基于多边缘服务器的个性化搜索隐私保护方法

张强[1]，王国军[2]，张少波[3]

（1. 中南大学信息科学与工程学院，湖南 长沙 410083；2. 广州大学计算机科学与网络工程学院，广东 广州 510006；
3. 湖南科技大学计算机科学与工程学院，湖南 湘潭 411201）

摘 要：在明文环境下根据用户的兴趣模型以及查询关键词能够获得用户个性化的搜索结果会导致敏感数据和用户隐私信息的泄露，不利于含有敏感数据的云搜索服务的推广，鉴于此，数据通常以密文的形式存储在云服务器中。用户在使用云搜索服务时，希望在海量的密文中快速地获得自己想要的搜索结果。为了解决这一问题，在个性化搜索中提出了一种基于多边缘服务器的隐私保护方法，该方法通过引入多个边缘服务器，并通过切割索引与查询矩阵，实现了在边缘服务器上计算部分用户查询与部分文件索引之间的相关性得分，云服务器只需要将边缘服务器上得到的相关性得分做简单处理即能返回与用户查询最相关的前 K 个文件，使其特别适用于大量用户在海量密文中的个性化搜索。安全分析和实验结果表明，该方法能很好地保护用户的隐私以及数据的机密性，并具有高效的搜索效率，能为用户提供了更好的个性化搜索体验。

关键词：个性化搜索；隐私保护；边缘服务器；索引切割；可搜索加密

中图分类号：TP391

文献标识码：A

doi: 10.11959/j.issn.1000−436x.2019024

Method of privacy protection based on multiple edge servers in personalized search

ZHANG Qiang[1], WANG Guojun[2], ZHANG Shaobo[3]

1. School of Information Science and Engineering, Central South University, Changsha 410083, China
2. School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, China
3. School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China

Abstract: In the plaintext environment, users' personalized search results can be obtained through users' interest model and query keywords. However, it may possibly result in the disclosure of sensitive data and privacy, which prevents using sensitive data in cloud search. Therefore, data is generally stored in the form of ciphertext in the cloud server. In the process of cloud search service, users intend to quickly obtain the desired search results from the vast amount of ciphertext. In order to solve the problem, it was proposed that a method of privacy protection based on multiple edge servers in personalized search shall be used. By introducing multiple edge servers and cutting the index as well as the query matrix, the computing relevance scores of partial query and partial file index are achieved on the edge server. The cloud server only needs to get the relevance score on the edge server and make a simple processing that can return to the most relevant Top K files by user query, so as to make it particularly suitable for a large number of users in the massive personalized ciphertext search. Security analysis and experimental results show that this method can effectively protect users' privacy and data confidentiality. In addition, it can guarantee high efficiency in search to provide better personalized search experience.

Key words: personalized search, privacy protection, edge server, index cutting, searchable encryption

---

收稿日期：2018−01−18；**修回日期**：2018−08−14

通信作者：王国军，csgjwang@gmail.com

基金项目：国家自然科学基金资助项目（No.61632009, No.61472451）；广东省自然科学基金资助项目（No.2017A030308006）；广东省高等教育高层次人才计划基金资助项目（No.2016ZJ01）；中南大学中央高校基本科研业务费专项基金资助项目（No.2017zzts141）

Foundation Items: The National Natural Science Foundation of China (No.61632009, No.61472451), The Guangdong Provincial Natural Science Foundation (No.2017A030308006), The High-Level Talents Program of Higher Education in Guangdong Province (No.2016ZJ01), The Fundamental Research Funds for the Central Universities of Central South University (No.2017zzts141)

1　引言

随着时代的发展，信息量呈指数级增长趋势，为了快速地从庞大的数据中找到所需要的信息，搜索成为了人们共同的选择，搜索技术也从最开始的分类目录时代渐渐进入了以用户为中心的时代。同时，随着数据量的剧增，存储和计算问题也越来越突出，为了解决这一问题，云计算技术应运而生。

如今，云服务越来越便捷。然而，随着云服务的普及，其安全和隐私泄露问题已然成为了人们关注的焦点[1]。因为黑客及云服务器本身的不可信，当数据以明文形式存储在云服务器上时，很可能会导致数据的泄露，鉴于此，数据拥有者倾向于先加密数据，再将密文外包到云服务器中。然而，传统的明文检索技术在密文环境下将毫无用处。与此同时，用户在实时检索时希望以最短的时间获得自己最需要的检索结果，但随着数据量与用户数的激增，云服务器可能会成为云服务的性能瓶颈，增加用户的等待时间，这将严重影响用户的搜索体验。因此，如何在浩瀚的密文中快速地获得自己所需要的检索结果成为密文环境下个性化搜索技术的研究方向。

2　相关工作

为了在密文环境下检索信息，可搜索加密技术应运而生。Song 等[2]采用流密码对关键词进行加密，通过关键词与密文文件之间的一一匹配，即可获悉该密文中是否包含该关键词，开启了密文关键词检索的新篇章。随后，研究者们提出了许多的改进方案[3-5]，为可搜索加密注入了新的活力。Dan 等[6]最早提出了公钥加密的关键词搜索方案，以解决服务器不可信时的路由问题。在这个方案中，用户只需要拥有私钥即可以通过搜索获得经过公钥加密的数据。随后，研究者们[7-8]提出了应用于各种场景的可搜索加密方案，这些方案推动了可搜索加密技术的发展。

然而，以上的方案只考虑了搜索关键词，并没有考虑每个人在提交相同关键词时的真实需求。世界上没有相同的两片树叶，同样也不会存在着兴趣完全相同的人，因此，如何根据用户的兴趣及关键词返回用户满意的搜索结果，关乎着用户搜索体验的好坏。文献[9]结合内容过滤及协同过滤的方法为用户提供个性化的搜索结果，实验结果表明该方法能够提供精确的搜索结果，提升用户的搜索体验。文献[10]通过挖掘用户的点击数据获取用户的兴趣偏好，同时引入了用户的位置信息，并采用熵来平衡用户偏好与位置信息之间的权重，该方法提高了搜索的精确度，提升了用户的搜索体验。

但上述方法却只限于明文搜索，如何很好地实现密文环境下的个性化搜索，提升用户的搜索体验，还是一个任重而道远的事情。文献[11]通过用户的搜索历史，并根据语义网（WordNet）构建用户模型，通过关键词优先级将用户兴趣融入用户的查询关键词，然后对存储在云服务器上的密文进行搜索，并返回相关性得分最高的前 K 个搜索结果给用户，实现在密文环境下个性化搜索的目的，但该方法存在 3 个不足：1) 索引构建时间太长，不仅加大了数据拥有者构建索引的负担，也不利于索引的更新；2) 云服务器需要计算每个查询与所有文件索引的相关性得分，云服务器的计算负担不容小觑，这可能使云服务器成为性能瓶颈；3) 为了保护用户的隐私信息不被云服务器知晓，引入的假关键词不仅增加了云服务器的开销，还降低了查询的精确度，而高的查询精确度是提高用户搜索体验的保证。

