基于分块的移动边缘计算密文检索方法

王娜[1]，郑坤[1]，付俊松[2]，李剑[1]
（1. 北京邮电大学计算机学院，北京 100876；2. 北京邮电大学网络安全学院，北京 100876）

摘 要：针对云计算密文检索方案的效率问题，提出了基于分块的移动边缘计算密文检索方案。首先，引入了边缘服务器计算文档相似性得分，从而减少了云服务器的计算开销，提升了云服务器的处理效率；其次，在 MRSE 方案基础上通过分块方法过滤掉大部分与查询无关的关键词，从而提高了计算文档相似性得分的效率。理论分析和实验结果表明，所提方案在已知背景威胁模型下是安全的，与现有方案相比，所提方案在具有相同安全性的同时具有更高的检索效率。

关键词：云计算；移动边缘计算；密文检索；隐私保护
中图分类号：TP309
文献标识码：A
doi: 10.11959/j.issn.1000−436x.2020142

Method of ciphertext retrieval in mobile edge computing based on block segmentation

WANG Na[1], ZHENG Kun[1], FU Junsong[2], LI Jian[1]
1. School of Computer, Beijing University of Posts and Telecommunications, Beijing 100876, China
2. School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract: Aiming at the efficiency of cloud computing ciphertext retrieval scheme, a method of ciphertext retrieval in mobile edge computing based on block segmentations was proposed. Firstly, the edge server was introduced to calculate the document similarity score, thereby the computational cost of cloud server was reduced and the processing efficiency of cloud server was improved. Secondly, most keywords that are not related to the query were filtered out by a method of block segmentations based on the MRSE scheme, thereby the efficiency of calculating the document similarity score was improved. Theoretical analysis and experimental results show that the solution is safe under the known background threat model. Compared with the existing scheme, the proposed scheme has the same security and higher retrieval efficiency.
Key words: cloud computing, mobile edge computing, ciphertext retrieval, privacy protection

1 引言

随着云计算应用的逐渐普及，越来越多的企业和用户选择将数据存储在云服务器上，云计算极大地方便了数据的存储和使用[1-2]，但是随着时代的进步，海量数据带来的通信负担和计算压力使现有的云计算在很大程度上已经不能够满足用户的需求。与此同时，物联网的崛起和 5G 时代的到来，带动了移动边缘计算和雾计算的快速发展[3-8]。移动边缘计算和雾计算通过在网络的边缘设置一些边缘服务器，可以实现快速响应用户的需求或是提前进行一部分数据处理来减轻云服务器的通信负担或者计算压力。

移动边缘计算上的密文检索与云计算上的密文检索大体上是相似的，不同之处在于边缘计算环境下，若干个移动边缘服务器分布在网络的边缘，数据使用者选择最近的边缘服务器发送查询请求，

收稿日期：2020–04–14；**修回日期**：2020–06–11
通信作者：付俊松，fujs@bupt.edu.cn
基金项目：北京市自然科学基金资助项目（No.4204107）
Foundation Item: The Natural Science Foundation of Beijing (No.4204107)

由边缘服务器负责计算文档与查询的相关性，将排好序的查询结果返回给云服务器，云服务器再将对应的加密文档发送给用户，这种方式减轻了云服务器的计算压力。同云服务器一样，边缘计算也被认为是半可信的、“诚实而好奇”的实体[5]，云服务器上的数据隐私安全问题[9-11]在边缘服务器上同样存在，数据需要在上传给云服务器、边缘服务器之前进行加密，这样一来便限制了数据的使用[12-13]，所以寻找一个可以应用于移动边缘计算的密文检索方案是十分有必要的。

2 相关工作

现有的可搜索加密方案大多是基于云计算环境的研究，在移动边缘计算上的方案很少。而边缘计算可以认为是云计算的拓展，所以本文方案主要是与现有的云计算上的可搜索加密方案进行比较。

早些时候，可搜索加密方案集中在单关键词检索的研究[14-16]，由于单关键词的搜索结果过于粗糙，用户为了获得更加准确的检索结果，通常会输入多个关键词进行检索，尤其是在云计算的按使用收费机制下，返回高相关性的文档是有必要的，不仅节省了流量，也方便了用户的使用，基于这些原因，支持多关键字排序的密文检索方案陆续被提出。

Cao 等[17]提出了安全的、多关键字可排序的密文检索方案（MRSE, multi-keyword ranked search over encrypted cloud data），该方法基于安全 k 最近邻（kNN，k-nearest neighbor）技术[18]，利用可逆矩阵来加密索引和查询向量，通过计算查询向量与文档向量的内积得到文档在本次查询下的相似性得分，但该方法对每一个无关的关键词都进行了计算，在检索效率上有改进的空间。Sun 等[19]在此基础上提出了基于多维 B 树的索引结构，使用贪婪的深度优先搜索来加快检索效率，但是该方法的检索效率与查询的关键词分布情况有关，当关键词分布在多维 B 树的底部时查询效率严重退化，在一定程度上影响了数据使用者的体验。Fu 等[20]提出了基于二叉树的索引结构，该方法自下而上建树，通过中间节点存储辅助向量来标识关键词是否出现，出现为 1，未出现则为 0，在检索过程中排除不包含查询向量的子树，这种方法过滤掉了不包含查询关键词的文档，只对包含查询关键词的文档计算相关性得分，从而加快查询。Xia 等[21]同样提出了基于二叉树的索引结构，该方法也是自下而上构建索引树，但中间节点存储的是其孩子节点关键词权值的最大值，在检索过程中利用贪婪的深度优先搜索过滤掉文档的最大可能得分小于当前阈值的子树，以此加快检索。这 2 种基于二叉树的检索方法没有考虑到文件之间的相似性，叶子节点是随机放置的，因此在检索效率上还有提升的空间。文献[22-24]采用聚类的方法，利用基于 k-means 的方法自顶向下构建层次聚类索引树，该方法效率较高，但会存在一些检索误差。文献[25-26]采用了两段式的索引结构来提升查询效率，在查询的第一阶段筛选掉无关的文档，在第二阶段计算剩余文档的相关性得分，效率较高，但在第一阶段其检索模式是“半保护的”，会暴露查询陷门的隐私。文献[27-29]引入了布隆过滤器来过滤不包含查询关键词的文档，效率较高，但会有一定的误差。

