云存储环境中多关键词加密排序搜索方法研究

黄 健[1],铁治欣[1,2],宋滢锟[1]

(1.浙江理工大学 信息学院,浙江 杭州 310018;2.浙江理工大学科学与艺术学院,浙江 绍兴 312369)

摘 要:随着可搜索加密技术的发展,用户输入多个查询关键词即可对云服务器中的数据进行检索。但是随着数据量的增加,云服务器的检索效率不断降低,其安全性也难以得到保障。为此,提出一种云存储环境中多关键词加密排序搜索方法。首先,通过对文档的关键词进行聚类,获得特征较集中的索引向量;其次,对索引和查询向量构建标记,根据查询标记的位置过滤无关文档,减少搜索时间;最后,将索引向量按照相应标记所属类别进行分组,将高维的加密密钥降为多个低维密钥,进一步减少索引的加密时间。随着文档分组数量的增加,查询时间将减少50%以上。实验结果表明,该方案在保证安全性和查询准确性的同时,能提高查询效率。

关键词:云存储;可搜索加密;关键词聚类;索引分组;降维

DOI:10.11907/rjdk.211208 **开放科学(资源服务)标识码(OSID)**:

中图分类号:TP391 **文献标识码**:A **文章编号**:1672-7800(2022)001-0226-07

Research on Multi-Keyword Encrypted Sorting Search Method in Cloud Storage Environment

HUANG Jian[1], TIE Zhi-xin[1,2], SONG Ying-kun[1]

(1. *School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou* 310018, *China*;
2. *Keyi College of Zhejiang Sci-Tech University, Shaoxing* 312369, *China*)

Abstract: With the development of searchable encryption technology, users can enter multiple query keywords to retrieve data in the cloud server. However, as the amount of data increases, the retrieval efficiency of the cloud server continues to decrease, and its security is difficult to guarantee. To this end, this paper proposes a multi-keyword encrypted sort search method in cloud storage environment. Firstly, by clustering the keywords of the documents, an index vector with more concentrated features is obtained. Second, build tags for the index and query vector, filter irrelevant documents based on the location of the query tags, and reduce search time. Finally, the index vector is grouped according to the category of the corresponding mark, and the high-dimensional encryption key is reduced to multiple low-dimensional keys, which further reduces the encryption time of the index. As the number of document groups increases, query time will be reduced by more than 50%. Experiments show that the scheme can improve query efficiency while ensuring safety and query accuracy.

Key Words: cloud storage environment; searchable encryption; keyword clustering; index grouping; dimensionality reduction

0 引言

随着大数据与云计算技术的发展,云存储也逐渐受到人们重视。很多用户和企业开始将复杂的数据从本地站点外包给商业公共云进行存储,以获得极大的灵活性,并节约成本,同时实现信息共享[1]。然而,在云存储环境下存在一些数据安全隐患,个人信息或机密文件等一些隐私也很容易被泄露出去。为保证隐私信息的安全性,数据在存储到云端前需对其进行加密。尽管加密后的数据可免受不法用户、非授权用户及不可信云服务商的攻击,但同时也带来了如检索效率低、检索难度高等问题[2]。

基于明文的关键词检索方案对密文是不可行的,但如果每次查询时都将密文数据下载到本地,在本地解密后再

收稿日期:2021-02-19

基金项目:国家自然科学基金项目(61170110);浙江省自然科学基金项目(LY13F020043);浙江省教育厅科研项目(21030074-F)

作者简介:黄健(1996-),男,浙江理工大学信息学院硕士研究生,研究方向为智能计算与数据挖掘;铁治欣(1972-),男,博士,浙江理工大学信息学院教授、硕士生导师,研究方向为数据挖掘、嵌入式系统;宋滢锟(1998-),女,浙江理工大学信息学院硕士研究生,研究方向为计算机应用技术。本文通讯作者:铁治欣。

执行查询,不仅浪费了本地存储空间,而且提高了检索难度。如何在保护隐私的前提下实现加密文件搜索,仍是目前迫切需要解决的问题。迄今已有很多学者对可搜索加密技术进行了研究,并提出一些有效方法,使用户能够输入关键词对加密文档进行检索[3-6]。然而直接将这些方法应用于复杂的文档系统并不科学,因为这些方法不仅检索效率低,而且不适用于要求更高的检索情景。一些相关研究也尝试提高密文检索的灵活性,但仍不能按照用户需求排序筛选出数据[7]。现有云存储环境下的密文排序搜索方案执行创建与更新加密索引时间较长,且随着文档数量的增加,检索效率会逐步降低。因此,寻找一种既能降低检索开销,又能提高检索效率的方案是目前的主要研究方向。

1　相关工作

可搜索加密技术是将数据和索引加密后存储到远程服务器中,根据用户提交的加密后的搜索请求生成特定陷门对加密数据进行搜索,之后返回匹配文档。在整个过程中,云服务器除负责存储与搜索外,不能获得任何有关的数据信息。

