面向高效加密云数据排序搜索的类别分组索引方法

刘良桂,孙 辉,贾会玲,张 宇
(浙江理工大学信息学院,浙江杭州 310018)

摘 要: 针对现有可搜索加密领域所遇到的加密密钥维度高、更新不灵活和搜索速度慢等问题,我们提出了一种新型类别分组索引方法——CGIM. 新方法将数据分类后,按类提取关键词建立分组索引,并采用分组加密方式实现以若干低维加密密钥代替高维加密密钥以缩短索引和查询请求的加密时间. 此外,分组索引方法的每个组向量对应不同的类别,这样不仅可以实现分类更新以改善更新文档的灵活性,而且能够在检索过程中生成针对性陷门,从而进一步提高搜索的速度和效率. 理论和实验分析表明,该方法是可行且有效的.

关键词: 可搜索加密; 分组索引; 高维密钥转换; 分类更新; 针对性搜索
中图分类号: TP309.7 **文献标识码:** A **文章编号:** 0372-2112(2019)02-0331-06
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2019.02.011

CGIM: Classificatory Group Index Method for Efficient Ranked Search of Encrypted Cloud Data

LIU Liang-gui, SUN Hui, JIA Hui-ling, ZHANG Yu
(*School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou, Zhejiang* 310018, *China*)

Abstract: In order to solve the problems such as high dimension of encrypted key, low degree of update flexibility and low search speed in the field of encrypted search, we propose a novel classificatory group index method-CGIM. The method extracts category keywords from classified data to create group index, and uses group encryption method to transform a high-dimensional secret key into several low-dimensional keys to reduce the encryption time of indexes and query requests. In addition, each group in the index is corresponding to different category. Thus, the method can not only achieve classification update to improve the flexibility of document update, but also can generate a targeted trapdoor in the retrieval process to improve the search speed and efficiency further. Through security analysis and performance test, we prove that the method is feasible and effective.

Key words: encrypted search; group index; key transformation; category update; targeted search

1 引言

随着计算机技术的快速发展,人们对云存储服务提出了更高要求. 在云计算时代,很多私人用户和企业通过将庞杂的数据存储到云端来使用快捷的云服务[1]. 随着云服务的推广应用,许多云消费者担心存储到云端的隐私数据会被泄露[2]. 因此,为保护隐私信息安全,数据在存储到云端之前需要进行加密[3],然而这同时也将增加数据利用的难度,因为加密数据并不像明文数据那样便于检索[4]. 当前,在快速增长的云存储需求和不可信的云存储背景下,这已成为一个极具挑战性的问题.

在应对急剧增长的数据量的问题上,现有密文检索方案时间复杂度高,更新不灵活. 为此,在处理大量数据时,可以通过特征提取、计算关键词权重和聚类将内容相似的文档归为一类[5,6],但是分类后如何在确保信息安全的同时提高搜索效率已成为一个关键问题. 此外,在可搜索加密技术中,加密后的数据和索引在存储到远程服务

收稿日期:2018-03-12;修回日期:2018-08-25;责任编辑:蓝红杰
基金项目:国家自然科学基金(No. 61002016, No. 61711530653);国家自然科学基金委员会——中国民航联合研究基金(No. U1533133);教育部人文社科项目(No. 15YJCZH095);中国国家留学基金(No. 201708330439)

器之后,可以用特定的陷门对其进行搜索[7,8].然而,现有的密文排序搜索方案[9~11]均存在加密密钥维度高,执行创建和更新索引时间长以及检索效率低等问题.因此,如何针对用户需求,寻找一种既能降低计算开销又能提高检索效率的方案成为一个急需解决的问题.

我们在 MRSE(Multi-keyword ranked search over encrypted cloud data)方案[12]基础上提出了一种面向高效加密云数据排序搜索的类别分组索引方法 CGIM.本文的贡献总结如下:

(1)本文首次通过对文档分类以在索引中建立组向量.组向量的创建不仅可以实现以若干低维度加密密钥代替高维度的加密密钥以加速加密过程,而且提高了更新操作的灵活性.

(2)在 CGIM 的查询过程中,我们进一步引入“针对性搜索”方法来提高搜索的速度和效率.

2 相关工作

2000 年,Song 等[13]最早引入使用不可信云服务器对加密数据进行远程搜索的技术,并提出一种安全的可搜索加密方案,但是全文搜索效率较低.Li 等[14]设计了基于通配符的技术,构造出高存储效率的模糊关键词集,首次正式解决了支持隐私保护的模糊搜索问题.Curtmola 等[15]给出了更强的对称可搜索加密安全定义,并考虑到查询方案的可扩展性.