综上所述，现有的各种检索方案都是从过滤低相关性文档的角度进行改进来提高检索效率，而本文从过滤无关关键词的角度进行改进，提出一种分块的多关键字排序密文检索方案。本文的创新如下。

1) 由于只需要加密和发送包含查询关键词的分块，与其他加密和发送整个字典长度的查询向量方法相比，减少了数据使用者的计算开销和通信开销。

2) 通过分块过滤掉大部分无关的关键词，大幅减少了边缘服务器计算文档相似性得分时的计算量，提升了查询效率。

3) 通过理论分析和仿真实验，证明了所提方法在保证一定安全性的同时提升了查询效率。

3 问题描述

3.1 系统模型

系统模型如图 1 所示，各部分功能介绍如下。

1) 云服务器

云服务器存储数据拥有者上传的密文文档集合和对应的加密后的索引，将索引分发给所有的边缘服务器，接收边缘服务器传来的查询结果索引号，将对应的加密文档发送给指定的数据使用者。

2) 边缘服务器

边缘服务器存储云服务器传来的加密后的索引，接收数据使用者传来的查询陷门，利用查询陷门和索引计算出最相关的 K 个文档索引编号，发送给云服务器。

3) 数据拥有者

数据拥有者拥有明文文档集合 F，基于明文文

档集合构建可搜索加密索引 I，对明文文档加密得到密文文档集合 C，将加密后的索引 I 和密文文档集合 C 上传的云服务器，负责对数据使用者发放检索密钥和文档解密密钥。

4) 数据使用者

数据使用者从数据拥有者获得密钥，生成查询陷门 T，上传查询陷门给边缘服务器，得到云服务器返回的 K 个文档，利用数据拥有者发来的密钥来解密文档得到想要的明文文档。

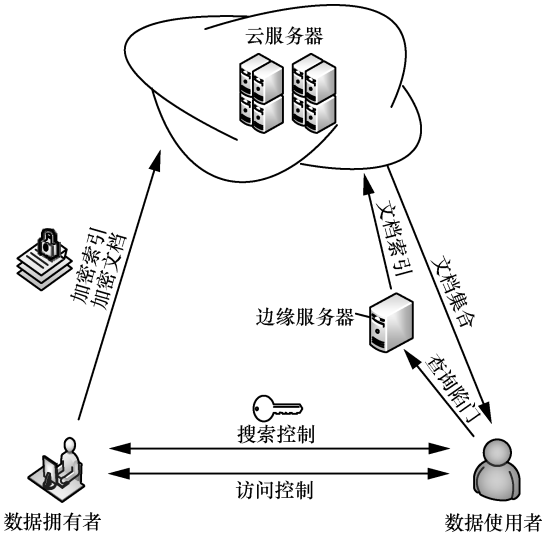

图 1　系统模型

3.2　威胁模型

本文假设云服务器和边缘服务器是“诚实而好奇”的：服务器会按照规定执行指定的操作，同时对存储其上的数据和收到的查询是好奇的，会试图根据得到的信息进行推理、分析文档数据和用户的隐私，但不会对数据进行恶意篡改、破坏。

根据云服务器和边缘服务器能够获得的有效信息，本文考虑以下 2 种威胁模型。

1) 已知密文模型，云服务器能够获得密文文档集合、密文索引。除此之外，云服务器不知道任何其他信息，云服务器只能发动密文攻击。同样，边缘服务器只能获得加密的查询陷门，不知道其他信息。

2) 已知背景知识模型，在已知密文模型的基础上，边缘服务器还会根据用户的查询请求，统计分析用户查询记录中的隐含信息，结合已有的相关背景知识，试图挖掘出一些其他有用信息，如用户的关键词查询偏好、关键词被使用的频率、文档与关键词之间的关联关系等。

3.3　符号说明

表 1 为本文主要符号说明。

表 1　　主要符号说明

符号	说明
F	明文文档集合 $F=\{F_1,F_2,\cdots,F_m\}$，共有 m 个文档
C	密文文档集合 $C=\{C_1,C_2,\cdots,C_m\}$
W	字典集合 $W=\{W_1,W_2,\cdots,W_n\}$，共有 n 个关键词
I	可搜索加密索引 $I=\{I_1,I_2,\cdots,I_n\}$
$\tilde{W}$	查询的关键词集合，$\tilde{W}$ 是 W 的子集
$T_{\tilde{W}}$	$\tilde{W}$ 的查询陷门
K	用户指定返回文件的数量

4　预备知识

4.1　TF×IDF

TF×IDF 是一种统计方法，其中 TF 是词频，表示关键词出现的频率，TF 值越大，说明关键词在文档中越具有代表性。IDF 是逆文档频率，计算方法是文档总数除以包含该词语的文档数量，再将得到的结果取对数，关键词的 IDF 值越大，意味着文档集合中包含该关键词的文档数量越少，该关键词就越具有区分性。

