**Homomorphic Encryption Based Data Storage and Query Algorithm**

DI Chun-Yun1,2,3+, CAO Chen-Lei1,2,3, ZHANG Ru1,2,3, LIU Jian-Yi1,2,3

1(State Engineering Laboratory of Recovery Technology, Beijing University of Posts and Telecommunications,

Beijing 100876, China)

2(Key Laboratory of network and information attack & defense technology of MOE, Beijing University of Posts and Telecommunications,

Beijing 100876, China)

3(Key Laboratory of Trustworthy Distributed Computing and Service (BUPT), Ministry of Education, Beijing University of Posts and

Telecommunications, Beijing 100876, China)

**Abstract** To solve the problems of ciphertext storage and query on cloud, this paper proposes a ciphertext query algorithm according to the mutiplication homomorphic encryption [characteristic](http://cn.bing.com/dict/clientsearch?mkt=zh-cn&setLang=match&form=BDVEHC&q=%E7%89%B9%E6%80%A7) of RSA. Moreover, we have sorted and displayed the query result of the ciphertext efficiently by using the calculation method of the ciphertext similarity ,which is the combination of  vector cosine inclined angle,vector mapping length and the weight between words. Finally, the ciphertext query time is investigated in a lab environment. The ciphertext query time is considered as acceptable to users.

**Keywords** ciphertext query; homomorphic encryption; ciphertext index vector; ciphertext query vector; similarity

**1 Introduction**

With the rapid development of the Internet as well as the arrival of data-intensive society, cloud computing and related technologies attract increasingly attention. Cloud computing combines the resources on the Internet as a whole, providing users with powerful computing resources and adequate storage space. While users enjoy the services provided by the cloud, they don’t care the process about data computing, storage and management on the cloud. Large-scale data is migrated from the client to the cloud storage servers, which greatly reducing the user’s burden about data management .

However,once the data stroed on the cloud server, the user won’t resist the inside attack from the cloud system. To protect data security, users store the encrypted data on the cloud. While storing the ciphertext improved security, but it also complicates the process of data query quickly.

To solve the above problems, this paper proposes a ciphertext query algorithm according to the mutiplication homomorphic encryption [characteristic](http://cn.bing.com/dict/clientsearch?mkt=zh-cn&setLang=match&form=BDVEHC&q=%E7%89%B9%E6%80%A7) of RSA. On this basis, we sort and display the query result of the ciph-ertext efficiently by using the calculation method of the ciphertext similarity,which is the combination of  vector cosine inclined angle,vector mapping length and the weight between words.

**2 Related Work**

In the ciphertext retrieving field, the main solutions as follows: linear search algorithm, security index method, key-based public key encryption and homomorphic encryption technology.

In terms of linear search, Song *et al.*[1] proposed the linear retrieving algorithm, encrypting every word in the document.This algorithm has excellent features as follows: simple, quick, flexible, extensible and better resistance to statistic analysis.

在安全索引方面，Goh 等人在其论文Secure Indexs[2]中提出了安全索引方法。安全索引是一种数据结构，它允许当索引中包含某一单词时，以O(1)的时间复杂度，采用“门限”对该单词进行查询，是一种较为安全的加密索引及密文检索方法。随后，Curtmola等人在[3]中提出了一种自适应安全定义的加密搜索方法，解决了方法[2]面临的攻击者根据查询结果产生查询操作这一问题。

在基于关键字的公钥加密搜索方面，Boneh 等人在2007年又提出了一种基于关键字的公钥加密搜索算法（PEKS）[4]，实现邮件服务器在不知道秘密信息的前提下即可向用户返回所有包含查询关键字的信息。这种方法采用公钥加密算法对数据进行加密，可以实现多人加密一人检索，增强了安全性。同时，公钥加密算法的采用也解决了用户和服务器之间的密钥分发问题。但是PEKS有加解密效率较低、运算复杂、检索效率较低、消耗客户端大量的CPU和内存、不适用于多个关键字的搜索等问题。Qin Liuy等人在[5]和[6]中分别提出了一种在云计算中的有效的隐私保护关键字搜索方法（EPPKS）和云存储的安全的隐私保护关键字搜索方法（SPKS），进一步解决PEKS方法中的问题。

