一种云环境中密文数据的模糊多关键词检索方案

何 亨 夏 薇 张 继 金 瑜 李 鹏
(武汉科技大学计算机科学与技术学院 武汉 430065)
(武汉科技大学智能信息处理与实时工业系统湖北省重点实验室 武汉 430065)

摘 要 越来越多的企业和个人用户将大量的数据存储在云服务器。为了保障数据隐私,重要数据以密文形式存储在云端,但却给数据检索操作带来严峻挑战。传统的基于明文的检索方案不再适用,已有的基于密文的检索方案存在不支持模糊检索或多关键词检索、效率较低、空间开销较大、不支持检索结果排序等问题。因此,研究安全高效的密文检索方法具有重要意义。提出了一种新的云环境中密文数据的模糊多关键词检索方案,该方案能够从云服务器上检索出包含有指定多个关键词的密文,支持模糊关键词检索,并且不会向云服务器和其他攻击者泄露与数据和检索相关的任何明文信息;使用计数型布隆过滤器和MinHash算法构建索引向量和查询向量,使得索引构建和查询过程更加高效,且排序结果更加准确。安全性分析和性能评估表明该方案具有高安全性、可靠性、检索效率和准确率。

关键词 云计算,模糊检索,多关键词检索,密文数据,安全索引
中图法分类号 TP393.08 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.05.026

Fuzzy Multi-keyword Retrieval Scheme over Encrypted Data in Cloud Computing

HE Heng XIA Wei ZHANG Ji JIN Yu LI Peng
(School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065, China)
(Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, Wuhan 430065, China)

Abstract Nowadays more and more enterprise and individual users store a large amount of data in the cloud. To protect data privacy, important data has to be encrypted before being stored in the cloud, which brings a severe challenge to the retrieval of data. Traditional retrieval schemes based on plaintext have not been applicable, and existing retrieval schemes based on ciphertext have many shortcomings, some of which do not support fuzzy search or multi-keyword search, poor efficiency or large space overhead, or do not return ranking results. Therefore, researching secure and efficient retrieval scheme on ciphertext is of great significance. A new fuzzy multi-keyword retrieval scheme over encrypted data in cloud computing was proposed, which can retrieve the ciphertext containing multiple keywords from cloud server, support fuzzy keyword search, and will not leak any plaintext information of data and retrieval to cloud server and other attackers. In the scheme, counting bloom filter and MinHash algorithm are used to construct index vectors and query vectors, which makes the process of building index and querying more efficient, and the ranking results more accurate. The security analysis and performance evaluation show that our scheme has high security, reliability, retrieval efficiency and accuracy.

Keywords Cloud computing, Fuzzy search, Multi-keyword search, Encrypted data, Secure index

1 引言

随着云计算[1]技术的飞速发展,越来越多的企业和个人用户选择将他们的本地数据文件存储在云服务器中,以有效降低本地数据存储和管理的成本,同时获得高质量的应用服务。然而,云服务器被认为是"诚实但好奇"的,即云服务器会诚实地执行系统协议和功能,但是其会主动探测存放在其上的重要文件内容[2]。为了保障数据安全和用户隐私,用户通常先对敏感的数据进行前端加密,再将密文数据存储在云服务器中[2]。密文数据则不再具有明文的可检索特性。这一过程使得用户如何在加密数据中检索出他们所需要的内容成为一个非常具有挑战性的难题。

到稿日期:2016-09-13 返修日期:2017-01-07 本文受国家自然科学基金(61602351,61502359,61303117),武汉科技大学国家自然科学基金预研项目(2015XG005),智能信息处理与实时工业系统湖北省重点实验室开放基金项目(2016znss10B)资助。
何 亨(1981-),男,博士,讲师,CCF会员,主要研究方向为云计算、网络安全、信息检索,E-mail:heheng@wust.edu.cn(通信作者);夏 薇(1992-),女,硕士生,主要研究方向为云计算、信息检索;张 继(1992-),男,硕士生,主要研究方向为云计算、网络安全;金 瑜(1973-),女,博士,副教授,主要研究方向为云计算、网络安全;李 鹏(1981-),男,博士,副教授,主要研究方向为云计算、移动计算。

传统的信息检索方案已经不适用于云环境中密文数据的检索。针对云环境中的密文检索问题[3]，国内外研究人员进行了深入的研究，已有方案分为精确关键词检索和模糊关键词检索两大类，其中大多数方案的主要过程都是数据拥有者先对每个文件关键词建立索引，再将加密的索引和数据存储到云服务器端，数据使用者在授权后通过输入加密的查询关键词进行检索。

Song 等人[4]首次提出基于对称密钥的单关键词可检索加密方案，该方案并不支持索引，而是对每个文件进行操作，效率较低。Curtmola 等人[5]在 Song 的基础上给出了更严格的安全性定义，并构建了更高效的对称密钥可检索加密方案。Boneh 等人[6]首次提出非对称可检索加密方案，该方案仅考虑单个关键词查询，只支持精确匹配，并且会泄漏用户的访问模式。Wang 等人[7]提出基于对称密钥保序加密技术的单关键词分级密文检索方案，该方案需要扫描所有文件，不易进行索引更新。此外，单关键词检索不足以满足用户的个性化检索需求，且涉及过多检索结果，会导致巨大的网络通信负载。Cao 等人[8]第一次提出基于多关键词的密文检索方案，基于安全 kNN 查询技术[9]中索引向量与查询向量间的内积相似度来实现检索结果的排序。Li 等人[10]提出了更为高效的支持排序的多关键词密文检索方案。冯贵兰等人[11]提出了一种基于多属性排序的密文检索方案，以提高检索速度和准确性。上述方案都属于精确关键词检索，即只能提供关键词的精确查找，不允许出现任何细微的拼写错误或者格式的不一致性问题，也未考虑数据之间的相关性，在实际应用中不足以满足用户的检索需求。

