doi:10.3969/j.issn.1000-436x.2014.z2.020

云计算环境下支持排名的关键词加密检索方法

张鹏[1,2]，李焱[3,4]，林海伦[4]，杨嵘[1,2]，刘庆云[1,2]

(1. 中国科学院 信息工程研究所，北京 100093；2. 信息内容安全技术国家工程实验室，北京 100093；
3. 国家计算机网络应急技术处理协调中心，北京 100029；4. 中国科学院 计算技术研究所，北京 100049)

摘 要：随着云计算的出现，越来越多的数据开始集中存储到云端，为了保护数据隐私，敏感数据需要在外包到云端之前进行加密，使在加密数据上进行有效检索成为一个挑战性任务。尽管传统的加密检索模型支持在加密数据上进行关键词检索，但是它们没有描述检索结果的相关度，导致返回所有包含关键词的检索结果占用了大量的网络带宽，并且用户从返回的检索结果中再次选择最相关的结果也会产生大量的时间开销，为此，提出了云计算环境下支持排名的关键词加密检索方法。该方法根据相关度返回排序后的检索结果，其中的保序对称加密模型不仅防止了相关度信息的泄漏，而且提供了高效的检索性能。实验表明了该方法的有效性。

关键词：隐私保护；云计算；保序对称加密；一对多的保序映射

中图分类号：TP319 **文献标识码**：A **文章编号**：1000-436X(2014)Z2-0147-07

Approach to keyword search over encrypted data in cloud

ZHANG Peng[1,2], LI Yan[3,4], LIN Hai-lun[4], YANG Rong[1,2], LIU Qing-yun[1,2]

(1. Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China;
2. National Engineering Laboratory for Information Security Technologies, Beijing 100093, China;
3. National Computer Network Emergency Response and Coordination Center, Beijing 100029, China;
4. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: With the advent of cloud computing, large-scale data are being increasingly outsourced to the cloud. For the protection of data privacy, sensitive data has to be encrypted before outsourcing, which makes effective data utilization a very challenging task. Although traditional searchable encryption approaches allow users to search over encrypted data through keywords, they don't capture any relevance of data files, so users have to spend much time on post-processing every retrieved file in order to find ones most matching their interest. Moreover, retrieving all files containing the queried keyword further incurs unnecessary network traffic, which is not accord with pay-as-you-use cloud paradigm. An approach to ranked keyword search over encrypted data in cloud is proposed. Ranked search greatly mitigate the user's effort by returning the matching files in a ranked order, and also protects the data privacy by order-preserving encryption. Extensive experimental results demonstrate the efficiency.

Key words: privacy-preserving; cloud computing; order-preserving encryption; one-to-many order-preserving mapping

1 引言

云计算作为效用计算的延伸，其愿景是用户能够把他们的数据存储到远程的云端，并且可以从可伸缩的计算和存储资源池中获得高性能和可靠的服务[1]，这种方式所带来的灵活性以及成本的节约正在推动个人和企业把他们本地复杂的数据管理系统外包到云端[2]，特别是当需要存储和利用的数据快速增长的时候。为了保护数据隐私以及防止云中敏感数据，例如邮件、个人健康档案、金融交易等的恶意访问，数据所有者在把这些数据外包到云端之前需要对其进行加密[3]，然而，加密会使基于普通文本的检索方式不再可用，主要原因是下载所有数据并且在本地进行解密会占用巨大的网络带

收稿日期：2014-07-17
基金项目：国家自然科学基金资助项目（61402464）；中国博士后基金资助项目（2013M541076）
Foundation Items: The National Natural Science Foundation of China (61402464); The Ph.D. Programs Foundation of China (2013M541076)

