Impact Factor Value: 4.029 ISSN: 2349-7084
IJCERT International Journal of Computer Engineering In Research Trends
Volume 4, Issue 8, August -2017, pp. 356-360 www.ijcert.org

Secure Multi Keyword Dynamic Search Scheme Supporting Dynamic Update.

[1]Vadla Jhansi Rani, [2]K.Samson Paul

[1]*M.Tech(CSE),Dr.K.V.Subba Reddy Institute of Technology.Kurnool,Andhra Pradesh*

[2]*Assistant Professor, Department of CSE, Dr.K.V.Subba Reddy Institute of Technology.Kurnool,Andhra Pradesh*

Abstract: -Cloud computing is becoming predominant; data owners are motivated to delegate complex data managements to the commercial cloud for economic savings. Sensitive data is usually encrypted before being uploaded to the cloud, which unfortunately makes the frequently-used search function a challenging problem. In this paper, we present a new multi-keyword dynamic search scheme with result ranking to search encrypted data more secure and practical. In the scheme, we employ a powerful function-hiding inner product encryption to enhance the security by preventing the leakage of the search pattern. For the concern of efficiency, we adopt a tree-based index structure to facilitate the searching process and updating operations. A comprehensive security analysis is provided, and experiments over the real world data show that our scheme is efficient.

Keywords: secure search; ranked search; dynamic update; cloud computing.

----

1. Introduction

Cloud computing is a fascinating IT service which enables its customers to remotely store their data economically and manage them online anywhere and anytime [1]. Due to its unprecedented development, more and more sensitive data (e.g., financial reports, medical records, private photos) are centralized into the cloud servers. Since the users lose their control over the data, their privacy may be compromised by various factors, such as curious employees of the cloud service provider (CSP) or powerful attackers. To address this concern, sensitive data need to be encrypted before being outsourced to the cloud. However, although the data encryption can satisfy difficulty of utilizing data effectively, which will lower CSP's service quality? Among all the utilization forms, keyword search is a fundamental one who can return the files including certain keywords that the users feel interested in. It has frequently been used in our daily life (e.g., Google search engine) and enjoyed various mature techniques developed by the information retrieval (IR) community. Nevertheless, the encryption disables search techniques designed for the plain text data. The remote storage will be meaningless if the cloud data cannot be efficiently searched. Thus, exploring an efficient and secure search scheme over the encrypted cloud data is very necessary. The concept of searchable encryption has been brought up aiming at this issue [2-3]. Informally speaking, searchable encryption enables users to retrieve their interested files from the cloud server without the server knowing the content of records or the users' interests (i.e., what keywords the users submit). However, the problem of satisfying the efficiency requirements is still challenging as the cloud server usually has large-scale data storage. Particularly, for data storage with the enormous amount, the number of retrieved results for a certain query is correspondingly large. This fact can cause excessively long search time and great complexity of users' final processing, such as decryption of every retrieved item. Furthermore, the CSP may face thousands of updating requests (adding or deleting files) every day. All of the above issues will reduce the quality of users' search experience. There have been several searchable encryption schemes trying to alleviate this problem. They facilitate the search process by providing flexible functionalities. Some of the efforts have been put into a multi-keyword feature [4] which will help the users to express their interests more accurately. To further narrow down the searching scope, some researchers exploit the relevance between the queried keywords and the target files to achieve ranked document retrieval, i.e., secure ranked search filtered out and returned to the users with much less communication cost. The users' computation overhead will also be lessened as they do not have to decrypt every document with the queried keywords to choose most interested ones. Essentially speaking, it improves the efficiency by sacrificing a certain degree of security (exposure of the relevant information). Also, few existing schemes of secure ranked search consider a dynamic scenario where the updating of searchable encryptions, unfortunately, none of them has been extended to support ranked search. Finding a ranked search scheme on the encrypted data with suitability to the dynamic scenarios is still in high demand.

In this paper, we propose a secure search scheme over the large-scale encrypted data in the cloud, which supports multi-keyword ranked search and dynamic document updating. Particularly, we adopt the vector space model and the *TF X IDF* criterion to support multi-keyword queries and result ranking. To improve the efficiency of searching and file updating, we have designed a search index structure with an elaborate modification of the tree which is very I/O efficient More Importantly, to realize the ranked pursuit securely, we adopt the function hiding encryption vector space model and protect the query privacy. Our contributions are summarized as follows

1) We propose a secure ranked search scheme for large-scale cloud data, achieving multi-keyword search as well as the data updating with high efficiency

2) We leverage the function-private encryption over inner product in a novel way, combining with a fast tree-based search index. Rigorous security analysis shows that our scheme is secure under a weakened security definition for ranked search pattern, which few other related works can achieve.

2. Problem Formulation

2.1 System Model

As illustrated in Fig. 1, the system model of our scheme includes three entities: data user, and cloud server. The data owner has a document collection F to be outsourced to cloud server in the encrypted form C. To enable search capability for C, before outsourcing, the data owner first generates a secure, searchable index by keyword set W extracted from F. The safe index is transferred together with document collection C to the cloud server. We assume the access control between the data user and data owner is done successfully [13]. To search over C with t interested keywords, the user needs to ask for a corresponding trapdoor T from the data owner [14]. With T submitted by the user, the cloud server starts to search over index tree and return the required encrypted documents. To make the document