最早的可搜索加密技术是由 Song 等[8]提出的,这种对称可搜索加密方案的主要流程是先将文档拆分成多个词,然后用流密码对这些关键词进行双层加密,当用户输入查询关键词后,云服务器会将陷门与加密后的关键词依次进行匹配,并根据匹配结果返回相应的加密文档。该方案虽然满足了安全性要求,但是全文搜索效率较低。Goh[9]首次采用布隆过滤器作为安全索引结构,对提取的关键词利用哈希函数计算其对应哈希值,之后映射到布隆过滤器中。当用户输入查询关键词时,通过相同的哈希函数计算、查询关键词的哈希值,并验证其在布隆过滤器相应位置处的值是否相同。该方案虽然提高了查询效率,但在检索时会存在误差,从而影响结果的准确性。Chai 等[10]首次提出“半诚实且好奇”的云服务器模型,由于为了节省计算量和带宽资源,服务器提供商可能仅执行了部分搜索操作并返回部分搜索结果,因此提出基于单词查找树索引结构的可验证可搜索加密方案。Mahajan 等[11]提出层次聚类方法用于云数据保护,该框架的重要组成部分是数据复制以及使用 SHA1 哈希策略进行检查。Chen 等[12]提出一种新的基于关键字搜索的双服务器公钥加密框架,通过光滑投影构造散列函数,可防止来自两个不可信服务器的关键词猜测攻击。Tariq 等[13]协调了对称和非对称加密算法,设计一种新的基于服务器认证的双重加密框架,利用通配符技术寻找加密数据,同时对服务器进行验证,从而提高了数据的安全性。

随着数据文件的增加,关键词词典中词的数量也不断增加,使得通过构建索引进行关键词检索的计算量增大,效率很低。Cao 等[14-15]解决了加密数据的排序搜索问题,增强了系统的可用性,并提出基于多关键词排序的搜索方案(Multi-keyword Ranked Search over Encrypted cloud data, MRSE),通过对索引向量和请求向量计算内积得分对文档进行排序,但对大量文档而言,搜索计算量过大、耗时长,且精度不高。Saini 等[16]提出关键词模糊搜索方案,通过构造关键词模糊集,以容忍用户搜索时输入拼写错误与格式不一致的情况,但无法搜索与关键词语义相关的文档。Ahmed 等[17]通过使用加密动态索引以提高搜索效率,在加密数据集发生变化时能够对索引进行动态更新。Chen 等[18]提出支持高效、动态多关键字排序搜索的方案,首先通过协调匹配获得外包文件以查询关键词相关性,然后利用内积相似性进行分析,最后采用块稀疏对角矩阵与置换矩阵提高搜索速度。Fu 等[19]利用基于树的索引结构组织所有的文档索引向量,提出一种新的基于概念层次与语义的搜索方案。该方案使用一台服务器用于存储外包数据集,并将排名结果返回给数据用户,再使用另一个服务器计算文档及查询关键词之间的相似性分数,并将分数发送到第一个服务器。

一些学者提出基于树的搜索方案,Krishna 等[20]提出基于树的排名搜索方案,通过二叉树建立动态索引,以减少索引生成与查询时间。Peng 等[21]利用双线性映射构建用加法顺序与隐私保护函数族加密的基于树的索引,云服务器通过合并这些索引,用深度优先算法搜索文档。黄新宇[22]基于分治思想为数据集构建 B+树结构的索引树组,之后对对查询向量进行分组并在相应索引树上进行检索,在提高效率的同时降低了存储开销。陈垚等[23]提出一种基于优先级排序的动态安全可搜索加密方案,首先采用预处理字典树结构提高搜索效率,然后通过对新添加的文件标识符进行加密,使动态更新算法的时间复杂度变为 O(1),最后增加自定义搜索结果排序功能,提高了搜索准确率。徐光伟等[24]注意到查询关键词与索引之间的关联性,提出一种基于语义扩展的多关键词可搜索加密算法,首先根据依存句法区分多关键词并进行语义扩展,然后基于凝聚层次聚类与关键词平衡二叉树构建索引关联的索引树,最后通过剪枝参数和相关性得分阈值过滤索引无关的子树。

在 MRSE 方案[14]的基础上,本文提出一种新的云存储环境中多关键词加密排序搜索方法。首先从聚类后的文档中提取关键词,并将关键词按所属类别随机排列构成词典,然后利用向量空间模型,根据词典中的关键词为每篇文档建立特征较集中的向量索引。在此基础上,通过对索引和查询向量构建标记,根据查询关键词的位置过滤掉大量无关文档,从而减少搜索时间。最后,将索引向量按照关键词所属类别进行分组,以提高了索引加密速度。将加密后的索引上传至云服务器,云服务器通过计算文档组索引向量与组查询向量的内积,降序返回用户所需的前 θ 个文档。

本文的贡献总结如下:①对聚类后的文档提取关键词,构建特征集中的索引向量;②对索引和查询向量构建标记,根据查询标记的位置过滤掉大量无关文档;③对过滤后的索引向量分组加密,使每个加密密钥的维度降低,

从而提高搜索效率。

2 问题描述

2.1 系统模型

如图1所示，系统模型包括数据所有者、用户和云服务器。这3个实体和密文搜索方法组成一个系统模型，其中数据所有者和用户是诚实可信的，云服务器是半可信的。

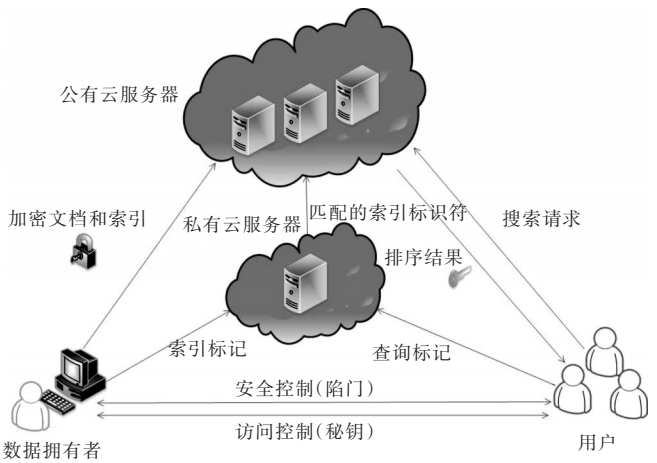

Fig. 1 System model of encrypted search scheme

图1 加密检索方案系统模型

(1)数据所有者：数据所有者首先根据文献[25]中的方法将文档聚类，然后提取所有类中的文档关键词生成词典，据此为每一篇文档建立索引向量，并对文档进行加密，最后将加密后的文档和索引上传给云服务器。当需要修改数据时，重复以上过程。