而同态加密算法近些年得到了极大的关注。秘密同态也叫同态加密，其思想最早由 Rivest 等人于1978年提出[7]，算法的同态性可以保证对加密数据进行运算又不会泄露数据信息。随后，Sander和 Tschudin在[8]中提出了整数环上的加法、乘法同态加密机制。直到2009年，Gentry基于理想格提出了第一个全同态加密方案[9]，并在文[10]中对全同态加密做了详尽的研究。现实中有很多加密算法可满足同态加密算法的加法或乘法特性[11]，RSA和ElGamal密码体制满足乘法同态，椭圆曲线密码体制本身即不满足乘法特性也不满足加法特性，但是对明文嵌入形式的椭圆曲线加密算法满足加法同态特性，Paillier密码体制具有加法同态性。

本文结合同态加密技术的原理和RSA公钥加密算法的乘法同态特性，提出了一种基于同态加密的密文数据查询方法。该方法只需根据查询条件中的密文查询项和待匹配元素个数，对密文索引元素进行相乘运算并将其结果与密文查询项进行比对处理，即可返回查询结果。整个查询过程只需对密文索引元素进行计算处理，无需解密，减少了解密开销，提高了计算性能，同时又不泄露数据信息，加强了算法的抗攻击性。针对查询结果的排序问题，本文充分考虑向量余弦夹角、向量映射长度及词项权值对数据排序的重要性，提出了一种相似度计算方法，实现对查询结果的有效排序。

**3 系统模型与算法设计**

**3.1数据存储与查询模型**

本文使用的密文数据存储与查询方法的整体模型如图1所示，由数据所有者Owner、数据使用者User和云存储服务器Cloud Storage Service三个实体组成。



图1 数据存储与查询模型

存储数据时，数据所有者对原始明文数据进行相应处理，生成密文数据和密文索引向量，并将密文数据和密文索引向量发送至云服务器进行存储。查询数据时，数据使用者对原始查询语句进行处理，生成密文查询向量，并将查询向量和其他相关信息发送至云服务器端进行密文查询。云服务器端则根据相关查询条件对密文数据进行计算和处理，向数据使用者返回密文查询结果。数据使用者的客户端得到密文查询结果后，将解密密文数据，并按相应的顺序将明文查询结果显示给数据使用者。

**3.2密文存储方法**

为了定义本文提出的算法设计内容，在此首先给出如下定义：

**定义1.** 索引向量。表示某一原始明文文件的密文索引向量：

，其中表示第k个密文索引元素，表示第k个密文索引元素的权值。

**定义2.** 索引元素频率。表示索引向量中索引元素k的频率[12]：，其中是索引元素k在文档中的出现次数。

**定义3.** 倒置索引频率。反映了文档数据集中索引元素的重要性。其典型形式如下[12]：，其中是索引元素k的倒置文档频率，N是文档数据集中文档的个数，是索引元素k出现过的文档个数。

**定义4.** 权值。根据典型的文档词项权值的形式[12]，可知索引元素权值的形式如下：

，其中索引元素频率加1是为了保证频率为1的索引元素具有非零权值。

**定义5.** 查询向量Q。

，其中是第j个查询元素。对于查询向量而言其每一个查询元素具有相同的重要性。

**方法1.** 密文数据存储方法。

**步骤1.** 生成密文索引向量：

数据所有者对原始文件进行分词处理生成明文关键词，同时计算该明文关键词在当前文件中的权值。采用RSA公钥加密算法对每个明文关键词进行加密生成密文关键词，其加密密钥为。生成文件的密文索引向量：