宽的开销，不仅如此，如果无法支持高效和便利的检索，数据存储到云端除了不需要本地存储管理外，将不会带来任何好处。因此，探索在云计算环境下支持排名的关键词加密检索技术至关重要。由于云端可能存储大量的数据，加密检索技术需要重点考虑性能问题。一方面，为了有效地满足用户的检索需求，云端需要支持检索结果的相关度排名，而不是返回没有差别的结果，这种支持排名的检索系统能够使用户快速查找到他们最感兴趣的信息，而不用在大量的返回结果中通过多次比较来进行排序；另一方面，由于支持排名的检索系统只返回最相关的结果，因此减少了不必要的网络传输，降低了网络带宽的开销，非常适合按需付费的云计算范型。然而，传统的加密检索技术[4]重点关注加密安全的形式化定义以及性能改进。为了丰富检索词，在加密数据上的联合关键词检索已经在文献[5]中被提出。近年来，为了适应用户在输入关键词时出现格式和内容上的错误，加密数据上的模糊关键词检索也已经被提出来。但是所有这些技术都没有给出检索结果与关键词的相关度。尽管支持排名的检索技术在信息检索领域已经得到很多关注，但是在云计算环境中，支持排名的关键词加密检索技术却很少有人研究。为此，提出了一种高效的支持排名的关键词加密检索方法，其中在加密数据的文件外包到云端之前的索引构建的过程中，通过利用信息检索和数据挖掘的统计度量方法，计算每个文件的权重信息，也就是相关度。由于把相关度外包到云端很可能会泄露关键词词频等敏感信息，因此该方法在文献[6]的基础上提出了一对多保序映射模型，防止了关键词词频信息的泄漏，定义了云计算环境下支持排名的关键词加密检索问题，并且提出了一个有效的方法，该方法在不泄露敏感信息的情况下实现了支持排名的关键词检索；通过实验证明了该方法的有效性和性能。

2 问题描述

2.1 体系结构

如图 1 所示，云计算环境下加密检索体系结构包含 3 个实体，数据所有者(*O*)，用户(*U*)和云服务器(CS)。数据所有者拥有一个包含 n 个文件集合 $C=(F_1,F_2,\cdots,F_n)$，他们把这些文件加密外包到云服务器中并且希望还能进行关键词检索。为此，在外包这些文件前，数据所有者首先从文件集合 C 中抽取 m 个不同的关键字 $W=(w_1,w_2,\cdots,w_m)$，构建了一个安全的可检索的索引 I，并且在云服务器中存储索引 I 以及加密的文件集合 C。

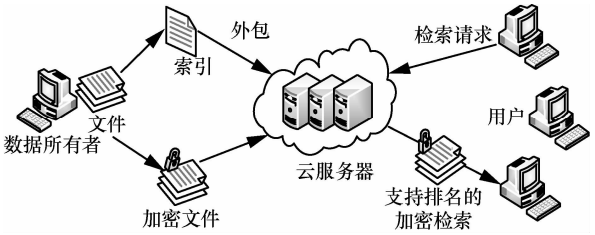


图 1 云计算环境下加密检索体系结构

假设数据所有者和用户之间已经达成授权协议。为了在文件集合中检索一个给定的关键词 w，一个被授权的用户提交一个加密的查询请求：关键词 w 的暗门 T_w 到云服务器上。云服务器一旦接收到查询请求 T_w 后，就会开始检索索引 I 并且把相应的文件返回给用户。支持排名的关键词加密检索问题描述如下：在用户没有事先了解文件集合内容的情况下，云服务器应该根据相关度的排名返回检索结果，例如基于关键词词频的相关度，以提高用户查找匹配文件的效率。然而，由于相关度也会包含一些敏感信息，因此，云服务器不应该获得任何关于相关度的敏感信息。同时，为了减少网络带宽的开销，用户只需要云服务器返回 k 个与关键词 w 最相关的文件。这里的云服务器假设是一个“honest-but-curious”的模型[7]，其中“honest”表示云服务器能够按照指示正确的执行，“curious”表示云服务器会推理和分析执行过程中接收到的数据，从而获取更多的信息，但是云服务器不会主动修改数据或者干扰正常执行。

2.2 设计目标

在上述架构下，为了对云服务器中的外包加密文件进行高效的检索，在设计加密检索方法时应该确保实现以下的安全和性能的目标：1) 防止云服务器获取文件或检索的关键词内容；2) 尽可能减少关键词检索过程中的通信和计算开销。