(2)私有云服务器：私有云服务器用于存储数据拥有者上传的索引标记向量，之后与用户发送过来的查询标记向量进行匹配，将匹配度高的类的文档标识符发送给公共云服务器。

(3)公共云服务器：公共云服务器为半可信服务器，用于存储数据拥有者上传的文档索引和加密后的文档集，并对用户发送的陷门和私有云服务器发送的文档标识符对应索引向量执行搜索操作，之后向用户返回所需的前θ个文档。

(4)用户：用户是数据使用者，当需要访问某文档时，则向数据所有者发送请求。收到数据所有者返回的密钥后，根据检索内容生成陷门，并发送给云服务器。云服务器据此返回用户检索的相应加密文档，用户再根据密钥对文档进行解密。

2.2 攻击模型

在数据所有者、用户、云服务器之间的通信过程中，攻击者可拦截通信，从所拦截的信息中推导出额外信息。云服务器被认为是“诚实且好奇”的[26]，具体而言，云服务器会诚实地执行指定操作，但同时也会试图从文件、索引或陷门中获取并分析隐私信息。在本研究中，仅要求云服务器知道加密后的数据文档、索引及查询陷门，而不知道具体密钥。但由于云服务器是“好奇”的，会在搜索过程中学习更多信息，如查询关键词和加密文档的信息，从而根据陷门与查询关键词的关联性推导出加密密钥。

根据云服务器获取的信息数量，将服务器攻击类别分成两种：①已知密文：云服务器只知道加密信息，如加密后的数据集、索引以及陷门；②已知背景：云服务器可知道更多信息，如搜索请求(陷门)的关联关系，或通过陷门与查询结果推测查询关键词。

2.3 符号说明

本文使用的符号及其说明如表1所示。

Table 1 Symbol description

表1 符号说明

符号	说明
F	文档集 $F=(F_1,F_2\cdots F_k)$
F_i	第i类文档集 $F_i=(F_{i1},F_{i2}\cdots F_{im})$
W	关键词集 $W=(W_1,W_2\cdots W_k)$
W_i	第i类关键词集 $W_i=(w_{i1},w_{i2}\cdots w_{ij})$
D	文档索引向量 $D=(D_1,D_2\cdots D_k)$
D_i	第i类中索引向量 $D_i=(D_{i1},D_{i2}\cdots D_{in})^T$
D_{ij}	第i类中第j篇文档的索引向量
P_i	扩维后第i类文档的索引向量 $P_i=(P_{i1},P_{i2}\cdots P_{in})^T$
P_{ij}	扩维后第i类中第j篇文档的索引向量 $P_{ij}=(p_1,p_2\cdots p_k)^T$
p_i	分组后的第i个组索引向量
I	加密后的索引集合 $I=(I_1,I_2\cdots I_k)$
W'	查询关键词集
q	查询向量 $q=(q_1,q_2\cdots q_n)^T$
Q_i	扩维后查询向量的第i个组向量
Q	分组后的查询向量 $Q=(Q_1,Q_2\cdots Q_k)^T$
T	陷门

2.4 词语解释

关键词词典：根据文档类别分别提取文档关键词，经去重处理后按类别排列组成关键词词典。

向量空间模型：将所有文档和搜索关键词用向量表示，维度为关键词词典大小，向量中每一维的值为该位置关键词得分。在加密搜索领域，通常采用词频tf和反词频idf表示该得分，其中词频是指关键词在文档中出现的频率，频率越高说明关键词对该文档越重要；反词频表示包含某关键词的文档数，反映该关键词在整个数据集中的重要程度，文档数越多说明关键词对文档的区分度越低。

文档得分：文档得分反映了查询关键词与文档中关键词之间的匹配程度，云服务器会计算文档得分，并按照分数高低进行排序，之后返回搜索结果。当云服务器收到查询请求q后，可用式(1)计算文档F_{ij}的得分[27]。

$$Score(F_{ij},q)=\frac{1}{\left|F_{ij}\right|}\sum_{w_{ij}\in\tilde{w}}(1+\ln f_{i,j})\cdot\ln(1+\frac{m}{f_j}) \tag{1}$$

其中，$\left|F_{ij}\right|$是文档的欧几里得长度，作为归一化因子，计算公式为$\sqrt{\sum_{j=1}^{n}(1+\ln f_{i,j})^2}$，$w_{ij}$是文档$F_{ij}$中的关键词，$f_{i,j}$是关键词$w_{ij}$在文档$F_{ij}$中的出现次数，$\tilde{w}$是文档$F_{ij}$中包含的

关键词集合,m是文档总数,f_j是包含关键词w_{ij}的文档数。

3　云存储环境中多关键词加密排序搜索方法

以下将分为5部分详细介绍本文提出的云存储环境中多关键词加密排序搜索方案(Multi-Keyword Encrypted Sorting Search Method,MESM)。