为了提高检索的适用性和灵活性，一些模糊关键词检索方案被提出。Li 等人首次提出了一种基于通配符的模糊关键词检索方案[2]，使用通配符和编辑距离进行匹配，服务器能够返回与所查关键词相似的结果。Liu 等人[12]进一步提出了一种基于字典的模糊关键词检索方案，使用字典包含所有可用的英语单词。然而，这些方案效率较低，存储开销较大，并且都不支持多关键词检索。Wang 等人[13]首次提出了云环境中基于隐私保护的多关键词模糊检索方案，使用局部敏感哈希及布隆过滤器[14]构建索引，大幅提高了检索效率及精度，但该方案中没有确切的方法来支持检索结果排序。因此研究设计云环境中具有高安全性、可靠性、检索效率和准确率的密文数据模糊多关键词检索方案具有重要意义和实用价值。

基于上述研究成果，本文提出了一种新的云环境中密文数据的模糊多关键词检索方案。云服务器可以按照用户需求，检索出包含有指定多个关键词的密文，支持模糊关键词检索，并且不会向云服务器和其他攻击者泄露与数据和检索相关的任何信息。相比现有相关方案，索引构建和检索过程更加高效，并且排序结果更加准确。具体包括：

1）在 Wang 等人[13]方案的基础上引入 MinHash 算法[15]来建立索引。在不影响检索精确度的前提下，通过 MinHash 算法对关键词集合进行降维，并基于 Jaccard 距离[15]计算向量相似度，能极大提高检索效率，也能减少空间开销。

2）使用计数型布隆过滤器[16]作为索引结构，使用其计数表示关键词权重。在文件中出现频率高的关键词在对应的索引结构中权重高，在返回的排序结果集中，具有较高关键词权重的文件排序位置更加靠前，从而可以返回更精确的结果。

2　问题描述

2.1　系统与安全模型

在云环境中密文数据的模糊多关键词检索系统中主要存在 3 个实体：数据拥有者（Data Owner，DO）、数据使用者（Data User，DU）和云服务器（Cloud Server，CS）。任何系统用户都可以作为 DO 在 CS 上存储并发布数据，也可以同时作为被授权的 DU 获得其他用户的数据访问权限。

图 1 示出了系统总体结构图。首先，DO 为文件集合中每个文件抽取关键词生成关键词集合，为关键词集合建立可检索的安全索引，并将加密后的文件和安全索引一起上传至 CS。接着，当 DU 需要对密文检索时，如果 DU 是授权用户，直接使用预先分配的密钥对查询语句进行加密得到陷门函数；如果 DU 预先没有被 DO 授权，则需要把查询语句发送给 DO 以获取陷门函数。最后，DU 将陷门函数发送给 CS，CS 根据安全索引和陷门为 DU 的请求进行查询，将匹配到的密文文件返回给 DU。

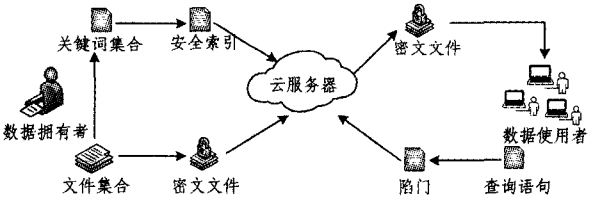

图 1　云环境中密文数据的模糊多关键词检索系统总体结构图

本文认为 CS 是半可信的，即是“诚实但好奇”的，它会正确地执行程序（不会添加或篡改来自 DO 或 DU 的指令集），但它也会窥探关键词信息并根据陷门函数之间的关联来推测文件内容。正如文献[2]所述，本文亦采用如下两种威胁模型：1）已知密文模型，即 CS 只拥有密文文件、安全索引和 DU 提交的陷门函数，并且 CS 只知道记录查询结果；2）已知背景模型，在已知密文模型的基础上，CS 会统计收到的陷门函数，并分析其中包含的信息来推测文件中包含的关键词。

为了确保用户数据和检索的隐私性，避免 CS 收集有关背景信息来推断关键词，本文方案将实现：1）索引机密性：CS 不能获得加密的索引中包含的明文信息，如关键词和关键词的词频；2）数据机密性：CS 不能获得密文文件中包含的明文数据信息；3）查询机密性：CS 不能得到加密的请求中包含的明文信息，如请求中的关键词和词频；4）陷门函数无关联：不能推断两个或两个以上的陷门是否来自同一个请求。

2.2　设计目标

本方案主要实现以下安全和性能目标：

多关键词模糊检索：能够检索出 CS 中包含有指定多个关键词的密文，支持模糊关键词检索，即用户输入拼写错误的关键词能够检索出包含正确关键词的文件。

隐私保护：不会将数据文件、索引文件和检索关键词的明文信息泄露给 CS 和其他攻击者。

检索高效：索引构建和检索过程保持较高的计算效率，不需要预先定义的词典，空间开销小。

结果精确：返回尽可能精确的排序结果。

2.3 预备知识

2.3.1 LSH 算法

局部敏感哈希(Locality Sensitive Hashing，LSH)算法[17]主要用于高效求解最近邻搜索问题。该算法的基本思想是通过一组满足特定约束条件的哈希函数将数据集映射到多个哈希表的不同冲突桶中，从而建立多个哈希表，使得在某种相似度量条件下，距离越近(相似度越高)的点映射到相同冲突桶中的概率越大，而距离越远(相似度越低)的点映射到相同冲突桶的概率越小。通过这种方式，将数据集分成了多个子集，而每个子集中的数据间是相邻的且该子集中的元素个数较少，因此将一个在超大集合内查找相邻元素的问题转化为了在一个很小的集合内查找相邻元素的问题，从而极大地减少了检索时要比较的数据量。