C——外包到云服务器中包含 n 个文件的集合 $C=(F_1,F_2,\cdots,F_n)$。

W ——从文件集合 C 中抽取不同的关键词，表示成包含 m 个关键词集合 $W=(w_1,w_2,\cdots,w_m)$。

$id(F_j)$ ——文件 F_j 的标识符，它能够唯一定位一个文件。

I——从文件集合中构建的索引，包含一组关键词的倒排索引表 $\{I(w_i)\}$。

T_{w_i} ——关键词 w_i 生成的对应暗门。

$F(w_i)$ ——C 中包含关键词 w_i 文件的标识符集合。

N_i —— 包含关键词 w_i 的文件个数，并且 $N_i = |F(w_i)|$。

如表 1 所示，倒排索引是存储一组关键词到包含该关键词文件映射的索引结构，为了实现支持排名的检索，一般通过赋予一个基于排名函数计算的得分数值来确定相关度。

表 1　　倒排索引的存储结构

关键词	w_i				
文件 ID	F_{i_1}	F_{i_2}	F_{i_3}	F_{i_4}	F_{i_5}
相关度	2.3	3.5	4.1	5.4	10.3

在信息检索中[8]，一个排名函数被用来计算一个给定关键词与文件匹配的程度。目前，大部分都使用 TF×IDF 作为计算相关度的统计指标，不失一般性，选择文献[9]中提出并且普遍被使用的 TF×IDF 公式作为相关度的计算公式。它的定义如下

$$\text{score}(Q,F_d)=\sum_{t\in Q}\frac{1}{|F_d|}(1+\ln f_{d,t})\ln\left(1+\frac{N}{f_t}\right) \qquad (1)$$

其中，Q 表示检索的关键词；$f_{d,t}$ 表示文件 F_d 中词项 t 的 TF；f_t 表示包含词项 t 的文件个数；N 表示文件集合中的文件个数；$|F_d|$ 表示 F_d 的长度，作为归一化因子，它可以通过计算索引项的个数得到。

3　支持排名的关键词加密检索模型

加密检索是指数据所有者以加密方式外包数据后，支持其在加密数据上进行一定检索的能力。通常情况下，加密检索可以使用无关随机存储器实现全部功能[10]。然而，对于每次检索请求，无关随机存储器虽然能够对来自恶意服务器的检索隐藏所有信息，但是它的加密检索方式会在用户和服务器之间产生对数增长的交互次数。因此，为了实现更加高效的加密检索，几乎所有工作都放宽了对隐私保护的要求。对现有加密检索有一个正确的认识对本文提出的支持排名的关键词加密检索模型具有重要意义。类似文献[4]提出的加密检索模型，本文提出的支持排名的关键词加密检索模型由 KeyGen、BuildIndex、TrapdoorGen、SearchIndex 4 个算法组成。它们在配置阶段和查询阶段 2 个阶段创建该模型。

配置阶段：数据所有者通过执行 KeyGen 来初始化用于加密的公钥和私钥的参数，并且通过 BuildIndex 来预处理文件集合 C，从中抽取不重复的关键词以构造用于检索的索引。然后，数据所有者对文件集合 C 进行加密，并且把加密后的 C 和加密后的基于关键词词频相关度的索引一起发布到云服务器上。作为配置阶段的一部分，数据所有者通过线下的公钥加密技术或者更加高效的广播加密技术把私钥的参数，即暗门生成密钥，分发给被授权的用户。

查询阶段：用户使用 TrapdoorGen 生成检索关键词所对应的暗门，并且把它提交给云服务器，云服务器一旦接收到该暗门，它就会通过 SearchIndex 查询索引以发现匹配的文件 ID 以及它们所对应的加密的相关度的列表。这些匹配的文件以相关度排序的方式发送给用户，并且云服务器除了相关度排序外不会知道任何其他信息。