联合使用 TF 与 IDF，可以避免单独使用 TF 导致一些不具有区分性的常用词被赋予过高的权重，又可以避免单独使用 IDF 造成生僻词或者噪声词被赋予过高的权重问题。

本文采用被广泛应用的归一化的 TF×IDF 方法为每个文档中的关键词进行评分，计算式为

$$\mathrm{score}_{i,j}=\frac{1+\ln \mathrm{TF}_{i,j}}{\sqrt{\sum_{W_j\in W}(1+\ln \mathrm{TF}_{i,j})^2}}\ln\left(1+\frac{m}{f_j}\right) \tag{1}$$

其中，$\mathrm{score}_{i,j}$ 和 $\mathrm{TF}_{i,j}$ 分别表示关键词 W_j 在文档 F_i 中的相关性得分和词频，f_j 表示包含关键词 W_j 的文档数量。

4.2　向量空间模型

向量空间模型（VSM, vector space model）由 Salton 等[30]于 20 世纪 70 年代提出。VSM 利用向量表示文本，把对文本的处理转化为向量空间中的内积运算。VSM 中向量的维度大小等于关键词的长度，每一维度表示一个关键词。VSM 中每个项的值表示关键词在文档中的权重。

通过 VSM，2 个文件的相似度可以利用这 3 个

文件的向量夹角来衡量，夹角越小，相似度越高。查询请求可以看作一个文档，根据查询的关键词生成一个与向量空间维数相同的向量，通过比较查询请求与所有文档的相似度得到文档与查询的相似性得分，从而按照需求得到与查询最相关的若干个文档。对于查询来说，除了通过比较向量的夹角偏差程度来计算相关程度外，还可以通过计算向量内积来计算每个文档与关键词的相似性得分。有了上述的向量空间模型，文本数据就转换成了计算机可以处理的结构化数据，文档与查询关键词之间的相关性得分就转换成了 2 个向量的内积结果。

本文使用了文献[17]中的应用于向量空间模型的基于安全 kNN 技术[18]的安全内积计算方案。这个技术可以使边缘服务器通过加密的索引和加密的查询陷门计算出陷门与每个文档的明文相关性分数，有效地保护了服务器上索引和查询的隐私。

5 本文方案

5.1 初始化

数据拥有者遍历明文文档集合 F 中的所有文档，提取所有关键词，去掉停用词后得到关键词字典 W，长度为 n。数据拥有者随机生成一个长度为 $n+2$、由 0 和 1 组成的向量 $\boldsymbol{S}$ 作为分裂指示器，再生成两组可逆矩阵 $\boldsymbol{M}'_{v_1},\cdots,\boldsymbol{M}'_{v_h}$ 和 $\boldsymbol{M}''_{v_1},\cdots,\boldsymbol{M}''_{v_h}$，每组前 v_h-1 个矩阵大小为 $\left\lceil\frac{n+2}{v_h}\right\rceil\times\left\lceil\frac{n+2}{v_h}\right\rceil$，第 v_h 个矩阵大小为 $\left(n+2-(v_h-1)\times\left\lceil\frac{n+2}{v_h}\right\rceil\right)\times\left(n+2-(v_h-1)\times\left\lceil\frac{n+2}{v_h}\right\rceil\right)$，最终生成的密钥为 $\left\{\boldsymbol{S},\boldsymbol{M}'_{v_1},\cdots,\boldsymbol{M}'_{v_h},\boldsymbol{M}''_{v_1},\cdots,\boldsymbol{M}''_{v_h}\right\}$。

5.2 构建索引

数据拥有者为每个文档 F_i 构建一个长度为 n 的向量 $\boldsymbol{D}_i$，向量的第 j 位是关键词 W_j 在这个文档中的 TF×IDF 值，然后将向量 $\boldsymbol{D}_i$ 扩展至 $n+2$ 维得到 $\bar{\boldsymbol{D}}_i$，其中第 $n+1$ 位为随机数 ε，第 $n+2$ 位为 1，即 $\bar{\boldsymbol{D}}_i=\{\boldsymbol{D}_i,\varepsilon,1\}$。接下来，对 $\bar{\boldsymbol{D}}_i$ 根据分裂指示器 $\boldsymbol{S}$ 进行随机分裂得到 2 个长度为 $n+2$ 的向量 $\bar{\boldsymbol{D}}'_i$ 和 $\bar{\boldsymbol{D}}''_i$，如果 $\boldsymbol{S}[j]=0$，则使 $\bar{\boldsymbol{D}}'_i[j]=\bar{\boldsymbol{D}}''_i[j]=\bar{\boldsymbol{D}}_i[j]$，否则使 $\bar{\boldsymbol{D}}'_i[j]=\bar{\boldsymbol{D}}''_i[j]=\bar{\boldsymbol{D}}_i[j]$，然后将 $\bar{\boldsymbol{D}}'$ 和 $\bar{\boldsymbol{D}}''$ 按照与密钥相同的方式分割，得到 $\bar{\boldsymbol{D}}'_{v_1},\cdots,\bar{\boldsymbol{D}}'_{v_h}$ 和 $\bar{\boldsymbol{D}}''_{v_1},\cdots,\bar{\boldsymbol{D}}''_{v_h}$，并计算其与 $\boldsymbol{M}'_{v_1},\cdots,\boldsymbol{M}'_{v_h}$ 和 $\boldsymbol{M}''_{v_1},\cdots,\boldsymbol{M}''_{v_h}$ 的乘积，最终得到加密后的索引 $I=\{\bar{\boldsymbol{D}}'_{v_1}\boldsymbol{M}'_{v_1},\cdots,\bar{\boldsymbol{D}}'_{v_h}\boldsymbol{M}'_{v_h},\ \bar{\boldsymbol{D}}''_{v_1}\boldsymbol{M}''_{v_1},\cdots,\bar{\boldsymbol{D}}''_{v_h}\boldsymbol{M}''_{v_h}\}$。