基于以上研究，本文通过引入边缘服务器，提出一种基于多边缘服务器的个性化搜索隐私保护方法，实现了密文环境下的个性化搜索。具体的创新点如下。

1) 文件索引存储在边缘服务器中，而文件的密文存储在云服务器中，从源头上保证了文件索引不被云服务器知晓。

2) 通过索引的切割，边缘服务器只能得到部分索引信息，通过在边缘服务器上加密索引，大大减轻了数据拥有者构建索引的负担。

3) 通过引入随机数后，将用户查询矩阵进行切割、矩阵加密，在优化整个系统查询性能的同时保护了用户的查询隐私。

3　系统模型和相关定义

3.1　系统模型

图 1 为基于多边缘服务器的隐私保护模型，该模型由用户、数据拥有者、边缘服务器和云服务器这 4 类实体构成。数据拥有者负责生成密钥 sk 并通过安全信道将查询加密密钥与密文密钥 key 传送给用户，同时还负责构建文件的索引并将索引切割后将 p_i 与相应的索引加密密钥 M_i^{T} 发送给边缘服务

器 i，同时数据拥有者将密文 C 传送至云服务器，以便后续的查询。在用户端，用户输入查询关键词以生成查询矩阵，经过用户兴趣模型后，用户查询矩阵发生变化使其不仅带有用户本次查询关键词的信息，同时带有用户的兴趣信息，然后用户将转换后的查询矩阵进行切割并用相对应的密钥加密生成 $\boldsymbol{T}_i$，最后将 $\boldsymbol{T}_i$ 发送至相应的边缘服务器 i。边缘服务器负责索引的加密，根据安全内积计算用户查询与索引的相关性得分 S_i，随后将计算得到的相关性得分矩阵发送到云服务器中。云服务器负责将边缘服务器发送来的相关性得分进行累加，并根据累加后的相关性得分的高低，将相关性得分最高的前 K 个密文给用户。该方法通过引入多边缘服务器，并将索引以及用户查询矩阵进行切割后加密，大大地减少了计算量，同时使用多边缘服务器计算用户查询与索引之间的相关性得分能够大幅度地减少查询的时间，进而提升用户体验。通过引入多边缘服务器，减少了云服务器以及数据拥有者的计算开销，提升了整个系统的效率，同时保护了用户的查询隐私。

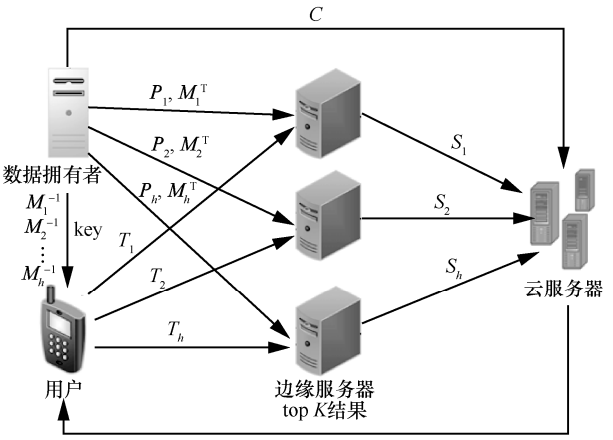


图 1 基于多边缘服务器的隐私保护模型

3.2 相关定义

定义 1 TF-IDF（term-frequency-inverse document frequency）索引构建方法。TF-IDF 方法构建的索引能够很好地反映文件中关键词的重要程度，这在很多文献中都得到了证实[12-14]。本文采用 TF-IDF 构建文件的索引，以期关键词能更好地反映文件。

定义 2 相关性得分。相关性得分的高低反映了加密后的查询矩阵 $\boldsymbol{T}_i$ 与加密后的索引 $\boldsymbol{I}_i$ 的相关性程度，得分越高，说明两者的相关性程度越高。

定义 3 安全内积计算[15]。为了达到保护用户的隐私不被泄露及完成相关性得分计算的目标，采用了先对用户查询及索引进行加密，然后通过安全内积计算的方法获得两者的相关性得分。安全内积计算如式(1)所示。

$$E(\boldsymbol{p}(i,:))E(\boldsymbol{q})=\boldsymbol{p}(i,:)\boldsymbol{q} \tag{1}$$

定义 4 用户兴趣模型。用户兴趣模型反映了用户的偏好为了返回和用户搜索意图最匹配的搜索结果，需要构建用户的兴趣模型，张强等[16]通过对用户的搜索历史捕捉用户的偏好；Du 等[17]根据用户的喜好程度建立了多层次的用户模型，以使其能充分地反映用户的真实需求；本文为了能返回符合用户兴趣的密文，采用了文献[11]的方法构建用户兴趣模型，通过用户的查询历史及 WordNet[18]英语词汇数据库来建立具有语义信息的用户兴趣模型。

定义 5 索引切割。本文采用多个边缘服务器，用于索引的加密及用户查询与其上索引的相关性得分计算。为了实现这一功能，数据拥有者需要根据索引加密密钥的维度对索引进行切割。为了简单起见，假设需要将索引切割成 3 份，索引加密密钥为 3 阶方阵，数据拥有者根据索引加密密钥方阵的维数，将索引切割成 3 份。其中，索引中的每一行代表一个文件的索引，行中的每一个元素代表文件中相应关键词的权重。数据拥有者根据索引加密矩阵的维数，以 3 列为一单位对整个索引进行切割。如图 2 所示，整个索引被切割成了 3 个15×3的矩阵。

	1	2	3	4	5	6	7	8	9
1	0.007 9	0.003 7	0	0	0	0	0	0	0
2	0	0	0.009 5	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0.006 8
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0.001 9	0	0	0.007 9	0	0
6	0	0	0	0	0.007 0	0	0	0.030 5	0
7	0	0	0	0	0	0	0	0	0
8	0.008 3	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0.007 9	0	0	0
10	0	0	0.028 8	0	0	0	0	0	0
11	0	0	0	0	0	0.009 4	0	0	0.003 3
12	0.012 9	0	0	0	0	0	0	0	0
13	0	0.009 7	0	0	0	0	0	0	0
14	0	0	0.004 1	0	0	0	0	0	0
15	0.015 0	0	0	0.004 7	0	0	0	0	0

图 2 索引切割

定义 6 查询矩阵切割。查询矩阵为行矩阵，维度等于索引的列数，其切割方法与索引切割方法相同，也是根据相应密钥的维数进行切割，因为查

询矩阵的加密密钥与索引的加密密钥一一对应，因此经过切割后的查询矩阵也会和切割后的索引矩阵一一对应。

3.3 攻击模型

本文假设边缘服务器不会进行合谋攻击，且边缘服务器不会将索引以及索引加密密钥泄露给第三方。同时假设数据拥有者能够通过可信信道安全地将密钥发送给用户，将索引以及索引加密密钥发送给边缘服务器，将密文发送给云服务器。这个是很容易实现的，如通过 SSL/TLS（secure socket layer/transport layer security）通信方式就能很容易地达到这一目的[19-20]。