随着可搜索加密技术发展,又有很多学者提出更加高效的搜索方案.Wang 等[10]首次解决了加密数据排序搜索问题,通过搜索排序返回匹配文件,极大增强了系统的可用性.Fu 等[16]首次研究并解决了加密数据个性化多关键词排序搜索问题.Zhang 等[17]提出多所有者模型中多关键字排序搜索方案.Wang 等[18]提出了一种基于符号树的可验证模糊关键词搜索方案,该方案具有搜索结果的可验证性.

以上相关工作虽然在不同方面对加密数据搜索技术进行完善和改进,但是在大数据的背景下,关键词词典尺寸大,因而加密密钥维度高,直接采用高维密钥加密将会带来加密计算量大、时间复杂度高等问题,因此设计新的方法迫在眉睫.

3 问题叙述

3.1 系统模型

可搜索加密系统由三部分组成:数据所有者,云服务器和授权用户.它们之间关系如图 1 所示.数据所有者首先为每篇文档创建索引并加密;然后将加密后的文档和索引上传到云服务器保存.授权用户输入关键词查询时,先通过搜索控制操作生成陷门;再将陷门提交给云服务器进行检索.云服务器接收到陷门后,计算查询陷门与每篇文档索引的内积,并按内积数值大小对搜索结果排序返回给授权用户.授权用户接收加密数据并通过获取控制操作解密数据.

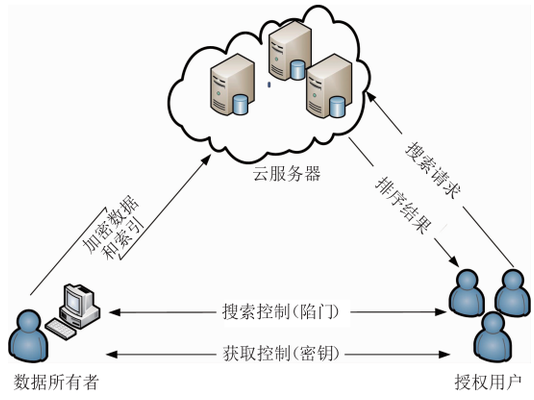

图1 可搜索加密系统模型

3.2 符号说明

本文使用的符号及说明如表 1 所示.

表 1 符号说明

符号	符号说明
F	明文文档集,表示为 $F=\{F_1,F_2,\cdots,F_m\}$
C	加密文档集,表示为 $C=\{C_1,C_2,\cdots,C_m\}$
n_j	第 j 类关键词总数
$tf_{ij}^{(x_j)}$	第 j 类第 x_j 个关键词在 F_i 中标准化词频
$idf_j^{(x_j)}$	第 j 类第 x_j 个关键词的标准化反词频
W_j	第 j 个类关键词集合
W	关键词集,表示为 $W=\{W_1,W_2,\cdots,W_k\}$
$\boldsymbol{p}_{ij}$	文档 i 的第 j 个组向量
$\boldsymbol{p}_i$	文档 i 的索引向量
P	明文索引集,表示为 $P=\{\boldsymbol{p}_1,\boldsymbol{p}_2,\cdots,\boldsymbol{p}_m\}$
I	加密索引集,表示为 $I=\{I_1,I_2,\cdots,I_m\}$
$\boldsymbol{q}_j$	查询请求中第 j 个组向量
$\boldsymbol{q}$	查询请求向量
T	查询陷门

3.3 词语解释

类关键词集和关键词集:文档分类后,每类文档分别提取关键词构成各类文档的类关键词集;类关键词集依次排列构成关键词集.如果有 k 类,则关键词集就包含 k 个类关键词集.

文档得分:每个关键词对一篇文档的重要程度是不同的.为更好满足用户检索需求,我们引入文档得分.文档得分是排序和返回搜索结果的依据.词频和反词频(TF-IDF)是计算文档得分一种常用方法.我们用式(1)中标准化的 TF · IDF 来计算提交查询请求 $\boldsymbol{q}$ 时文档 F_i 的得分[12].

$$\mathrm{Score}(F_i,\boldsymbol{q})=\frac{1}{|F_i|}\sum_{w_b\in\bar{W}}(1+\ln f_{i,b})\cdot\ln\left(1+\frac{m}{f_b}\right) \tag{1}$$

其中,$f_{i,b}$表示关键词 w_b 在文档 F_i 中出现的次数,$\widetilde{W}$ 表示查询请求中关键词的集合,f_b 表示包含关键词 w_b 的文档数,m 表示文档总数,$|F_i|$是文档 F_i 的欧氏长度,表示为 $\sqrt{\sum_{b=1}^{n}(1+\ln f_{i,b})^2}$. 创建索引时,我们将组向量每一维设置成对应关键词的标准化词频;创建查询请求时,我们将组向量每一维设置成对应关键词的标准化反词频.