由于本文只关注单个关键词的检索，因此在这种情况下，式(1)中的因子 IDF 总是一个常数，因此，式(1)只要获得了词项的词频信息以及文件的长度信息就能够精确地对检索结果进行排名，如式(2)所示。数据所有者可以通过记录这 2 个数值预先计算相关度，同时，这不会给索引构建带来多少开销。

$$\text{score}(Q,F_d)=\frac{1}{|F_d|}(1+\ln f_{d,t}) \qquad (2)$$

加密检索模型：假设 k、l、l'、p 是 KeyGen(·)中使用到的参数，r 是对称加密算法 ε 使用到的参数

$$\varepsilon:\{0,1\}^l\times\{0,1\}^r\to\{0,1\}^r$$

假设 v 是最多的包含关键词 w_i 的文件个数，$v=\max_{i=1}^{m}N_i$，这个数值不需要提前知道。f 表示一个伪随机函数，π 表示一个抵抗冲突的散列函数，其中

$$f:\{0,1\}^k\times\{0,1\}^*\to\{0,1\}^l$$

$$\pi:\{0,1\}^k\times\{0,1\}^*\to\{0,1\}^p$$

其中，$p>\log m$，实际中，$\pi(\cdot)$ 可以通过离线的散列函数，例如 SHA-1 来创建，其中 p 是 160 位。

1）配置阶段

① 数据所有者通过调用

$$\text{KeyGen}(1^k,1^l,1^{l'},1^p)$$

初始化这个模型，生成随机键 x、y、z，其中 R 表示值域范围

$$x, y \xleftarrow{R} \{0,1\}^k, z \xleftarrow{R} \{0,1\}^l$$

同时输出

$$K = \{x, y, z, 1^l, 1^{l'}, 1^p\}$$

② 数据所有者通过调用 BuildIndex(K, C)从文件集合 C 中构建一个倒排索引，如算法 1 所示，其中 l'个填充 0 表示这是一个有效的倒排记录。

在查询阶段

① 对于一个关键词 w，用户通过调用 Trapdoor Gen(w)生成一个对应的暗门 T_w= $(\pi_x(w), f_y(w))$。

② 云服务器一旦接收到 T_w，它就会调用 SearchIndex(I, T_w)，首先，通过 $\pi_x(w)$ 找到索引中匹配的倒排记录，然后，通过 $f_y(w)$ 对倒排记录解密，然后根据 $F(w)$ 发回对应的文件以及它们关联的被加密的相关度。

③ 用户通过密钥 z 对相关度进行解密并且得到排名后的检索结果。

上述模型支持加密检索过程中的隐私保护，然而，将排名过程交给了用户可能会给用户带来大量的计算和后期处理的时间开销。不仅如此，发回所有文件还会占用大量的网络带宽。一种可能减少通信开销的方式是：云服务器首先发回所有有效的倒排记录 $\langle id(F_{ij}) \| \varepsilon_z(S_{ij}) \rangle$，其中 $1 \leqslant j \leqslant N_i$。然后，用户对相关度进行解密，并且利用被解密的排序的相关度，再次发送给云服务器一个检索 *Top-k* 个最相关文件的请求。由于有效的倒排纪录 $\langle id(F_{ij}) \| \varepsilon_z(S_{ij}) \rangle$ 的大小远远小于对应的文件大小，因此，当用户不查询所有匹配的文件时，这种方式可以节约大量网络带宽。然而，这种方式的一个明显缺陷是：每个用户的一次请求都需要来回处理 2 次，同时，虽然云服务器无法获得相关度，但是它仍然能够推测出请求后返回的文件肯定比请求后没有返回的文件具有更高的相关度，这样可能会泄露一些敏感信息。

算法 1 索引构建

BuildIndex(K,C)

① 初始化

扫 C 并且从 C 中抽取出不同的关键词 $W = (w_1, w_2, \cdots, w_m)$，对每一个 $w_m \in W$ 构建 $F(w_i)$。

② 构建倒排索引表

对每个 $w_m \in W$

for $1 \leqslant j \leqslant |F(w_i)|$：

a) 根据式(2)计算文件 F_{ij} 的相关度，表示为 S_{ij}；

b) 计算 $\varepsilon_z(S_{ij})$，并且用 F_{ij} 的标识符 $\langle id(F_{ij}) \| \varepsilon_z(S_{ij}) \rangle$ 把 $\varepsilon_z(S_{ij})$ 存储到倒排索引表 $I(w_i)$。