3.1　词典构建

数据所有者首先根据文献[25]中的方法提取文档特征,并根据关键词权重构建相应特征向量,然后利用k-$means$算法[28]将m篇文档分成k类。按类分别提取关键词并去重,同一类关键词由于相关性较强,可将这些关键词按所属类别随机排列,令相关性较强的类间词语排列在一起,继而构建大小为$n = n_1 + n_2 \cdots n_k$的关键词词典。其中,n是词典中的关键词总数,n_i是第i类中的关键词个数。

3.2　索引创建

第一步:创建索引与标记向量。计算词典中每一维关键词在每篇文档中的词频得分,对于每一类文档而言,通过向量空间模型将其中第i篇文档表示为$D_i' = (d_{i1},d_{i2} \cdots d_{in})^T$,其中$d_{ij}$是词典中第$j$个词在第$i$篇文档对应位置的词频。与创建索引向量类似,如果词典中的关键词在第i篇文档中对应位置的词频得分不为0,则标记该位置的值为1,最终得到标记向量$B_{\varphi i} \in \{0,1\}^{(n)}$,其中$\varphi$表示该标记向量所属类别。

第二步:维度扩展。对D_i'进行扩维,从n维扩到$n + u + 1$维,其中将第$n + 1$维到$n + u$维设置为任意随机数ε_i,而将第$n + u + 1$维设置为常数1。扩维后第i篇文档的索引向量$P_i' = (D_i'^T,\varepsilon_1,\varepsilon_2 \cdots \varepsilon_u,1)$。

第三步:向量分组。将扩展后的向量P_i'按照每一维得分所在组的不同分为$k + 1$组,表示为$(P_{i1}',P_{i2}' \cdots P_{i(k+1)}')^T$,其中$P_{ij}'$即为第$j$个组向量,其前$k$个组向量维度即是该组的关键词数$n_i$,而$P_{i(k+1)}'$的维度为$u + 1$。

第四步:生成密钥。数据拥有者随机生成$2k$个维度为$n_i \times n_i$的可逆矩阵$M_{11},M_{12} \cdots M_{1k}$和$M_{21},M_{22} \cdots M_{2k}$,2个$(u + 1) \times (u + 1)$维的$M_{1(k+1)}$、$M_{2(k+1)}$以及1个$n + u + 1$维的分割指示向量$S$。其中,$u + 1$是扩展维度,$S \in \{0,1\}^{(n+u+1)}$。与前面$P_i'$的分组方法相同,将分割指示向量$S$也分成$k + 1$组,最后表示为$S = (S_1,S_2 \cdots S_{k+1})$,密钥则表示为$M_1 = (M_{11},M_{12} \cdots M_{1(k+1)})$和$M_2 = (M_{21},M_{22} \cdots M_{2(k+1)})$。

第五步:随机分割。根据指示向量S的每个组指示向量S_i对索引向量P_i'的对应组向量$P_{ij}'(j = 1,2 \cdots k + 1)$进行随机分割,分割成$\tilde{P}_{ij}'$和$\tilde{P}_{ij}''$,分别表示为$P_i' = (\tilde{P}_{i1}',\tilde{P}_{i2}' \cdots \tilde{P}_{i(k+1)}')$和$P_i'' = (\tilde{P}_{i1}'',\tilde{P}_{i2}'' \cdots \tilde{P}_{i1(k+1)}'')$。对于该组第$\omega$个位置上的值,分割规则如式(2)所示。

$$\begin{cases} \tilde{P}_{ij}'[\omega] = \tilde{P}_{ij}''[\omega] = P_{ij}'[\omega] & 若S_i[\omega] = 0 \\ \tilde{P}_{ij}'[\omega] + \tilde{P}_{ij}''[\omega] = P_{ij}'[\omega],\ \tilde{P}_{ij}'[\omega] \neq \tilde{P}_{ij}''[\omega] & 若S_i[\omega] = 1 \end{cases} \tag{2}$$

第六步:索引加密。用组密钥M_{1i}和$M_{2i}(i = 1,2 \cdots k + 1)$分别对分割后的组索引$\tilde{P}_{ij}'$与$\tilde{P}_{ij}''(j = 1,2 \cdots k + 1)$进行加密,第$i$篇文档索引加密过程可分别表示为$P_i'M_1 = \{\tilde{P}_{i1}'M_{11},\tilde{P}_{i2}'M_{12} \cdots \tilde{P}_{i(k+1)}'M_{1(k+1)}\}$和$P_i''M_2 = \{\tilde{P}_{i1}''M_{21},\tilde{P}_{i2}''M_{22} \cdots \tilde{P}_{i(k+1)}''M_{2(k+1)}\}$,整个加密结果表示为$I_i = \{P_i'M_1,P_i''M_2\}$,$m$篇加密结果则表示为$I = (I_1,I_2 \cdots I_m)$。最后,将加密文档C与加密索引$I$上传到服务器。

3.3　陷门创建

步骤1:创建查询与标记向量。用户输入查询关键词,根据关键词词典生成相应查询向量$q = (q_1,q_2 \cdots q_n)^T$。如果查询关键词与词典中对应位置的关键词相匹配,则设置q_i为该词对应的反词频,否则设置$q_i = 0$。与创建查询向量类似,但将q_i不为0的值标记为1,得到标记向量$b \in \{0,1\}^{(n)}$。

步骤2:维度扩展。首先对q进行维度扩展,从n维扩展到$n + u + 1$维,其中在第$n + 1$维到$n + u$维间任意选择v维设置为1,其余设为0,然后将前$n + u$维数值乘以一个非零随机数r,最后将第$n + u + 1$维设置为随机数t,则扩展后的最终查询向量表示为Q。