1) 诚实而好奇 (HBC, honest-but- curious) 模型[21-23]。在 HBC 模型中，攻击者严格遵照协议的约定执行整个流程，但为了某些利益或满足自己的好奇心，会试图从已知的信息中挖掘用户的隐私信息（例如通过用户的快递信息推测用户的家庭住址，或者通过用户的网购信息推测用户的喜好）。本文假设云服务器和边缘服务器是诚实而好奇的攻击者，其试图获取用户的隐私信息。

2) 恶意攻击模型（malicious model）。恶意攻击者完全不遵守协议的约定，可能发起拒绝服务攻击(DoS, denial of service attack)，也可能发起重放攻击，甚至利用社会工程学进行人身攻击。本文主要关注的是保护用户的隐私，因而这些主动攻击不是本文的重点，并且有相当多的文献对恶意攻击做了相关研究[24-25]。

4 基于多边缘服务器的隐私保护方法

本文提出了一种基于多边缘服务器的个性化搜索隐私保护方法，该方法的基本思想是引入多个边缘服务器。与文献[11]相比，在保护用户隐私的前提下，引入边缘服务器有如下作用。

1) 在边缘服务器上加密索引，能大幅度地减少数据拥有者的负担，同时提升索引的加密效率。

2) 在边缘服务器上计算用户查询矩阵和索引的相关性得分，在减轻云服务器计算开销的同时能够提升搜索的效率，进而缩短用户获得搜索结果的时间，从而提高用户的搜索体验。

如图 3 所示，本文所提方法主要包括 6 个阶段，依次是系统初始化阶段、索引加密阶段、查询生成阶段、得分计算阶段、查询处理阶段和结果解密阶段。相关的符号描述如表 1 所示。

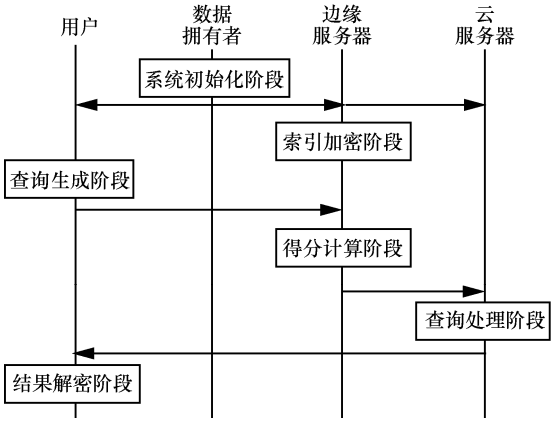


图 3　基于多边缘服务器的隐私保护方法流程

表 1　　符号描述

符号	描述
sk	密钥
K	用户提交的参数 K
a	大于 0 的随机数
p, p_i	明文索引
$\boldsymbol{I}, \boldsymbol{I}_i$	加密后的索引
C	加密后的文件集
$\boldsymbol{R}_i(j,:)$	第 i 个边缘服务器上的第 j 行索引的相关性得分
$\boldsymbol{S}_i$	第 i 个边缘服务器计算的相关性得分矩阵
$\boldsymbol{q}$, $\boldsymbol{q}_i$	用户查询
n , n_i	关键词数
$\boldsymbol{U}$	用户模型
m	文件数量
$\boldsymbol{T}, \boldsymbol{T}_i$	加密后的查询矩阵
$\boldsymbol{S}$	云服务器上计算获得的相关性得分矩阵
h	边缘服务器数量
key	文件密文解密密钥

4.1 系统初始化阶段

在系统初始化阶段，数据拥有者随机切割索引，并设第 i 部分索引的关键词数为 n_i ($i \in [1,h]$)，为了尽量减少系统的开销，数据拥有者选取 $n_i = \frac{n}{h}$，为了简单起见，本文假设 $n \bmod h = 0$，然后生成 h 个 $n_i \times n_i$ 的可逆矩阵 $\boldsymbol{M}_i$ ($i \in [1,h]$)作为密钥 sk，因而密钥 sk 是一个 h 元组 $\{\boldsymbol{M}_i\}$。

数据拥有者根据“ TF×IDF ”模型构建文件集的索引 $\boldsymbol{p}$，并通过索引切割方法将其切割为 h 份，然后数据拥有者利用安全信道将索引 $\boldsymbol{p}_i$ 以及索引

加密密钥 $\boldsymbol{M}_i^{\mathrm{T}}$ 发送到第 i 个边缘服务器。数据拥有者利用安全信道将 $\{\boldsymbol{M}_i^{-1}\}$ 以及文件密文解密密钥 key 发送给用户。

为了降低加密与解密文件的开销，数据拥有者使用对称密码技术（如 3DES、AES）对文件集 C 中的每个文件进行加密，并将加密后的文件集 C 发送给云服务器。

4.2 索引加密阶段

大数据时代的到来使数据量呈指数级增长，为了减轻数据拥有者的负担，利用边缘服务器加密索引。由于多边缘服务器能并行加密索引，并且对切割后的索引加密降低了时间复杂度，这大大地缩短了索引加密的时间。边缘服务器利用式(2)对索引进行加密。

$$\boldsymbol{I}_i = \boldsymbol{p}_i \boldsymbol{M}_i^{\mathrm{T}},\ i \in [1,h] \tag{2}$$

4.3 查询生成阶段

步骤 1 用户查询的转换

系统会根据用户提交的查询关键词生成查询矩阵，查询矩阵根据用户兴趣模型进行变化，当用户兴趣模型中关键词的权重不为 0 时，查询矩阵对应的关键词权重作为用户模型中的权重；当用户兴趣模型中关键词权重为 0 时，查询矩阵中对应的关键词权重不变，为初始值 1。转换后的查询矩阵不仅包含用户的查询信息，还包含用户的兴趣信息，例如用户的查询矩阵 $\boldsymbol{q}=[1,1,0,0,0,0,1,0,0]$，用户兴趣模型 $\boldsymbol{U}=[9,0,8,1,0,0,2,0,7]$，经过转换后的查询矩阵 $\boldsymbol{q}=[9,1,0,0,0,0,0,2,0,0]$。

步骤 2 查询矩阵的切割与加密

为了迷惑边缘服务器和云服务器，用户首先随机地生成值 $a(a>0)$，并让 a 和转换后的查询矩阵 $\boldsymbol{q}$ 相乘，使 $\boldsymbol{q}$ 中元素的值变为 a 倍，记为 $a\boldsymbol{q}$，然后根据加密密钥 $\{\boldsymbol{M}_i^{-1}\}$ 的维度，将 $a\boldsymbol{q}$ 切割为 h 份，记为 $a\boldsymbol{q}_i\ (i\in[1,h])$，接下来根据 $\boldsymbol{T}_i=a\boldsymbol{q}_i\boldsymbol{M}_i^{-1}(i\in[1,h])$ 加密用户的查询，最后用户将 $\boldsymbol{T}_i$ 发送到第 i 个边缘服务器，参数 K 随机发送到其中的一个边缘服务器。