③ 确保索引 I 的安全

对每个 $I(w_i)$，其中 $1 \leqslant i \leqslant m$：

a) 用 $key = f_y(w_i)$ 加密所有具有 l' 个填充 0 的 N_i 个记录，$\langle 0^{l'} \| id(F_{ij}) \| \varepsilon_z(S_{ij}) \rangle$，其中 $1 \leqslant j \leqslant v$；

b) 对于剩下的 $v - N_i$ 条记录，把它们设置成与 $I(w_i)$ 现有的 N_i 条记录具有相同规模的随机值；

c) 用 $\pi_x(w_i)$ 替换 w_i。

④ 输出 I

为了让云服务器在不知道相关度的情况下进行排序。采用文献[6]提出的保序对称加密模型实现加密文件上的排名检索。保序对称加密模型是一个确定性的加密模型，其中，文件的相关度排序能够被加密函数所保留。由于采用了保序对称加密，云服务器可以获得相关度排序信息，因此，该模型的隐私保护能力会降低，然而，这也是为了提高加密检索性能所做出的折中处理。下面用 GHD 表示保序对称加密模型。然而，保序加密模型是一个确定性加密模型，这个确定性的性质，使它如果不能被合适处理的话，就会像所有确定性的加密模型一样泄露很多信息，例如攻击者无需知道暗门的构造就可以通过加密的相关度分布逆向推出检索的关键词。为了减少从确定性的性质中泄露相关度信息的风险，提出了一对多的保序映射，它通过混淆原始的相关度分布，增加了随机性，但是仍然能够保序。在 GHD 的加密过程中，定义域 D 中的一个文件 m 总是被映射到值域 R 中同一个随机化大小非覆盖的区间桶，这个映射是被一个在值域 R 上键入的 BinarySearch 和一个随机的 GHD 抽样函数所确定的，m 是密文选取到的随机选取函数的种子。然而，一对多的保序映射不仅利用已有的保序映射中随机文件到桶的映射，而且把文件标识符 ID 连同文件 m 一起作为密文选取过程中的随机选取函数种子。由于文件标识符 ID 作为随机选取函数种子的一部分，同一个文件 m 将不再被确定性地分配同一个密文 c，而是分配一个在值域 R 中随机分配桶中的一个随机

数。整个过程如算法 2 所示。这里 CoinGen(·)是一个随机硬币产生器，同时 GHYEINV(·)是用 MATLAB 实现的 GHD(·)抽样函数。一对多的保序映射的正确性能够通过算法 2 得到。现在，用 *OPM* 表示一个带有参数的一对多的保序映射函数：$OPM=\{0,1\}^{l}\times\{0,1\}^{\log|D|}\rightarrow\{0,1\}^{\log|R|}$，基于该函数，支持排名的关键词加密检索模型的具体实现如下。

1) 配置阶段

① 数据所有者通过调用

$$\mathrm{KeyGen}(1^{k},1^{l},1^{l'},1^{p},|D|,|R|)$$

生成随机键 $x,y,z\xleftarrow{R}\{0,1\}^{k}$，同时输出

$$K=\{x,y,z,1^{l},1^{l'},1^{p},|D|,|R|\}$$

② 然后，数据所有者通过调用 BuildIndex(*K*,*C*)，构建文件集合 *C* 的倒排索引，其中使用 $OPM_{f_z(w_i)}(.)$ 而不是 $\varepsilon(.)$ 对相关度进行加密