4　类别分组索引方法

以下将分六个部分详细介绍所提出的类别分组索引方法.

4.1　文档分类

根据文档分类方法[6],首先提取聚类关键词(表示文档类别特征的关键词),然后计算关键词权重并创建特征空间向量,最后用"K-Means"聚类把文档分成 k 类. 选择参数 k 时,尽量让每类文档数分布均匀,以使各项性能达到最优.

4.2　生成密钥

在这一阶段,数据所有者生成密钥 $\boldsymbol{M}_1,\boldsymbol{M}_2$ 和分割指示器 $\boldsymbol{S}$,它们分别可以表示为

$$\boldsymbol{M}_1=\begin{pmatrix}\boldsymbol{M}_{11}&0&\cdots&0\\0&\boldsymbol{M}_{12}&\cdots&0\\\vdots&\vdots&\ddots&0\\0&0&\cdots&\boldsymbol{M}_{1k}\end{pmatrix},$$

$$\boldsymbol{M}_2=\begin{pmatrix}\boldsymbol{M}_{21}&0&\cdots&0\\0&\boldsymbol{M}_{22}&\cdots&0\\\vdots&\vdots&\ddots&0\\0&0&\cdots&\boldsymbol{M}_{2k}\end{pmatrix},$$

$\boldsymbol{S}=(\boldsymbol{s}_1,\boldsymbol{s}_2,\cdots,\boldsymbol{s}_k)$. 其中 $\boldsymbol{M}_{1j}$和 $\boldsymbol{M}_{2j}(j=1,2,\cdots,k)$是$(n_j+u+1)\times(n_j+u+1)$维随机可逆的块矩阵,$\boldsymbol{s}_j\in\{0,1\}^{(n_j+u+1)}(j=1,2,\cdots,k)$,$n_j$ 是第 j 类提取的关键词总数,$u+1$ 是扩展的维度.

4.3　创建分组索引

第一步:数据所有者首先创建类关键词集和关键词集,然后分别计算类关键词集中的关键词在文档 F_i 中的标准化词频,生成索引 $\boldsymbol{p}_i$. $\boldsymbol{p}_i$ 可以表示为 $\boldsymbol{p}_i=(\boldsymbol{p}_{i1}^{\mathrm{T}},\cdots,\boldsymbol{p}_{ij}^{\mathrm{T}},\cdots,\boldsymbol{p}_{ik}^{\mathrm{T}})$,其中 $\boldsymbol{p}_{ij}$是组向量,它的每一维都是第 j 类的类关键词集中关键词在文档 F_i 中标准化词频.

第二步:对索引 $\boldsymbol{p}_i$ 中每个组向量 $\boldsymbol{p}_{ij}(j=1,2,\cdots,k)$进行维度扩展,从 n_j 维扩展到 n_j+u+1 维. 并且将每组的 $n_j+l(l\in[1,u])$维设置成任意随机数 $\varepsilon_{ij}^{(l)}$,将 n_j+u+1 维设置成常数 1. 扩展后的组向量可以表示为 $\boldsymbol{p}_{ij\sim}=(tf_{ij}^{(1)},\cdots,tf_{ij}^{(n_j)},\varepsilon_{ij}^{(1)},\cdots,\varepsilon_{ij}^{(u)},1)^{\mathrm{T}}$. 其中,$\varepsilon_{ij}^{(1)},\cdots,\varepsilon_{ij}^{(u)}$表示任意随机数且它们服从同一均匀分布 $U(\mu'-c,\mu'+c)$. 最终,扩展后的索引可以表示为 $\boldsymbol{p}_{i\sim}=(\boldsymbol{p}_{i1\sim}^{\mathrm{T}},\cdots,\boldsymbol{p}_{ij\sim}^{\mathrm{T}},\cdots,\boldsymbol{p}_{ik\sim}^{\mathrm{T}})$.