对于集合 S，任意两点 $x,y\in S$，如果哈希函数集 H 中的每一个函数 h 满足以下两个条件，则 H 称为 (d_1,d_2,p_1,p_2)-敏感：

1)如果 $d(x,y)\leqslant d_1$，则 $Pr[h(x)=h(y)]\geqslant p_1$；

2)如果 $d(x,y)\geqslant d_2$，则 $Pr[h(x)=h(y)]\leqslant p_2$。

其中，$d(x,y)$表示 x 和 y 之间的距离，$h(x)$和 $h(y)$表示对 x 和 y 进行哈希映射，$Pr[h(x)=h(y)]$表示 $h(x)=h(y)$ 的概率，$d_1<d_2$，$p_1>p_2$。

通过一个或多个 (d_1,d_2,p_1,p_2)-敏感的哈希函数对数据集进行哈希映射生成一个或多个哈希表的过程即为局部敏感哈希。

2.3.2 MinHash 算法

MinHash[15]是一种基于 Jaccard 相似度的算法，属于 LSH，用于快速估算两个集合的相似度。对于集合 A 和 B，其相似度可以用 Jaccard 系数定义：

$$J(A,B)=\frac{|A\cap B|}{|A\cup B|}$$

当集合较大或者集合数量过多时，直接计算集合交集与并集过于耗时，因此提出了 MinHash 方法，其基本过程为：令 $h(x)$为把 x 映射成一个整数的哈希函数，$h_{\min}(S)$为集合 S 中的元素经过 $h(x)$哈希后，具有最小哈希值的元素，那么对集合 A,B，$h_{\min}(A)=h_{\min}(B)$成立的条件是 $A\cup B$ 中具有最小哈希值的元素也在 $A\cap B$ 中。此处假设 $h(x)$是一个良好的哈希函数，其具有较好的均匀性，能够将不同元素映射成不同的整数。所以有 $\Pr[h_{\min}(A)=h_{\min}(B)]=J(A,B)$，即集合 A 和集合 B 的相似度为集合 A 和集合 B 经过哈希后最小哈希值相等的概率。

在本文中，为了减少计算开销，定义一个 MinHash 系数 $\lambda,\lambda\in(0,1]$，设集合 S 中包含 n 个元素，则对 S 进行一次哈希后，取其中具有最小哈希值的 $\lambda\times n$ 个元素作为 $h_{\min}(S)$。

2.3.3 计数型布隆过滤器

布隆过滤器(Bloom Filter，BF)[16]由一组哈希函数和一个二进制向量组成，用于在低错误率的前提下高效地判断某个元素是否属于某个集合。BF 的基本原理是：用 k 个值域为 $[0,m)$整数的哈希函数对数据集合 S 中的每个对象计算一个地址序列$(h_1,h_2,\cdots,h_k)$，然后设二进制向量对应的地址序列的位置为 1。计数型布隆过滤器(Counting Bloom Filter，CBF)由 BF 扩展而来，主要解决 BF 无法进行插入和删除操作的问题。CBF 将 BF 的二进制向量的每一位扩展成一个计数器，初值为 0，如图 2 所示，在得到每个数据对象在 CBF 向量中的地址序列后，将向量对应地址序列上的计数器值加 1，从而将数据对象映射到 CBF 中。

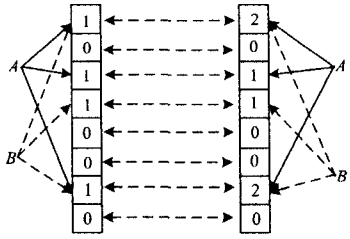

图 2 BF 和 CBF 结构图

CBF 在判断一个元素是否属于它表示的集合时会有一定的误报率，即元素在经过哈希计算后得到的索引值错误，导致相应的地址序列没有被正确地置位。

假设每个元素都被等概率地哈希到对应的地址序列，则在每个地址序列中一个确定位被设置为 1 的概率为 $1/m$，相反被设置为 0 的概率为 $1-1/m$，k 个哈希函数中没有一个将确定位设置为 1 的概率为$(1-1/m)^k$，当把集合中的 n 个元素全部插入 CBF 后，每一个确定位为 0 的概率为 $p=(1-1/m)^{nk}$，p 即 CBF 的误报率。m,n,k 在数值上存在一定的关系，DO 可以对三者进行适当取值，保证误报率最小。