4.4 得分计算阶段

边缘服务器 i 在收到用户的查询请求 $\boldsymbol{T}_i$ 后，其利用安全内积计算并根据式(3)计算获得索引 $\boldsymbol{I}_i$ 与 $\boldsymbol{T}_i$ 的相关性得分 $\boldsymbol{S}_i$，计算得到的 $\boldsymbol{S}_i$ 是一个列矩阵。

$$\boldsymbol{S}_i = \boldsymbol{I}_i\boldsymbol{T}_i = \boldsymbol{p}_i\boldsymbol{M}_i^{\mathrm{T}}a\boldsymbol{q}_i\boldsymbol{M}_i^{-1}=a\boldsymbol{p}_i\boldsymbol{q}_i \tag{3}$$

从式(3)可以看出，边缘服务器计算得到的 $\boldsymbol{S}_i$ 确实为 $\boldsymbol{I}_i$ 与 $\boldsymbol{T}_i$ 内积的 a 倍，这说明能通过该方法将文件按相关性得分进行排序，从而返回用户最相关的搜索结果。

4.5 查询处理阶段

当边缘服务器将计算得到的 $\boldsymbol{S}_i$ 以及参数 K 上传到云服务器后，云服务器将所有边缘服务器计算得到的 $\boldsymbol{S}_i$ 进行累加，从而得到 $\boldsymbol{S}$，即 $\boldsymbol{S}=\sum_1^h \boldsymbol{S}_i$，所得到的 $\boldsymbol{S}$ 是一个列矩阵，$\boldsymbol{S}(j)(1\leqslant j\leqslant m)$ 代表用户查询与第 j 个文件的相关性得分。云服务器根据 K 将相关性得分最高的前 K 个密文文件返回给用户。

4.6 结果解密阶段

当用户收到云服务器返回给自己的 top K 密文后，用户使用解密密钥 key 对文件进行解密，从而完成整个搜索的过程。

5 安全性分析

5.1 抵御诚实而好奇的边缘服务器

挑战 1 在本文中，为了充分利用边缘服务器的计算能力，边缘服务器将获得部分索引以及加密该索引的密钥。如果边缘服务器可以知道用户的查询矩阵或用户查询矩阵与文件索引的相关性得分，那么边缘服务器将赢得这个挑战。

定理 1 本文所提方法可以抵御边缘服务器诚实而好奇的窥视。

证明 数据拥有者通过索引切割方法将索引切割成 h 份子信息 $\{\boldsymbol{p}_1,\boldsymbol{p}_2,\cdots,\boldsymbol{p}_h\}$，边缘服务器 i 从数据拥有者处获得了部分索引 $\boldsymbol{p}_i$ 及加密该索引的密钥 $\boldsymbol{M}_i^{\mathrm{T}}$，根据 $\boldsymbol{M}_i^{\mathrm{T}}$，边缘服务器能够计算得到 $\boldsymbol{M}_i^{-1}$。用户的查询经过转换、切割与加密后，形成 h 份子信息 $\{\boldsymbol{T}_1,\boldsymbol{T}_2,\cdots,\boldsymbol{T}_h\}$，边缘服务器 i 从用户处获得了部分搜索信息 T_i，边缘服务器 i 根据 $\boldsymbol{M}_i^{-1}$ 能够计算出用户上传到该边缘服务器的 $a\boldsymbol{q}_i$，但每次发送查询前，用户都会在用户端随机地生成 a，因而边缘服务器 i 不可能根据 $a\boldsymbol{q}_i$ 推断出 $\boldsymbol{q}_i$，假设边缘服务器不共谋，因而，边缘服务器 i 只能获得 $a\boldsymbol{q}_i$。然而即使边缘服务器共谋，攻击者能够获得 $a\boldsymbol{q}$，但用户在每次发送查询时，都会随机地选择 $a(a>0)$，因此，攻击者不可能获得 $\boldsymbol{q}$，并且攻击者没有关键词词典，其不可能根据 $a\boldsymbol{q}$ 的值推断出用户的查询关键词。

由式(4)~式(6)知，即使用户两次查询的查询向量 $\boldsymbol{q}_i$ 相同，由于 $a_1\neq a_2$，边缘服务器通过计算获得

的相关性得分 $\boldsymbol{S}_{i1} \neq \boldsymbol{S}_{i2}$，因此其不可能知道用户的查询矩阵，更不可能推断出查询矩阵与文件索引的相关性得分。

$$\boldsymbol{S}_{i1} = \boldsymbol{I}_i \boldsymbol{T}_{i1} = \boldsymbol{p}_i \boldsymbol{M}_i^{\mathrm{T}} a_1 \boldsymbol{q}_i \boldsymbol{M}_i^{-1} = a_1 \boldsymbol{p}_i \boldsymbol{q}_i \tag{4}$$

$$\boldsymbol{S}_{i2} = \boldsymbol{I}_i \boldsymbol{T}_{i2} = \boldsymbol{p}_i \boldsymbol{M}_i^{\mathrm{T}} a_2 \boldsymbol{q}_i \boldsymbol{M}_i^{-1} = a_2 \boldsymbol{p}_i \boldsymbol{q}_i \tag{5}$$

$$a_1 \neq a_2 \Rightarrow \boldsymbol{S}_{i1} \neq \boldsymbol{S}_{i2} \tag{6}$$

经过以上的分析可知，边缘服务器不能确定地猜出用户的查询以及查询矩阵与文件索引的相关性得分。

5.2　抵御诚实而好奇的云服务器

挑战 2　云服务器负责密文的存储、查询处理并将最符合用户本次查询请求的前 K 个密文返回给用户。云服务器希望从中获取用户的隐私信息，进而获得经济效益或满足自己的好奇心，同时也希望获得文件的明文。如果云服务器能够从中获得用户的具体查询信息或者将存储在其上的明文解密，那么云服务器将赢得这个游戏。

定理 2　本文算法可以抵御云服务器诚实而好奇的攻击。

证明　由于云服务器只知道密文以及加密、解密算法，文件采用对称密码技术（如 3DES、AES）进行加密，其安全性已被证明，云服务器没有用于文件解密的密钥 key，因此其不可能将密文解密成明文，更不可能获悉返回用户的密文的具体信息。云服务器通过从边缘服务器发送来的 $\boldsymbol{S}_i$ 计算用户查询与文件之间的相关性得分 $\boldsymbol{S}$，而 $\boldsymbol{S}$ 的值与 a 相关，a 为用户随机生成的大于 0 的数，由式(7)~式(9)知，即使用户两次查询的查询向量 $\boldsymbol{q}_i$ 相同，对于同一文件，$\boldsymbol{S}_1 \neq \boldsymbol{S}_2$。因而云服务器不可能根据相关性得分对用户的具体查询信息进行揣测。

$$\boldsymbol{S}_1 = \sum_1^h \boldsymbol{S}_{i1} = \sum_1^h a_1 \boldsymbol{p}_i \boldsymbol{q}_i \tag{7}$$

$$\boldsymbol{S}_2 = \sum_1^h \boldsymbol{S}_{i2} = \sum_1^h a_2 \boldsymbol{p}_i \boldsymbol{q}_i \tag{8}$$

$$a_1 \neq a_2 \Rightarrow \boldsymbol{S}_1 \neq \boldsymbol{S}_2 \tag{9}$$

从以上分析可知，云服务器不能猜测出返回给用户的密文信息以及用户的具体查询信息。

5.3　抵御恶意攻击者的窃听攻击

挑战 3　攻击者通过窃听不安全的通信信道，试图从这些数据中挖掘出用户的某些敏感信息，如果攻击者能够恢复用户的查询矩阵或者获知返回用户的文件信息，那么攻击者将赢得这个游戏。