步骤3:查询向量分组。将查询向量Q分成$k + 1$个组向量$Q_1,Q_2 \cdots Q_{k+1}$,其中$Q_i(i = 1,2 \cdots k)$的维度是其所属类别关键词个数n_i,而Q_{k+1}的向量维度为$u + 1$。

步骤4:随机分割。根据指示向量S的每个组指示向量S_i对查询向量Q的组向量Q_i进行随机分割,分割成Q_i'和Q_i'',分别表示为$Q' = (Q_1',Q_2' \cdots Q_{k+1}')$和$Q'' = (Q_1'',Q_2'' \cdots Q_{k+1}'')$。对于该组第$\varpi$个位置上的值,分割规则如式(3)所示。

$$\begin{cases} Q_i'[\varpi] + Q_i''[\varpi] = Q_i[\varpi],Q_i'[\varpi] \neq Q_i''[\varpi] & 若S_i[\varpi] = 0 \\ Q_i'[\varpi] = Q_i''[\varpi] = Q_i[\varpi] & 若S_i[\varpi] = 1 \end{cases} \tag{3}$$

步骤5:生成陷门。用组密钥M_{1i}^{-1}和$M_{2i}^{-1}(i = 1,2 \cdots k + 1)$分别对分割后的查询组索引$Q_i'$与$Q_i''$进行加密,加密索引为$M_1^{-1}Q' = \{M_{11}^{-1}Q_1',M_{12}^{-1}Q_2' \cdots M_{1(k+1)}^{-1}Q_{k+1}'\}$和$M_1^{-1}Q'' = \{M_{21}^{-1}Q_1'',M_{22}^{-1}Q_2'' \cdots M_{2(k+1)}^{-1}Q_{k+1}''\}$,最终生成陷门$T = \{M_1^{-1}Q',M_2^{-1}Q''\}$。

3.4　文档筛选

对于数据拥有者上传的索引标记向量$B_{\varphi i} \in \{0,1\}^{(n)}$($\varphi$表示所属类别),当用户输入查询关键词时,这些关键词之间往往不会是完全无关的,由于其是对所需文档的特征描述,因而这些词的相关度较高,其查询标记向量$b \in \{0,1\}^{(n)}$的特征值也是集中的。根据标记向量之间的匹配,返回匹配度高的标记$B_{\varphi i}$。

查询标记与索引标记匹配过程如图2所示。假设文档分类数为3,每类文档数分别为2、3、2,关键词词典及标记向量维度为15。第2类是有关云计算的文档集,从中提取的关键词可能包括cloud、computing、encrypted、search等,这些词将按顺序排列在词典中的特定位置,如词典的最后,因此生成的索引标记1也都会集中于最后一部分。当用户输入包括相关联的如cloud、search等多个查询关键词时,查

询标记向量 b 与第 2 类文档集的索引标记 B_{21}、B_{22}、B_{32} 的匹配度更高，而最终需要返回的是前 θ 个相关性最高的文档。所以对于匹配度较低的属于第 1、3 类的文档，可将其过滤，从而避免对所有文档进行不必要的得分计算，提高了搜索效率。

B_{11}: 1 0 1 1 1 0 0 0 0 1 0 0 0 0 0
B_{12}: 1 1 1 0 1 0 0 0 0 0 0 0 0 **1** 0
B_{21}: 0 0 1 0 0 0 0 0 0 0 0 **1** **1** 1 **1** **1**
B_{22}: 0 0 0 0 0 0 0 0 0 0 0 **1** 0 1 **1** **1**
B_{23}: 0 0 0 1 0 0 1 0 0 0 0 **1** **1** 1 **1** 0
B_{31}: 0 0 0 0 0 0 1 1 **1** 0 1 0 0 0 0 **1**
B_{32}: 0 1 0 0 0 1 1 **1** 1 1 0 0 0 0 0

b: 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1

Fig. 2 Matching process of query mark and index mark

图 2 查询标记与索引标记匹配过程

3.5 查询

用户将陷门 T 上传给公共云服务器进行查询，公共云服务器接收后将依次计算私有云服务器发送的文档标识符 $B_{\varphi i}$ 对应索引向量与陷门的内积得分，之后根据内积大小对其进行降序排列，将得分较高的前 θ 个加密文档返回给数据使用者。内积计算过程如式(4)所示。

$$\begin{aligned} I_i \cdot T &= (P_i' M_1, P_i'' M_2)(M_1^{-1} Q', M_2^{-1} Q'') \\ &= P_i' Q' + P_i'' Q'' \\ &= P_i Q \\ &= r(D_i' q + \sum_{i \in v} \varepsilon_i) + t \\ &= r(Score(F_i, q) + \sum_{i \in v} \varepsilon_i) + t \end{aligned} \tag{4}$$

4 性能分析

4.1 复杂度分析

随着数据量的增加，关键字数量增多，生成的关键词词典也越大，导致索引的加密矩阵维度很高。而在 MRSE 方案中，加密矩阵维度直接取决于词典大小，当词典中关键词个数为 n、扩展维度为 $u+1$ 时，m 篇文档的加密时间复杂度为 $O(m(n+u+1)^2)$。为降低加密矩阵维度，进一步减少加密时间，本方案将所有索引向量与查询向量按其特征值所属类别分成 $k+1$ 组，其加密时间复杂度可表示为 $O(m(\sum_1^k n_i^2 + (u+1)^2))$。由于 $(n+u+1)^2 = (\sum_1^k n_i + u + 1)^2 > (\sum_1^k n_i)^2 + (u+1)^2 > \sum_1^k n_i^2 + (u+1)^2$，故本方案的加密效率更高。

