云计算中保护数据隐私的快速多关键词语义排序搜索方案

杨 旸[1),2),3)] 刘 佳[1),2)] 蔡圣暐[1),2)] 杨书略[2),4)]

[1)](福州大学数学与计算机科学学院 福州 350108)

[2)](网络系统信息安全福建省高校重点实验室(福州大学) 福州 350108)

[3)](福建省信息处理与智能控制重点实验室(闽江学院) 福州 350121)

[4)](福州大学物理与信息工程学院 福州 350108)

摘 要 可搜索加密技术主要解决在云服务器不完全可信的情况下,支持用户在密文上进行搜索.该文提出了一种快速的多关键词语义排序搜索方案.首先,该文首次将域加权评分的概念引入文档的评分当中,对标题、摘要等不同域中的关键词赋予不同的权重加以区分.其次,对检索关键词进行语义拓展,计算语义相似度,将语义相似度、域加权评分和相关度分数三者结合,构造了更加准确的文档索引.然后,针对现有的 MRSE(Multi-keyword Ranked Search over Encrypted cloud data)方案效率不高的缺陷,将创建的文档向量分块,生成维数较小的标记向量.通过对文档标记向量和查询标记向量的匹配,有效地过滤了大量的无关文档,减少了计算文档相关度分数和排序的时间,提高了搜索的效率.最后,在加密文档向量时,将文档向量分段,每一段与对应维度的矩阵相乘,使得构建索引的时间减少,进一步提高了方案的效率.理论分析和实验结果表明:该方案实现了快速的多关键词语义模糊排序搜索,在保障数据隐私安全的同时,有效地提高了检索效率,减少了创建索引的时间,并返回更加满足用户需求的排序结果.

关键词 云计算;可搜索加密;语义相似度;域加权评分;快速 KNN(K-Nearest Neighbor)算法

中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2018.01346

Fast Multi-Keyword Semantic Ranked Search in Cloud Computing

YANG Yang[1),2),3)] LIU Jia[1),2)] CAI Sheng-Wei[1),2)] YANG Shu-Lue[2),4)]

[1)](*College of Mathematics and Computer Science, Fuzhou University, Fuzhou* 350108)

[2)](*University Key Laboratory of Information Security of Network Systems (Fuzhou University), Fujian Province, Fuzhou* 350108)

[3)](*Fujian Provincial Key Laboratory of Information Processing and Intelligent Control (Minjiang University), Fuzhou* 350121)

[4)](*College of Physics and Information Engineering, Fuzhou University, Fuzhou* 350108)

Abstract With the growing popularity of cloud computing, the individuals and corporations are motivated to outsource their data to the public cloud server for economic savings and accessing to data at any time, any place, and with any device. Note that the outsourced data may be private and contain sensitive information, such as financial trading files, electronic health records, private secret logs and individual sensitive multimedia data. To minimize the probability of the risk of sensitive data leakage, it is desirable for the data owners to encrypt sensitive data before sending them to cloud. However, it also hinders the usability of outsourced data, such as data retrieval operation. Searchable encryption (SE) technology is an important approach to deal with this problem, which enables the users to search over encrypted data to realize effective data

收稿日期:2016-11-28;在线出版日期:2017-05-31.本课题得到国家自然科学基金(61402112,61472307,61472309,61303198)、福建省教育厅科技项目(JA12028)、闽江学院福建省信息处理与智能控制重点实验室开放课题(MJUKF201734)、福建省重大区域产业项目(2014H4015)、福建省重大科技项目(2015H6013)资助.**杨 旸**,女,1984 年生,博士,副教授,主要研究方向为信息安全和隐私.E-mail:yang.yang.research@gmail.com.**刘 佳**(通信作者),女,1993 年生,硕士研究生,主要研究方向为可搜索加密.E-mail:1142101626@qq.com.**蔡圣暐**,男,1993 年生,硕士研究生,主要研究方向为可搜索加密.**杨书略**,男,1990 年生,硕士研究生,主要研究方向为可搜索加密.

utilization. In searchable encryption schemes, the cloud storage server is assumed honest-but-curious (or say, semi-trusted), who is honest to execute the required storage and retrieval operations, but also curious to discover the plaintext keyword or file of users. The security requirement of SE should guarantee that only authorized users can decrypt encrypted data with decryption keys and then obtain plaintext files. In recent years, diverse SE schemes are proposed, which pay attention to both privacy and practicability of the system. However, most of the existing multi-keyword searchable encryption schemes have neither taken into consideration the location information of the keywords nor measured the similarity of the synonym keywords. At the same time, the search efficiency is low and the index construction time is too long. In this paper, we propose a fast multi-keyword semantic ranked search scheme. Firstly, for the first time, the concept of weighted domain scoring is introduced to searchable encryption to calculate the document relevance scores. The keywords in different domains (title, abstract, etc.) are measured by different weighted domain score. Secondly, the retrieved keywords are semantically expanded to their synonym sets and the semantic similarities of the synonyms are calculated. Combining the semantic similarity, the weighted domain score and the relevance scores, we construct the encrypted document index with higher accuracy. To improve the efficiency of MRSE (multi-keyword ranked search over encrypted cloud data), we partition the document index vector into several pieces and generate mark vector according to these pieces. Comparing the document mark vector and the query mark vector, we effectively filter a large number of irrelevant documents. The time for calculating the relevance scores and ranking is greatly reduced. Finally, document index vectors are partitioned into several sub-vectors, which are encrypted by the matrices with smaller dimensions. The method greatly reduces the computation overhead of generating encrypted indices and further improves the system efficiency. Theoretical analysis and experimental results demonstrate that the proposed scheme achieves the multi-keyword semantic ranked search with high efficiency. It improves the retrieval efficiency, reduces the encrypted index generation time and returns more accurate ranking results. It also guarantees the privacy and security of data. Both the basic and enhanced schemes proposed in this paper are proved secure in known background model.

Keywords　cloud computing; searchable encryption; semantic similarity; weighted domain scoring; fast K-nearest neighbor algorithm

1　引　言

随着云计算技术[1]的飞速发展，敏感数据被越来越多的存储到云中，如电子邮件、私人聊天记录和财务报表等[2]. 云服务器提供了高质量的数据存储服务，将数据上传到云端，可减少用户的数据存储和维护开销. 但是数据拥有者和云服务器不在同一个信任域中会使外包数据处于危险之中，为了保护用户的隐私安全，将数据加密后再上传到云端是一种常见的解决方法. 然而数据经过加密后不再具有原有的特性，当用户需要某些数据时，无法直接在密文中分辨出所需要的数据，在数据量很小的情况下，可以将所有的密文数据下载至本地，解密后在明文中搜索自己想要的数据. 然而随着云端数据规模的急剧增长，这种浪费了大量时间开销与带宽功耗的做法显然已经不能满足用户的实际需求，因此，如何在大量密文中搜索到所需要的文档成为了一个难题.

为了更好的解决密文中检索的问题，Song 等人[3]率先开始进行可搜索加密技术的研究. Chang 等人[4]为了改善搜索效率，为每一篇文档构建对应的索引来实现有效的可搜索加密方案. Cao 等人[5]为每篇文档创建文档向量，利用可逆矩阵与文档向量相乘来加密文档向量，实现了多关键词的排序搜索. 但是这些方案要求用户输入的关键词必须与预定义的关键词完全匹配，才能返回搜索结果，这使得搜索方案具有较大局限性. 为了改善用户的搜索体验，使得用户在输入与预定义的关键词不完全匹配

时,也能找到相关文档,Li 等人[6]实现了模糊搜索方案.然而,现有的模糊搜索方案中,很多只考虑了关键词字符上的模糊,而忽视了关键词语义上的模糊.

在实际应用中,存在大量的同义词现象,用户不一定能够输入关键词的所有同义词来查询所需的文档,这导致了搜索结果的不精确,达不到用户的要求.例如:用户输入 search,精确搜索或者模糊搜索的方案只能返回包含关键词 search 的文档,但实际包含关键词 retrieve 和 seek 的文档可能也满足用户的搜索需求,因此,支持语义的搜索方案更能符合当下用户的智能搜索要求.

现有的多关键词排序搜索方案存在以下缺陷:

(1)未考虑关键词位置信息.关键词出现在一篇文档的标题和正文的重要性有很大差别,但现有方案都只是把一整篇文档视为一个域,关键词出现在标题和正文并没有区分,使得构造出的索引不能准确地反映出关键词在文章中的权重,从而导致了搜索结果的不精确.

(2)未考虑语义拓展后关键词的相似度.例如,用户输入搜索关键词 search,通过语义拓展得到了语义相关词 retrieve 和 seek.在语义相似度计算中,search 的语义相似度最高(因为 search 就是查询关键词本身),而 retrieve 和 seek 的语义相似度则相对较低,本应赋予它们不同的权重,提高搜索精确度.然而在现有方案中,search、retrieve 和 seek 却被当做 search 的语义相关词,赋予了相同的权重,降低了搜索的精确性.特别是在拓展出的语义相关词数量较多且语义相似度差异较大时,赋予它们相同的权重,就会使得搜索结果不能很好的满足用户的需求.

(3)搜索效率不够高.目前多关键词的排序方案大多采用经典的 MRSE(Multi-keyword Ranked Search over Encrypted cloud data)算法[5],创建的字典集一般很大,向量空间模型构造出的比特串一般非常长,而一般一篇文档只会包含字典集中少量的关键词.在检索时,云服务器无法知道哪些是相关文档,因而要对所有的文档进行相似度分数的计算和排序,消耗了大量的时间.

在按需付费的云存储环境下,如果将用户的搜索结果全部返回,这种浪费了大量时间开销与带宽功耗的做法显然已经不能满足用户的实际需求,应采用更加合理化的排序方案,返回给用户最相关的文档.