。

**步骤2.** 数据加密：数据所有者用对称加密算法加密原始文件生成密文数据。

**步骤3.** 关联密文查询关键字和密文：建立密文与密文索引向量之间的关系，以便于数据查找；

**步骤4.** 数据存储：数据所有者将密文数据和密文索引向量发送至云服务器端进行存储。

**3.3密文数据查询方法**

密文数据的查询和匹配方法采用RSA公钥加密算法的乘法同态特性进行处理，设RSA算法的公钥为。

对两个明文数据和进行加密，得密文和，表示加密。





密文和满足如下操作：，即，可知RSA公钥加密算法具有同态乘法特性。

**算法1.** 密文数据查询算法.数据使用者查询存储在云服务器上的密文数据，其具体步骤如下：

**步骤1.** 首先对原始查询语句进行处理，提取查询关键字，生成查询向量。设查询向量有三个查询元素，即。

**步骤2.** 数据使用者用和数据所有者相同的公钥(e，n)对查询元素进行相应处理得到密文查询项。包含三个查询元素的密文查询项为：

，

包含两个查询元素的密文查询项为：

，

其中。只包含一个查询元素的密文查询项为：

，

其中。将密文查询项和查询元素个数k发送至云服务器，供云服务器处理。

**步骤3.** 云服务器端根据数据使用者发送的数据进行查询匹配。其具体步骤如下：（1）根据查询元素个数k，从密文索引向量中任取k个密文索引元素，进行相乘运算得到密文索引元素的乘积，

，

其中分别表示第x，y，z个密文索引元素。以k=3为例，则，其中x，y，z互不相等。

（2）对密文查询项和进行比较运算得flag。

（3）若密文查询项和密文索引元素的乘积相等，则flag=1，返回密文索引元素相对应的权值信息及其他相关信息，供计算相似度和返回对应密文数据。否则flag=0，密文查询项和密文索引元素的乘积不匹配，不返回数据。

同理可查与查询向量中的任意k个元素相匹配的密文数据及与查询元素相匹配的k个索引元素的权值。

**3.4查询结果排序方法**

本文通过计算密文索引向量和查询向量之间的相似度，对查询结果进行有效排序。

相似度(Similarity) [13]，是指两个文档内容相关程度的大小。目前已经有很多相似度计算方法，其中常用的方法有计算文档向量间的内积或夹角的余弦，两个文档向量之间的夹角越小说明相似度越高。

本文充分考虑向量余弦夹角、向量映射长度及词项权值对数据排序的重要性，提出了一种相似度计算方法，实现对查询结果的有效排序。为详细说明相似度算法的实现过程给出如下定义。

**定义6.** 向量夹角余弦.指查询向量和索引向量形成夹角的余弦值，设查询向量共有t个查询元素，某一文档D的索引向量有n个索引元素，查询向量Q有k个查询元素与中的索引元素匹配，则对Q进行归一化处理后得到的空间坐标值为；对于向量而言，首先与查询向量Q的查询元素相匹配的索引元素位置应置为1，否则为0；其次，对其进行归一化处理可得如下形式的空间坐标值：



则向量与Q的余弦值可表示为如下形式：

(1)

可知，查询向量中的查询元素被命中越多，其余弦值就越大，查询向量与索引向量就越倾向于相近。

**定义7.** 查询总权值。若查询向量Q有k个查询元素与中的索引元素匹配，则查询总权值即为索引向量中与查询向量Q匹配的各索引元素的权值之和，其表示形式如下：。其中，是与查询向量Q的第j个查询元素相匹配的索引元素的权值。

**例1.** 若查询向量的三个查询元素在、两个密文索引向量中相匹配的索引元素的权值分别为(0.5, 0.8, 0.3)和(0.9, 0.4, 0.2)，则查询总权值：

= 0.5+0.8+0.3=1.6，

= 0.9+0.4+0.2=1.5。

在相同情况下查询总权值越大，查询向量与索引向量之间的匹配度越高。

**定义8.** 匹配长度。即索引向量在查询向量上的映射长度，如下图所示。



图2 匹配长度

可表示为如下形式：



其中，即为两个向量的夹角的余弦值；返回一个集合的大小；是索引向量的大小；是查询向量Q的大小。

**定义9.** 匹配间距。当匹配长度大于查询向量大小时，匹配间距为与之差；当小于时，匹配间距为与之差加最大索引向量的大小。当的情况下，在与的差值上加是因为大于时，Q更易于在其对应的索引向量中找到相匹配的索引元素。匹配间距的表示形式如下所示：

(2)

匹配间距越小，索引向量与查询向量就越接近，其差异性就越小，与查询元素相匹配的索引元素在这个索引向量中的重要性就越大，在其他条件都相等的情况下查询向量与索引向量之间的匹配度越高。

**算法2.** 查询向量与第个索引向量之间的相似度计算方法，可表示为如下形式：

 (3)

在相似度计算方法中采用是因为：（1）可放大向量夹角在相似度计算中的决定性作用；（2）加大不同索引向量的相似度之间的差值；相似度之间的差值越大，返回查询结果时更易于排序。

匹配间距越小，索引向量与查询向量就越接近，对相似度具有反相关的作用，计算相似度时采用的倒数。在相似度计算方法中采用的倒数是因为：（1）保证时计算仍有意义；（2）避免0时，的倒数趋于无限大。另外，采用是因为“1”是算法的原本系数，表明在原本数据基础上产生的作用，既可以反应对相似度的反作用，又可以避免当值很大时导致值极小的情况出现。