第三步:对索引 $\boldsymbol{p}_{i\sim}$ 中的每个组向 $\boldsymbol{p}_{ij\sim}(j=1,2,\cdots,k)$进行随机分割,分割成 $\boldsymbol{p}'_{ij\sim}$和 $\boldsymbol{p}''_{ij\sim}$. 分割规则如下:如果 $\boldsymbol{s}_j[n]$等于0,则将 $\boldsymbol{p}'_{ij\sim}[n]$和 $\boldsymbol{p}''_{ij\sim}[n]$设置成两个相等的数且都等于 $\boldsymbol{p}_{ij\sim}[n]$;如果 $\boldsymbol{s}_j[n]$等于1,则将 $\boldsymbol{p}'_{ij\sim}[n]$和 $\boldsymbol{p}''_{ij\sim}[n]$设置成两个不相等且不为零的随机数,且它们的和等于 $\boldsymbol{p}_{ij\sim}[n]$. 分割后的索引可以分别表示为 $\boldsymbol{p}'_{i\sim}=(\boldsymbol{p}'^{\mathrm{T}}_{i1\sim},\cdots,\boldsymbol{p}'^{\mathrm{T}}_{ij\sim},\cdots,\boldsymbol{p}'^{\mathrm{T}}_{ik\sim})$和 $\boldsymbol{p}''_{i\sim}=(\boldsymbol{p}''^{\mathrm{T}}_{i1\sim},\cdots,\boldsymbol{p}''^{\mathrm{T}}_{ij\sim},\cdots,\boldsymbol{p}''^{\mathrm{T}}_{ik\sim})$.

第四步:用密钥 $\boldsymbol{M}_1$ 和 $\boldsymbol{M}_2$ 对分割后的索引 $\boldsymbol{p}'_{i\sim}$和 $\boldsymbol{p}''_{i\sim}$进行加密,加密过程为 $\boldsymbol{p}'_{i\sim}\boldsymbol{M}_1=(\boldsymbol{p}'^{\mathrm{T}}_{i1\sim}\boldsymbol{M}_{11},\cdots,\boldsymbol{p}'^{\mathrm{T}}_{ij\sim}\boldsymbol{M}_{21},\cdots,\boldsymbol{p}'^{\mathrm{T}}_{ik\sim}\boldsymbol{M}_{k1})$和 $\boldsymbol{p}''_{i\sim}\boldsymbol{M}_2=(\boldsymbol{p}''^{\mathrm{T}}_{i1\sim}\boldsymbol{M}_{12},\cdots,\boldsymbol{p}''^{\mathrm{T}}_{ij\sim}\boldsymbol{M}_{22},\cdots,\boldsymbol{p}''^{\mathrm{T}}_{ik\sim}\boldsymbol{M}_{k2})$. 因此最终加密索引 $I_i=\{\boldsymbol{p}'_{i\sim}\boldsymbol{M}_1,\boldsymbol{p}''_{i\sim}\boldsymbol{M}_2\}$.

4.4　创建陷门

第一步:授权用户输入关键词查询时,首先生成查询请求 $\boldsymbol{q}$,查询请求 $\boldsymbol{q}$ 也是由 k 个组向量构成且与类关键词集相对应,可以表示为 $\boldsymbol{q}=(\boldsymbol{q}_1,\cdots,\boldsymbol{q}_j,\cdots,\boldsymbol{q}_k)$. 组向量 $\boldsymbol{q}_j(j=1,2,\cdots,k)$每一维的值规定如下:如果查询关键词与第 j 类关键词集中关键词相匹配,则将相应维度设置成匹配关键词的标准化反词频;如果查询关键词与第 j 类关键词集中关键词不匹配,则将相应维度设置成零.

第二步:对查询请求 $\boldsymbol{q}$ 中每个组向量 $\boldsymbol{q}_j(j=1,2,\cdots,k)$进行维度扩展,从 n_j 维扩展到 n_j+u+1 维,可以表示为 $\boldsymbol{q}_{j\sim}$. 扩展规则如下:如果查询关键词不在第 j 类关键词集中,则 $\boldsymbol{q}_{j\sim}$的所有维都将设置成 0. 如果查询关键词在第 j 类关键词集中,则首先从 $\boldsymbol{q}_{j\sim}$的 n_j+1 维到 n_j+u 维随机选择 v 个位置设置成 1 其余设置成 0;然后将(n_j+u+1)维设置成随机数 t,并将其余维乘以随机数 r. 扩展后组向量表示为 $\boldsymbol{q}_{j\sim}=(r\cdot idf_j^{(1)},\cdots,r\cdot idf_j^{(n_j)},r\cdot b_j^{(1)},\cdots,r\cdot b_j^{(u)},t)^{\mathrm{T}}$, 其中 $b_j^{(1)},b_j^{(2)},\cdots,b_j^{(u)}$ 为0 和1 随机数,且有 v 个为1,其余为0. 扩展后的查询请求可以表示为 $\boldsymbol{q}_{\sim}=(\boldsymbol{q}_{1\sim}^{\mathrm{T}},\cdots,\boldsymbol{q}_{j\sim}^{\mathrm{T}},\cdots,\boldsymbol{q}_{k\sim}^{\mathrm{T}})$.