为了进一步改善多关键词搜索结果的排序效果,提高整体方案的效率,返回更符合用户搜索需求的相关文档,本文提出了一种快速的多关键词语义排序搜索方案,主要贡献如下:

(1)多关键词语义模糊搜索.本文提出了支持语义拓展的多关键词排序可搜索加密方案,首先对用户输入的查询关键词进行语义拓展,并计算查询词与拓展词之间的语义相似度,将语义相似度分数引入到文档的评分当中.当授权用户希望搜索到查询关键词语义相关的文档,或者由于各种原因无法输入准确的关键词时,本方案也可以匹配到语义相关的文档并返回给授权用户,实现了语义模糊检索,满足用户的搜索需求.

(2)引入域加权评分.本文引入了域加权评分,对处于文档不同域中的关键词赋予不同的权重,将语义相似度、域加权评分和相关度分数三者结合,构造了更加准确的文档索引.

(3)提高搜索效率.可搜索加密方案创建的字典集一般很大,这使得 MRSE 方案[5]中创建的文档向量的维度通常很大.在检索时,云服务器无法知道哪些是相关文档,因而要对所有文档进行相似度分数的计算和排序,浪费了大量的时间.因此,本方案分别对文档向量和查询向量进行分块,生成维数较小的文档标记向量和查询标记向量.通过文档标记向量和查询标记向量的匹配,快速过滤掉大量无关文档,减少了计算文档相似度分数的时间和排序的时间,提高了搜索的效率.

(4)减少创建索引时间.MRSE 方案[5]中创建的文档向量的维度通常很大,所以方案构建索引的时间主要花费在文档向量和矩阵的相乘上.本文将文档向量分段,将每一段分别与维度大大减小的矩阵相乘,这使得此方案的索引构建时间大大减少.在本文实验中,当文档数量为 5000,关键词数为 3000 时,本方案索引的创建时间大约为 MRSE 方案的 1/2.随着文档数和关键词数的不断增多,本方案的优势会更加显著.

(5)实验验证方案可行性.本文实验在真实数据集上进行.实验结果表明本方案实现了快速的多关键词语义排序搜索,在保障数据隐私的前提下,不仅有效地改善了搜索效率,缩短了创建索引的时间,而且返回了更加满足用户需求的搜索结果.

2 相关工作

在 Song 等人[3]提出通过密钥流加密数据并

实现可搜索加密方案后，Cao 等人[5]通过使用向量空间模型[7]为每篇文档创建文档向量，采用 KNN (K-Nearest Neighbor)方法[8]将可逆矩阵与文档向量相乘来加密文档向量，通过文档向量和查询向量计算内积得到相关度分数，实现多关键词的排序搜索. Xu 等人[9]利用保序加密[10-11]后的相关度分数进行排序，实现了一种支持两轮排序的多关键词排序搜索方案，提高了排序结果的精确性. Yang 等人[12]提出了支持动态更新的快速多关键词搜索方案. Li 等人[13-14]提出了支持布尔查询的可搜索加密方案，将执行“或”、“与”、“非”查询词的权重设置为三组正数递增序列 a_i, b_j, c_k，并且序列间满足超递增，即 $\sum a_i < b_1$，$\sum a_i + \sum b_j < c_1$，接着将执行“非”查询词的权重 c_k 置换成对应的相反数 $-c_k$，则相关度分数大于 0 的文档即为满足布尔查询的相关文档. 但是上述方案没有考虑到文档向量的高度稀疏性，一篇文档中往往只会出现少量的关键词，因此文档向量中会出现大量的 0，在检索时，云服务器无法知道哪些是相关文档，因而要对所有的文档进行相似度分数的计算和排序，浪费了大量的时间，降低了方案的查询效率.

但是以上方案要求用户输入的关键词必须与预定义的关键词完全匹配才能返回搜索结果. 因此，Chuah 等人[15]提出了基于 BedTree 的多关键词模糊搜索方案，提高了搜索效率. Wang 等人[16]和 Fu 等人[17]将局部敏感哈希[18]和安全 KNN 方法(K-Nearest Neighbor)[8]结合，实现了一种新型的多关键词模糊搜索方案. Wang 等人[19]将局部敏感哈希[18]、布隆过滤器[20]和保序加密[10-11]方法结合，实现了支持范围查询的多关键词模糊排序搜索方案. 但是以上的模糊搜索方案仅仅考虑了关键词字符上的相似，并没有考虑到关键词语义上的相似. 因此，Fu 等人[21]对文档关键词进行同义词拓展，通过计算内积的方式，实现了支持同义词查询的多关键词排序搜索方案. Xia 等人[22]为文档集创建倒排索引，利用语义库扩展查询关键词，并且通过一对多保序加密函数对相关度分数进行加密，实现了多关键词语义排序搜索方案. 然而，这些语义模糊搜索方案没有将语义相似度参与到文档的评分中，并且忽略了不同域[23]中的关键词的权重差异.

3　问题描述

3.1　系统模型

本文的系统模型如图 1 所示：

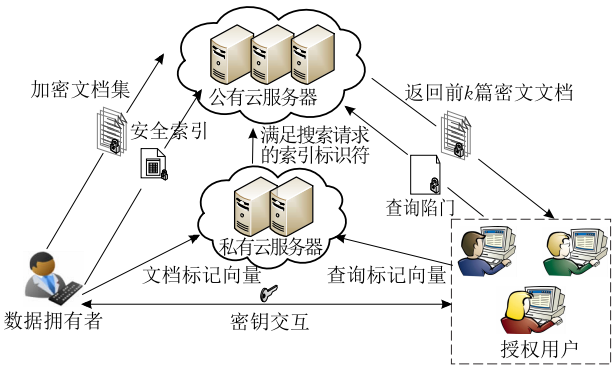


图 1　系统模型

(1) 数据拥有者. 首先从明文文档集中提取关键词集合 $W=(w_1, w_2, \cdots, w_n)$，根据关键词集合构造出安全索引和文档标记向量，然后将明文文档集合 $F=(f_1, f_2, \cdots, f_m)$ 加密成密文 $C=(c_1, c_2, \cdots, c_m)$，最后将文档标记向量发送给私有云服务器，将加密后的文档集 C 和安全索引一起上传到公有云服务器上.

(2) 授权用户. 可以根据需要输入感兴趣的关键词进行搜索. 授权用户对搜索关键词 $\Gamma=(q_1, q_2, \cdots, q_\eta)$ 进行语义扩展，得到语义扩展集合 $\Lambda=(q_1, q_2, \cdots, q_\eta, \delta_1, \cdots, \delta_\sigma)$. 查询标记向量根据语义扩展集合 Λ 得到. 然后使用数据拥有者提供的密钥，对语义扩展集合 Λ 构造相应的陷门 T_Q，之后将查询标记向量提交给私有云服务器，将陷门提交给公有云服务器完成搜索请求.

(3) 私有云服务器. 主要负责存储数据拥有者上传的文档标记向量，执行授权用户的查询请求. 在接收到授权用户发送的查询标记向量后，将查询标记向量与文档标记向量进行匹配，将可能满足用户查询请求的安全索引标识符集合 $SID_\in=(\cdots, sid_i, \cdots, sid_j, \cdots, sid_z, \cdots)$ 发送给公有云服务器，让公有云服务器进行进一步的查询操作.

(4) 公有云服务器. 主要负责存储数据拥有者外包的加密文档和安全索引并且支持授权用户进行搜索. 在接收到授权用户上传的陷门 T_Q 以及私有云服务器发送的可能满足搜索请求的候选安全索引标识符集合后，公有云服务器通过候选安全索引标识符集合 $SID_\in=(\cdots, sid_i, \cdots, sid_j, \cdots, sid_z, \cdots)$ 找到对应的安全索引集合 $\mathcal{I}_\in=(\cdots, I_i, \cdots, I_j, \cdots, I_z, \cdots)$，用安全索引与陷门 T_Q 计算文档的相似度分数. 为了提高查询的准确性和减少带宽开销，公有云服务器需对查询结果排序，返回最相关的前 k 篇文档给授权用户.

最后,授权用户用加密文件密钥对返回的密文文档进行解密.

3.2 模型威胁分析

在本文中,私有云服务器假定是"诚实且可信"的,不会泄露任何相关的标记向量信息,并且会诚实地执行授权用户提交的搜索请求,也会将搜索结果如实的发送给公有云服务器.

公有云服务器假定是"诚实但是好奇"的,不会删除数据拥有者上传的数据,会如实地根据授权用户上传的陷门进行查询,也会按照私有云服务器发送的索引标识符找到对应的安全索引,完成授权用户的搜索请求后会照实地返回结果.但是公有云服务器又是好奇的,它可能会采取一些措施来获取一些其余的信息.本文主要探讨的是已知密文模型,假定公有云服务器只知道数据拥有者上传的加密索引、密文文档和授权用户提交的查询陷门.

3.3 主要符号说明

表 1 为本文使用的主要符号说明.