图 2 展示了构建索引的过程，首先数据拥有者为每个文档生成文档-单词矩阵，然后进行分块操作，得到分块后的文档-单词矩阵，将分块后的矩阵作为索引上传给云服务器，云服务器将收到的索引分发给边缘服务器。

文档-单词矩阵

文档索引1	1	2	0	3	1	2
文档索引2	0	0	2	2	8	4
文档索引3	1	1	3	2	2	5
文档索引4	0	1	3	5	7	5
文档索引5	2	2	5	4	2	9

分块

分块后的文档-单词矩阵

文档索引编号	块1	块2	块3
文档索引1	1 2	0 3	1 2
文档索引2	0 0	2 2	8 4
文档索引3	1 1	3 2	2 5
文档索引4	0 1	3 5	7 5
文档索引5	2 2	5 4	2 9

图 2 构建索引

5.3 生成查询陷门

数据使用者首先根据查询关键词集合 $\tilde{W}$ 生成一个长度为 n 的向量 $\boldsymbol{Q}$，将向量的第 $j(W_j\in\tilde{W})$ 位设为 1，表示该次查询中包含关键词 W_j，其他位为 0，将向量 $\boldsymbol{Q}$ 扩展至 $n+2$ 位得到 $\bar{\boldsymbol{Q}}$，其中第 $n+1$ 位为 1，第 $n+2$ 位为随机数 t。然后将 $\bar{\boldsymbol{Q}}$ 向量的前 $n+1$ 位乘以一个随机数 $r(r>0)$，即 $\bar{\boldsymbol{Q}}=\{r\boldsymbol{Q},r,t\}$。最后根据分裂指示器 $\boldsymbol{S}$ 对向量 $\bar{\boldsymbol{Q}}$ 进行随机分裂得到 2 个长度为 $n+2$ 的向量 $\bar{\boldsymbol{Q}}'$ 和 $\bar{\boldsymbol{Q}}'$，如果 $\boldsymbol{S}[j]=0$ 并且 $\bar{\boldsymbol{Q}}[j]\neq 0$，则使 $\bar{\boldsymbol{Q}}'[j]+\bar{\boldsymbol{Q}}''[j]=\bar{\boldsymbol{Q}}[j]$，否则使 $\bar{\boldsymbol{Q}}'[j]=\bar{\boldsymbol{Q}}''[j]=\bar{\boldsymbol{Q}}[j]$。接下来，对向量 $\bar{\boldsymbol{Q}}'$ 和 $\bar{\boldsymbol{Q}}'$ 按照与密钥相同的方式分别分割得到 $\bar{\boldsymbol{Q}}'_{v_1},\cdots,\bar{\boldsymbol{Q}}'_{v_h}$ 与 $\bar{\boldsymbol{Q}}''_{v_1},\cdots,\bar{\boldsymbol{Q}}''_{v_h}$，同时生成一个列表 L，将子向量中不全为 0 的块的序号加入列表 L 中，再将 $\boldsymbol{M}'^{-1}_{v_u}$，$\boldsymbol{M}''^{-1}_{v_u}$ $(v_u\in L)$ 分别乘以 $\bar{\boldsymbol{Q}}'^{\mathrm{T}}_{v_u}$，$\bar{\boldsymbol{Q}}''^{\mathrm{T}}_{v_u}$，最终生成陷门 $T_{\tilde{W}}=\left\{\boldsymbol{M}'^{-1}_{v_u}\bar{\boldsymbol{Q}}'^{\mathrm{T}}_{v_u},\boldsymbol{M}''^{-1}_{v_u}\bar{\boldsymbol{Q}}''^{\mathrm{T}}_{v_u},L\right\}$。

图 3 是生成查询陷门的过程。数据拥有者首先根据查询关键词生成查询向量，然后对查询向量进行分块，同时生成列表 L，将与查询相关的块号加入列表 L 中。接下来，将分块后的向量加密并连同列表 L 一起上传到边缘服务器。

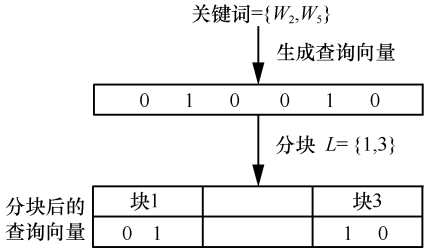

图 3　生成查询陷门的过程

5.4　查询

数据使用者将陷门上传给边缘服务器，边缘服务器首先根据陷门中的 L，找出对应位置的索引块，然后按照式(2)计算所有文档的得分

$$\mathbf{score}_{T_{\tilde{W}}} = \sum_{v_u \in L} \bar{\boldsymbol{D}}'_{v_u} \boldsymbol{M}'_{v_u} \boldsymbol{M}'^{-1}_{v_u} \bar{\boldsymbol{Q}}'^{\mathrm{T}}_{v_u} + \sum_{v_u \in L} \bar{\boldsymbol{D}}''_{v_u} \boldsymbol{M}''_{v_u} \boldsymbol{M}''^{-1}_{v_u} \bar{\boldsymbol{Q}}''^{\mathrm{T}}_{v_u} = \sum_{v_u \in L} \bar{\boldsymbol{D}}'_{v_u} \bar{\boldsymbol{Q}}'^{\mathrm{T}}_{v_u} + \sum_{v_u \in L} \bar{\boldsymbol{D}}''_{v_u} \bar{\boldsymbol{Q}}''^{\mathrm{T}}_{v_u} = \sum_{v_u \in L} \bar{\boldsymbol{D}}_{v_u} \bar{\boldsymbol{Q}}^{\mathrm{T}}_{v_u} \tag{2}$$