第三步:根据分割指示器 $\boldsymbol{S}$,对查询请求 $\boldsymbol{q}_{\sim}$中每个组向量 $\boldsymbol{q}_{j\sim}(j=1,2,\cdots,k)$进行随机分割,分割成 $\boldsymbol{q}'_{j\sim}$和 $\boldsymbol{q}''_{j\sim}$. 分割规则如下:当查询关键词不在第 j 类关键词集中,则 $\boldsymbol{q}'_{j\sim}$和 $\boldsymbol{q}''_{j\sim}$都将设置成零向量. 当查询关键词在第 j 类关键词集中,则如果 $\boldsymbol{s}_j[n]$等于 0 时,$\boldsymbol{q}'_{j\sim}[n]$和 $\boldsymbol{q}''_{j\sim}[n]$将设置成任意两个不相等且不为零的随机数,且它们的和等于 $\boldsymbol{q}_{j\sim}[n]$;如果 $\boldsymbol{s}_j[n]$等于1 时,则将 $\boldsymbol{q}'_{j\sim}[n]$和 $\boldsymbol{q}''_{j\sim}[n]$设置成两个相等的数,且它们都等于 $\boldsymbol{q}_{j\sim}[n]$. 最后,分割后的查询请求可以分别表示为 $\boldsymbol{q}'_{\sim}=$

$(\boldsymbol{q}'^{\mathrm{T}}_{1\sim},\cdots,\boldsymbol{q}'^{\mathrm{T}}_{j\sim},\cdots,\boldsymbol{q}'^{\mathrm{T}}_{k\sim})$, $\boldsymbol{q}''_{\sim}=(\boldsymbol{q}''^{\mathrm{T}}_{1\sim},\cdots,\boldsymbol{q}''^{\mathrm{T}}_{j\sim},\cdots,\boldsymbol{q}''^{\mathrm{T}}_{k\sim})$.

第四步:用密钥 $\boldsymbol{M}_1$ 和 $\boldsymbol{M}_2$ 对查询请求 $\boldsymbol{q}'_{\sim}$ 和 $\boldsymbol{q}''_{\sim}$ 加密生成陷门 $\boldsymbol{T}$. $\boldsymbol{T}=\{\boldsymbol{M}_1^{-1}\boldsymbol{q}'^{\mathrm{T}}_{\sim},\boldsymbol{M}_2^{-1}\boldsymbol{q}''^{\mathrm{T}}_{\sim}\}$,加密过程可以表示为 $\boldsymbol{M}_1^{-1}\boldsymbol{q}'^{\mathrm{T}}_{\sim}=(\boldsymbol{M}_{11}^{-1}\boldsymbol{q}'_{1\sim},\cdots,\boldsymbol{M}_{j1}^{-1}\boldsymbol{q}'_{j\sim},\cdots,\boldsymbol{M}_{k1}^{-1}\boldsymbol{q}'_{k\sim}{}^{\mathrm{T}})$, $\boldsymbol{M}_2^{-1}\boldsymbol{q}''^{\mathrm{T}}_{\sim}=(\boldsymbol{M}_{12}^{-1}\boldsymbol{q}''_{1\sim},\cdots,\boldsymbol{M}_{j2}^{-1}\boldsymbol{q}''_{j\sim},\cdots,\boldsymbol{M}_{k2}^{-1}\boldsymbol{q}''_{k\sim})^{\mathrm{T}}$.

4.5 查询和解密

授权用户提交陷门给云服务器查询. 云服务器接收到陷门后,首先计算查询陷门中不为零的组向量与每篇文档索引中对应的组向量的内积,即执行“针对性搜索”;然后将前 k 篇得分最高的加密文档返回给授权用户. 我们以对第 j 类文档查询为例,介绍检索过程如下:

$$\begin{aligned}I_i\cdot T&=\{\boldsymbol{p}'_{i\sim}\boldsymbol{M}_1,\boldsymbol{p}''_{i\sim}\boldsymbol{M}_2\}\cdot\{\boldsymbol{M}_1^{-1}\boldsymbol{q}'^{\mathrm{T}}_{\sim},\boldsymbol{M}_2^{-1}\boldsymbol{q}''^{\mathrm{T}}_{\sim}\}\\&=\boldsymbol{p}'^{\mathrm{T}}_{ij\sim}\boldsymbol{q}'_{j\sim}+\boldsymbol{p}''^{\mathrm{T}}_{ij\sim}\boldsymbol{q}''_{j\sim}\\&=\boldsymbol{p}^{\mathrm{T}}_{ij\sim}\boldsymbol{q}_{j\sim}\\&=r(\boldsymbol{p}_{ij}^{\mathrm{T}}\boldsymbol{q}_j+\sum\varepsilon_i^{(v)})+t\\&=r(\mathrm{Score}(F_i,\boldsymbol{q})+\sum\varepsilon_i^{(v)})+t\end{aligned}$$

授权用户在接收到云服务器返回的加密文档之后,首先通过图 1 中的获取控制操作来获取密文文档的解密密钥 sk,然后用该密钥 sk 将密文文档解密成明文文档.