3 方案实现

3.1 设计概述

本方案的整体流程图如图 3 所示。

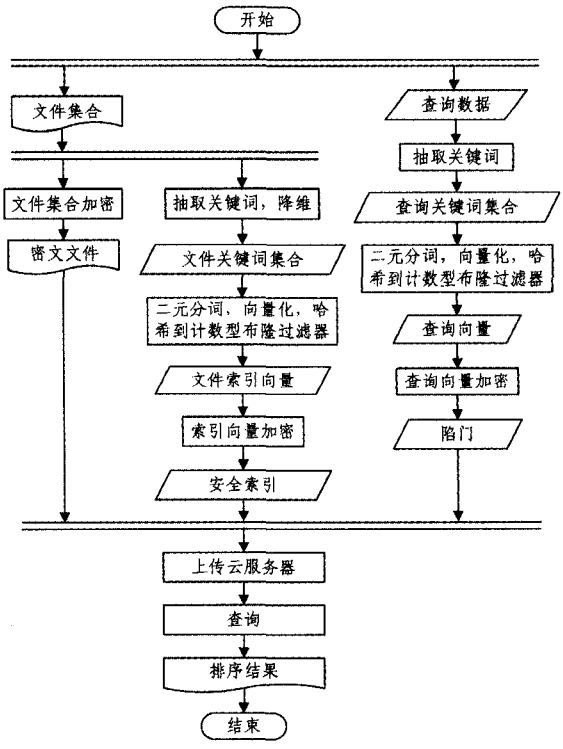

图 3 云环境中密文数据的模糊多关键词检索方案流程

首先，DO 对文件集合中每个文件抽取关键词生成关键

词集合,使用 MinHash 算法对关键词集合进行降维,再对降维后的每个关键词进行二元分词和向量化,并以 Jaccard 距离为相似度度量,将关键词的二元向量哈希到 CBF 中,其中相似度在一定范围内的二元向量组被哈希到 CBF 的同一地址序列中,从而生成文件的索引向量;然后,DO 对文件集合进行加密,并使用随机安全参数生成密钥对文件索引向量进行加密,生成密文文件和安全索引,将其发送到 CS;接着,预先被授权的 DU 或 DO 对要查询的数据抽取关键词,进行二元分词和向量化后,与生成索引时类似的操作,得到查询向量,并加密得到陷门函数,将其发送给 CS;最后,CS 根据陷门和安全索引进行查询,找到相关度最高的一定的数量文件并进行排序后返回给 DU。

3.2　核心算法

本方案包括 KeyGen, IndexEnc, QueryEnc, BuildIndex_B, BuildIndex_E, Trapdoor, Search 共 7 种核心算法,下面分别进行描述:

KeyGen(m):使用安全参数 m 生成安全密钥 $SK=(M_1, M_2, S)$,M_1, M_2 是 m 阶可逆矩阵,$S=\{0,1\}^m$ 是一个 m 位向量。

IndexEnc(SK, I):使用密钥 SK 将文件索引向量 I 加密得到安全索引向量,步骤为:

1)遵循以下规则将 I 分解为两个向量 I' 和 I'':

依次遍历向量 $S=\{0,1\}^m$ 的每一位,记作 $S[t]$,而 $I[t]$, $I'[t]$, $I''[t]$分别表示对应向量的每一位,$t\in\{1,\cdots,m\}$,则有:

规则 1　当 $S[t]=1$ 时,$I'[t]=I''[t]=I[t]$;

规则 2　当 $S[t]=0$ 时,$I'[t]=I[t]/2+r$, $I''[t]=I[t]/2-r$, r为一个随机数;

2)使用矩阵 M_1, M_2 将 I', I''加密得到$(M_1^{\mathrm{T}} I', M_2^{\mathrm{T}} I'')$;

3)输出 $Enc_{sk}(I)=(M_1^{\mathrm{T}} I', M_2^{\mathrm{T}} I'')$作为文件的安全索引向量。

QueryEnc(SK, Q):使用密钥 SK 将查询向量 Q 加密得到陷门函数,步骤为:

1)遵循以下规则将 Q 分解为两个向量 Q' 和 Q'':

依次遍历向量 $S=\{0,1\}^m$ 的每一位,记作 $S[t]$,而 $Q[t]$, $Q'[t]$, $Q''[t]$分别表示对应向量的每一位,$t\in\{1,\cdots,m\}$,则有:

规则 3　当 $S[t]=0$ 时,$Q'[t]=Q''[t]=Q[t]$;

规则 4　当 $S[t]=1$ 时,$Q'[t]=Q[t]/2+r'$, $Q''[t]=Q[t]/2-r'$, r'为一个随机数;

2)使用矩阵 M_1, M_2 将 Q', Q'' 加密得到 $(M_1^{-1}Q', M_2^{-1}Q'')$;

3)输出 $Enc_{sk}(Q)=(M_1^{-1}Q', M_2^{-1}Q'')$作为查询的陷门函数。

BuildIndex_B(D, SK, H):使用包含 k 个独立哈希函数的集合 $H=\{h_j \mid h_j:\{0,1\}^{26\times 26}\to\{0,1\}^m, j=\{1,2,\cdots,k\}\}$和密钥 SK 为文件 D 构建安全索引向量,步骤为:

1)对 D 抽取关键词集合为 $W_D=\{w_{D1}, w_{D2}, \cdots, w_{Dn}\}$,选取 MinHash 系数 λ,$\lambda\in(0,1]$,使用 MinHash 算法对 W_D 降维,使得生成具有最小哈希值的关键词集合为 $h_{\min}(W_D)$,其中 $h_{\min}(W_D)$中的关键词个数 $n'=\lambda\times n$;

2)对 $h_{\min}(W_D)$中的每个关键词进行二元分词及向量化,得到其二元向量 $w_{Di}, w_{Di}\in\{0,1\}^{26\times 26}$;

3)为 D 构造一个 m 位 CBF 作为 D 的索引向量 I;

4)使用 H 中的每个哈希函数 h_j 对每个 w_{Di}进行运算,将其映射到 I 中;

5)使用 Index_Enc(SK, I)对 I 加密得到安全索引向量,输出 $Enc_{SK}(I)$。

BuildIndex_E(D, SK, H):使用包含 k 个独立哈希函数的集合 $H=\{h_j \mid h_j:\{0,1\}^{26\times 26}\to\{0,1\}^m, j=\{1,2,\cdots,k\}\}$和密钥 SK 为文件 D 构建安全索引向量,步骤为:

1)对 D 抽取关键词集合为 $W_D=\{w_{D1}, w_{D2}, \cdots, w_{Dn}\}$,选取 MinHash 系数 λ,$\lambda\in(0,1]$,使用 MinHash 算法对 W_D 降维,生成具有最小哈希值的关键词集合 $h_{\min}(W_D)$,其中 $h_{\min}(W_D)$中的关键词个数 $n'=\lambda\times n$,并将随机选取的 d 个虚拟关键词加入集合 $h_{min}(W_D)$中,d 的范围可以根据 CBF 误报率来确定;

步骤 2)—步骤 5)同 BuildIndex_B(D, SK, H)。

Trapdoor(q, SK, H):使用与 BuildIndex(D, SK, H)中相同的哈希函数集 H 和密钥 SK 为查询语句 q 生成陷门函数,步骤为:

1)对 q 抽取关键词集合为 $W_q=\{w_{q1}, w_{q2}, \cdots, w_{qt}\}$;

2)对每个关键词进行二元分词及向量化得到其二元向量 $w_{qi}, w_{qi}\in\{0,1\}^{26\times 26}$;

3)为 q 构造一个 m 位 CBF 作为 q 的查询向量 Q;

4)使用 H 中的每个哈希函数 h_j 对每个 w_{qi}进行运算,将其映射到 Q 中;

5)使用 Query_Enc(SK, Q)对 Q 加密得到陷门函数,输出 $Enc_{SK}(Q)$。

Search($Enc_{SK}(Q), Enc_{SK}(I)$):计算安全索引向量 $Enc_{SK}(I)$和陷门函数 $Enc_{SK}(Q)$之间的内积为:

$$(M_1^{\mathrm{T}} I')^{\mathrm{T}} \cdot M_1^{-1} Q' + (M_2^{\mathrm{T}} I'')^{\mathrm{T}} \cdot M_2^{-1} Q' = I'^{\mathrm{T}} \cdot Q' + I''^{\mathrm{T}} \cdot Q''$$

根据规则 1—规则 4:

当 $S[t]=1$ 时,

$$\begin{aligned} &I'[t]^{\mathrm{T}} Q'[t] + I''[t]^{\mathrm{T}} Q''[t] \\ &\quad = I[t](Q[t]/2+r') + I[t](Q[t]/2-r') \\ &\quad = I[t]Q[t] \end{aligned}$$

当 $S[t]=0$ 时,

$$\begin{aligned} &I'[t]^{\mathrm{T}} Q'[t] + I''[t]^{\mathrm{T}} Q''[t] \\ &\quad = (I[t]/2+r)Q[t] + (I[t]/2-r)Q[t] \\ &\quad = I[t]Q[t] \end{aligned}$$

即 $I'^{\mathrm{T}} \cdot Q' + I''^{\mathrm{T}} \cdot Q' = I^{\mathrm{T}} Q$。

3.3　执行步骤

本方案包括预处理、安全索引和密文文件生成、密文查询、动态更新 4 个主要执行步骤,下面分别对其进行描述。

3.3.1　预处理

DO 选取一个安全参数 m,调用算法 KeyGen(m),生成安全密钥 SK。

3.3.2　安全索引和密文文件生成

DO 基于 LSH 算法选取包含 k 个独立哈希函数的集合 $H=\{h_j \mid h_j:\{0,1\}^{26\times 26}\to\{0,1\}^m, j=\{1,2,\cdots,k\}\}$,对文件集

合中每个文件 D 调用算法 BuildIndex_B(D,SK,H),对 D 抽取包含 n 个关键词的集合 W_D,对 W_D 进行降维后得到 $h_{min}(W_D)$,包含 $\lambda \times n$ 个关键词,对其中的关键词进行二元分词及向量化,构建文件索引向量 I,并生成安全索引向量 $Enc_{SK}(I)$,所有文件的安全索引向量构成文件集的安全索引。考虑到 CS 可能会基于接收的陷门函数进行统计攻击,对于这种安全性相对较低的 CS,DO 也可以调用算法 BuildIndex_E(D,SK,H)来完成相关工作。接着 DO 使用 AES 或 DES 算法对文件集合进行加密,生成密文文件,并将密文文件和安全索引一起发送给 CS。

本方案使用 26^2 位的向量来表示关键词的二元分词对应的二元向量,该向量中的每个元素代表 26 个字母两两排列组合的一种,如果对应元素存在于给定关键词的二元分词中,则将这个元素值设为 1,否则设为 0。例如,对关键词"network"经过二元分词后为{ne,et,tw,wo,or,rk},将其表示为二元向量为{1,…,1,…,1,1,…,1,1,0},而拼写错误的"netword"的二元分词为{ne,et,tw,wo,or,rd},将其表示为二元向量为{1,…,1,…,1,1,…,1,0,1}。这种二元向量可以表示任何单词,并对拼写错误具有高容忍度,一个关键词的错误拼写都可以被映射为一个二元向量,并且与原始向量保持较低的差异。

本方案为每个文件构建一个 CBF 作为文件的索引向量,将关键词的二元向量哈希到 CBF 中,并将相似度在一定范围内的二元向量哈希到 CBF 同一个地址序列中并计数,使用其计数表示关键词权重。在本方案中,以 Jaccard 距离度量向量之间的相似度。对于两个二元向量,它们之间 Jaccard 距离的计算公式为:$D=1-f_{11}/(f_{11}+f_{00}+f_{10}+f_{01})$,其中 f_{11} 表示两个向量同一位置都是 1 出现的次数,f_{00} 表示两个向量同一位置都是 0 出现的次数,f_{10} 表示同一位置一个向量为 1,另一个向量为 0 出现的次数,f_{01} 则反之;D 越小,表示两个向量之间的相似度越高;当两个向量之间的 D 小于预设的阈值时,它们将被哈希到 CBF 的同一地址序列中。例如,关键词"network"与"netword"的二元向量将被哈希到 CBF 的同一地址序列中,对应的计数器值将被设置为 2。