其中，$\mathbf{score}_{T_{\tilde{W}}}$ 是一个 m 维的向量，表示在查询陷门 $T_{\tilde{W}}$ 下的所有文档的得分。文档 F_i 的得分为 $\mathbf{score}[i]$，然后边缘服务器选出得分最高的 K 个文档索引号，发送给云服务器，云服务器根据索引号找到对应的加密文档返回给用户。

图 4 是边缘服务器上的文档相关性分数计算过程。数据使用者传来的查询请求包括了一组查询向量块和一个列表 L，其中 $L=[1,3]$ 表示本次计算文档相关性分数需要对块 1 和块 3 进行计算，边缘服务器收到查询请求后，对索引的块 1 和块 3 及对应的查询向量分别计算内积，得到 2 个内积结果，再对内积结果求和得到文档的相关性分数。

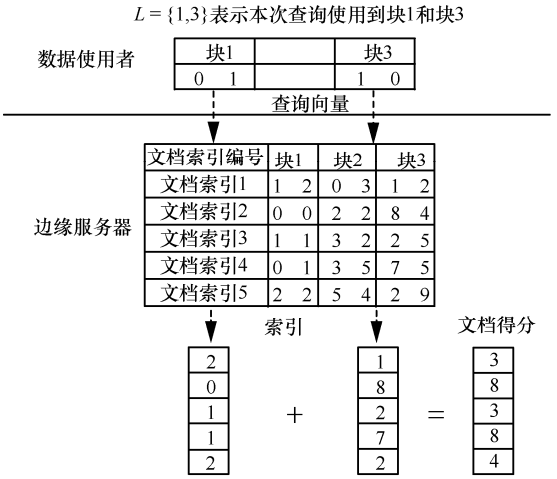

图 4　计算文档相关性分数

6　安全性分析

6.1　索引安全

索引安全指的是服务器无法根据加密后的索引推断出文档与关键词之间的关联关系。如果服务器能够获得文档与关键词之间的关系，那么服务器就能够得知文档中的一些信息，如果是短文本，甚至能够推断出整个文本内容[31]。

本文方法中存储在边缘服务器上的索引是 $2h$ 个加密的矩阵，其密钥是一个长度为 $n+2$ 的向量 $\boldsymbol{S}$ 和 $2h$ 个矩阵。云服务器在不知道密钥的情况下只能进行穷举攻击，文献[18]中已证明，在对称加密中对称密钥长度超过 80 bit 可以认为是安全的。在本文方法中，如果使每个矩阵块的维数为 80，则对于每个矩阵块来说都有一个长度为 80 的向量和 2 个 80×80 的矩阵作为密钥，由于 2 个转换矩阵的存在使维数为 80 的矩阵块的破解难度超过了普通的 80 bit 对称密钥的对称加密。在这种情况下可以认为本文方案能够保证索引的安全性。

此外，由于本文的分割方法使最后一个分块的维数可能很小，为了保证索引的安全性，可以人为地将最后一个分块通过填充 0 扩充维度来满足安全要求，这样就可以在不影响查询结果的同时保证整个索引的安全。

6.2　陷门的不可链接性

如果相同的查询生成的陷门也是相同的，那么云服务器就能够统计不同查询请求出现的频率，从而结合背景知识推断出一些关键字。

本文方法的陷门是 $T_{\tilde{W}}=\{\boldsymbol{M}'^{-1}_{v_u}\bar{\boldsymbol{Q}}'^{\mathrm{T}}_{v_u}, \boldsymbol{M}''^{-1}_{v_u}\bar{\boldsymbol{Q}}''^{\mathrm{T}}_{v_u}, L\}$，其中包括 2 个部分，第一部分 $\boldsymbol{M}'^{-1}_{v_u}\bar{\boldsymbol{Q}}'^{\mathrm{T}}_{v_u}, \boldsymbol{M}''^{-1}_{v_u}\bar{\boldsymbol{Q}}''^{\mathrm{T}}_{v_u}$，其中 $\boldsymbol{M}'^{-1}_{v_u}$ 与 $\boldsymbol{M}''^{-1}_{v_u}$ 在每次查询时都是相同的，而 $\bar{\boldsymbol{Q}}'^{\mathrm{T}}_{v_u}$ 和 $\bar{\boldsymbol{Q}}''^{\mathrm{T}}_{v_u}$ 由于随机分裂的存在，即使是相同的查询，$\boldsymbol{M}'^{-1}_{v_u}\bar{\boldsymbol{Q}}'^{\mathrm{T}}_{v_u}, \boldsymbol{M}''^{-1}_{v_u}\bar{\boldsymbol{Q}}''^{\mathrm{T}}_{v_u}$ 也是不同的。密钥的第二部分 L 是一个包含查询关键词出现的块位置的列表，在相同的查询中是相同的，而本文方法为了索引的安全性使每个块的维度大小至少为 80，意味着每个块内包含的关键词至少是 80 个，根据相同的 L，服务器不

能判断出是否为相同的查询。所以服务器无法根据陷门判断出两次查询之间的关系。除此之外，服务器也无法根据文档相关性得分判断出两次查询之间的关系，这是由于在生成查询陷门时引入了随机生成的 r 和 t，使相同的查询得到的文档分数是不一样的，进一步确保了云服务器不能识别出相同的查询陷门。