4.6 索引更新

更新操作包括三部分,增加新文档、修改现有文档和删除现有文档. CGIM 在执行更新操作时,首先判断更新文档的类别;然后对每个索引中与这些类对应的组向量进行更新,不对应的组向量将保持不变. 我们以更新第 j 类为例,介绍更新过程如下.

第一步:对第 j 类更新后的文档重新提取关键词构建第 j 类的类关键词集,然后根据第 j 类提取关键词数量和扩展维度重新生成加密密钥 $\boldsymbol{M}_{j1}$ 和 $\boldsymbol{M}_{j2}$,以及分割指示器 $\boldsymbol{s}_j$.

第二步:重新构建所有文档索引的第 j 类组向量,组向量长度与更新后的关键词总数相同. 每个索引的对应组向量创建规则如下:如果文档 F_i 属于第 j 类,则重新计算索引 $\boldsymbol{p}_i$ 的第 j 个组向量中每一维关键词的标准化词频. 如果文档 F_i 不属于第 j 类,则 $\boldsymbol{p}_i$ 的第 j 个组向量的每一维都将设置成零. 随后,对 $\boldsymbol{p}_i$ 的第 j 个组向量进行维度扩展、随机分割和加密生成加密组索引. 特别地,当添加新文档时,还需计算这些新文档索引的其它组向量,当删除文档时,也要删除对应文档索引.

第三步:更新每个索引中第 j 组的组向量,然后上传到云服务器.

需要注意,更新文档会影响现有词典中关键词 IDF 值的变化. 数据所有者添加、修改或删除文档都需要重新计算所有关键词的 IDF 值,并且将它们发送给授权用户. 已有研究表明[19],当添加或删除文档数量在 300 之内时,关键词的 IDF 值将不会发生太大变化,因此造成的误差可忽略,此时,数据所有者无需重新计算所有关键词 IDF 值. 当更新操作导致关键词 IDF 值变化较大时,为减小误差,数据所有者可灵活选择关键词 IDF 值变化较大的进行更新.

5 实验分析

我们用 RFC(Request For Comments)[20]数据作为测试数据集,采用 Java 语言来实现实验系统,将 Windows 7 服务器设为实验环境,其中 CPU 为英特尔酷睿 i5(2.5GHz)处理器.

实验过程如下:首先采用文档聚类方法将文档分成两类和四类. 然后针对 MRSE[12]和 CGIM,从每篇文档中提取 3 个关键词构成关键词集和类关键词集,关键词总数为 15630 个. 给定系统参数 $\omega=2000$,在 MRSE 中,每个索引采用 $2\omega+1$ 个维度扩展;在 CGIM 中,索引的每个组向量采用 $2\omega/k+1$ 个维度扩展,k 为分组数. 最后进行建立索引和创建陷门验证加密速度,进行更新索引验证更新性能、进行搜索验证搜索速度,并分别与 MRSE 方案对比. 实验结果如图 2,3,4,5 所示,图中 CGIM-2(1)和 CGIM-4(1)表示在类别分组索引方法中将文档分成两类、四类时,每组关键词数近似均匀时的实验图;CGIM-2 和 CGIM-4 表示将文档分成两类和四类时,每组关键词数差异较大时的实验图.

