V}' \cdot \tilde{\boldsymbol{I}}$,并将其返回给用户.

用户接收到 $P_{\mathrm{R}}(x)$ 之后,在数据拥有者的帮助下解密该多项式,之后对其进行因式分解并找到 $P_{\mathrm{R}}(x) = 0$ 的根,这些根就是搜索结果,即关键字对应的文件标签.

4　安全性证明和性能分析

4.1　定理 1 的正确性证明

根据 $P_{\mathrm{Q}}(x)$ 的定义可以得出查询陷门 $P'_{\mathrm{Q}}(x)=\prod_{W_i\in\Omega}(x-t_{W_i})\prod_{q+1}^{m}(x-r_j)//\prod_{W_i\in Q}(x-t_{W_i})$，从而可以看出 $P'_{\mathrm{Q}}(x)=0$ 的根就是不包括在查询中的所有关键字的标签，即 $\{x\mid x=t_{W_i},W_i\notin Q\}$.

因为 $v'_i=\mathrm{Enc}_{(n,g)}(v_i)+_h\mathrm{Enc}_{(n,g)}(a_0)$ 是 $P'_{\mathrm{Q}}(t_{W_i})$，$i\in[m]$ 的密文，而且对于关键字 $W_i\notin Q$，$P'_{\mathrm{Q}}(W_i)=0$，则，结果多项式 $P_{\mathrm{R}}(x)=\boldsymbol{V}'\cdot\tilde{\boldsymbol{I}}^{\mathrm{T}}=\sum_{W_j\in Q}v'_jP_{W_j}$ 只包含查询相关的文件标签，所以 $P_{\mathrm{R}}(x)$ 的根实际上是包含所有查询关键字的文件集合. 因此，用户将会从云服务器中获取到正确的文件标签集合.

4.2　定理 2 的安全性证明

MKSE 的安全性依赖于加法同态加密算法的语义安全. 如果 MKSE 所基于的加法同态加密算法即 Paillier 同态加密方法是语义安全的加密系统，那么 MKSE 是语义安全的.

假设多项式时间算法 A 可以以一个不可忽略的优势赢得 2.3 节中安全性实验，则可以利用 A 构建一个算法 B，B 可以破坏基于随机预言模型(ROM)加密算法的语义安全. B 可以访问预言机 O_f，在预言机中方法 f 要么是一个随机算法，要么是加法同态加密算法. 用 B 的计算代替该方案中的加密操作，然后通过 2.3 节安全性定义中给定的安全性实验来证明 MKSE 方案的安全性.

安全性实验的具体过程如下：

$\mathrm{Game}_{A,B}(k)$：

$(\Omega,\Sigma,\boldsymbol{I})\leftarrow B(k)$

$\mathrm{MK}\leftarrow\mathrm{Setup}(k)$

$(\boldsymbol{M}'_{\mathrm{D}},\tilde{\boldsymbol{I}})\leftarrow\mathrm{IndexGen}(\mathrm{MK},\boldsymbol{I})$

for $1\leqslant i\leqslant q$ do

$q_i\xleftarrow[\text{each}]{\text{one query time}}A(Q_1,Q_2,\cdots,Q_q)$

$T_{Q_i}\longleftarrow\mathrm{Trapdoor}(\mathrm{MK},Q_i)$

$P_{\mathrm{R}}(x)\longleftarrow\mathrm{Search}(\tilde{\boldsymbol{I}},T_{Q_i})$

$b\leftarrow A(V_0\in\Omega^*,V_1\in\Omega)\in\{0,1\}$

$T_{Q_b}\longleftarrow\mathrm{Trapdoor}(\mathrm{MK},V_b)$

$P_{\mathrm{R}}(x)\longleftarrow\mathrm{Search}(\tilde{\boldsymbol{I}},T_{Q_b})$

output b'.