定理 3　本文算法能抵御恶意攻击者的窃听攻击。

证明　假设恶意攻击者拦截到用户发送给所有边缘服务器的查询请求 T_i，由于其不知道加密密钥 $\{\boldsymbol{M}_i^{-1}\}$ $(i \in [1,h])$，并且在每次查询时随机数 a 的值均不相同，因而恶意攻击者不可能恢复用户的查询矩阵 $\boldsymbol{q}$。假设攻击者通过侦听云服务器与用户之间的通信信道，获得了云服务器返回给用户的密文，但由于没有密文的解密密钥 key，其不可能破解经过对称密码技术（如 3DES、AES）进行加密后的文件，因此本文算法能抵御恶意攻击者的窃听攻击。

从以上分析可知，恶意攻击者既不能获得用户的查询矩阵，也不能猜测出云服务器返回给用户的密文信息。

6　实验及结果分析

实验主要从索引加密、用户查询生成、得分计算及查询处理这 4 个方面分析本方法的性能，并将其与 MRSE[26]以及 PRSE[11]方法进行比较，由于边缘服务器数量为 1 时会造成部分隐私的泄露，因此实验中边缘服务器为 1 的实验数据仅作性能对比。采用 Yelp 数据集中的“business”与“review”数据作为本文实验的数据集，以一条“business”数据以及与该“business”数据相关的所有“review”数据作为一个文件，进而形成文件集，并采用 TF×IDF 模型构建这些文件的索引，以期关键词能更好地反映文件。在文件中，每个关键词的重要程度是不一样的，关键词的权重越大，说明该关键词越重要，因此只需要选取权重较大的关键词即能很好地反映文件，同时避免了因关键词字典过于庞大而增加计算开销与存储开销。在构建完关键词字典后，一个文件便可以按照关键词字典中关键词的顺序，形成用关键词的权重表示的文件索引。实验的硬件环境为 2.6 GHz Intel (R) Core (TM) i7-6700HQ CPU，16.00 GB 内存，操作系统为 Microsoft Windows 10，软件为 Matlab R2016b，并使用 OriginPro 2017 对实验数据进行仿真。

6.1　精确度

如今，数据量呈指数级增长势，在这样的环境下，快速地获得需要的信息变得越来越迫切，因而

快速地获得高精度的搜索结果成为了提高用户搜索体验的法宝。本文所提方法能够在保护用户隐私的同时返回用户满意的搜索结果，而用户对返回结果的满意程度在一定程度上反映了精确度的高低。为了评价所提方法的精确度，随机选择 10 位用户使用本方法进行搜索，结果表明，有 9 人对搜索结果满意，这从一定程度上反映了本方法是可行的。

为了简单明了地说明本方法能够在密文环境下实现个性化搜索的目标，随机选择 Yelp 数据集中的 200 条“business”数据以及与这些“business”数据相关的所有“review”数据，生成了 200 个文件。然后根据 TF-IDF 模型获得各个文件中的关键词权重，并选择各个文件中关键词权重最大的前 10 个关键词生成关键词字典，实验中关键词词典共有 1 732 个关键词。最后，根据关键词字典中关键词的顺序，采用关键词权重表示文件的索引，因此，每个文件可以表示为1×1 732 的矩阵。

为了说明本方法中云服务器返回的搜索结果不仅和用户的查询相关，也与用户的历史查询相关，即与用户的兴趣相关。考虑了 2 种情况下云服务器返回用户的搜索结果列表，具体如下。

1) 只考虑用户提交的查询关键词。本文实验中假设用户选择的关键词数占总关键词数的 30%（由于本文实验中总文件数为 200 个，为了有更多的相关文档，因此假设用户选择的关键词数较多。）。

2) 既考虑用户的查询关键词，也考虑用户的兴趣模型。在实验中，假设用户进行了 1 000 次的历史查询，并设置这 1 000 次查询中的用户兴趣偏好如表 2 所示：即有 20%的概率选择前 500 个关键词，15%的概率选择第 501~1 000 个的关键词，10%的概率选择第 1 001~1 500 个的关键词，1%的概率选择第 1 501~1 732 个关键词。

表 2 用户兴趣偏好设置

关键词/个	选择概率
1~500	20%
501~1 000	15%
1 001~1 500	10%
1 501~1 732	1%

设置参数 K=10，即云服务器返回得分最高的前 10 个搜索结果给用户，实验结果如表 3 所示。

表 3 2 种情况下搜索结果对比

兴趣模型	密文按得分降序排序的文件编号
忽略模型	135 85 41 32 34 57 13 69 90 141
考虑模型	90 85 135 32 189 64 41 45 13 199

从表 3 可以看出，云服务器在 2 种情况下返回给用户的搜索结果列表是不同的，说明了用户兴趣模型真真切切地影响了搜索结果。同时也可以看到，在这 2 种情况下，云服务器返回给用户的文件有 6 个是相同的，分别是 135、85、41、32、13、90。

从这个实验可以看出，即使对于相同的查询关键词，由于每个人的兴趣偏好、搜索历史不相同，云服务器返回的搜索结果也不会相同，因此本方法适用于个性化搜索，尤其适用于密文环境下的个性化搜索。同时，由于返回的搜索结果不仅与用户的搜索关键词相关，也与用户的搜索历史相关，这无疑能够提高本方法的精确度。

召回率衡量搜索到所有相关文档的能力，但本方法中，云服务器是根据用户提交的参数 K 返回相关性得分最高的前 K 个文件给用户。当 K 越大，召回率显然会相应地提高，比如当用户设置 K=10 时，而与用户此次查询相关的文档有 100 个时，召回率会很低。但随着 K 的增大，召回率也会相应地增加，因而在本文中讨论召回率的大小是没有必要的，也是没有意义的。

6.2 个性化搜索

6.1 节的实验表明，在考虑或忽略用户兴趣模型时，2 种情况下返回的搜索结果是不同的。为了更进一步地说明本文所提方法能很好地实现个性化搜索的目标，考虑当用户的搜索关键词相同时，云服务器返回给具有不同兴趣偏好的 3 个用户的搜索结果。

本节采用 6.1 节中的数据集进行相关实验。

用户兴趣偏好设置如表 4 所示，既考虑用户的查询关键词，也考虑用户的兴趣模型。在实验中，假设用户进行了 1 000 次的历史查询，并设置这 1 000 次查询中的用户兴趣偏好如表 4 所示，即用户 1、用户 2、用户 3 分别以 20%、12%、5%的概率选择前 500 个关键词，分别以 15%、20%、12%的概率选择第 501~1 000 的关键词；以 10%、15%、5%的概率选择第 1 001~1 500 的关键词，分别以 1%、6%、10%的概率选择第 1 501~ 1 732 个关键词。

表 4　　用户兴趣偏好设置

关键词/个	用户 1	用户 2	用户 3
1~500	20%	12%	5%
501~1 000	15%	20%	12%
1 001~1 500	10%	15%	5%
1 501~1 732	1%	6%	10%