表 1 主要符号说明

符号	说明
$F=(f_1,f_2,\cdots,f_m)$	明文文档集合
$C=(c_1,c_2,\cdots,c_m)$	密文文档集合
$W=(w_1,w_2,\cdots,w_n)$	关键词集合
$\mathcal{I}=(I_1,I_2,\cdots,I_m)$	安全索引集合
$\mathcal{I}_\in=(\cdots,I_i,\cdots,I_j,\cdots,I_z,\cdots)$	候选安全索引集合
$SID=(sid_1,sid_2,\cdots,sid_m)$	索引标识符集合
$SID_\in=(\cdots,sid_i,\cdots,sid_j,\cdots,sid_z,\cdots)$	候选索引标识符
T_Q	查询关键词陷门
Z	域加权得分
SC	语义相似度
$\boldsymbol{B}=(b_1,b_2,\cdots,b_m)$	文档标记向量集合
$\boldsymbol{\varphi}$	查询标记向量

4 预备知识

4.1 域加权评分

域(zone)可以由任意的、数目无限制的文本构成,例如,通常可以把文档的标题、摘要和正文看作不同的域[23].不同域中的关键词通常具有不同的重要性(标题>摘要>正文).若每篇文档有 l 个域,其对应的权重系数分别是 $g_1,\cdots,g_l\in[0,1]$,它们满足:

$$\sum_{i=1}^{l} g_i=1 \tag{1}$$

令 s_i 为关键词在文档第 i 个域的匹配得分(1 和 0 分别表示是否匹配),域加权评分定义为

$$Z=\sum_{i=1}^{l} g_i s_i \tag{2}$$

4.2 WordNet

WordNet 是一种基于认知语言学的英语词汇语义网①.传统词典主要是根据词形来组织词汇信息,忽视了词汇之间的语义关系.而 WordNet 则是根据单词的意义来组织词汇信息的.WordNet 中将英语分为四种词性,分别为名词、形容词、动词和副词.WordNet 中涉及的语义关系包括:整体与部分、同义、反义、上下位义等.

图 2 是名词网络的一部分,其中涉及的语义关系分别为:下位义、反义、部分义(整体与部分).

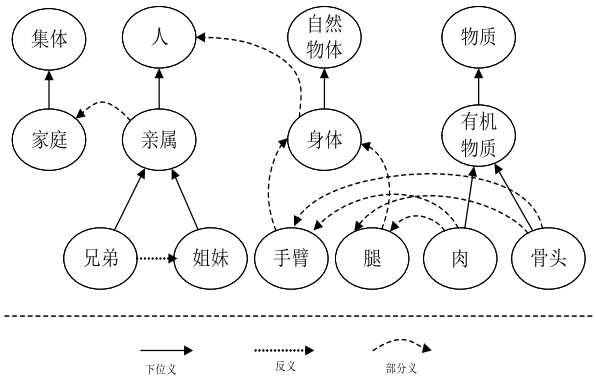


图 2 WordNet 名词网络的部分

4.3 语义相似度计算方法

本文采用基于信息内容的 Resnik 算法[24]来计算语义相似度.Resnik 提出两个概念最近公共祖先节点(Least Common Ancestors,LCA)决定了两个概念的相似度.即将两个概念相似度的计算转换为求两个概念 LCA 的信息内容.

图 3 为 WordNet 分类树[24]的子树,该子树的叶节点为 hydrogen 和 gold. Chemical element 和 metal 为 hydrogen 和 gold 的最近公共祖先节点,substance 为 chemical element 和 metal 的最近公共祖先节点.Substance 为该子树的根节点,显然在

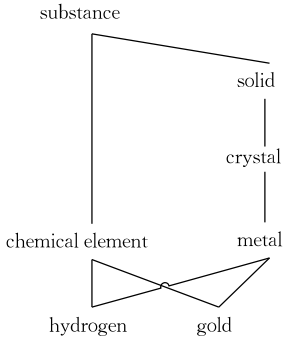


图 3 WordNet 分类树部分

① WordNet—A lexical database for English,. http://wordnet.princeton.edu/wordnet/documentation 2016,3,27

chemical element 和 metal 的上层，因此，hydrogen 和 gold 的相似度比 chemical element 和 metal 的要大.

Resnik 算法计算两个概念 c_1 和 c_2 相似度的公式如下：

$$sim(c_1,c_2)=-\log p(lso(c_1,c_2))=\mathrm{IC}(lso(c_1,c_2)) \quad (3)$$

其中 $lso(c_1,c_2)$ 表示概念 c_1 和 c_2 在 WordNet 分类树中的最近公共祖先节点.

在信息理论中，信息内容(Information Content，IC)表示为 $-\log p(c)$，$p(c)$ 是 WordNet 语料库中出现概念 c 的名词的概率，其计算方法如下：

$$p(c)=\frac{freq(c)}{N} \quad (4)$$

其中，N 表示 WordNet 语料库中名词的个数，$freq(c)$ 表示语料库中包含概念 c 的单词个数，其计算公式如下：

$$freq(c)=\sum_{n\in words(c)} count(n) \quad (5)$$

其中，$words(c)$ 表示包含概念 c 的单词集合.

Resnik 算法计算两个单词 w_1 和 w_2 相似度的公式如下：

$$sim(w_1,w_2)=\max_{c_1\in s(w_1),c_2\in s(w_2)}[sim(c_1,c_2)] \quad (6)$$

其中，$s(w_1)$ 和 $s(w_2)$ 分别表示单词 w_1 和 w_2 包含的概念集合.

4.4　相关度分数

本文采用 $tf-idf$ 权值计算方法[23]计算关键词的相关度分数. $tf_{t,f}$(词项频率)表示关键词 t 在文档 f 中出现的次数，idf_t(逆文档频率)表示关键词 t 在文档集合中的罕见程度，其定义如下：

$$idf_t=\log\frac{N}{df_t} \quad (7)$$

其中 N 表示所有文档的篇数，df_t 为包含关键词 t 的文档篇数.

对于每篇文档中的每个关键词 t，可以将其 tf 和 idf 组合计算其相关度分数：

$$tf-idf=tf_{t,f}\times idf_t \quad (8)$$

4.5　向量空间模型

向量空间模型(Vector Space Model)[7]是文档集在同一向量空间中的表示. 每篇文档对应一个文档向量，向量的维度等于关键词集合的长度，向量的每一维对应一个关键词并且值等于该维对应关键词的权重. 用户的查询也看成同一个空间下的向量，称为查询向量. 向量每一维对应的关键词和文档向量保持一致，向量维度和文档向量相同. 查询和每个文档的相关度分数等于文档向量和查询向量的内积的值.

5　方案构建

5.1　基础方案

在多关键词排序可搜索加密方案 MRSE[5]中，创建的字典集一般很大，向量空间模型构造出的比特串一般非常长，考虑到一般一篇文档只会包含字典集中极少量的关键词，即其向量中会包含大量的 0. 针对这一特点，对向量做一些处理来提高检索效率，具体步骤如下：

(1) Setup：

① 抽取关键词. 数据拥有者从明文文档集合 $F=(f_1,f_2,\cdots,f_m)$ 中抽取关键词，得到关键词集合 $W=(w_1,w_2,\cdots,w_n)$.

② 生成密钥. 数据拥有者随机产生一个 $(n+2)$ 比特的向量 $\boldsymbol{S}$ 和两个 $(n+2)\times(n+2)$ 维的可逆矩阵 $\{\boldsymbol{M}_1,\boldsymbol{M}_2\}$，密钥 SK 由四元组 $\{\boldsymbol{S},\boldsymbol{M}_1,\boldsymbol{M}_2,u\}$ 组成，u 是一个正整数并且 $u\mid n$. 接着，数据拥有者生成一个加密文档的密钥 sk.

(2) Index(F,SK). 本方案创建索引过程如图 4 所示，具体操作步骤如下：

① 计算词频权重 $wf_{t,f}$ 和逆文档频率 idf_t. 假定一个关键词在某篇文档中出现的次数是另一个关键词的 30 倍. 在 $tf-idf$ 权值计算方法中，由于关键词的相关度分数随词项频率线性增长，则此关键词的权重是另一个关键词的 30 倍. 然而，一个关键词携带的信息重要性不与权重成线性关系，因此，这种权值计算方法并不准确. 本文采用 tf 亚线性尺度变换方法[23]计算词频权重：

$$wf_{t,f}\begin{cases}1+\log tf_{t,f}, & tf_{t,f}>0\\ 0, & tf_{t,f}=0\end{cases} \quad (9)$$

采用式(7)的方式计算关键词的逆文档频率 idf_t. 则关键词在文档中的相关度分数的计算公式如下：

$$wf-idf=wf_{t,f}\times idf_t \quad (10)$$

② 计算域加权得分 Z. 在本方案中，设定每篇文档有 3 个域，分别为标题、摘要和正文. 其对应的权重系数分别为 g_1,g_2,g_3，满足式(1)且 $g_1>g_2>g_3$. s_i 为关键词在某篇文档的第 i 个域的匹配得分，$s_i=1$ 表示匹配，$s_i=0$ 表示未匹配. 接着根据式(2)计算该关键词的域加权得分.

例如，在某篇文档中，关键词 cloud 出现在标题

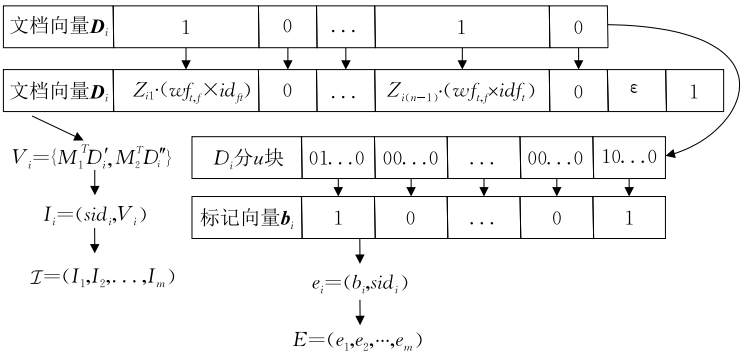


图 4 创建索引过程

和正文中,未在摘要中出现,则 3 个域的得分为 $s_1=1$, $s_2=0$, $s_3=1$,那么,关键词 cloud 在该文档中的域加权得分为 $Z=g_1\times s_1+g_2\times s_2+g_3\times s_3=g_1+g_3$.