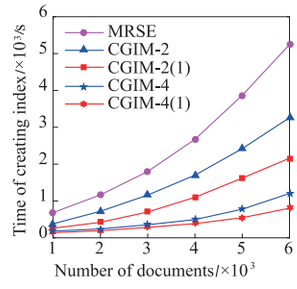

图2 创建索引加密用时

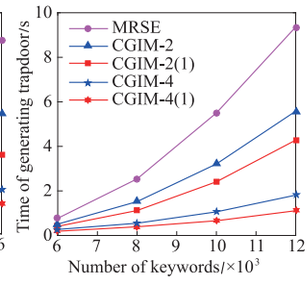

图3 创建陷门加密用时

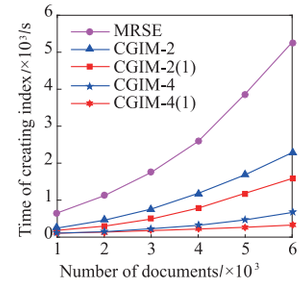

图4 更新索引加密用时

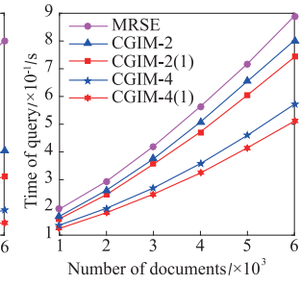

图5 检索用时

创建索引和陷门加密用时如图 2 和图 3 所示. 一方面,文档数量越多,关键词词典的尺寸越大,对应的索引

维度和加密矩阵维度也就越高,二者乘积的计算量也随之剧增. 因此,两种方案用时随文档数量呈增长趋势. 另一方面,对于相同数量文档,MRSE 和 CGIM 的关键词词典长度近似相等,即 $N \approx \sum_{j=1}^{k} n_j$,且总扩展维度相同,满足 $U = \sum_{j=1}^{k} u$. MRSE 加密 m 篇文档索引需要计算 $2m(N+U+1)^2$ 个乘法运算和 $2m(N+U)(N+U+1)$ 个加法运算. 由于$(N+U)$远大于 1,所以加密计算总量近似为 $4m(N+U)^2$,也可近似表示成 $4m(\sum_{j=1}^{k}(n_j+u))^2$. 同理,CGIM 加密 m 篇文档索引的计算总量近似为 $4m\sum_{j=1}^{k}(n_j+u)^2$. 当 $k>1$ 时,MRSE 加密计算量大于 CGIM 加密计算量,所以 CGIM 创建索引和查询陷门的加密时间均小于 MRSE. 此外,根据柯西不等式可知 $4m\sum_{j=1}^{k}(n_j+u)^2 \geq 4m(\sum_{j=1}^{k}(n_j+u))^2/k$,因此各组关键词数分布越均匀,加密计算量越接近最小值 $4m(N+U)^2/k$;且分组数越多,加密用时越小,如图中 CGIM-2(1)和 CGIM-4(1). 当每组关键词数差异增大时,加密用时会在原变化趋势基础上增大,如图中 CGIM-2 和 CGIM-4.

更新加密时间随文档数量变化关系如图 4 所示. 首先,文档数越多,关键词词典尺寸越大,索引和密钥维度也就越高,从而更新加密用时越长;其次,MRSE 更新索引需要重新加密,所以更新加密计算量仍近似为 $4m(\sum_{j=1}^{k}(n_j+u))^2$;CGIM 仅更新部分包含更新关键词的组向量,当更新 c(实验中取 $c=1$)类文档时,更新加密计算量近似为 $4m\sum_{c}(n_j+u)^2$. 所以更新相同的文档时,CGIM 平均更新加密用时小于 MRSE. 此外,每组关键词数分布越均匀,平均更新加密用时越小;且分组数越多,更新加密用时越少,如图中 CGIM-2(1)和 CGIM-4(1). 当每组关键词数分布差异增大时,平均更新加密用时也会在原变化趋势基础上增大,如图中 CGIM-2 和 CGIM-4.

图 5 给出搜索时间随文档数量变化关系图. 从图中可以看出,搜索用时随文档数呈增长趋势. 主要是因为两种方案检索过程均需计算查询陷门与每篇文档索引的向量内积;且文档数越多,计算量越大,因而用时越长. 对于相同数量文档,MRSE 查询计算量为 $4m\sum_{j=1}^{k}(n_j+u)$;CGIM 采用针对性搜索方法,省去不相关组向量的计算. 当查询关键词包含在 c(实验中取 $c=1$)类的类关键词集中,查询计算量近似为 $4m\sum_{c}(n_j+u)$. 因此,输入相同关键词查询时,CGIM 平均查询用时不大于 MRSE. 此外,每组关键词数分布越均匀,平均查询用时越少;且随分组数增大而减小,如图中 CGIM-2(1)和 CGIM-4(1). 当每组关键词数分布差异增大时,平均查询用时会在原变化趋势基础上增大,如图中 CGIM-2 和 CGIM-4.

6　结论

本文提出的类别分组索引方法将数据分类后建立分组索引,实现以若干低维加密密钥代替高维加密密钥,缩短了加密时间,同时提高了更新操作的灵活性. 在 CGIM 搜索过程中,我们引入“针对性搜索”方法,进一步提高检索的速度和效率.

参考文献

[1] Armbrust M, Fox A, Griffith R, et al. A view of cloud computing[J]. Communications of the ACM, 2010, 53(4): 50 – 58.

[2] Heiser J, Nicolett M. Assessing the security risks of cloud computing[J]. Gartner Report, 2008, 27: 29 – 52.