综上所述，云服务器不能推断出两次查询的关系，从而保证了陷门的不可链接性。

7 效率分析及仿真实验

本节分别对构建查询陷门效率和查询效率进行了理论分析和实验验证。实验使用从网络上爬取的中文语料作为数据集，使用 Python 3 语言进行仿真实验。仿真实验的硬件环境为 AMD Ryzen 5 3550H CPU，16 GB 内存，Microsoft Windows 10 操作系统。

将本文方案与文献[17,19,21]方案进行比较，分别是 MRSE 方案、基于多维 B 树的检索方案和基于平衡二叉树的检索方案。实验中将查询的关键词固定为 10 个；文献[19]方案中每层的索引向量固定为 100 维，每层按照欧氏距离 Ed ≤ 0.02 进行聚类；文献[21]方法采用单线程实现。

7.1 构建查询陷门

由式(3)可知，构建陷门是查询的重要一环，构建陷门需要在用户的本地进行，会给用户带来一定的计算负担，通常来说性能较差的设备会对陷门构建的时延更敏感，为让尽可能多的用户有良好的体验，高效的构建陷门是有必要的。

查询总时间=构建陷门时间+
计算文档相似性得分时间+
通信时间 (3)

假设包含查询关键词的块总数为 $|L|$，块长为 l。构建查询陷门，首先要根据字典构建查询向量并进行随机分裂和块的分割，时间复杂度为 $O(n)$，然后通过矩阵相乘来加密 $|L|$ 个块，时间复杂度为 $O(|L|l^2)$，所以本文方案构建查询陷门的时间复杂度为 $O(n+|L|l^2)$。

图 5 是本文方案与现有方案关于构建陷门时间的比较。从图 5 可以看出，本文方案在构建查询陷门的效率上优于其他方案，虽然文献[19]中的分层操作与本文方案中的分块方法类似，但其对所有层都进行了加密，而本文方案只加密与查询有关的块，因此本文方案构建陷门的效率更高。

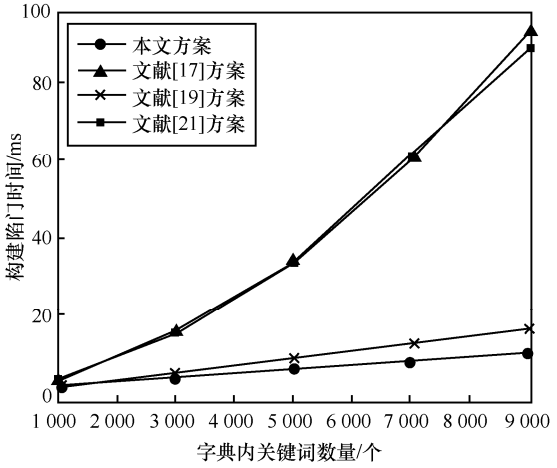

图 5 构建陷门时间随字典内关键词数量的变化

此外，由于本文方案中数据使用者只需要上传相关的分块，比其他方法拥有更小的通信开销。

7.2 计算文档相似性得分

本文方案的查询效率主要由计算文档相似性得分决定。每个索引分块的大小是 ml，对应的查询向量是长度为 l 的向量，共有 $|L|$ 对这样的分块，所以查询的时间复杂度为 $O(|L|ml)$。

首先对本文方案的查询时间与索引块数 v_h 的关系在固定的文档数量、字典长度和返回文档数量下进行实验。从图 6 可以看出，本文方案的查询时间随着索引块数的增加而减少。分块数量越多，每个块的长度就越小，云服务器计算文档相似性得分时需要计算的次数越少，因此提升了查询的效率。由此可知，分块数越多查询效率越高，而随着分块数量的增加，系统的安全性也会减弱，所以在实际应用中，可以根据对查询效率与安全性的不同需求来确定适宜的分块数量。

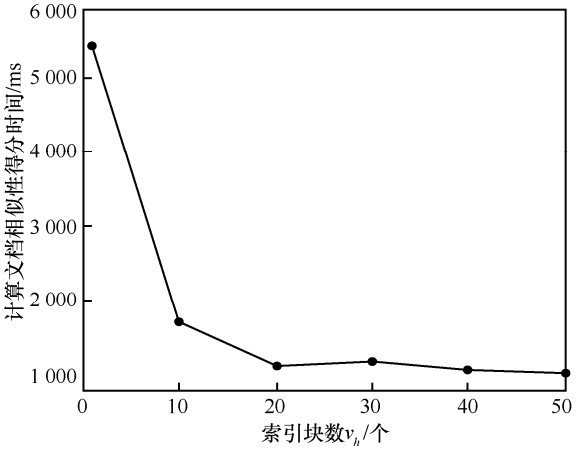

图 6 本文方案计算文档相似性得分时间随索引块数 v_h 的变化（m=3 000，n=5 000，K=100）

从图 7 可以看出，随着文档集合的增大，各种方法的查询耗时都随之增加。文献[19]方案在文档数量较大时表现较好，但该方案在构建索引的过程中用到了聚类，对于给定的聚类阈值来说，文档越多，聚类导致的效果提升越显著，这样做提升了效率但牺牲了一定准确性。本文方案的计算文档相似性得分的时间随着文档数的增加而线性增加，与分析结果一致。