③加密文档向量. 基于向量空间模型,数据拥有者为每篇文档 f_i 生成一个文档向量 $\boldsymbol{D}_i$,若文档 f_i 中包含关键词 w_j,则 $\boldsymbol{D}_i[j]=1$,否则 $\boldsymbol{D}_i[j]=0$. 接着将文档向量 $\boldsymbol{D}_i$ 均分为 u 块,如果某个块全为 0,则标记值 $bb_s=0$,否则 $bb_s=1$,得到文档标记向量 $\boldsymbol{b}_i=(bb_1,bb_2,\cdots,bb_u)$, $\boldsymbol{e}_i=(\boldsymbol{b}_i,sid_i)$. 将 $\boldsymbol{D}_i[j]$ 中 1 的值置为 $(Z_{ij}\cdot(wf_{t,f}\times idf_t))$ 后,对 $\boldsymbol{D}_i$ 进行维度扩展,其中第 $(n+1)$ 位设置为一个随机数 ε,第 $(n+2)$ 位设置成 1,那么 $\boldsymbol{D}_i$ 表示为 $(\boldsymbol{D}_i,\varepsilon,1)$. 接着,根据指示向量 $\boldsymbol{S}$ 将文档向量 $\boldsymbol{D}_i$ 分裂成 $\boldsymbol{D}_i'$ 和 $\boldsymbol{D}_i''$:若 $\boldsymbol{S}$ 的第 j 位为 0,则令 $\boldsymbol{D}_i'[j]=\boldsymbol{D}_i''[j]=\boldsymbol{D}_i[j]$;若 $\boldsymbol{S}$ 的第 j 为 1,则将 $\boldsymbol{D}_i'[j]$ 和 $\boldsymbol{D}_i''[j]$ 设为随机值,满足公式 $\boldsymbol{D}_i'[j]+\boldsymbol{D}_i''[j]=\boldsymbol{D}_i[j]$. 然后,用密钥 SK 对 $\boldsymbol{D}_i'$ 和 $\boldsymbol{D}_i''$ 加密,得到 $V_i=\{\boldsymbol{M}_1^{\mathrm{T}}\boldsymbol{D}_i',\boldsymbol{M}_2^{\mathrm{T}}\boldsymbol{D}_i''\}$, $I_i=(sid_i,V_i)$. 最后,将 $\boldsymbol{E}=(e_1,e_2,\cdots,e_m)$ 发送给私有云服务器,$\mathcal{I}=(I_1,I_2,\cdots,I_m)$ 上传给公有云服务器.

(3) Encrypt(F,sk). 数据拥有者使用对称加密算法对文档集合 $F=(f_1,f_2,\cdots,f_m)$ 进行加密,得到密文集合 $C=(c_1,c_2,\cdots,c_m)$ 并上传给公有云服务器.

(4) Trapdoor(Γ,SK):本方案陷门生成过程如图 5 所示,具体操作步骤如下:

①关键词语义拓展. 当授权用户进行搜索时,首先输入 η 个搜索关键词 $\Gamma=(q_1,q_2,\cdots,q_s,\cdots,q_\eta)$. 接着,基于 WordNet 语料库进行语义扩展.

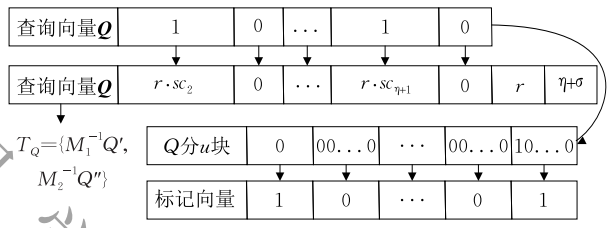


图 5 陷门生成过程

②语义相似度计算. 如图 6 为本方案的查询拓展过程,将语义拓展出的关键词用基于信息内容的 Resnik[24] 算法,计算出原单词 q_s 和拓展词之间的语义相似度并排序,选取最相关的前 σ 个拓展词作为最终拓展词,得到语义拓展集合 $\Lambda=(q_1,q_2,\cdots,q_\eta,\delta_1,\cdots,\delta_\sigma)$ 及其对应的语义相似度分数 $SC=(sc_1,sc_2,\cdots,sc_\eta,sc_{\eta+1},\cdots,sc_{\eta+\sigma})$.

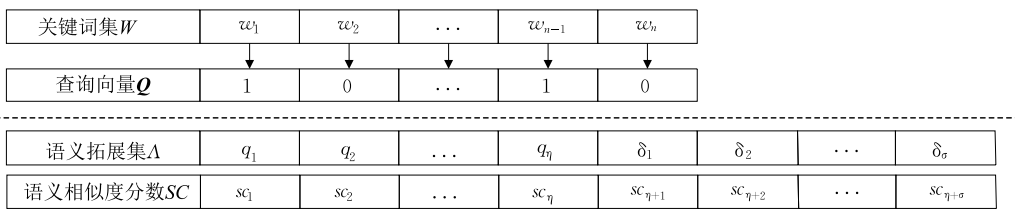


图 6 查询拓展过程

③加密查询向量. 根据语义扩展集 Λ 创建查询向量 $\boldsymbol{Q}$,若 $w_j\in\Lambda$,则将 $\boldsymbol{Q}[j]=1$,否则 $\boldsymbol{Q}[j]=0$. 将 $\boldsymbol{Q}$ 分为 u 块,若某个块全为 0,则此块标记值为 0,否则标记为 1,得到查询标记向量 $\boldsymbol{\varphi}$. 接着将向量 $\boldsymbol{Q}$ 中的 1 置为对应的语义相似度分数 sc_j,然后将 $\boldsymbol{Q}$ 扩展成 $(n+1)$ 维且 $(n+1)$ 位设置为 1,用大于 0 的随机数 r 对 $\boldsymbol{Q}$ 缩放,并扩展成 $(n+2)$ 维,第 $(n+2)$ 位设置成 $(\eta+\sigma)$,因此 $\boldsymbol{Q}$ 表示为 $(r\boldsymbol{Q},r,(\eta+\sigma))$. 用创建文档索引时类似的方法,将 $\boldsymbol{Q}$ 分裂为 $\boldsymbol{Q}'$ 和 $\boldsymbol{Q}''$:如果指示向量 $\boldsymbol{S}$ 的第 j 位为 0,则将 $\boldsymbol{Q}'[j]$ 和 $\boldsymbol{Q}''[j]$ 设为随机值,满足 $\boldsymbol{Q}'[j]+\boldsymbol{Q}''[j]=\boldsymbol{Q}[j]$;如果 S 的第 j 位为 1,则 $\boldsymbol{Q}'[j]=\boldsymbol{Q}''[j]=\boldsymbol{Q}[j]$. 使用密钥 SK 进行

加密，生成陷门 $T_Q=\{\boldsymbol{M}_1^{-1}\boldsymbol{Q}',\boldsymbol{M}_2^{-1}\boldsymbol{Q}''\}$. 最后，授权用户将查询标记向量 $\boldsymbol{\varphi}$ 发给私有云服务器，将陷门 T_Q 上传到公有云服务器.

(5) Query($\boldsymbol{\varphi},T_Q,k,E,\mathcal{I}$). 私有云服务器接收到授权用户发送的查询标记向量 $\boldsymbol{\varphi}$ 后，依次用 $\boldsymbol{\varphi}$ 中每一位 1 去匹配 $\boldsymbol{e}_i$ 中对应的块，即块的标记值 bb_s 是否为 0. 若为 0 则说明该文档对应的块没有用户搜索的关键词，如果为 1 则将对应的索引标识符 sid_i 记录下来，得到可能包含搜索关键词的候选索引标识符集合 $SID_\in=(\cdots,sid_i,\cdots,sid_j,\cdots,sid_z,\cdots)$. 接着私有云服务器将 $SID_\in$ 上传给公有云服务器，公有云服务器根据索引的标识符 sid_i 找到对应的安全索引 I_i，将对应的 V_i 和陷门 T_Q 计算文档的相似度分数并排序，返回前 k 篇文档给用户.

文档得分计算公式如下：

$$\begin{aligned}V_i\cdot T_Q&=\{\boldsymbol{M}_1^{\mathrm{T}}\boldsymbol{D}_i',\boldsymbol{M}_2^{\mathrm{T}}\boldsymbol{D}_i''\}\cdot\{\boldsymbol{M}_1^{-1}\boldsymbol{Q}',\boldsymbol{M}_2^{-1}\boldsymbol{Q}''\}\\&=\boldsymbol{D}_i'\cdot\boldsymbol{Q}'+\boldsymbol{D}_i''\cdot\boldsymbol{Q}''\\&=\boldsymbol{D}_i\cdot\boldsymbol{Q}\\&=r\Big(\sum_{j=1}^{n}(Z_{ij}\cdot(wf_{t,f}\times idf_t))\cdot sc_j+\varepsilon\Big)+(\eta+\sigma).\end{aligned}$$

(6) Decrypt(C,sk). 授权用户使用密钥 sk，对返回的 top-k 篇密文进行解密，得到明文文档集.