在本方案中,在选定较低的误报率 p 后,根据确定的参数 m 及关键词个数,再选取合适的哈希函数个数 k,在保证方案检索准确性的同时,空间和时间上取得较高的效率。由于 CBF 的误报率随着被插入的关键词个数以及哈希函数个数的增加而减少,当 CS 安全性相对较低时,在保证误报率较低并且不影响方案整体效率的情况下,选取合适的 k 值,并计算引入的虚拟关键词 d 的取值,以防止 CS 进行统计攻击。在为每个文件构造 CBF 作为其索引向量时,向关键词的集合中插入 d 个随机虚拟关键词,对所有的关键词进行二元分词向量化,并哈希到 CBF 中。

为了更清晰地描述本方案的关键步骤,本文以英文文件内容为示例进行说明,对于中文文件,在得到关键词集合后,将所有关键词全切分转换为对应的五笔字型码,接下来对关键词的五笔字型码进行操作,具体步骤与对英文关键词的操作完全相同。

3.3.3 密文查询

预先未被授权的 DU 将查询语句 q 发送给 DO,DO 调用算法 Trapdoor(q,SK,H)为 q 抽取关键词,对每个关键词进行二元分词及向量化,构建查询向量 Q,并生成陷门函数 $Enc_{SK}(Q)$,返回给 DU;预先被授权的 DU 则直接使用 DO 分配的密钥 SK 等安全参数生成陷门函数即可;接着 DU 把陷门函数发送给 CS。

CS 在接收到 $Enc_{SK}(Q)$ 后,调用算法 Search($Enc_{SK}(Q)$,$Enc_{SK}(I)$),计算每个安全索引向量与陷门函数之间的内积,并对内积结果进行排序,将内积最高的一定数量的文件按顺序返回给 DU。

例如,文件 D 包含关键词"computer,network,data,network",按照 3.3.2 节中的操作步骤,可以为其生成索引向量,如图 4 所示,其中 network 的权重为 2,其他关键词的权重为 1;当查询语句 q 包含关键词"conputer,netword"时,可以为其生成查询向量,如图 4 所示。因此,CS 最后计算该文件索引向量与查询向量的内积为 6。

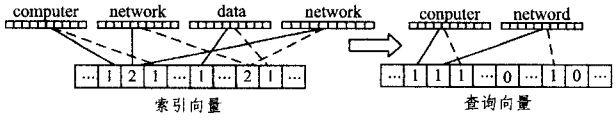

图 4 索引向量与查询向量示例图

3.3.4 动态更新

当需要进行文件更新时,DO 根据实际需求对文件内容进行增加、修改或删除操作,由于本方案不需要预先定义的词典,只需要根据生成索引操作变更索引,同时生成新的密文文件和安全索引,并将其发送至 CS 即可。

4 安全性分析和性能评估

4.1 安全性分析

本文在算法设计上采用了文献[13]中的安全密钥组成 $SK=(M_1,M_2,S)$,M_1,M_2 是 m 阶可逆矩阵,$S=\{0,1\}^m$ 是一个 m 位向量,分别使用 M_1,M_2 的转置矩阵和可逆矩阵对索引向量及查询向量进行加密得到安全索引及陷门。由于每一个可逆矩阵只有唯一的一个可逆矩阵与其互为逆矩阵,利用密钥矩阵空间无穷大的特性,在每次初始化时对密钥矩阵进行随机选择,因此非法用户求得密钥矩阵的可能性为 0。当非法用户进行查询时,在对查询数据 q 抽取关键词并二元分词后,生成 m 位伪造查询向量 Q,由 CBF 的特性可得 Q 为正确查询向量的可能性为 $1/16^m$,且非法用户推断出向量 S 的概率为 $1/2^m$。即使非法用户推断出正确的 Q 和 S,构造了两个 m 位的可逆矩阵 M_3,M_4,并使用 QueryEnc(SK,Q)相同的步骤计算陷门函数 $Enc_*(Q)'=(M_3^{-1}Q',M_4^{-1}Q')$,然后将陷门函数发送至云服务器,由于 M_3,M_4 与 M_1,M_2 相同的可能性为 0,云服务器使用该陷门函数与安全索引进行内积运算得到正确结果的可能性亦为 0,从而不会得到正确的查询结果,即保证了本方案的安全性。

本文在已知密文模型下能够确保用户数据和检索的隐私性,主要包括:1)索引机密性,将文件关键词的二元向量映射到 CBF 隐藏关键词信息,且使用矩阵加密文件索引向量时使用随机数保证了索引机密性;2)数据机密性,使用 AES 或

DES算法对明文数据加密，保证了数据机密性；3）查询机密性，只有预先被授权的 DU 或是通过 DO 才能执行查询操作，同样将查询关键词的二元向量映射到 CBF 隐藏关键词信息，并使用矩阵加密查询向量保证了查询机密性；4）陷门函数无关联，在生成陷门函数时使用随机数，使得对于同一用户的不同查询关键词、同一用户针对不同 DO 的同一查询关键词、不同用户的任何查询关键词，其陷门函数不同且无关联性。