4.2 隐私分析

保证数据信息的安全性对于可搜索加密过程十分重要，在已知背景的攻击模型下，本文从密钥安全性、关键词信息、查询信息与陷门非关联性保护几方面进行安全性分析。

密钥安全性：对于文档集中的第 i 篇文档，云服务器并不知道索引加密过程 $I_i = \{P_i' M_1, P_i'' M_2\}$，其第 $j(j = 0,1 \cdots k+1)$ 组向量的加密过程表示为 $\tilde{P}_{ij}' M_{1j}$ 和 $\tilde{P}_{ij}'' M_{2j}$。设 ω_j 为每组向量维度，其值为 n_j 或 $u+1$。云服务器不知道降维、分组及分割具体过程，对于加密后的组向量 $\widetilde{P_{ij}}'$ 和 $\tilde{P}_{ij}''$，只能建立式(5)：

$$\begin{cases} \tilde{P}_{ij}' M_{1j} = \widetilde{P_{ij}}' \\ \tilde{P}_{ij}'' M_{2j} = \tilde{P}_{ij}'' \end{cases} \tag{5}$$

其中，M_{1j}、M_{2j} 各有 ω_j^2 个未知变量，$\widetilde{P_{ij}}'$、$\tilde{P}_{ij}''$ 则各有 ω_j 个未知变量。方程组(5)有 $2(\omega_j^2 + \omega_j)$ 个未知数，而等式个数只有 $2\omega_j$，方程组则会有无数个解，因而云服务器无法据此推出加密矩阵 M_{1j} 和 M_{2j}。

关键词信息保护：云服务器不知道词典中的关键词个数，通过分组可使得降维后的矩阵也是多变的，从而提高数据安全性。此外，在已知背景的攻击模型下，为有效提高安全性，可引入系统参数 w，保证索引向量至少有 2^w 个不同的 $\sum \varepsilon_i^{(v)}$，使得拥有相同值 $\sum \varepsilon_i^{(v)}$ 的概率小于 $1/2^w$，不同 $\sum \varepsilon_i^{(v)}$ 的数量 C_u^v 不大于 u/v，且在 $u/v = 2$ 时达到最大值。考虑到 $C_u^v \geqslant (u/v)^v$，需设置 $u = 2w$ 和 $v = w$。ε_i 还需满足均匀分布 $N(\mu' - c\mu' + c)$，其均值为 μ'，方差 $\sigma^2 = c^2/3$。为保证 ε_v 符合正态分布 $N(\mu, \sigma^2)$，应设置 $\mu' = \mu/w$，$c = \sqrt{3/w}\,\sigma$。在正态分布中，标准差 σ 作为折中参数，σ 较小时搜索精度较高，但是相对带来的混淆较小，降低了安全性，因此需要合理地设置 σ 以获得安全性与精度的平衡。

查询信息与陷门非关联性保护：为防止云服务器从陷门中推知用户查询信息，本方案对查询向量进行特征集中、分组、扩展、随机分割与加密处理，使查询关键词信息不会表现在查询陷门中，从而保护了查询信息。此外，由于随机数的引入，使得不同甚至相同的查询请求都会有不同得分，从而保护了陷门非关联性。

4.3 实验分析

本实验使用 RFC(Request For Comments)[14] 数据作为实验数据集，实验环境为 Windows 7 服务器，CPU 为英特尔酷睿 i5(2.5GHZ)处理器。

影响实验效率的主要因素是文档数量 m 及词典中的关键词数量 $n(n = n_1 + n_2 \cdots n_k)$，其决定了通过向量空间模型构建的索引与查询向量维度，从而决定了加密密钥维度及最终的检索效率。本实验通过减少查询文档数量以及降低加密维度两方面解决该问题，以下将分别分析 MESM 方案得到的结果，并与 MRSE 方案结果作对比。实验结果如图 3-图 6 所示，其中 MESM-5、MESM-7、MESM-9、MESM-6 000 分别表示分组数为 5、7、9 及文档数为 6 000 时 MESM 方案的实验结果。

随着文档数量的增加，生成的关键词词典也越大，通过向量空间模型构造的查询向量和索引向量维度也就越高，导致加密的时间复杂度提升。为此，可将索引与查询向量按其特征值所属类别进行分组，从而降低了加密密钥维度。如图 3 所示，对于 MRSE 方案和分组数分别为 5、7、9

的MESM方案,当文档数从1 000增加到6 000时,陷门生成时间持续增加,这是因为生成陷门的索引向量和加密密钥维度增加,但文档数量相同时,MESM方案所用时间少于MRSE方案。如文档数为6 000,分组数为5、7、9的MESM方案陷门生成时间分别为2.4s、2.1s、1.8s,而MRSE方案为2.9s。

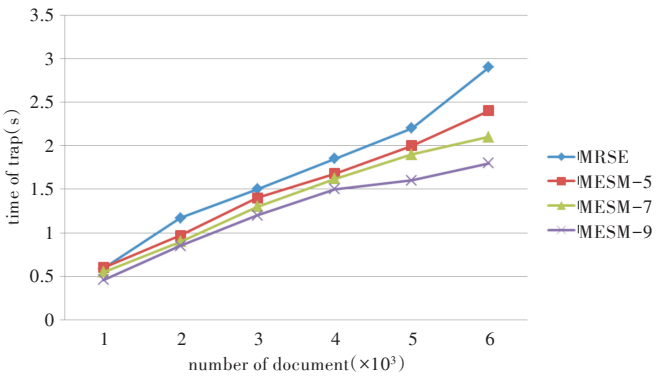

Fig. 3　Variation of trapdoor generation time with increasing number of documents in different groups