为了更好的阐述本方案，我们假设文档数量 $m=10$，关键词数 $n=100$，标记向量维度 $u=10$ 来简化说明文档分块标记及查询标记向量和文档标记向量的匹配过程. 如图 7 为文档向量 $\boldsymbol{D}_1$ 分块标记的过程. 关键词数 $n=100$，则文档向量 $\boldsymbol{D}_1$ 的维数为 100 维. 将 $\boldsymbol{D}_1$ 分为 $u=10$ 块，每一块有 10 个元素，若 10 个元素全为 0，则将此块标记为 $bb_s=0$，10 个元素中只要有一个元素为 1，此块的标记就为 1，得到文档标记向量 $\boldsymbol{b}_1=(1,0,0,0,0,1,0,0,1,0)$. 将 10 个文档依次按此操作得到文档标记向量 $\boldsymbol{b}_1$ 到 $\boldsymbol{b}_{10}$. 图 8 为私有云服务器上这 10 个文档标记向量与查询标记向量的匹配过程，当查询关键词的标记向量为 $\boldsymbol{\varphi}=(0,1,0,0,1,0,0,0,0,1)$ 时，将 $\boldsymbol{\varphi}$ 中的第二位 1 同 10 篇文档标记向量对应位置的值比较，得到 b_3 所对应的文档可能包含查询的关键词，记录其索引的标识符 sid_3，依次将 $\boldsymbol{\varphi}$ 中的 1 与 10 篇文档的标记向量对应位置的值比较得到候选的索引标识符的集合 $SID_\in=(sid_3,sid_8,sid_5)$，私有云服务器将候选索引标识符集合 $SID_\in$ 发送给公有云服务器，进行相

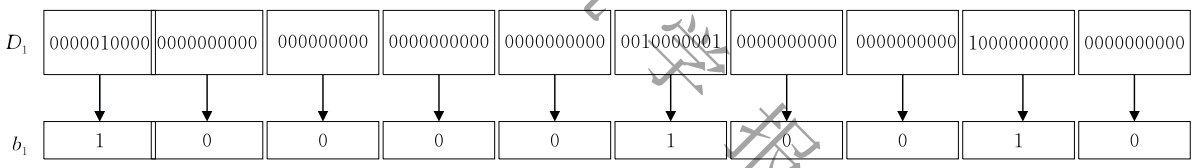


图 7　文档分块标记

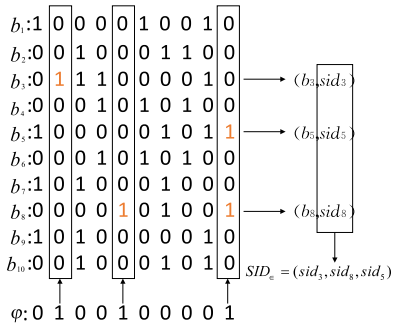


图 8　查询标记向量和文档标记向量匹配过程

在此方案中，对文档和查询的向量分块标记，在私有云服务器上将标记向量 $\boldsymbol{b}_i$ 和 $\boldsymbol{\varphi}$ 进行匹配，过滤不可能包含检索关键词的索引. 因此，在公有云服务器上无需进行不必要的相关度分数的计算及大量文档的排序，这极大提高了检索效率，特别是在包含海量数据的云环境中进行检索时，能过滤大量的无关文档，检索效率的提高更加显著.

5.2　增强方案

基础方案虽然提高了检索效率，但分块标记操作使构建索引和陷门时间相较于 MRSE 方案[5]有所增加. 针对上述问题，对基础方案进行进一步改进. 图 9 为本方案的基本过程.

首先，数据拥有者随机产生一个 $(n+2)$ 比特的向量 $\boldsymbol{S}$ 和 $2h$ 个维度为 $((n+2)/h)\times((n+2)/h)$ 的可逆矩阵 $\{\boldsymbol{M}_{11},\boldsymbol{M}_{12},\cdots,\boldsymbol{M}_{1h},\boldsymbol{M}_{21},\boldsymbol{M}_{22},\cdots,\boldsymbol{M}_{2h}\}$，其中，$h$ 是一个正整数并且满足 $h\mid(n+2)$. 生成密钥 SK 为 $SK=\{\boldsymbol{S},\boldsymbol{M}_{11},\boldsymbol{M}_{12},\cdots,\boldsymbol{M}_{1h},\boldsymbol{M}_{21},\boldsymbol{M}_{22},\cdots,\boldsymbol{M}_{2h},u,h\}$，其中，$u$ 是一个正整数并且 $u\mid n$.

其次，文档向量 $\boldsymbol{D}_i$ 分裂成 $\boldsymbol{D}_i'$ 和 $\boldsymbol{D}_i''$ 后，将 $\boldsymbol{D}_i'$ 和 $\boldsymbol{D}_i''$ 分别分成 h 段，得到 $\boldsymbol{D}_i'=(\boldsymbol{D}_{i1}',\boldsymbol{D}_{i2}',\cdots,\boldsymbol{D}_{ih}')$，$\boldsymbol{D}_i''=(\boldsymbol{D}_{i1}'',\boldsymbol{D}_{i2}'',\cdots,\boldsymbol{D}_{ih}'')$. 然后使用密钥 SK 进行加密，得到 $V_i=\{\boldsymbol{M}_{11}^{\mathrm{T}}\boldsymbol{D}_{i1}',\boldsymbol{M}_{12}^{\mathrm{T}}\boldsymbol{D}_{i2}',\cdots,\boldsymbol{M}_{1h}^{\mathrm{T}}\boldsymbol{D}_{ih}',\boldsymbol{M}_{21}^{\mathrm{T}}\boldsymbol{D}_{i1}'',\boldsymbol{M}_{22}^{\mathrm{T}}\boldsymbol{D}_{i2}'',\cdots,\boldsymbol{M}_{2h}^{\mathrm{T}}\boldsymbol{D}_{i2}''\}$ 和对应的索引 $I_i=(sid_i,V_i)$.

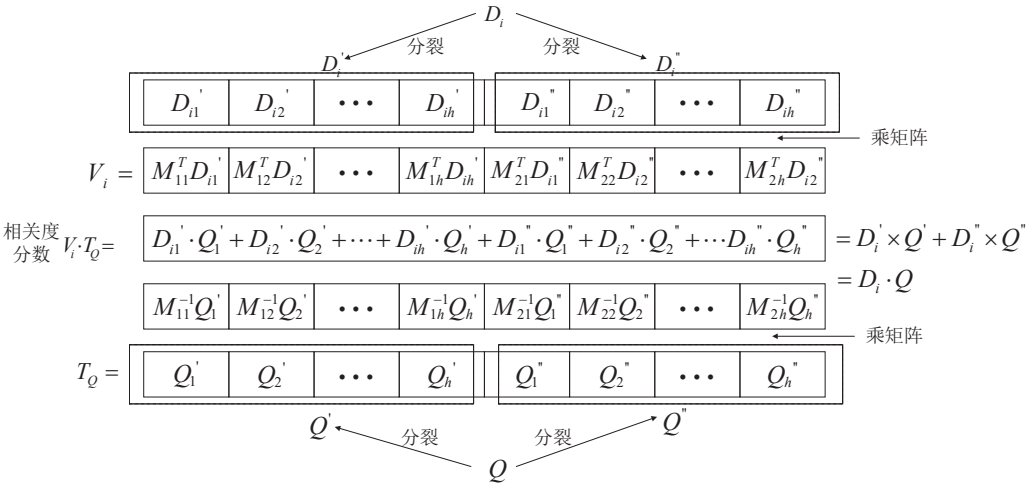


图 9 增强方案基本过程

数据拥有者将文档的标记向量集 $\boldsymbol{E}=(e_1,e_2,\cdots,e_m)$ 发送私有云服务器,将索引集合 $\mathcal{I}=(I_1,I_2,\cdots,I_m)$ 和密文集合 $C=(c_1,c_2,\cdots,c_m)$ 上传给公有云服务器.

接着,授权用户采用构建索引时类似的操作,得到陷门 $T_Q=\{\boldsymbol{M}_{11}^{-1}\boldsymbol{Q}_1',\boldsymbol{M}_{12}^{-1}\boldsymbol{Q}_2',\cdots,\boldsymbol{M}_{1h}^{-1}\boldsymbol{Q}_h',\boldsymbol{M}_{21}^{-1}\boldsymbol{Q}_1'',\boldsymbol{M}_{22}^{-1}\boldsymbol{Q}_2'',\cdots,\boldsymbol{M}_{2h}^{-1}\boldsymbol{Q}_h''\}$,将陷门发送给公有云服务器,将查询标记向量 $\boldsymbol{\varphi}$ 发送给私有云服务器.

相关度分数的公式如下:

$$\begin{aligned}V_i \cdot T_Q &= \{\boldsymbol{M}_{11}^{\mathrm{T}}\boldsymbol{D}_{i1}',\cdots,\boldsymbol{M}_{1h}^{\mathrm{T}}\boldsymbol{D}_{ih}',\boldsymbol{M}_{21}^{\mathrm{T}}\boldsymbol{D}_{i1}'',\cdots,\boldsymbol{M}_{2h}^{\mathrm{T}}\boldsymbol{D}_{ih}''\}\cdot \\ &\quad \{\boldsymbol{M}_{11}^{-1}\boldsymbol{Q}_1',\cdots,\boldsymbol{M}_{1h}^{-1}\boldsymbol{Q}_h',\boldsymbol{M}_{21}^{-1}\boldsymbol{Q}_1'',\cdots,\boldsymbol{M}_{2h}^{-1}\boldsymbol{Q}_h''\} \\ &= \boldsymbol{D}_i' \cdot \boldsymbol{Q}' + \boldsymbol{D}_i'' \cdot \boldsymbol{Q}'' \\ &= \boldsymbol{D}_i \cdot \boldsymbol{Q} \\ &= r\Big(\sum_{j=1}^{n}(Z_{ij} \cdot (wf_{t,f} \times idf_t)) \cdot sc_j + \varepsilon\Big) + (\eta+\sigma).\end{aligned}$$

最后,授权用户使用密钥 sk,对返回的 top-k 篇密文进行解密,得到明文文档.

在增强方案中,将文档向量 $\boldsymbol{D}_i$ 分裂后的 $\boldsymbol{D}_i'$ 和 $\boldsymbol{D}_i''$ 分成 h 段,然后用 $2h$ 个 $((n+2)/h)\times((n+2)/h)$ 维的可逆矩阵与之计算得到索引. 可逆矩阵的维数为基础方案的 $1/h$ 倍,而 MRSE 方案和基础方案的索引构建时间主要花费在文档向量和矩阵的相乘上,这使得此方案的索引构建时间大大减少,构建陷门时间也是如此.