为了进行对比，用户查询和用户 1 的兴趣模型与 6.1 节中相同，并设置参数 K=10，即云服务器返回得分最高的前 10 个搜索结果给用户，实验结果如表 5 所示。

表 5 不同用户相同查询时返回的搜索结果对比

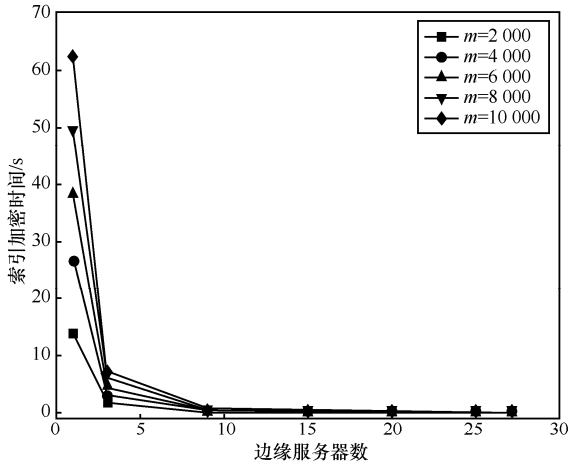
兴趣模型	返回用户的 top 10 结果
忽略模型/用户 1~用户 3	135 85 41 32 34 57 13 69 90 141
考虑模型/用户 1	90 85 135 32 189 64 41 45 13 199
考虑模型/用户 2	135 85 32 41 189 13 69 181 57 34
考虑模型/用户 3	135 57 141 181 69 34 146 85 189 32

从表 5 可以看出，当用户的查询相同时，在忽略用户兴趣模型的情况下，返回给用户 1 至用户 3 的查询结果是完全相同的；当考虑用户的兴趣模型时，即使用户的查询完全相同，返回给用户的查询结果也是不相同的，因为云服务器返回给用户的查询结果不仅与用户的查询相关，也与用户的搜索历史相关，这无疑会使查询结果更符合用户的需求，进而提升用户的搜索体验。同时，本文算法完全在密文环境下进行，很好地保护了用户的隐私。

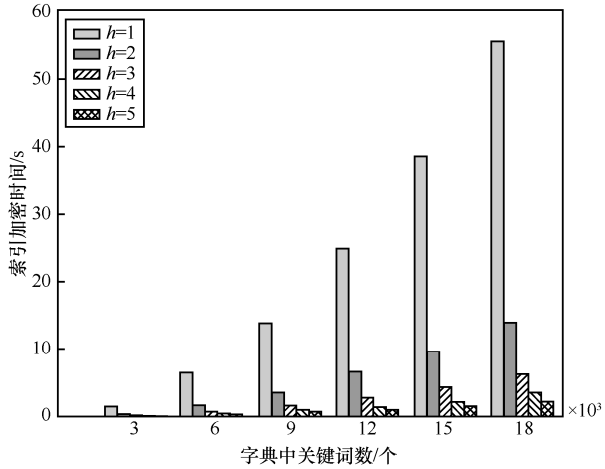
6.3　索引加密

不同于 MRSE[26]和 PRSE[11]在数据拥有者处进行索引的加密，本文利用边缘服务器对索引进行加密。同时，本文对 PRSE 中的索引加密方法进行了改进。边缘服务器的引入既降低了索引加密的时间复杂度，也保护了索引信息不被云服务器知晓。

图 4(a)为字典中的关键词数 n=18 711，文件数 m 为 2 000~10 000 时，索引加密时间随边缘服务器数的变化情况，其表明索引加密的时间随着文件数的增加而增加，而随着边缘服务器的增加而减小。在图 4(a)中，当边缘服务器数量 h=3，文件数从 2 000 增加到 10 000 时，索引加密时间从 1.72 s 增加到了 7.03 s。图 4(b)为当边缘服务器的数量一定、文件数为 10 000 时，索引加密时间随着字典中的关键词数的增加呈二次曲线增长。当字典中的关键词数与文件数一定时，索引加密时间随着边缘服务器的增加而急剧减小。



(a) n=18 711，索引加密时间随边缘服务器数变化情况



(b) m=10 000，索引加密时间随关键词数变化情况

图 4　基于多边缘服务器的隐私保护方法索引加密时间模型

为了更好地验证本方法的性能，将本文算法的索引加密时间与 MRSE 以及 PRSE 方法进行对比，并取字典中的关键词数 n=18 711，如表 6 所示，在对比实验中，采用 3 个边缘服务器对文件的索引进行加密。从表 6 可以看出，当文件数为 2 000 时，本文算法用时仅为 1.72 s，而 MRSE 和 PRSE 方法分别用时 6 211.11 s 和 5 531.48 s，当文件数为 10 000 时，本文算法加密用时为 7.03 s，而 MRSE 和 PRSE 方法分别用时 32 243.81 s 和 30 360.43 s。本文算法索引加密用时不超过 MRSE 方案的 3‰，不超过 PRSE 方案的 4‰。

在整个索引的构建开销中，索引加密占了主要的部分，当索引加密时间降得足够低时，有利于采用 TF-IDF 模型更新索引，因而，本方法也为索引的更新提供了新的思路。

表 6 3 种方案加密时间对比

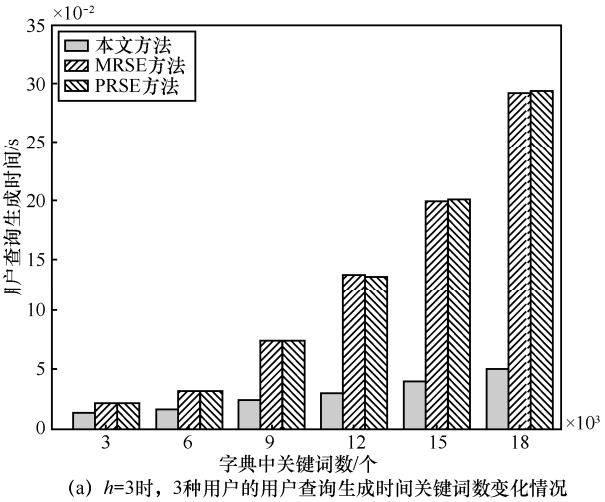
文件数/个	本文算法/s	MRSE/s	PRSE/s
2 000	1.72	6 211.11	5 531.48
4 000	3.09	13 921.08	12 094.42
6 000	4.64	20 111.26	18 364.29
8 000	6.01	27 212.51	25 130.54
10 000	7.03	32 243.81	30 360.43