图3　不同分组时陷门生成时间随文档数量的变化

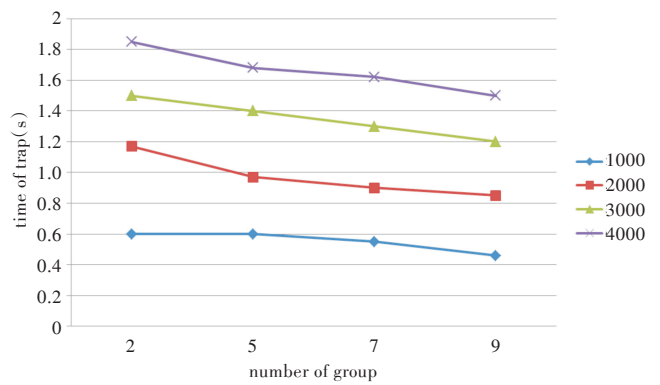

Fig. 4　Change of trapdoor generation time with different number of groups

图4　陷门生成时间随分组数的变化

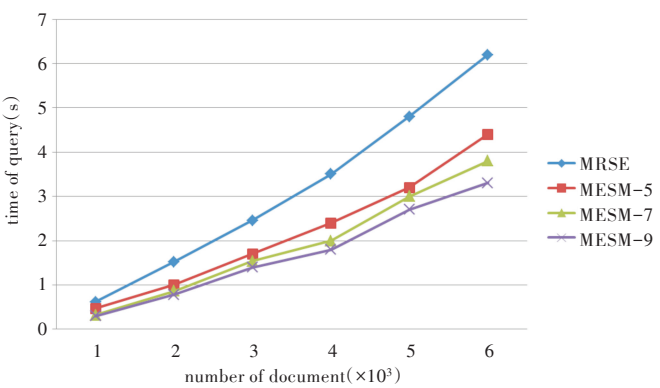

Fig. 5　Change of query time with different number of documents

图5　查询时间随文档数量的变化

图4为MESM方案在文档数为1 000~4 000,分组数从2增加到9时的陷门生成时间。从图中可以看出,在文档数一定时,陷门生成时间将随着分组数的增加而减少。如当文档数为4 000时,分组数为2、9时的陷门生成时间分别为1.85s和1.5s,减少了近20%。随着分组数的增加,陷门生成时间会进一步减少,加密效率会更高。

通过对相关性强的词进行聚类,然后将索引标记向量与查询标记向量进行匹配,以过滤相关度低的文档,对这些文档不再计算得分,从而减少了检索时间,提高了检索效率。如图5所示,随着文档数从1 000增加到6 000,MRSE方案的查询时间从0.7s增加到6.3s,近似一次系数增长,而MESM方案在分组数为5、7、9时,查询时间均远少于前者。其中,当分组数为9时,查询时间仅从0.3s增加到3.3s,查询效率提高了近一倍。当文档数量一定时,MESM方案的查询时间均少于MRSE方案,如文档数为6 000,分组数为5、7、9的MESM方案查询时间分别为4.4s、3.8s、3.3s,而MRSE方案为6.2s。随着分组数的增加,查询效率会进一步提升。

图6为在查询关键词相同而返回文档数不同的情况下,6 000篇文档在分组数为9时的MESEM方案与MRSE方案查询精度变化情况。

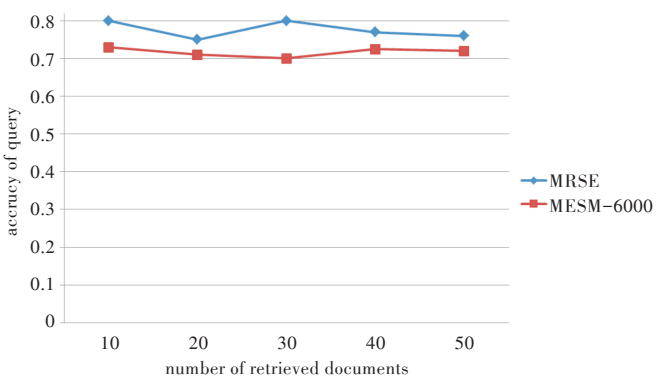

Fig. 6　Change of query accuracy with different number of returned documents

图6　查询精度随返回文档数的变化

在计算前θ个与查询关键词匹配度最高的文档得分时,通过标记向量过滤掉很多相关性弱的文档,但由于聚类时的偏差以及用户输入查询关键词的差异性,可能导致查询结果并不都符合查询要求。本实验通过公式(6)计算查询结果精度。

$$\lambda = \alpha/\beta \tag{6}$$

其中,β表示返回的文档数量,α表示包含查询关键词的文档。设置返回文档数从10增加到50,MRSE方案查询精度在0.76~0.8之间,相比之下,MESM方案精度略低,在0.7~0.72之间,但还是比较稳定。

5　结语

本文提出一种云存储环境中多关键词加密排序搜索方法——MESM,首先对关键词进行聚类,使相关性强的词排列在一起,并获得特征较集中的索引向量;然后对索引与查询向量进行标记处理,根据查询向量中的特征位置匹配相应索引向量,从而过滤掉无关文档,提高搜索效率;最

后对向量按其特征所属类别进行分组,降低了加密密钥维度,减少了加密时间。实验结果表明,本文提出的MESM方案是可行的,在相同条件下,MESM方案的查询效率高于MRSE方案。在下一步研究中,将主要致力于解决用户输入关键词相关度差异带来的查询精度下降问题。

参考文献:

[1] CAO N, YU S, YANG Z Y, et al. LT codes-based secure and reliable cloud storage service[C]//Orlando: IEEE INFOCOM, 2012.