6 性能测试与安全分析

6.1 性能测试

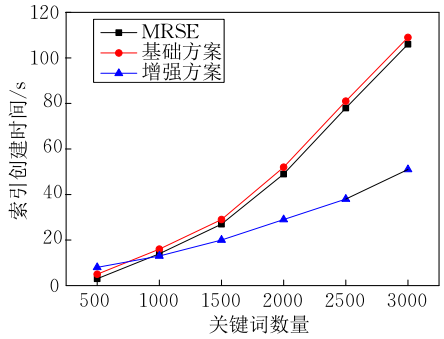
本文使用 IEEE 数据库中的英文论文作为测试数据集,仿真实验使用 Java 语言实现,使用 PDFBox 对 PDF 文档进行关键词提取并计算词频信息和其对应的相关度分数. 系统的性能测试在 64 位操作系统的台式机上执行,运行环境为 Windows 8,处理器主频为 3.40GHz,内存 16GB. 本实验基于 WordNet 3.0 语料库,采用包含 Resnik、Lin、JiangAndConrath、WuAndPalmer 等多种算法的软件包 Java WordNet Similarity,并选用 Resnik 算法[24]计算语义相似度. 为了方便得到关键词的信息内容,使用了 WordNet-InfoContent-2.1①,配合 WordNet 语料库计算语义相似度分数. 在本文实验中,当关键词数量 $n=3000$ 时,每个块的维数应该不小于 30. 标记向量维度 u 等于关键词数量除以每个块的维数. 因此,标记向量维度 $u=100$. 文档向量分段数量 h 满足 $h\mid(n+2)$,为了方便计算,实验中设置 $h=50$.

图 10(a)表示给定文档数量 $m=5000$ 时,随着关键词数量的增多,三个方案索引创建时间的变化趋势. 关键词数量越多,每个文档向量 $\boldsymbol{D}_i$ 的维度 n 越大,矩阵的维度也越大,需要进行的文档向量分裂和矩阵相乘的计算开销就越大,因此索引创建时间就会越多.

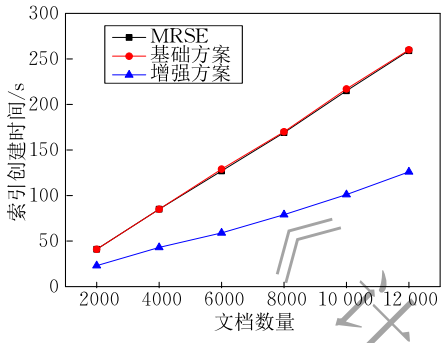
图 10(b)表示给定关键词数量 $n=3000$ 时,随着文档数量的增多,三个方案索引创建时间的变化趋势. 由于每个文档向量 $\boldsymbol{D}_i$ 的维度固定为 $n=3000$,因此创建每个文档向量 $\boldsymbol{D}_i$ 对应的索引时间相同,文档数量越多,索引创建时间越多.

由图 10(a)和图 10(b)两幅图可以看出,基础方案由于要对每个文档向量 $\boldsymbol{D}_i$ 分 u 块标记,相对于 MRSE 方案[5]索引创建时间有所增加,而增强方案

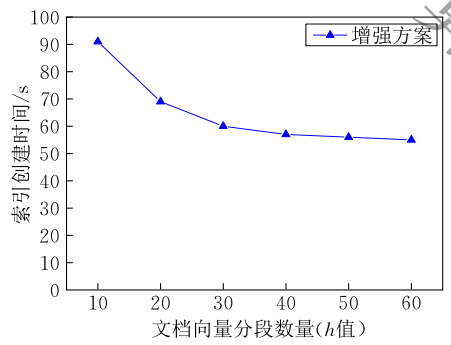
① Information content computed on various corpora. http://www.d.umn.edu/~tpederse/similarity.html 2016,4,21



(a) 关键词数量不同，文档数量m=5000



(b) 文档数量不同，关键词数量n=3000



(c) 文档向量分段数量不同

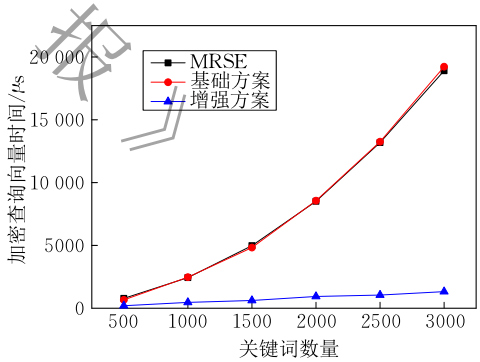
图 10　索引创建时间

将每一个文档向量 $\boldsymbol{D}_i$ 分裂后的向量 $\boldsymbol{D}_i'$ 和 $\boldsymbol{D}_i''$ 分成 h 段，用 $2h$ 个 $((n+2)/h)\times((n+2)/h)$ 维的可逆矩阵的转置与之相乘，使得矩阵相乘的计算开销大幅度降低，大大减少了索引的创建时间. 当文档数量 $m=5000$，关键词数 $n=3000$ 时，增强方案的索引创建时间为 51 s，MRSE 方案[5] 的索引创建时间为 106 s. MRSE 方案[5] 的索引创建时间约为本文中增强方案的 2 倍.

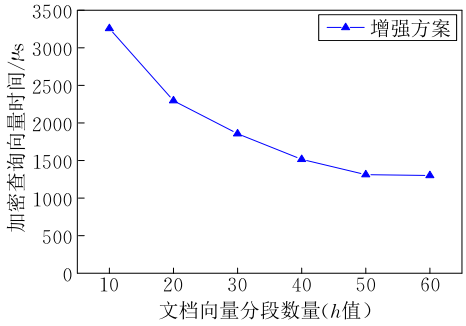
图 10(c)表示给定文档数量 $m=5000$，关键词数量 $n=3000$ 时，增强方案的索引创建时间随文档向量分段数量 h 值的变化趋势. 文档向量分的段数越多，每一段的维数就越小，与之相乘的矩阵维度也越小，计算矩阵相乘的时间也就相应越少，索引创建时间降低. 由于 MRSE 方案和基础方案不涉及 h 的取值，因此图 10(c)中只考虑 h 值对增强方案索引创建时间的影响.

图 11(a)是随着关键词数量的增多，三个方案加密查询时间的变化趋势. 关键词数量越多，则查询向量 $\boldsymbol{Q}$ 的维度 n 越大，矩阵维度也越大，需要进行的查询向量分裂和矩阵相乘的计算开销越大，因此加密查询的时间越多. 图 11(a)中 MRSE 方案和基础方案两条曲线几乎重合，这是因为基础方案比 MRSE 方案进行了额外的查询向量的分块标记操作，而标记向量的维度通常较小，因而分块标记操作花费的时间很少. 因此相对于 MRSE 方案，基础方案加密查询时间的增加并不明显. 增强方案将查询向量 $\boldsymbol{Q}$ 分裂后的向量 $\boldsymbol{Q}'$ 和 $\boldsymbol{Q}''$ 分成 h 段，用 $2h$ 个 $((n+2)/h)\times((n+2)/h)$ 维的可逆矩阵的逆与之相乘，使得矩阵相乘的计算开销大幅度降低，大大减少了加密查询向量的时间. 当关键词数 $n=3000$ 时，MRSE 方案[5] 的加密查询时间为 18 910 μs，而增强方案的加密查询时间仅为 1324 μs. MRSE 方案[5] 的加密查询时间是本文中增强方案的 14 倍.

由于 MRSE 方案和基础方案不涉及 h 的取值，因此图 11(b)只考虑 h 值对增强方案加密查询时间的影响. 图 11(b)表示给定文档数量 $m=5000$，关键词数量 $n=3000$ 时，增强方案加密查询时间随文档向量分段数量 h 值的变化趋势. 文档向量分的段数越多，每一段的维数就越小，与之相乘的矩阵维度也



(a) 关键词数量不同,文档数量m=5000



(b)文档向量分段数量不同

图 11　加密查询时间

越小，计算矩阵相乘的时间也就相应越少，加密查询时间降低.

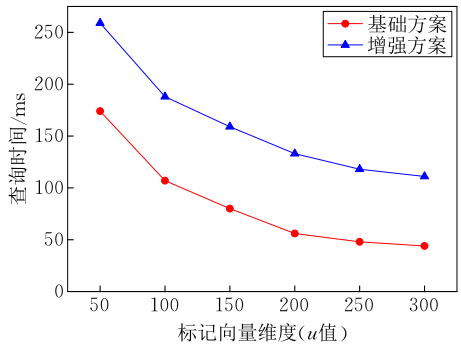
MRSE 方案不涉及 u 的取值，基础方案和增强方案都有涉及 u 的取值，因此，图 12(a)考虑了基础方案和增强方案中 u 值对查询时间的影响. 图 12(a)表示给定文档数量 $m=5000$，关键词数量 $n=3000$ 时，基础方案和增强方案查询时间随标记向量维度，即 u 值的变化趋势. u 值越大，也就是每个文档向量 $\boldsymbol{D}_i$ 分的块数越多，那么每个块的维数就越小，标记为 0 的可能性就越大，私有云过滤时，能更多的过滤不相关的文档，减少了计算不相关文档相关度分数和排序的时间. 因此查询时间随着 u 值增大而降低. 但是由于标记向量是以明文存储在私有云上，u 值越大，每个块就越小，越可能会暴露 $\boldsymbol{D}_i$ 和 $\boldsymbol{Q}$ 的信息，当 u 值等于关键词数量时，文档向量 $\boldsymbol{D}_i$ 和查询向量 $\boldsymbol{Q}$ 就以明文形式存储在私有云服务器上了，因此，u 的取值要在查询效率和保护隐私之间进行权衡，在保证查询效率的情况下，也不暴露用户的隐私. 例如，关键词数量 $n=3000$ 时，每个块的维数应该不小于 30. 而增强方案的查询时间比基础方案有所增加，是因为相关度分数的计算由原来的两个元素的内积变成 $2h$ 个元素的内积，计算开销加大了. 由于 h 值不变，随着 u 值的增大，增强方案和基础方案查询时间的差距是固定的. 在本文实验中，$h=50$ 时，增强方案的查询时间比基础方案增加了约 90 ms.