在已知背景模型中，CS 可能进行统计攻击，即根据部分陷门函数的关联、部分关键词或关键词词频以及部分关键词与其他关键词的关联等信息来分析查询过程，推测不同陷门函数之间的关系，以及文件中包含的关键词信息。在已知背景模型中，向每个文件中引入 d 个随机选取的虚拟关键词，打乱索引关键词的映射，虚拟关键词的重复率低，使 CS 无法根据关键词或关键词词频之间的关联推断出包含相同关键词的安全索引之间的关联，对关键词词频进行隐藏，保证了索引关键词的安全。在查询过程中，使用随机数保证查询关键词的安全。根据 CBF 的误报率来设置 d 的取值，达到隐藏关键词关联的目的，使 CS 难以进行统计攻击。

4.2　性能评估

对本方案的关键步骤进行系统性能评估，包括 MinHash 系数 λ 的选取、安全索引构建、陷门函数生成、查询结果返回。方案实现基于 Java 语言，执行在 Intel Core i5-3230M 2.60 GHz 的处理器及 Ubuntu 12.10 操作系统平台上。实验采用真实数据集：安然公司邮件数据集（Enron Email Dataset，EED）[19]，EED 是目前相关研究中使用最多的公开数据集，其邮件数据是安然公司（原是世界上最大的综合性天然气和电力公司之一）150 位高级管理人员的往来邮件。

（1）MinHash 系数 λ 的选取

在本方案中，MinHash 系数 λ 的选取决定降维后的关键词集合规模。为了确定合适的 λ 取值，在实现较高的检索成功率的同时保证较短的安全索引建立时间，分别使用不同的 λ 建立安全索引。从数据集中抽取 1000 个文件，使用不同的 λ 进行实验，对每个文件的关键词集合降维后建立安全索引，并使用不同的关键词进行密文检索。图 5、图 6 分别示出了检索成功率和文件安全索引向量构建时间随 MinHash 系数 λ 的变化情况，其中检索成功率是指检索到的文件中包含查询关键词的概率。由图 5 可见，当 $\lambda < 0.6$ 时，即降维后的关键词规模较小时，检索成功率较低；当 $\lambda \geq 0.6$ 时，即降维后的关键词规模较大时，成功率较高。由图 6 可知，$\lambda > 0.9$ 时，安全索引向量构建时间明显增大。综合图 5 和图 6 的实验结果，应该在 0.6～0.9 间选择 λ 的值。本文选择 $\lambda = 0.8$ 进行后续实验。

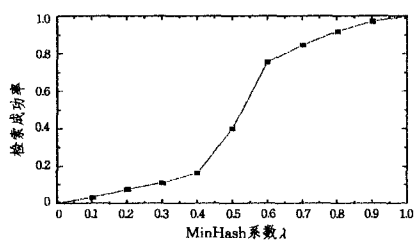

图 5　检索成功率随 MinHash 系数 λ 的变化情况

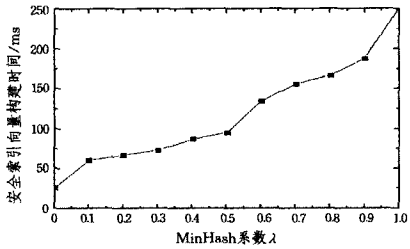

图 6　文件安全索引向量构建时间随 MinHash 系数 λ 的变化情况

（2）安全索引构建

从数据集中随机抽取文件，分别根据不同关键词规模及文件规模进行实验，并将本方案的安全索引构建时间与 Wang 等人方案[13]的结果进行对比分析。图 7 和图 8 分别示出了文件安全索引向量构建时间随关键词数及文件规模的变化情况。本方案文件安全索引向量构建主要包括关键词降维、关键词向量化、CBF 索引向量生成及索引向量加密；Wang 等人的方案[13]索引构建主要包括关键词向量化、BF 索引向量生成及索引向量加密。通过关键词降维可以使本方案去除一部分索引关键词，减少时间及空间代价，并且在索引构建过程中相较于 Wang 等人方案采用的欧氏距离，采用 Jaccard 距离计算相似度减少了计算时间。因此，对于任意的关键词规模及文件规模，本方案的文件安全索引向量构建时间都要明显优于 Wang 等人的方案。

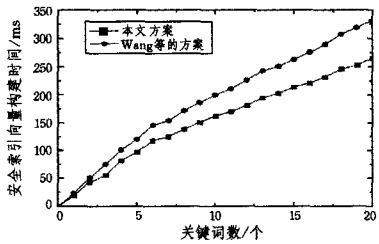

图 7　文件安全索引向量构建时间随关键词数的变化情况

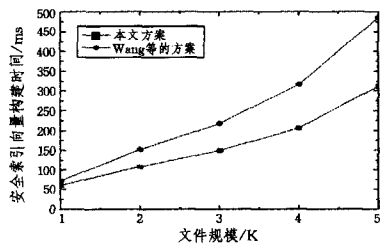

图 8　文件安全索引向量构建时间随文件规模的变化情况

（3）陷门函数生成

从数据集中随机抽取文件，由于陷门函数的生成只与查询关键词相关，因此根据不同关键词规模进行实验，并将本方案陷门函数生成时间与 Wang 等人方案[13]的结果进行对比分析。图 9 示出了陷门函数生成时间随关键词数的变化情况。本方案陷门函数生成主要包括查询关键词向量化、CBF 查询向量生成及查询向量加密，其中采用 Jaccard 距离计算相似度；Wang 等人的方案陷门函数生成主要包括查询关键词向量化、BF 查询向量生成及查询向量加密，采用欧氏距离计算相似度。因此，对于任意关键词规模，本方案的陷门函数生成时间都要明显优于 Wang 等人的方案。