A 输出 b'，作为他对 b 的猜想. 如果 A 输出 0，那么 B 猜想在 O_f 中的 f 是一个随机函数，表示为 $B_f=0$. 否则，B 猜想 f 是加密算法，表示为 $B_f=1$.

明显地，如果 f 是随机函数，有概率 $\Pr[B_f=0]=1/2$. 如果 f 是加密算法，那么 B 同 A 有相同概率输出 1. 因此 B 在从随机预言模型中区分语义安全加密算法同 A 赢得安全性实验，具有相同的优势. 然而，根据语义安全加密系统的定义，B 是不存在的. 因此，不存在具有不可忽略概率赢得安全性实验的算法 A.

另外讨论索引隐私和陷门隐私，MKSE 方案在生成安全索引 $\tilde{\boldsymbol{I}}$ 的过程中进行了随机填充，保证索引对应文件列表长度的一致性，并且采用 Paillier 加密方法保证了索引的机密性. 在陷门生成过程中，$T[1]$ 的系数是被加密的，在 $T[2]$ 中，虽然 a_0 是不加密的，但由于在陷门中添加了随机填充，a_0 每一次都是不同的，因此，一个明文查询每次都会被转换成不同的陷门，所以 MKSE 方案打破了陷门的可链接性，进而敌手无法从搜索操作中判断两个(或多个)加密的查询中是否使用了相同的关键词，所以 MKSE 保护了搜索模式.

4.3　效率分析

设 MKSE 方案中倒排索引的关键字个数为 n.

在初始化阶段，数据拥有者需要执行 2 个指数操作生成 Paillier 同态加密所需密钥，计算复杂度为 $O(1)$ 个指数操作.

在安全索引生成阶段，数据拥有者要对倒排索引进行加密，倒排索引包含 n 个文件列表，每个文件列表表示为 L 阶多项式，因为在加密操作时要用多项式系数表示多项式并对多项式系数进行加密，所以需要执行 $n\times L$ 个加密操作. 同时，生成 $\boldsymbol{M}'_{\mathrm{D}}$ 的过程需要 n^3 个乘法操作. 通常来讲，加密操作的计算复杂度最高，因此方案在生成安全索引时的计算复杂度为 $O(n)$ 个加密操作.

在陷门生成阶段，数据拥有者需要执行一个多项式除法操作生成 $P'_{\mathrm{Q}}(x)$，执行 n^2 个乘法操作生成 $T_{\mathrm{Q}}[1]$，执行 1 个指数操作生成 $T_{\mathrm{Q}}[2]$. 因此方案在陷门生成时的计算复杂度为 $O(1)$ 个指数操作和 $O(n^2)$ 个乘法操作.

在搜索阶段，云服务器执行 n^2 个乘法操作计算 $\boldsymbol{V}$，执行 n 个指数运算生成 $\boldsymbol{V}'$，并执行 n 个乘法操作进行同态求和. 计算复杂度为 $O(n)$ 个指数操作和 $O(n^2)$ 个乘法操作.

4.4　实验分析

本文对 MKSE 方案进行了实验分析. 实验使

用了 2015 年最新版本的 Enron 数据集 [http://www.cs.cmu.edu/~./enron/],在邮件集中选取 3 000 个邮件作为实验文件集合.

本文根据字典的大小对方案执行的三个阶段(索引生成、陷门生成、搜索)的耗时进行实验分析,所有记录的结果都是每组数据运行 20 次后的平均值. 本文按照字典大小的不同将测试数据分为 4 组:A 类字典(大小为 50),B 类字典(大小为 100),C 类字典(大小为 150)和 D 类字典(大小为 200). 测试过程中,每次搜索请求的关键字个数都为 4. 通过对以上 4 组数据的性能分析,得出如图 2 所示的索引生成、陷门生成以及搜索操作的耗时.