图 12(b)表示给定文档数量 $m=5000$ 时，随着关键词数量的增多，三个方案查询时间的变化趋势. 关键词数量越多，则 $\boldsymbol{D}_i$、$\boldsymbol{Q}$ 的维度 n 越大，需要进行内积运算的开销越大，因此查询时间越多.

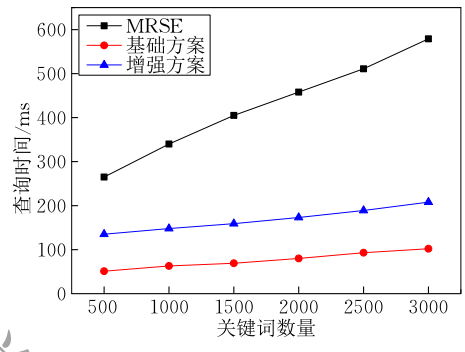
图 12(c)表示给定关键词数量 $n=3000$ 时，随着文档数量的增多，三个方案查询时间的变化趋势.

由于每个文档向量 $\boldsymbol{D}_i$、查询向量 $\boldsymbol{Q}$ 的维度固定为 $n=3000$，因此计算每个文档向量和查询向量的内积 $\boldsymbol{D}_i \cdot \boldsymbol{Q}$ 的时间相同，所以文档数量越多，查询的时间越多.

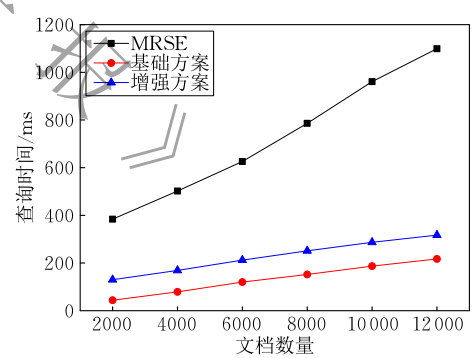
由图 12(b)和图 12(c)两幅图可以看出，基础方案的查询时间相对于 MRSE 方案大大降低是由于生成了标记向量，过滤了大量的不相关文档，减少了计算不相关文档的相关度分数及排序时间. 当文档数量 $m=12000$，关键词数 $n=3000$ 时，基础方案的查询时间为 217 ms，MRSE 方案[5]的查询时间为 1099 ms. MRSE 方案[5]的查询时间是本文中基础方案的 5 倍. 随着关键词数量和文档数量的增多，基础方案的优势会更加明显. 而增强方案的查询时间比基础方案有所增加，是因为相关度分数的计算由原来的两个元素的内积变成 $2h$ 个元素的内积，计算开销加大了. 当文档数量 $m=5000$，关键词数 $n=3000$ 时，增强方案的查询时间比基础方案增加了约 90 ms.



(a) 标记向量维度不同



(b) 关键词数量不同，文档数量都为$m=5000$



(c) 文档数量不同，关键词数量都为$n=3000$

图 12 查询时间

6.2 安全性分析

公有云服务器通过记录用户上传的陷门和查询的结果可创建访问模式. 在已知密文模型下，公有云服务器除了获取访问模式和查询结果外，不会获取任何额外信息.

定理 1. 基础方案在已知密文模型下是安全的.

在证明定理时，本文采用了 Curtmola 等人[25]使用的符号，具体符号及说明如下：

(1) $History$：$H=(\Delta, I, W)$，其中 Δ 表示文件

集，I 表示索引，查询词集 $W=(w_1,\cdots,w_k)$. H 由使用者上传.

(2) $View$：$V(H)=(Enc_{sk}(\Delta),Enc_{SK}(I),Enc_{SK}(W))$，其中，文件集 Δ 用对称密钥 sk 加密，索引 I 和查询词集用安全 KNN 算法[8]中的密钥 SK 加密. 公有云服务器只能看到 $View$.

(3) $Trace\ of\ a\ History$：公有云服务器获取到的敏感信息，例如访问模式和查询结果. 将 $Trace\ of\ a\ History$ 定义 $Tr(H)=\{Tr(w_1),\cdots,Tr(w_k)\}$，其中，$Tr(w_i)=\{(\delta_j,s_j)_{w_i\subset\delta_j},1\leq j\leq|\Delta|\}$，其中，$s_j$ 为查询词 w_i 和文件 δ_j 的相关度分数.

在已知密文模型下，给定两个相关度分数信息 $Trace\ of\ a\ History$ 相同的 $History$，分别生成 $V(H)$ 和 V'，如果公有云服务器无法分辨哪一个 $View$ 是由模拟者(Simulator)生成的，那么就可证明公有云服务器除了访问模式和查询结果，无法获取索引、文件集等任何信息. 接下来，证明定理 1.

证明.　定义 Sim 为模拟者(Simulator)，能够模拟(simulate)一个 V'，使得公有云服务器无法区分真实的 $V(H)$ 和 V'. 为了达到这个目的，Sim 要进行如下操作：

(1) Sim 随机选择 $\delta'_i\in\{0,1\}^{|\delta_i|}$，其中，$\delta_i\in\Delta$，$1\leq i\leq|\Delta|$，得到 $\Delta'=\{\delta'_i,1\leq i\leq|\Delta'|\}$.

(2) Sim 随机产生一个向量 $\boldsymbol{S}'\in\{0,1\}^{n+2}$，一个正整数 u'，$u'|n$，两个维度为 $(n+2)\times(n+2)$ 可逆矩阵 $\boldsymbol{M}'_1$ 和 $\boldsymbol{M}'_2$，密钥 $SK'=\{\boldsymbol{M}'_1,\boldsymbol{M}'_2,\boldsymbol{S}',u'\}$.

(3) Sim 进行如下操作创建 W' 和陷门 $Enc_{SK'}(W')$. 对每一个 $w_i\in W$，$1\leq i\leq k$，

① 创建 $w'_i\in\{0\}^{n+2}$，接着将 w'_i 的第 i 位设置成 1. 得到 $W'=\{w'_i,1\leq i\leq k\}$.

② 为每个 $w'_i\in W'$，$1\leq i\leq k$ 生成陷门，得到 $Enc_{SK'}(W')=\{Enc_{SK'}(w'_1),\cdots,Enc_{SK'}(w'_k)\}$.

(4) 为了生成 $I(\Delta')$，Sim 要执行以下操作：

① 为每个 $\delta'_j\in\Delta'$，$1\leq j\leq|\Delta'|$ 生成一个索引 $\boldsymbol{I}_{\delta'_j}$，$\boldsymbol{I}_{\delta'_j}$ 为一个向量且满足 $\boldsymbol{I}_{\delta'_j}\in\{0\}^{n+2}$.

② 遍历每个 $w_i\in W$，$1\leq i\leq k$，如果满足 $w_i\subset\delta_j$，$1\leq j\leq|\Delta|$，则 Sim 将向量 $\boldsymbol{I}_{\delta'_j}$ 和向量 $\boldsymbol{w}'_i$ 相加，即 $\boldsymbol{I}_{\delta'_j}=\boldsymbol{I}_{\delta'_j}+w'_i$.

③ 生成 $I(\Delta')=\{I_{\delta'_j},1\leq j\leq|\Delta'|\}$.

(5) Sim 用密钥 SK' 对索引 $I(\Delta')$ 进行加密，得到 $Enc_{SK'}(I(\Delta'))=Enc_{SK'}(\{\boldsymbol{I}_{\delta'_j}\},1\leq j\leq|\Delta'|)$.

(6) Sim 得到 $V'=(\Delta',Enc_{SK'}(I(\Delta')),Enc_{SK'}(W'))$.

经以上操作，可看出索引 $Enc_{SK'}(I(\Delta'))$ 和陷门 $Enc_{SK'}(W')$ 计算出的相关度分数 $Trace\ of\ a\ History$ 与公有云服务器拥有的 $V(H)=(Enc_{sk}(\Delta),Enc_{SK}(I),Enc_{SK}(W))$ 计算得到的 $Trace\ of\ a\ History$ 相同. 文件集 Δ 是用对称加密算法的密钥 sk 加密，得到 $Enc_{sk}(\Delta)$，在没有密钥 sk 的情况下，公有云服务器或攻击者无法分辨 $Enc_{sk}(\Delta)$ 和 Δ'. 而索引 $Enc_{SK'}(I(\Delta'))$ 和 $Enc_{SK}(I)$ 以及陷门 $Enc_{SK}(W)$ 和 $Enc_{SK'}(W')$ 使用了安全 KNN 算法[8]加密. 由于公有云服务器或攻击者没有密钥且加密过程中向量的分裂过程具有随机性，因此无法恢复出未加密前的索引和陷门信息. 结合以上的分析可知，公有云服务器或者攻击者无法分辨 $V(H)$ 和 V'，那么公有云服务器除了访问模式和查询结果，无法获取索引、文件集等任何信息.　　证毕.

定理 2.　增强方案在已知密文模型下是安全的.

证明.　定义 Sim 为模拟者(Simulator)，能够模拟(simulate)一个 V'，使得公有云服务器无法区分真实的 $V(H)$ 和 V'. 为了达到这个目的，Sim 执行的操作与定理 1 的证明类似，区别在于密钥 SK' 的生成过程，具体操作如下：

(1) Sim 随机产生一个向量 $\boldsymbol{S}'\in\{0,1\}^{n+2}$，两个正整数 u' 和 h'，分别满足 $u'|n$ 和 $h'|(n+2)$ 及 $2h'$ 个维度为 $((n+2)/h')\times((n+2)/h')$ 的可逆矩阵 $\{\boldsymbol{M}'_{11},\boldsymbol{M}'_{12},\cdots,\boldsymbol{M}'_{1h'},\boldsymbol{M}'_{21},\boldsymbol{M}'_{22},\cdots,\boldsymbol{M}'_{2h'}\}$，密钥 $SK'=\{\boldsymbol{M}'_{11},\boldsymbol{M}'_{12},\cdots,\boldsymbol{M}'_{1h'},\boldsymbol{M}'_{21},\boldsymbol{M}'_{22},\cdots,\boldsymbol{M}'_{2h'},\boldsymbol{S}',u',h'\}$.