6.4 用户查询生成

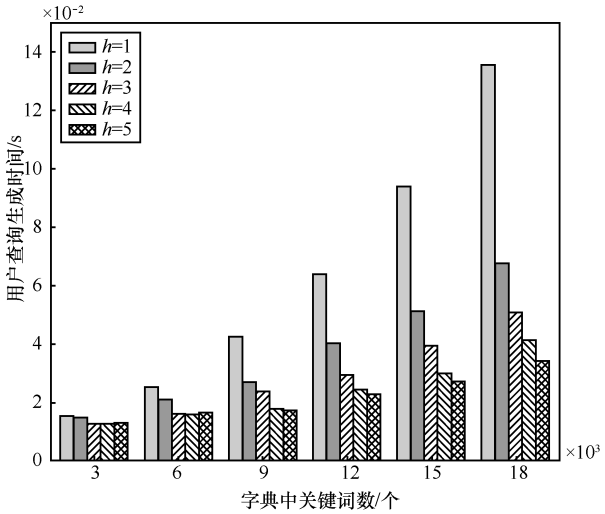
用户查询生成包括用户查询矩阵的转换与用户查询矩阵的切割与加密，为了得到“千人千面”的个性化查询结果，需要根据用户的兴趣模型对用户的查询矩阵进行转换，使转换后的用户查询矩阵不仅带有用户的查询信息，也带有用户的个性化信息，进而使用户得到个性化的搜索结果。以明文形式存在的用户查询矩阵在存储与传输的过程中容易泄露用户的查询隐私，因此需要对转换后的用户查询矩阵进行加密。为了迷惑边缘服务器与第三方攻击者，使用随机生成的迷惑数 a 乘以转换后的用户查询矩阵 $\boldsymbol{q}$ 后，采用类似于索引切割的方法，对转换后的用户查询矩阵 $\boldsymbol{q}$ 进行切割，生成 $\boldsymbol{q}_i\,(i\in[1,h])$，并将其用 $\boldsymbol{M}_i^{-1}(i\in[1,h])$ 加密后，将 $\boldsymbol{T}_i$ 发送到对应的服务器，同时将参数 K 发送到任意一个边缘服务器以告知云服务器所需要返回的查询结果数目。

图 5(a)为本文算法取边缘服务器数量 h=3 时，与 MRSE 以及 PRSE 方法在生成用户查询时的性能对比，其表明随着字典中关键词数的增加，3 种方法的用户查询生成时间均增加，并且本文算法性能明显优于 MRSE 以及 PRSE 方法，当字典中关键词数越多时，优势越明显，如当字典中关键词数为 18 000 时，使用本方法生成用户查询的时间为 0.051 s，而 MRSE 和 PRSE 方法生成用户查询的时间分别为 0.292 s 与 0.293 s。用户查询的生成为用户整个查询的一部分，用户查询的生成时间越短，用户的搜索体验越好。

图 5(b)展示了边缘服务器数量以及字典中关键词数与用户查询生成时间的关系。从图 5(b)可以看出，随着字典中关键词数的增加，用户查询生成时间也相应地增加。而随着边缘服务器的增加，用户查询生成时间减小，如当字典中的关键词数为 18 000，边缘服务器的数量从 1 增加到 5 时，用户查询生成时间从 0.136 s 减小到 0.034 s。



(a) h=3时，3种用户的用户查询生成时间关键词数变化情况



(b) 边缘服务器数变化时，用户查询生成时间随关键词数变化情况

图 5 基于多边缘服务器的隐私保护方法用户查询生成

6.5 得分计算

得分计算为每个边缘服务器计算出用户部分查询与其上的部分索引的相关性得分 $\boldsymbol{S}_i$，用户查询与文件索引之间的相关性得分决定了两者的相关性，进而决定了返回什么文件给用户。为了保护用户的查询隐私，在本文算法中，边缘服务器不会取为 1，但为了更好地进行性能的对比，取边缘服务器数 h=1、3、9、27，字典中的关键词数 n=18 711。从图 6 可以看出，在边缘服务器上进行得分计算的时间随着文件数的增加而增加，随着边缘服务器数量的增加而减小。当文件数为 500 时，边缘服务器数量从 1 增加到 27 的过程中，执行时间从 0.103 s 减小到 0.014 2 s。当文件数为 2 500 时，边缘服务器数量从 1 增加到 27 的过程中，执行时间从 0.503 s 减小到 0.033 s。同样地，得分计算也为用户搜索中的一部分，得分计算的时间越短，用户的搜索体验越好。

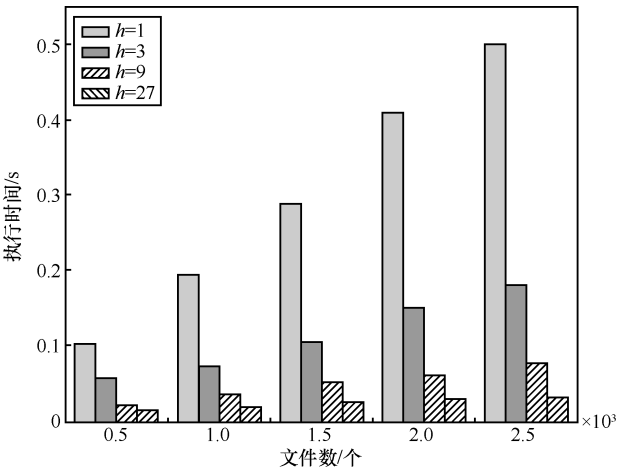


图 6　η=18 711 时，得分计算时间随文件数变化情况

6.6　查询处理

云服务器为了返回与用户此次查询最相关的前 K 个结果，至少需要知道是哪 K 个文件与用户的此次查询最相关，即至少需要知道与用户相关性得分最高的前 K 个文件是哪些。为了实现这一目标，云服务器运行查询处理进程，其将从边缘服务器计算得到的 $\boldsymbol{S}_i$ 进行加和，得到 $\boldsymbol{S}=\sum_1^h \boldsymbol{S}_i$，而 $\boldsymbol{S}$ 中存储了用户查询与文件索引的相关性得分，云服务器对 $\boldsymbol{S}$ 进行处理，即可知道是哪些文件与用户的此次查询最相关，进而返回与用户最相关的前 K 个密文。

在云服务器的查询处理阶段，本文将字典中的关键词数 n 设置为 18 711，返回用户的文件数 K 设置为 10。图 7 表明云服务器上查询处理的执行时间与文件数的多少关系不是很大，这是因为查询处理只需要根据 $\boldsymbol{S}=\sum_1^h \boldsymbol{S}_i$ 计算用户查询与文件之间的相关性得分，并根据相关性得分的大小对 $\boldsymbol{S}$ 中的元素进行排序后将相关性得分最高的前 K 个文件返回给用户，此过程的计算开销很小，因而从实验结果来看，文件数的多少对查询处理的执行时间几乎没有影响。

当文件数不变时，理论上说，随着边缘服务器数的增加，云服务器查询处理的执行时间会相应增加，图 7 也表明了这一变化趋势。但由于边缘服务器的增加只会影响 $\boldsymbol{S}=\sum_1^h \boldsymbol{S}_i$ 过程，并不会影响其后的相关性得分排序与结果返回，因而，边缘服务器数量会对查询处理时间有影响，但影响并不是特别大。从图 7 也可以看出，查询处理的时间很小，如当文件数为 2 500，边缘服务器数为 27 时，云服务器执行查询处理的时间仅为 0.026 s，这为创造良好的用户搜索体验提供了条件。