[3] 王小峰,陈培鑫,周寰,等. 一种可信安全的层次式基于身份加密系统[J]. 电子学报,2016,44(7):1521 – 1529.
WANG Xiaofeng, CHEN Peixin, ZHOU Huan, et al. A trustworthy and secure hierarchical identity-based encryption system[J]. Acta Electronica Sinica, 2016, 44(7): 1521 – 1529. (in Chinese)

[4] Wang C, Ren K, Yu S, et al. Achieving usable and privacy-assured similarity search over outsourced cloud data[A]. INFOCOM, 2012 Proceedings IEEE[C]. Orlando, FL, USA: IEEE, 2012. 451 – 459.

[5] Sanchez D, Batet M, Isern D, et al. Ontology-based semantic similarity: A new feature-based approach[J]. Expert Systems with Applications, 2012, 39(9): 7718 – 7728.

[6] Desai S S, Laxminarayana J A. WordNet and semantic similarity based approach for document clustering[A]. Computation System and Information Technology for Sustainable Solutions(CSITSS), International Conference on[C]. Bangalore, India: IEEE, 2016. 312 – 317.

[7] Li M, Yu S, Cao N, et al. Authorized private keyword search over encrypted data in cloud computing[A]. Distributed Computing Systems(ICDCS), 2011 31st International Conference on[C]. Minneapolis, MN, USA: IEEE, 2011. 383 – 392.

[8] 袁科,刘哲理,贾春福,等. 一对多场景下的公钥时控性可搜索加密[J]. 电子学报,2015,43(4):760 – 768.
YUAN Ke, Liu Zheli, JIA Chunfu, et al. Public key timed-release searchable encryption in one-to-many scenarios[J]. Acta Electronica Sinica, 2015, 43(4): 760 – 768. (in Chi-

nese)
[9] Fu Z, Sun X, Linge N, et al. Achieving effective cloud search services: multi-keyword ranked search over encrypted cloud data supporting synonym query[J]. IEEE Transactions on Consumer Electronics, 2014, 60(1): 164 – 172.
[10] Wang C, Cao N, Li J, et al. Secure ranked keyword search over encrypted cloud data[A]. Distributed Computing Systems(ICDCS), 2010 IEEE 30th International Conference on[C]. Genova, Italy: IEEE, 2010. 253 – 262.
[11] 杨旸,杨书略,柯闽. 加密云数据下基于 Simhash 的模糊排序搜索方案[J]. 计算机学报,2017,40(2):431 – 444. YANG Yang, YANG Shulue, KE Min. Ranked fuzzy keyword search based on simhash over encrypted cloud data[J]. Chinese Journal of Computers, 2017, 40(2): 431 – 444. (in Chinese)
[12] 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.
[13] Song D X, Wagner D, Perrig A. Practical techniques for searches on encrypted data[A]. Security and Privacy, 2000, S&P 2000, Proceedings, 2000 IEEE Symposium on[C]. Berkeley, CA, USA: IEEE, 2000. 44 – 55.
[14] Li J, Wang Q, Wang C, et al. Fuzzy keyword search over encrypted data in cloud computing[A]. INFOCOM, 2010 Proceedings IEEE[C]. San Diego, CA, USA: IEEE, 2010. 1 – 5.
[15] 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.
[16] Fu Z, Ren K, Shu J, et al. Enabling personalized search over encrypted outsourced data with efficiency improvement[J]. IEEE Transactions on Parallel and Distributed Systems, 2016, 27(9): 2546 – 2559.
[17] Zhang W, Lin Y, Xiao S, et al. Privacy preserving ranked multi-keyword search for multiple data owners in cloud computing[J]. IEEE Transactions on Computers, 2016, 65(5): 1566 – 1577.
[18] Wang J, Chen X, Ma H, et al. A verifiable fuzzy keyword search scheme over encrypted data[J]. J Internet Serv Inf Secur, 2012, 2(1/2): 49 – 58.
[19] Xia Z, Wang X, Sun X, et al. A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data[J]. IEEE Transactions on Parallel and Distributed Systems, 2016, 27(2): 340 – 352.
[20] Internet Society. Request for Comments[EB/OL]. http://www.ietf.org/rfc.html, 1969-04-07/2017-12-17.

作者简介

刘良桂 男,1975 年出生于江西泰和. 南京邮电大学博士. 研究方向为分布式计算、复杂网络、网络安全和自然计算.
E-mail: liangguiliu@126.com

孙 辉 男,1989 年出生于江苏徐州. 浙江理工大学信息学院硕士研究生,研究方向为云计算和信息安全.