算法 2　一对多的保序映射

① procedure $OPM_K(D, R, m, \mathrm{id}(F))$
② while $|D|$! = 1 do
③ $\{D, R\}\leftarrow$ BinarySearch(K, D, R, m);
④ end while
⑤ $coin \xleftarrow{R}$ CoinGen(K, (D, R, 1||m, id(F)))
⑥ $c \xleftarrow{coin} R$
⑦ return c
⑧ end procedure
⑨ procedure BinarySearch(K, D, R, m);
⑩ $M\leftarrow|D|$; $N\leftarrow|R|$;
⑪ $d\leftarrow$ min (D)-1; $r\leftarrow$ min(R)-1;
⑫ $y\leftarrow r+\lceil N/2\rceil$;
⑬ $coin\xleftarrow{R}$ CoinGen(K, (D, R, 0||y))
⑭ $x\xleftarrow{R} d+$ GHYEINV($coin$, M, N, $y-r$);
⑮ if $m\leqslant x$ then
⑯ $D\leftarrow\{d+1,\cdots,x\}$;
⑰ $R\leftarrow\{r+1,\cdots,y\}$;
⑱ else
⑲ $D\leftarrow\{x+1,\cdots,d+M\}$;
⑳ $R\leftarrow\{y+1,\cdots,r+N\}$;
㉑ end if
㉒ return $\{D, R\}$;
㉓ end procedure

2) 查询阶段

① 对于一个关键词 *w*，用户通过调用 Trapdoor Gen(*w*)生成并且发送一个暗门 $T_w=(\pi_x(w),f_y(w))$，云服务器一旦接收到这个 T_w，它就会调用 SearchIndex(*I*, T_w)，首先，通过 $\pi_x(w)$ 找到索引中匹配的倒排记录，然后，通过 $f_y(w)$ 对倒排记录进行解密。

② 云服务器查看文件标识符<$id(F_{ij})$>和它所关联的经过保序映射后的相关度：$OPM_{f_z(w_i)}(S_{ij})$。

③ 云服务器获取文件，并且根据加密的相关度发送给用户排序好的文件或者发送 *Top-k* 个最相关的文件。

在一对多的保序映射的帮助下，云服务器能够和相关度未加密时一样对文件进行排序。使用不同键值 $f_z(w_i)$ 对不同的倒排索引表加密相关度的原因是为了使一对多的保序映射不会被攻击者区分。因此，索引 *I* 的不同倒排记录中的同一个相关度将会被映射到 *R* 内不同的桶中。从整体角度来看，这将会随机化加密的数值，因此能够进一步减少泄露给云服务器敏感信息的风险，这些敏感信息可能会帮助云服务器通过对加密数值上的统计分析来推测出潜在的有用信息。

4　性能分析

通过在真实数据集上的检索请求验证提出方法的性能。使用 RFC 数据库[11]作为数据集，其中包含了 5 563 条普通文本记录，总共大约 277 MB。这个文件集包含了大量的技术性关键词，其中，很多关键词可以唯一区分文件。实验运行在 3.0 GHz 的双核 CPU 的 Linux 机器上。算法使用了 OpenSSL 和 MATLAB 库。实验的目的是验证一对多的保序映射的性能以及支持排名的关键词加密检索的性能，包括索引构建时间，检索时间以及获得相关度最高的文件的时间。

1) 保序映射：由于一对多的保序映射的性能是由定义域 *M* 和值域 *R* 的大小所决定的。*M* 会影响 BinarySearch(·)或者 GHD(·)被调用多少次 $O(\log M)$，同时，*M* 和 *R* 都影响每次 GHD(·)执行的时间开销，这也是为什么随着 *M* 增长，单个一对多的保序映射操作的时间开销比对数增长上升更快的原因。图 2 给出了一对多的保序映射性能的度量结果。这个结果是 100 次测试结果的平均值。由图 2 可知，当 *M* 被设置为 128 时，即使对于区间很大的值域 *R*，一次成功映射仍然能够在 200 ms 内完成。特别地，对

于$|R|=2^{46}$，这个时间开销会少于 70 ms。这个结果远远快于文献[12]和文献[13]提出的保序映射操作，其中文献[12]需要保存大量的元数据以支持数据所有者预先构建大量不同的桶，同时文献[13]需要把相关度的预先抽样和训练过程外包出去，然而，本文方法只需要预先生成随机键 key。