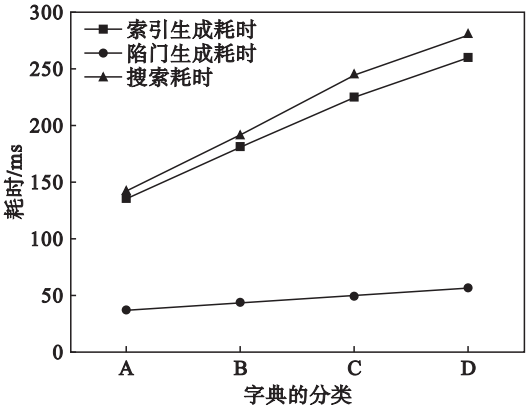

图 2 MKSE 算法性能

Fig. 2 Algorithm performance of MKSE

从图 2 可以看出,随着关键字字典大小的增大,索引生成和搜索过程的耗时会随着增加. 主要因为关键字字典增大的同时倒排索引也会变多,这样在索引生成和搜索过程中就需要相应进行更多的加密操作,因而耗时随之增加. 而在生成陷门过程中,字典的大小不会明显地影响陷门生成的耗时.

5 结 论

1) 提出一种基于同态加密和私有集合交集技术的多关键字可搜索加密方案 MKSE. 方案支持多关键字密文搜索,同时突破了现有方案只支持一次性搜索的限制,可以进行关键字重复搜索.

2) 在生成索引和搜索陷门的过程中引入了随机项,保护了索引隐私和陷门隐私,进而使得该方案能够有效地对搜索模式进行保护. 安全性分析表明该方案满足可搜索加密的语义安全.

3) 与其他基于双线性映射的公钥可搜索加密方案相比,该方案在整个搜索过程中仅仅使用了乘法和指数运算,因而具有较小的计算开销.

参考文献:

[1] 董晓蕾,周俊,曹珍富. 可搜索加密研究进展[J]. 计算机研究与发展,2017,54(10):2107-2120.
(Dong Xiao-lei, Zhou Jun, Cao Zhen-fu. Research advances on secure searchable encryption [J]. *Journal of Computer Research and Development*, 2017, 54(10): 2107-2120.)

[2] Song D X, Wagner D, Perrig A. Practical techniques for searches on encrypted data [C]// IEEE Symposium on Security and Privacy. Washington D C: IEEE Computer Society, 2000: 44.

[3] Dan B, Crescenzo G D, Ostrovsky R, et al. Public key encryption with keyword search[M]. Berlin: Springer, 2004: 506-522.

[4] Goh E J. Secure indexes [EB/OL]. (2004-03-16) [2017-12-17]. https://eprint.iacr.org/2003/216.pdf.

[5] Cao N, Wang C, Li M, et al. Privacy-preserving multi-keyword ranked search over encrypted cloud data[J]. *IEEE Transactions on Parallel & Distributed Systems*, 2013, 25(1): 222-233.

[6] Li M, Yu S, Lou W, et al. Toward privacy-assured cloud data services with flexible search functionalities [C]// International Conference on Distributed Computing Systems Workshops. Washington D C: IEEE, 2012: 466-470.

[7] Sun W, Yu S, Lou W, et al. Protecting your right: verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud [J]. *IEEE Transactions on Parallel and Distributed Systems*, 2014, 27(4): 1187-1198.

[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] Gentry C. Fully homomorphic encryption using ideal lattices [C]// Proceedings of the Annual ACM Symposium on Theory of Computing. New York: ACM, 2009: 169-178.

[10] Paillier P. Public-key cryptosystems based on composite degree residuosity classes[C]// International Conference on Theory and Application of Cryptographic Techniques. Berlin: Springer-Verlag, 1999: 223-238.

[11] Freedman M J, Nissim K, Pinkas B. Efficient private matching and set intersection[C]// Advances in Cryptology-EUROCRYPT 2004. Berlin: Springer, 2004: 1-19.

[12] Kolesnikov V, Matania N, Pinkas B, et al. Practical multi-party private set intersection from symmetric-key techniques [C]// ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2017: 1257-1272.