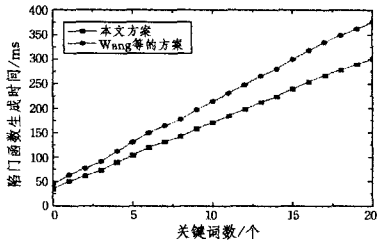

图 9 陷门函数生成时间随关键词数的变化情况

(4)查询结果返回

统计在查询返回结果集的文件中所具有的与查询关键词相关的关键词情况。图 10 示出了查询返回的前 10 个文件的检索精度,用文件与查询关键词的相关度表示检索精度,即文件中包含的相关查询关键词越多,检索精度越大。如图 10 所示,相比 Wang 等人的方案,本文方案中返回的文件与查询关键词的相关度更高,检索精度更大。

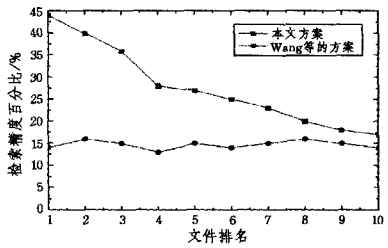

图 10 返回文件的检索精度

结束语 本文针对云环境中密文数据的检索需求,提出了一种适用于云中大规模密文数据的模糊多关键词检索方案。相比现有有关方案,该方案通过引入 MinHash 算法对关键词集合进行降维,并基于 Jaccard 距离计算向量相似度,有效减少了计算量和空间开销,提高了索引构建和查询效率,同时采用计数型布隆过滤器作为索引结构,使用其计数表示关键词权重,使得查询结果排序更加准确,并且有效保护了用户数据和检索的隐私性。

参 考 文 献

[1] ARMBRUST M,FOX A,GRIFFITH R,et al. Above the Clouds: A Berkeley View of Cloud Computing[J]. Communications of the ACM,2010,53(4):50-58.

[2] LI J,WANG Q,WANG C,et al. Fuzzy Keyword Search over Encrypted data in Cloud Computing [C]//Proceedings of IEEE INFOCOM,Mini-Conference. San Diego:IEEE Press,2010:441-445.

[3] XIANG F,LIU C Y,FANG B X,et al. Research on Ciphertext Search for the Cloud Environment[J]. Journal on Communications,2013,34(7):143-153. (in Chinese)
项菲,刘川意,方滨兴,等. 云计算环境下密文搜索算法的研究[J]. 通信学报,2013,34(7):143-153.

[4] SONG D,WAGNER D,PERRIG A. Practical Techniques for Searches on Encrypted data[C]//Proceedings of IEEE Symposium on Security and Privacy. Washington:IEEE Computer Society Press,2000:44-55.

[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] BONEH D,CRESCENZO GIOVANNI D,OSTROVSKY R,et al. Public Key Encryption with Keyword Search [C]//Proceedings of International Conference on Theory and Application of Cryptographic Techniques. Springer,Berlin,Heidelberg,2004:506-522.

[7] WANG C,CAO N,LI J,et al. Secure Ranked Keyword Search over Encrypted Cloud Data [C]//Proceedings of the IEEE 30th International Conference on Distributed Computing Systems. Genova:IEEE Computer Society,2010:253-262.

[8] CAO N,WANG C,LI M,et al. Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data [C]// IEEE Transactions on Parallel and Distributed Systems. Atlanta:IEEE Computer Society,2011:829-837.

[9] WONG W K,CHEUNG W L,KAO B,et al. Secure kNN Computation on Encrypted Databases [C]//Proceedings of the ACM SIGMOD International Conference on Management of Data. New York:ACM,2009:139-152.

[10] LI R X,XU Z Y,KANG W S,et al. Efficient Multi-keyword Ranked Query over Encrypted data in Cloud Computing [J]. Future Generation Computer Systems,2014,30(1):179-190.

[11] FENG G L,TAN L. Multi-attribute Ranked Keyword Search over Encrypted Cloud data [J]. Computer Science,2013,40(11):131-136. (in Chinese)
冯贵兰,谭良. 云环境中基于多属性排序的密文检索方案[J]. 计算机科学,2013,40(11):131-136.

[12] LIU C,ZHU L H,LI L Y J,et al. Fuzzy Keyword Search on Encrypted Cloud Storage Data with Small Index [C]//Proceedings of IEEE International Conference on Cloud Computing and Intelligence Systems. Beijing:Institute of Electrical and Electronics Engineers,2011:269-273.

[13] WANG B,YU S C,LOU W J,et al. Privacy-preserving Multi-keyword Fuzzy Search over Encrypted Data in the Cloud [C]// Proceedings of IEEE INFOCOM. Toronto:Institute of Electrical and Electronics Engineers,2014:2112-2120.

[14] BLOOM BURTON H. Space/time Trade-offs in hash Coding with Allowable Errors [J]. Communications of the Acm,2010,13(7):422-426.

[15] CHARIKAR M. Similarity Estimation Techniques from Rounding Algorithms [C]//Proceedings of the Thirty-fourth Annual ACM Symposium on Theory of Computing. New York:ACM,2002:380-388.

[16] FAN L,CAO P,ALMEIDA J,et al. Summary Cache:a Scalable Wide-area Web Cache Sharing Protocol [J]. IEEE/ACM Transactions on Networking,2000,8(3):281-293.

[17] INDYK P,MOTWANI R. Approximate Nearest Neighbors:towards Removing the Curse of Dimensionality [C]//Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing. New York:ACM,1998:604-613.

[18] SUN W H,LOU W J,HOU T,et al. Privacy-Preserving Keyword Search over Encrypted Data in Cloud Computing [C]//Secure Cloud Computing. Amsterdam:Springer,2014:189-212.

[19] Enron email dataset [EB/OL]. (2015-03-12) http://www.cs.cmu.edu/~./enron.