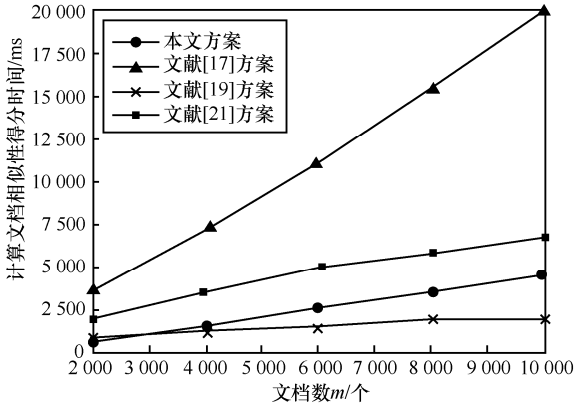

图 7　计算时间随文档数 m 的变化（n=5 000，v_h=50，K=100）

图 8 是各种方案的查询时间随着字典内关键词数量的变化情况。实验中将本文方案块的维数始终保持为 100，所以块的数量随着字典长度的增加而增加。从图 8 可以看出，随着字典的增大，本文方案在保持块大小不变、查询关键词数量不变的情况下查询效率与字典长度无关，而其他现有方案都是随着字典的大小线性变化。这表明了本文方案的查询效率与包含查询关键词的块数和每个块的块长有关，与字典长度无关。由此可以得出结论，本文方案比较适于字典维数很大的场景。

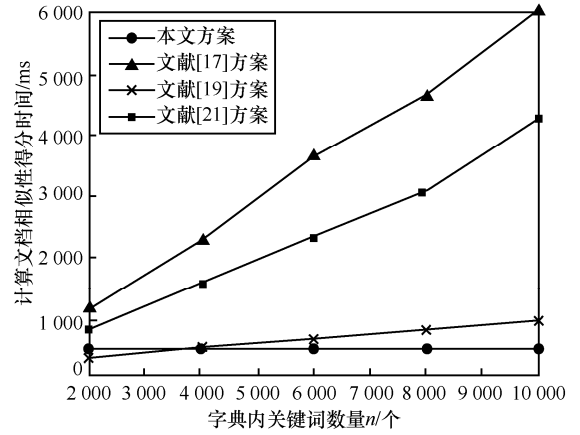

图 8　计算时间随字典内关键词数量 n 的变化

（m=2 000，$v_h=\frac{n}{100}$，K=100）

如图 9 所示，文献[19]方案和文献[21]方案受返回文档数量的影响比较大，因为这 2 个方案是树形索引，采用基于贪婪的深度优先搜索，K 值越大，算法在搜索中剪枝的阈值就越小，排除的子树越少，导致需要遍历的树范围就越大；而文献[17]方案和本文方案是先计算出所有文档的相似性得分，然后再选出相似性得分最大的 K 个文档索引。对于文献[17]方案和本文方案而言，K 值影响的是选出前 K 个文档这一过程，实验中采用的是利用快速排序的思想划分出前 K 个文档，这一步的时间复杂度是 $O(m)$，然后对这 K 个文档进行排序，所以总的时间复杂度是 $O(m+K\log K)$。随着 K 的增大，仅在排序前 K 个文档的过程中增大了一些计算，因此本文方案的查询时间受返回文档数量的影响较小。

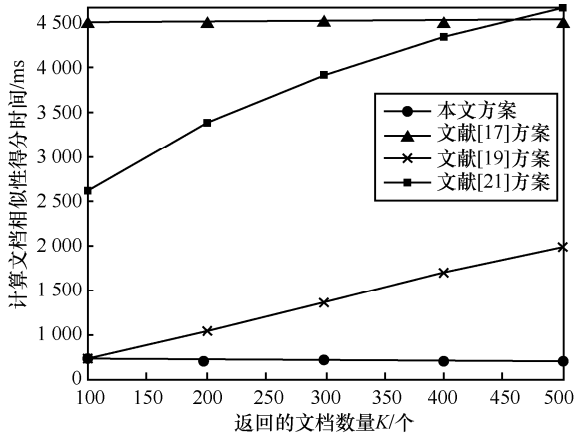

图 9　计算时间随返回的文档数量 K 变化（n=5 000，v_h=50，m=3 000）

8　结束语

针对传统的云计算密文检索方案难以满足当今用户需求的问题，本文提出了一种基于分块的移动边缘计算密文检索方案。引入了边缘服务器来分担云服务器的计算压力。在改进搜索效率上，不同于现有的方案从过滤低相关性文档的角度进行改进，本文从过滤无关关键词的角度进行了改进，为提高密文检索的效率研究提供了新思路。理论分析表明，本文方案具有索引安全性和陷门不可链接性。通过实验对本文方案与现有经典方案的查询效率进行了多个方面的比较，结果表明本文方案相对于现有经典方案有效地提高了查询效率，并且字典维数越大提高效果越显著。