下面举例说明相似度计算方法。

**例2.** 设被查询项索引的三个文档、和的索引向量分别为：，

，

，

其中与的相匹配，与的匹配，与的相匹配。则根据已给数据和公式(1)至(3)可得如表1所示的数据。其中，第一列表示与查询向量相匹配的索引向量的下标。二至五列分别表示与相对应的余弦值、匹配间距、查询总权值和相似度值。

表1 相似度计算表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| **1** | 0.816 | 0.266 | 1.3 | 1.551 |
| **2** | 1 | 2 | 1.5 | 2 |
| **3** | 0.816 | 5.551 | 1.4 | 1.076 |

由上表可知查询项与索引向量的相似度最高，返回查询结果时可排在第一；与索引向量的相似度最小，返回查询结果时排在最后。

至此，本文提出的方法1、算法1和算法2描述了如何建立密文索引向量，如何利用RSA的同态乘法特性进行密文查询和如何通过计算相似度的方法进行排序的全过程。

**4 系统实现与结果分析**

**4.1系统实现**

本文根据图1所示的系统模型及方法1、算法1和算法2中提出的实现步骤，设计并实现了密文数据的存储、查询与排序显示的测试系统。

本测试系统内，服务器端的硬件运行环境为：Intel(R) Core (TM) i7-3770 CPU@ 3.40GHz，8.00GB安装内存。服务器端的软件运行环境为：Windows 7, Web-logic 12C。开发语言为：Java，SQL。开发环境：Spring 3.1，Java JDK 1.7及部分第三方开源架构。

本系统在数据所有者的客户端实现了对敏感数据加密及密文索引向量的提取和生成功能，其中加密算法为AES-128，关键词加密算法采用RSA 1024位公钥加密。云服务器端接收到由数据使用者端发送的密文查询信息后，根据查询条件进行数据查询，返回相应的数据信息。从而模拟了面向云的密文数据存储与查询服务。

**4.2结果分析**

为了测试本文提出的密文查询算法的可行性和合理性，本文利用RSA公钥加密算法的乘法同态特性对密文查询项与密文索引向量进行比对，以测试密文数据的匹配查询时间。其中，加密密钥用RSA1024位公钥，待测试查询元素和明文索引元素可选取中文词组或英文词组。为了说明算法的性能，给出如下参数定义：

**定义10.** 密文查询时间. 从服务器端收到数据查询条件开始计时，到服务器端找到匹配数据的时间。

为了测试定义11中的指标，本文采用查询向量Q={"林肯","总统","宣言"}和分别包含查询向量中的三个元素的密文索引向量CIndex10、CIndex20、CIndex30、CIndex50 CIndex60、CIndex70、CIndex80、CIndex90和CIndex100，其分别包含10、20、30、50、60、70、80、90和100个索引元素。其具体步骤如3.3节中的算法1所示。测试所得的参数统计结果如表2所示。其中，第一行是匹配元素个数，第一列是索引元素个数，测试所得的查询时间的单位为毫秒（ms）。

表2 密文查询时间表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **（ms）** | **0** | **1** | **2** | **3** |
| **10** | 0 | 6 | 16 | 15 |
| **20** | 0 | 4 | 32 | 21 |
| **30** | 0 | 4 | 40 | 39 |
| **50** | 3 | 3 | 62 | 56 |
| **60** | 3 | 3 | 81 | 68 |
| **70** | 3 | 3 | 102 | 86 |
| **80** | 3 | 3 | 116 | 102 |
| **90** | 3 | 3 | 132 | 128 |
| **100** | 3 | 3 | 160 | 156 |

测试结果的图形表示如图3所示。其中，图3的横坐标表示密文索引元素的个数，纵坐标表示密文查询所消耗的时间。T3线是匹配查询向量中的三个元素所消耗的时间；T2线是匹配查询向量中的两个元素所消耗的时间；T1线是匹配查询向量中的一个元素所消耗的时间；T0表示与查询向量不匹配的情况下所消耗的时间。此外，T4是明文查询时间，即明文查询向量Q和密文索引向量相对应的明文索引向量进行查询匹配所消耗的时间。



图3 密文查询时间图

从表2和图3可知，当匹配元素大于1的情况下，索引元素的个数越多，系统查询数据所消耗的时间就越大。此外T2>T3是因为，在索引元素个数相同的情况下，索引元素的组合方式越多，查询时间就越大。从以上的测试数据可知，对三个查询元素，100个索引元素的查询最大时，基本满足用户需求。