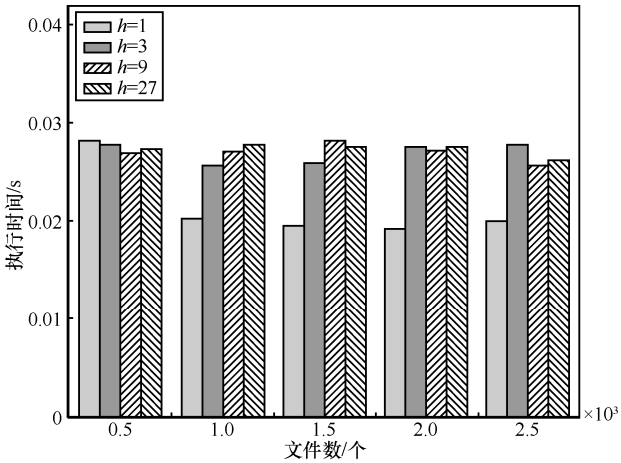


图 7　n=18 711，K=10 时，查询处理执行时间随文件数量变化情况

7　结束语

随着大数据时代的到来，信息过载与隐私保护问题越来越受到人们的关注，基于此，本文提出了一种基于多边缘服务器的个性化搜索方法，该方法同时实现了密文中的个性化搜索与用户隐私保护的统一，并且进行了安全性证明。该方法通过引入多个边缘服务器，并将文件索引存储于边缘服务器中，而将文件密文存储于云服务器中，然后通过切割索引与查询矩阵，成功实现了在边缘服务器上计算部分索引与部分查询之间的相关性得分的目的，有利于保护索引与用户隐私。实验表明，该方法能够实现个性化搜索并大幅地减小用户的搜索等待时间，有利于提高用户的搜索体验，同时该方法减小了云服务的存储开销与计算开销，能更加适用于大量用户的密文搜索环境。

参考文献：

[1] TANG J, CUI Y, LI Q, et al. Ensuring security and privacy preservation for cloud data services[J]. Acm Computing Surveys, 2016, 49(1):1-39.

[2] SONG D X, WAGNER D, PERRIG A. Practical techniques for searches on encrypted data[C]//IEEE Symposium on Security & Privacy. 2002:44-55.

[3] LI H W, LIU D X, DAI Y S, et al. Engineering searchable encryption of mobile cloud networks: when QoE meets QoP[J]. IEEE Wireless Communications, 2015, 22(4):74-80.

[4] CHANG Y C, MITZENMACHER M. Privacy preserving keyword searches on remote encrypted data[C]// International Conference on Applied Cryptography and Network Security. 2005:442-455.

[5] 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.

[6] DAN B, CRESCENZO G D, OSTROVSKY R, et al. Public key encryption with keyword search[C]// International Conference on the Theory and Applications of Cryptographic Techniques. 2004: 506-522.

[7] 李经纬, 贾春福, 刘哲理, 等. 可搜索加密技术研究综述[J]. 软件学报, 2015, 26(1):109-128.
LI J W, JIA C F, LIU Z L, et al. Survey on the searchable encryption[J]. Journal of Software, 2015, 26(1):109-128.

[8] CUI J, ZHOU H, ZHONG H, et al. AKSER: attribute-based keyword search with efficient revocation in cloud computing[J]. Information Sciences, 2017, 423:343-352.

[9] ZHAO F, YAN F W, JIN H, et al. Personalized mobile searching approach based on combining content-based filtering and collaborative filtering[J]. IEEE Systems Journal, 2017, 11(1):324-332.

[10] LEUNG W T, LEE D L, LEE W C. PMSE: a personalized mobile search engine[J]. IEEE Transactions on Knowledge & Data Engineering, 2013, 25(4):820-834.

[11] FU Z J, REN K, SHU J G, et al. Enabling personalized search over encrypted outsourced data with efficiency improvement[J]. IEEE Transactions on Parallel & Distributed Systems, 2016, 27(9): 2546-2559.

[12] FU Z J, WU X L, WANG Q, et al. Enabling central keyword-based semantic extension search over encrypted outsourced data[J]. IEEE Transactions on Information Forensics & Security, 2017, 12(12): 2986-2997.

[13] ALHABASHNEH O, IQBAL R, DOCTOR F, et al. Fuzzy rule based profiling approach for enterprise information seeking and retrieval[J]. Information Sciences, 2017, 394-395:18-37.

[14] ZHANG Q, WANG G J, LIU Q. Enabling cooperative privacy-preserving personalized search in cloud environments[J]. Information Sciences, 2019, 480: 1-13.

[15] WONG W K, CHEUNG W L, KAO B, et al. Secure kNN computation on encrypted databases[C]// ACM SIGMOD International Conference on Management of Data. 2009:139-152.

[16] 张强, 王国军. 个性化搜索中一种基于位置服务的隐私保护方法[J]. 电子与信息学报, 2018, 40(8): 1998-2005.
ZHANG Q, WANG G J. Privacy preserving method based on location service in personalized search[J]. Journal of Electronics & Information Technology, 2018, 40(8):1998-2005.

[17] DU Q, XIE H R, CAI Y, et al. Folksonomy-based personalized search by hybrid user profiles in multiple levels[J]. Neurocomputing, 2016, 204(C):142-152.

[18] MILLER G A. WordNet: a lexical database for english[C]//Communications of the ACM. 1995, 2(11):39-41.

[19] D'ORAZIO C J, CHOO K K R. A technique to circumvent SSL/TLS validations on iOS devices[J]. Future Generation Computer Systems, 2017,74:366-374.

[20] OPPLIGER R, HAUSER R, BASIN D. SSL/TLS session-aware user authentication[J]. Computers & Security, 2008, 27(3-4):64-70.

[21] LUO E T, LIU Q, ABAWAJY J H, et al. Privacy-preserving multi-hop profile-matching protocol for proximity mobile social networks [J]. Future Generation Computer Systems, 2017, 68:222-233.

[22] ZHANG Q, LIU Q, WANG G J. PRMS: A personalized mobile search over encrypted outsourced data[J]. IEEE Access, 2018, 6:31541-31552.

[23] 吴志强, 李肯立, 郑蕙. 高效可扩展的对称密文检索架构[J]. 通信学报, 2017, 38(8):79-93.
WU Z Q, LI K L, ZHENG H. Efficient and scalable architecture for searchable symmetric encryption[J]. Journal on Communications, 2017, 38(8): 79-93.

[24] GOSMAN C, CORNEA T, DOBRE C, et al. Controlling and filtering users data in intelligent transportation system[J]. Future Generation Computer Systems, 2016,78:807-816.

[25] WANG T, LI Y, CHEN Y, et al. Fog-based evaluation approach for trustworthy communication in sensor-cloud system[J]. IEEE Communications Letters, 2017, 21(11):2532-2535.

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

[作者简介]

张强（1988− ），男，湖南湘乡人，中南大学博士生，主要研究方向为个性化搜索、可信计算、云安全、隐私保护、大数据等。

王国军（1970− ），男，湖南长沙人，博士，广州大学博士生导师，主要研究方向为信息安全、可信计算、净室计算、信任推荐等。

张少波（1979− ），男，湖南邵东人，博士，湖南科技大学讲师，主要研究方向为移动社交网络隐私保护、云计算安全、大数据安全和隐私等。