参考文献：

[1]　WANG N, FU J, BHARGAVA B, et al. Efficient retrieval over docu-

ments encrypted by attributes in cloud computing[J]. IEEE Transactions on Information Forensics and Security, 2018, 13(10): 2653-2667.
[2] HASHEM I A T, YAQOOB I, ANUAR N B, et al. The rise of “big data” on cloud computing: review and open research issues[J]. Information Systems, 2015, 47: 98-115.
[3] 张佳乐, 赵彦超, 陈兵, 等. 边缘计算数据安全与隐私保护研究综述[J]. 通信学报, 2018, 39(3):1-21.
ZHANG J L, ZHAO Y C, CHEN B, et al. Survey on data security and privacy-preserving for the research of edge computing[J]. Journal on Communications, 2018, 39(3):1-21.
[4] 贾维嘉, 周小杰. 雾计算的概念、相关研究与应用[J]. 通信学报, 2018, 39(5):153-165.
JIA W J, ZHOU X J. Concepts, issues, and applications of fog computing[J]. Journal on Communications, 2018, 39(5):153-165.
[5] SATYANARAYANAN M. The emergence of edge computing[J]. Computer, 2017, 50(1): 30-39.
[6] ROMAN R, LOPEZ J, MAMBO M. Mobile edge computing, fog et al.: a survey and analysis of security threats and challenges[J]. Future Generation Computer Systems, 2018, 78: 680-698.
[7] FAN K, YIN J, ZHANG K, et al. EARS-DM: efficient auto correction retrieval scheme for data management in edge computing[J]. Sensors, 2018, 18(11): 3616.
[8] MIAO Y, MA J, LIU X, et al. Lightweight fine-grained search over encrypted data in fog computing[J]. IEEE Transactions on Services Computing, 2018, 12(5): 772-785.
[9] WANG N, FU J, LI J, et al. Source-location privacy protection based on anonymity cloud in wireless sensor networks[J]. IEEE Transactions on Information Forensics and Security, 2019, 15(1): 100-114.
[10] WANG C, WANG Q, REN K, et al. Privacy-preserving public auditing for data storage security in cloud computing[C]//Proceedings of International Conference on Computer Communications. Piscataway: IEEE Press, 2010: 1-9.
[11] ESPOSITO C, CASTIGLIONE A, POP F, et al. Challenges of connecting edge and cloud computing: a security and forensic perspective[J]. IEEE Cloud Computing, 2017, 4(2): 13-17.
[12] KHALIL I M, KHREISHAH A, AZEEM M. Cloud computing security: a survey[J]. Computers, 2014, 3(1): 1-35.
[13] SINGH S, JEONG Y S, PARK J H. A survey on cloud computing security: issues, threats, and solutions[J]. Journal of Network and Computer Applications, 2016, 75: 200-222.
[14] CHANG Y C, MITZENMACHER M. Privacy preserving keyword searches on remote encrypted data[C]//International Conference on Applied Cryptography and Network Security. Berlin: Springer, 2005: 442-455.
[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] WANG C, CAO N, REN K, et al. Enabling secure and efficient ranked keyword search over outsourced cloud data[J]. IEEE Transactions on Parallel and Distributed Systems, 2011, 23(8): 1467-1479.
[17] 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.
[18] WONG W K, CHEUNG D W, KAO B, et al. Secure kNN computation on encrypted databases[C]//Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2009: 139-152.
[19] SUN W, WANG B, CAO N, et al. Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking[C]// Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security. New York: ACM Press, 2013: 71-82.
[20] FU Z, SUN X, LIU Q, et al. Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing[J]. IEICE Transactions on Communications, 2015, 98(1): 190-200.
[21] 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, 2015, 27(2): 340-352.
[22] CHEN C, ZHU X, SHEN P, et al. An efficient privacy-preserving ranked keyword search method[J]. IEEE Transactions on Parallel and Distributed Systems, 2016, 27(4): 951-963.
[23] WODI B H, LEUNG C K, CUZZOCREA A, et al. Fast privacy-preserving keyword search on encrypted outsourced data[C]//IEEE International Conference on Big Data. Piscataway: IEEE Press, 2019: 1-10.
[24] 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.
XU G W, SHI C H, WANG W T, et al. Multi-keyword searchable encryption algorithm based on semantic extension[J]. Journal of Computer Research and Development, 2019, 56(10): 2193-2206.
[25] CHEN J, HE K, DENG L, et al. EliMFS: achieving efficient, leakage-resilient, and multi-keyword fuzzy search on encrypted cloud data[J]. IEEE Transactions on Services Computing, 2017, PP(99): 1.
[26] ZHONG H, LI Z, XU Y, et al. Two-stage index-based central keyword-ranked searches over encrypted cloud data[J]. Science China Information Sciences, 2020, 63(3): 1-3.
[27] FU S, ZHANG Q, JIA N, et al. A privacy-preserving fuzzy search scheme supporting logic query over encrypted cloud data[J]. Mobile Networks and Applications, 2020(4): 1-12.
[28] GUAN Z, LIU X, WU L, et al. Cross-lingual multi-keyword rank search with semantic extension over encrypted data[J]. Information Sciences, 2020, 514: 523-540.
[29] MU Y, YAO H. Encrypted data retrieval scheme based on bloom filter[C]//Proceedings of the 18th International Symposium on Distributed Computing and Applications for Business Engineering and Science. Piscataway: IEEE Press, 2019: 249-252.
[30] SALTON G, WONG A, YANG C S. A vector space model for automatic indexing[J]. Communications of the ACM, 1975, 18(11): 613-620.
[31] ZERR S, DEMIDOVA E, OLMEDILLA D, et al. Zerber: r-confidential indexing for distributed documents[C]//Proceedings of the 11th International Conference on Extending Database Technology: Advances in database technology. New York: ACM Press, 2008: 287-298.

[作者简介]

王娜（1988− ），女，湖南衡阳人，北京邮电大学在站博士后，主要研究方向为密码算法及安全协议、物联网安全、云计算安全、大数据隐私保护、移动互联网安全等。

郑坤（1996− ），男，辽宁葫芦岛人，北京邮电大学硕士生，主要研究方向为移动边缘计算安全、雾计算与云计算安全、大数据隐私保护、可信计算等。

付俊松（1989− ），男，河北唐山人，博士，北京邮电大学助理教授，主要研究方向为云计算、分布式网络安全、信息检索、隐私保护、软件安全等。

李剑（1976− ），男，陕西西安人，博士，北京邮电大学教授，主要研究方向为智能网络安全、量子密码学、移动通信安全、物联网安全等。