匹配元素个数是0和1时，明文查询时间T4总体大于密文查询时间T0和T1。这是因为密文的查询匹配是直接对数值进行比对计算，而明文的查询操作则是对字符串的比对计算，需消耗的时间多。匹配元素个数是2和3时，密文查询时间T2和T3大于明文查询时间，是因为匹配元素数大于1时，需要对密文索引元素进行每两个和每三个元素相乘后再和密文查询项进行比对计算，这个过程需要消耗时间，故密文查询时间大于明文查询时间。

**5 总结**

本文设计的密文数据存储与查询算法由密文索引向量的建立、密文数据的存储、密文数据的查询和对密文查询结果的排序等主要三部分内容构成。数据所有者通过RSA公钥加密算法生成并存储相应的密文索引向量。云服务器收到数据使用者发来的密文数据查询条件后，根据RSA算法的同态乘法特性和查询条件中的密文查询项和待匹配元素个数，对密文索引元素进行相乘运算并将其结果与密文查询项进行比对处理，实现无需解密即可对密文数据的查询，减少了解密开销，提高了计算性能，同时又不泄露数据信息，加强了算法的抗攻击性。再通过计算查询向量与密文索引向量间的余弦夹角、映射长度及词项权值，得到相似度值，并按相似度值的大小实现对查询结果的排序。本文通过测试系统测定了密文查询时间，索引元素的个数越多，索引元素的组合方式越多，系统查询数据所消耗的时间就越多。测试结果表明，该算法安全性较高，查询时间在用户可接受范围，基本符合用户需求。

为逐步提高系统的可用性与安全性，本文将在今后的工作中从以下两方面进行改进：一、提高查询算法的执行效率和查询准确率；二、在数据存储过程中制定更为安全有效的访问控制策略，并结合文献[14]中给出的方法实现密文数据的实名存储。

**参考文献**

[1] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. of the IEEE Symposium on Security and Privacy’00, 2000, pp. 44–55.

[2] E. Goh, "Secure indexes," in Cryptology ePrint Archive, Report 2003/216, 2003.

[3] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," in Cryptology ePrint Archive, Report 2006/210, 2006.

[4] Boneh D, Crescenzo G, Ostrovsky R, Persiano G. "Public key encryption with keyword search," In: Cachin C, Camenisch J, eds.LNCS 3027. Heidelberg: Springer-Verlag, 2004. 506−522.

[5] Qin Liuy, Guojun Wang, and Jie Wuz, "An Efficient Privacy Preserving Keyword Search Scheme in Cloud Computing," In：Computational Science and Engineering, 2009. CSE '09. International Conference on (Volume:2 ), Aug. 2009.715 - 720.

[6] Qin Liuy, Guojun Wang, and Jie Wuz, "Secure and privacy preserving keyword searching for cloud storage services," appear in Journal of Network and Computer Applications, 9 March 2011.

[7] Rivest R L, Adleman L M, and Dertouzos M L, "On Data Banks and Privacy Homomorphisms," in: Foundations of Secure Computation, 1978, pp. 169–178.

[8] T. Sander and C. Tschudin, "Towards mobile cryptography," In Proceedings of the IEEE Symposium on Security and Privacy. Oakland,CA,1998. IEEE Computer Society Press.

[9] Gentry C, "Fully homomorphic encryption using ideal lattices[C]"// Proceedings of 41st ACM Symposium on Theory of Computing (STOC 09), ACM press, New York, 2009: 169-178.

[10] Gentry C, "A fully homomorphic encryption scheme [D/OL]," Stanford University, 2009. http：//crypto.stanford.edu/craig.

[11] 闫世斗，刘念，李子臣.公钥密码体制的同态性分析.北京电子科技学院学报.2012年6月.

[12] W.Bruce Croft, Donald Metzler, Trevor Strohman著，刘挺、秦兵、张宇、车万翔译，《搜索引擎：信息检索实践》（美），机械工业出版社.

[13] 刘斌，陈桦，向量空间模型信息检索技术讨论，情报杂志.2006年第7期.

[14] Chenlei Cao, Ru Zhang, Mengyi Zhang, Yixian Yang, "IBC-based entity authentication protocols for federated cloud systems.(identity-based cryptography)(Report),"KSII Transactions on Internet and Information Systems,May 1, 2013.