(2) Sim 得到 $V'=(\Delta',Enc_{SK'}(I(\Delta')),Enc_{SK'}(W'))$.

经过定理 1 证明中的分析可知，公有云服务器或者攻击者同样无法分辨 $V(H)$ 和 V'，那么公有云服务器除了访问模式和查询结果，无法获取索引、文件集等任何信息.　　证毕.

另外，由于本文方案在查询向量 $\boldsymbol{Q}$ 中引入了随机数 r 和 $\eta+\sigma$，即使用户在两次搜索中输入了相同的关键词，也能够生成不同的陷门，实现了陷门的不可链接性. 同时在文档向量中引入了随机数 ε，公有云服务器计算出的相关度分数中包含了随机数，进一步干扰了服务器的分析.

7　结束语

现有的多关键词排序搜索方案未考虑关键词在标题、摘要和正文等不同域中的权重差异，也没有对语义扩展后的关键词进行相似度的区分，而且创建索引时间较多，搜索的效率也不够高. 为了进一步改善搜索结果的排序效果，提高整体方案的效率. 本文提出了一种快速的多关键词语义排序搜索方案.

首先引入域加权评分,对处于不同域中的关键词赋予不同的权重加以区分.然后计算查询词和拓展词之间的语义相似度.将语义相似度、域加权评分和相关度分数结合,构造更加准确的索引结构.其次分别对文档向量和查询向量分块,生成对应的文档标记向量和查询标记向量,通过文档标记向量和查询标记向量的匹配,快速的过滤了大量的无关文档,有效地提高了方案的检索效率.最后,通过对文档向量分段,将每一段与维数相同的矩阵相乘进行加密,缩短了创建索引的时间.从理论和实验两方面验证方案的可行性,该方案实现了快速的多关键词语义排序搜索,不仅提高了检索效率,缩短了索引创建时间,而且返回了更加满足用户需求的搜索结果.

参 考 文 献

[1] Mell P, Grance T. The NIST definition of cloud computing. Communications of the ACM, 2010, 53(6): 50

[2] Kamara S, Lauter K. Cryptographic cloud storage//Proceedings of the 14th Financial Cryptography and Data Security. Tenerife, Spain, 2010: 136-149

[3] Song D X, Wagner D, Perrig A. Practical techniques for searches on encrypted data//Proceedings of the IEEE Symposium on Security and Privacy. Oakland, USA, 2000: 44-55

[4] Chang Y C, Mitzenmacher M. Privacy preserving keyword searches on remote encrypted data//Proceedings of the Applied Cryptography and Network Security. New York, USA, 2005: 442-455

[5] Cao N, Wang C, Li M, et al. Privacy-preserving multi-keyword ranked search over encrypted cloud data. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(1): 829-837

[6] Li J, Wang Q, Wang C, et al. Fuzzy keyword search over encrypted data in cloud computing//Proceedings of the IEEE INFOCOM. San Diego, USA, 2010: 1-5

[7] Witten I H, Moffat A, Bell T C. Managing gigabytes: Compressing and indexing documents and images. IEEE Transactions on Information Theory, 1995, 41(6): 79-80

[8] Wong W K, Cheung W L, Kao B, et al. Secure kNN computation on encrypted databases//Proceedings of the ACM Sigmod International Conference on Management of Data. New York, USA, 2009: 139-152

[9] Xu J, Zhang W, Yang C, et al. Two-step-ranking secure multi-keyword search over encrypted cloud data//Proceedings of the International Conference on Cloud and Service Computing. Shanghai, China, 2013: 124-130

[10] Liu D, Wang S. Nonlinear order preserving index for encrypted database query in service cloud environments. Concurrency and Computation: Practice and Experience, 2013, 25(13): 1967-1984

[11] Boldyreva A, Chenette N, Lee Y, et al. Order-preserving symmetric encryption//Proceedings of the Theory and Applications of Cryptographic Techniques. Cologne, Germany, 2009: 224-241

[12] Yang C, Zhang W, Xu J, et al. A fast privacy-preserving multi-keyword search scheme on cloud data//Proceedings of the International Conference on Cloud and Service Computing. Shanghai, China, 2013: 104-110

[13] Li H, Yang Y, Luan T, et al. Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data. IEEE Transactions on Dependable and Secure Computing, 2016, 13(3): 312-325

[14] Li H, Liu D, Dai Y, et al. Engineering searchable encryption of mobile cloud networks: When QoE meets QoP. Wireless Communications IEEE, 2015, 22(4): 74-80

[15] Chuah M, Hu W. Privacy-aware BedTree based solution for fuzzy multi-keyword search over encrypted data//Proceedings of the 31st International Conference on Distributed Computing Systems Workshops (ICDCSW). Piscataway, USA, 2011: 273-281

[16] Wang B, Yu S, Lou W, et al. Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud//Proceedings of the IEEE INFOCOM. Piscataway, USA, 2014: 2112-2120

[17] Fu Z, Wu X, Guan C, et al. Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement. IEEE Transactions on Information Forensics and Security. 2016, 11(12): 2706-2716

[18] Indyk P, Motwani R et al. Approximate nearest neighbors: Towards removing the curse of dimensionality//Proceedings of the 30th ACM Symposium on Theory of Computing. Dallas, USA, 2000: 604-613

[19] Wang J, Yu X, Zhao M. Privacy-preserving ranked multi-keyword fuzzy search on cloud encrypted data supporting range query. Arabian Journal for Science and Engineering, 2015, 40(8): 2375-2388

[20] Bloom B H. Space/time trade-offs in hash coding with allowable errors. Communications of the ACM, 1970, 13(7): 422-426

[21] Fu Z, Sun X, Linge N, et al. Achieving effective cloud search services: Multi-keyword ranked search over encrypted cloud data supporting synonym query. IEEE Transactions on Consumer Electronics, 2014, 60(1): 164-172

[22] Xia Z, Zhu Y, Sun X, et al. Secure semantic expansion based search over encrypted cloud data supporting similarity ranking. Journal of Cloud Computing, 2014, 3(1): 1-11

[23] Manning C D, Raghavan P, Schütze H. Introduction to Information Retrieval. Cambridge, United Kingdom: Cambridge University Press, 2008

[24] Resnik P. Using information content to evaluate semantic similarity in a taxonomy//Proceedings of the 14th International Joint Conference on Artificial Intelligence. Montreal, Canada, 1995: 448-453

[25] Curtmola R, Garay J, Kamara S, et al. Searchable symmetric encryption: Improved definitions and efficient constructions//Proceedings of the 13th ACM Conference on Computer and Communications Security. Alexandria, USA, 2006: 79-88

YANG Yang, born in 1984, Ph. D., associate professor. Her research interest is information security and privacy.

LIU Jia, born in 1993, M. S. candidate. Her current research interest is searchable encryption.

CAI Sheng-Wei, born in 1993, M. S. candidate. His current research interest is searchable encryption.

YANG Shu-Lue, born in 1990, M. S. candidate. His current research interest is searchable encryption.

Background

With the growing popularity of cloud computing, more and more data has been outsourced to the public cloud for great flexibility and economic savings. To protect data privacy, it is desirable for data owner to encrypt sensitive data before sending to cloud, which makes the data utilization, such as data retrieval, a challenging task. Searchable encryption technology supports the users to search over encrypted data even though the cloud storage server is semi-trusted. Nowadays, a lot of searchable encryptions are proposed. However, most of the existing multi-keyword searchable encryption schemes have neither taken into consideration the location information of the keywords nor measured the similarity of the synonym keywords. At the same time, the search efficiency is low. In this paper, we propose a fast multi-keyword semantic ranked search scheme. Firstly, for the first time, the concept of weighted domain scoring is introduced to searchable encryption to calculate the document relevance scores. The keywords in different domains (title, abstract, etc.) are measured by different weighted domain score. Secondly, the retrieved keywords are semantically expanded to their synonym sets and the semantic similarities of the synonyms are calculated. Combining the semantic similarity, the weighted domain score and the relevance scores, we construct encrypted document index with higher accuracy. To improve the efficiency of MRSE (Multi-keyword Ranked Search over Encrypted cloud data), we partition the document index vector into several pieces and generate mark vector according to these pieces. Comparing the document mark vector and the query mark vector, we effectively filter a large number of irrelevant documents. The time for calculating the relevance scores and ranking is greatly reduced. Finally, document index vectors are partitioned into several sub-vectors, which are encrypted by the matrices with smaller dimensions. The method greatly reduces the time of generating encrypted indices and further improves the efficiency. Theoretical analysis and experimental results demonstrate that the proposed scheme achieves the multi-keyword semantic ranked search with high efficiency. It improves the retrieval efficiency, reduces the encrypted index generation time and returns more accurate ranking results. It also guarantees the privacy and security of data.

This paper is supported by the National Natural Science Foundation of China (Nos. 61402112, 61472307, 61472309, 61303198), the Science and Technology Project of Fujian Education Department (No. JA12028), the Open Fund Project of Fujian Provincial Key Laboratory of Information Processing and Intelligent Control (Minjiang University) (No. MJUKF201734), the Fujian Major Project of Regional Industry (2014H4015), and the Major Science and Technology Project of Fujian Province under Grant (No. 2015H6013). The research goals of these projects include searchable encryption, data protection and access control mechanisms.