[2] WANG C, REN K, YU S C, et al. Achieving usable and privacy-assured similarity search over outsourced cloud data [C]//Orlando: IEEE INFOCOM, 2012.

[3] GUO C, ZHUANG R, CHANG C C, et al. Dynamic multi-keyword ranked search based on bloom filter over encrypted cloud data [J]. IEEE Access, 2019,7: 35826-35837.

[4] DENG A Y, SHI J L, HUANG D, et al. KPP-CSS: a ciphertext sharing system supporting keywords privacy protection [J]. Software Guide, 2020, 19(10): 234-238.
邓安远,史姣丽,黄定,等. 支持关键词隐私保护的密文共享系统[J]. 软件导刊,2020,19(10):234-238.

[5] WANG K L. Research and improvement of symmetric searchable encryption scheme[D]. Shanghai: East China Normal University, 2020.
王康乐. 对称可搜索加密方案的研究与改进[D]. 上海:华东师范大学,2020.

[6] ZHANG M, CHEN Y, HUANG J. SE-PPFM: a searchable encryption scheme supporting privacy-preserving fuzzy multikeyword in cloud systems[J]. IEEE Systems Journal, 2020, 15(2): 2980-2988.

[7] BALLARD L, KAMARA S, MONROSE F. Achieving efficient conjunctive keyword searches over encrypted data[C]//Beijing: International Conference on Information & Communications Security, 2005.

[8] SONG D X, WAGNER D, PERRIG A. Practical techniques for searches on encrypted data[C]//Berkeley: IEEE Symposium on Security and Privacy, 2000.

[9] GOH E J. Building secure indexes for searching efficiently on encrypted compressed data[R]. Stanford, US IACR, 2003: 1-18.

[10] CHAI Q, GONG G. Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers[C]//Ottawa: IEEE International Conference on Communications, 2012.

[11] MAHAJAN N, BARKADE V. Clustering based efficient privacy preserving multi keyword search over encrypted data[C]//2018 Fourth International Conference on Computing Communication Control and Automation, 2018.

[12] CHEN R, MU Y, YANG G, et al. A new general framework for secure public key encryption with keyword search[C]//Brisbane: Australasian Conference on Information Security and Privacy, 2015.

[13] TARIQ H, AGARWAL P. Secure keyword search using dual encryption in cloud computing [J]. International Journal of Information Technology, 2018(7): 1-10.

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

[15] WANG C, CAO N, LI J, et al. Secure ranked keyword search over encrypted cloud data[C]//Genova: International Conference on Distributed Computing Systems, 2010.

[16] SAINI V, CHALLA R K, KHAN N S. An efficient multi-keyword synonym-based fuzzy ranked search over outsourced encrypted cloud data[C]//Singapore: International Conference on Advanced Computing and Communication Technologies, 2016.

[17] AHMED A, RAMACHANDRAM S, KHAN K U R. Privacy preserving dynamically indexed multi-phrase search over encrypted data [C]//Bangalore: 2018 International Conference on Advances in Computing, Communications and Informatics, 2018.

[18] CHEN L X, QIU L B, LI W B, et al. DMRS: an efficient dynamic multi-keyword ranked search over encrypted cloud data [J]. Soft Computing, 2017, 21(16): 4829-4841.

[19] FU Z, XIA L, SUN X, et al. Semantic-aware searching over encrypted data for cloud computing[J]. IEEE Transactions on Information Forensics and Security, 2018, 13(9): 2359-2371.

[20] KRISHNA C R, MITTAL S A. Privacy preserving synonym based fuzzy multi-keyword ranked search over encrypted cloud data[C]// Greater Noida: International Conference on Computing, 2017.

[21] PENG T, LIN Y, YAO X, et al. an efficient ranked multi-keyword search for multiple data owners over encrypted cloud data[J]. IEEE Access, 2018, 6: 21924-21933.

[22] HUANG X Y. Research on key technologies of ciphertext retrieval for cloud computing[D]. Nanjing: Nanjing University of Posts and Telecommunications, 2019.
黄新宇. 面向云计算的密文检索关键技术研究[D]. 南京:南京邮电大学,2019.

[23] CHEN Y, CHEN L Q, WU H. A dynamic secure searchable encryption scheme supporting priority ordering[J]. Cyberspace Security, 2020, 11(8): 51-55, 80.
陈垚,陈立全,吴昊. 一种支持优先级排序的动态安全可搜索加密方案[J]. 网络空间安全,2020,11(8): 51-55,80.

[24] XU G W, SHI C H, WANG W T, et al. Mult-keyword searchable encryption algorithm based on semantic extension[J]. Journal of Computer Research and Development, 2019, 56(10): 2193-2206.
徐光伟,史春红,王文涛,等. 基于语义扩展的多关键词可搜索加密算法[J]. 计算机研究与发展,2019,56(10):2193-2206.

[25] NAIK M P, PRAJAPATI H B, DABHI V K. A survey on semantic document clustering [C]//2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2015: 1-10.

[26] CHAI Q, GONG G. Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers [C]//Ottawa: IEEE International Conference on Communications, 2012.

[27] ZOBEL J, MOFFAT A. Exploring the similarity space[J]. ACM SIGIR Forum, 1998, 32(1): 18-34.

[28] STEINBACH M, KARYPIS G, KUMAR V. A comparison of document clustering techniques [C]//KDD Workshop on Text Mining, 2000: 525-526.

(责任编辑:黄 健)