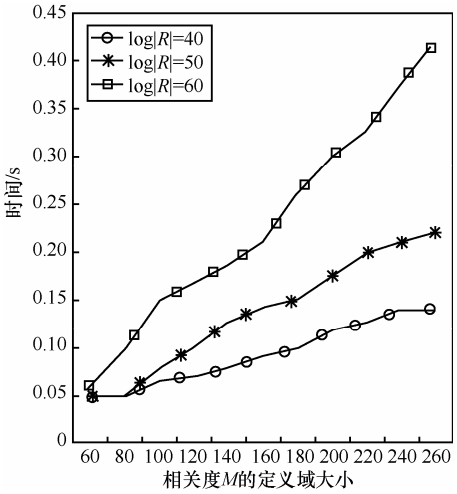


图 2 在不同参数下单个一对多的保序映射操作的时间开销

2) 索引构建：为了实现支持排名的关键词检索，倒排索引在每个倒排记录中都附上一个相关度。用一对多的保序映射后的数值替换原有的相关度的数值。相比原始的倒排索引构建，它只增加映射操作的时间开销，表示加密的相关度的比特位，以及整个倒排记录的加密开销。因此，只在表 2 中列出 1 000 个 RFC 文件集合的索引构建的时间开销。列出的索引大小和构建时间都是基于单个关键词的，这意味着倒排索引表的构建会随着关键词的变化而变化。之所以这样是因为它可以消除不同关键词集合构建的差异，支持整体性能的客观分析。由于云服务器的存储成本很低，所以存储表示加密的相关度的比特位不会给云服务器带来很大影响。然而，由于正常的索引构建时间平均只花费 2.31 s，而倒排索引表中每个倒排记录的一对多的保序映射大约需要 70 ms，因此，一对多的保序映射操作是索引构建的主要影响因素。

表 2 1 000 个 RFC 文件的索引构建时间

文件个数	每个关键词列表大小	每个关键词的索引构建时间
1 000	12.414 KB	5.44 s

3) 检索时间：检索时间包括在索引中获取倒排索引表的时间，对每条倒排记录进行解密以及排序时间。这里重点考虑 Top-k 查询。由于加密的相关度是保序的，云服务器能够像在普通文本一样快速地处理 Top-k 查询。注意对每一个给定的暗门，云服务器不必遍历每个倒排索引表，而是使用基于树的数据结构来获得对应的列表。因此，检索的时间开销和未加密数据上的检索的时间开销基本相同。图 3 显示了随着 k 的增加，上述相同索引结构下的检索的时间开销。

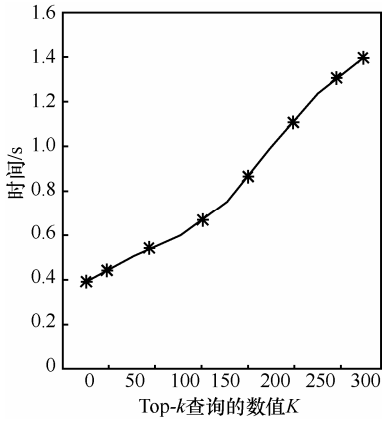


图 3 Top-k 查询的时间开销

4) 获得相关度最高文件的时间：这里比较了文献[4]的加密检索方法和本文提出的加密检索方法在不同测试文件数量下获得相关度最高的文件的时间开销，如图 4 所示，文献[4]的加密检索方法返回的结果由于没有描述相关度信息，所以用户需要在返回结果中再次查找，导致时间开销线形增长，用户人工筛选时间远远高于自动检索时间，而本文提出的加密检索方法支持排名检索，返回的结果按照相关度排序，所以返回的第一个结果就是相关度最高的文件，整个时间基本保持稳定，并且远远少于文献[4]的加密检索方法的时间。

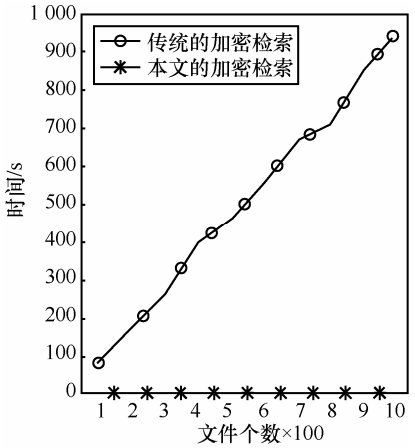


图 4 获得相关度最高文件的时间开销

5　结束语

为了有效地对存储在远程的云服务器中的加密数据进行检索，首先分析了现有的加密检索模型的缺陷，然后通过合理弱化隐私保护的要求，利用一对多的保序映射设计了支持排名的关键词加密检索方法。通过实验分析，该方法能够实现支持排名的关键词检索，并且具有良好的性能。

参考文献：

[1] VAQUERO L M, RODERO-MERINO L, CACERES J, *et al*. A break in the clouds: towards a cloud definition[J]. ACM SIGCOMM Computer Communication Review, 2009, 39(1):50-55.

[2] 孟小峰, 慈祥. 大数据管理: 概念, 技术与挑战[J]. 计算机研究与发展, 2013, 50(1):146-169.
MENG X F, CI X. Big data management: concepts, techniques and challenges[J]. Journal of Computer Research and Development, 2013, 50(1):146-169.

[3] KAMARA S, LAUTER K. Cryptographic Cloud Storage[M]. Financial Cryptography and Date Security Berlin, Heidelberg, 2010.136-149.

[4] CURTMOLA R, GARAY J A, KAMARA S, *et al*. Searchable symmetric encryption: improved definitions and efficient constructions[A]. Proceedings of the 13th ACM Conference on Computer and Communications Security[C]. 2006.79-88.

[5] HWANG Y H, LEE P J. Public key encryption with conjunctive keyword search and its extension to a multi-user system[A]. Pairing-Based Cryptography–Pairing 2007[C]. Springer Berlin Heidelberg, 2007. 2-22.

[6] BOLDYREVA A, CHENETTE N, LEE Y, *et al*. Order-preserving symmetric encryption[A]. Cryptology-EUROCRYPT 2009[C]. Springer Berlin Heidelberg, 2009.224-241.

[7] BOŽOVIĆ V, SOCEK D, STEINWANDT R, *et al*. Multi-authority attribute-based encryption with honest-but-curious central authority[J]. International Journal of Computer Mathematics, 2012.89(3): 268-283.

[8] SINGHAL A. Modern information retrieval: a brief overview[J]. IEEE Data Engineering Bulletin, 2001,24(4):35-43.

[9] WITTEN I H, MOFFAT A, BELL T C. Managing Gigabytes: Compressing and Indexing Documents and Images[M]. Morgan Kaufmann , 1999.

[10] PINKAS B, REINMAN T. Oblivious RAM revisited[A]. Cryptology—CRYPTO 2010[C]. Springer Berlin Heidelberg, 2010. 502-519.

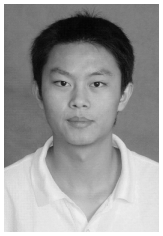
[11] RFC. Request for Comments Database[EB/OL]. http://www.ietf.org/rfc.html.

[12] SWAMINATHAN A, MAO Y, SU G M, *et al*. Confidentiality-preserving rank-ordered search[A]. Proceedings of the 2007 ACM Workshop on Storage Security and Survivability[C]. 2007.7-12.

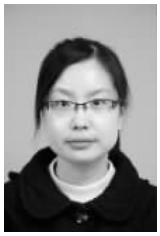
[13] ZERR S, OLMEDILLA D, NEJDL W, *et al*. Zerber+*r*: top-*k* retrieval from a confidential index[A]. Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology[C]. 2009.439-449.

作者简介：

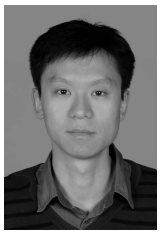
张鹏（1984-），男，安徽淮南人，中国科学院信息工程研究所助理研究员，主要研究方向为分布式系统和数据挖掘以及网络安全。



李焱（1984-），男，湖北随州人，国家计算机网络应急技术协调中心工程师，主要研究方向为分布式系统和云计算。



林海伦（1987-），女，山东临沂人，中国科学院博士生，主要研究方向为数据挖掘和信息检索。



杨嵘（1978-），男，山西运城人，中国科学院高级工程师，主要研究方向为网络安全和大数据处理。

刘庆云（1980-），男，河北邯郸人，中国科学院高级工程师，主要研究方向为网络安全和大数据处理。