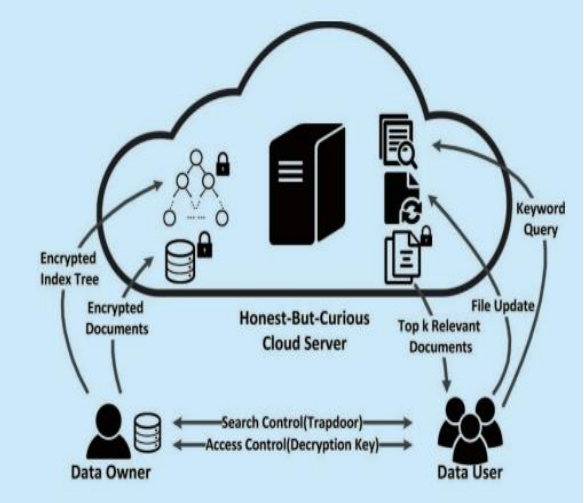

Fig.1 Architecture of dynamic ranked search over encrypted cloud data

retrieval more suitable to the user's demand, the cloud server is allowed to learn about the relevance between the queried keywords and documents. Thus the cloud server can rank the final results according to the relevance. To reduce to reduce the communication and computation cost, the user sends a parameter k along with the trap-door T to the cloud server and ask for k most relevant documents. W

When the document collection is too large, we can exploit the distributed computing feature of the cloud server. The collection will be divided into several sub-collections which are stored on different servers. The above search process can be deployed on each sub-collection independently, and the ranked results will be merged finally. The detail about how distributed servers collaborate is beyond the scope of this paper.

Also, the data owner may occasionally update the document collection. The owner generates update requirement (add or delete files) locally and sends it to the server. Upon receiving the update request, the cloud server updates the index and document collection C accordingly the cloud server is assumed to be "honest-but-curious," which means the cloud server will carry out the agreed protocol correctly. However, it always tries to infer the information about the encrypted document or user's queries by analyzing the stored data or the searching process. This assumption is consistent with the former related works on secure ranking search [5-6].

3. Construction of MKDRS

In this section, we describe the construction of our Secure Multi Keyword dynamic Search scheme Supporting Dynamic Update. (SMK-DSS) in details. We classify all the algorithms

In Definition 1 into three categories, namely initialization phase, search phase, and update phase.

In the initialization phase, the data owner makes necessary preparations for the outsourcing of its document collection F. It includes secret key generation, index extraction from F and encryption of index, corresponding to algorithms GenKey, TreeBuild, and Enc respectively. Then in the search phase, the data owner will generate secure search tokens from authorized users by algorithm SToken. On receipt of search tokens submitted by users, the server will honestly perform the search procedure following algorithm Search. In the update phase, the server deals with the update of the index when a document addition or deletion happens. The update consists of two Token to generate update token and Update to execute the updating. We describe the details of each algorithm in the following.Initialization Phase, Search Phase,Update Phase and acceleration strategy

3.1 Index Construction:

The building time of index tree is mainly affected by n as it costs time to build a tree. Fig. 2 shows that the building time is proportional to the number of documents.Xia's KBB scheme enjoy the same complexity of asymptotical analysis. However, as shown in Fig. 2, our keyword B+ tree with branch degree B=4 has less construction time than it. We can see that the branching degree of a tree plays a major role in influencing the construction time.KBB tree only has degree 2 while our tree's degree is 4 and can be further increased.The larger the degree is, the fewer nodes will be generated, which results in less construction time.

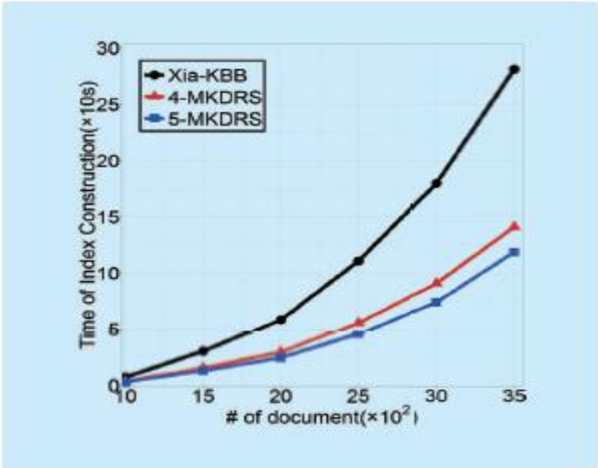

Fig 2. Index Construction Time

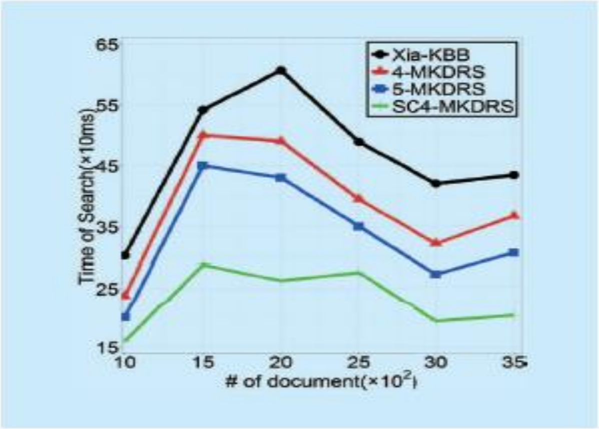

Fig.3 Search time for different sizes of document collection with k = 5

3.2 Search Process

The search time is not only affected by the size of document collection but also the number of retrieved documents, i.e. parameter k.The evaluation of search process will be implemented from the two aspects. With our design of index, for a top-1 search, as the height of a balanced tree is maintained as, we can achieve a search time. When retrieved number k is larger than 1, the earlier termination technique can also help to eliminate accesses for many nodes. As shown in figure 3.

3.3 Update Process:

When an insertion or deletion of document happens, we need to update nodes in the index tree. We take the insertion of a document as an example to demonstrate the update time in Fig. 4. It shows that the insertion time is dynamnearly logarithmic with the document number. Also, keyword tree performs better at update than KBB tree as fever nodes are encrypted and substituted.

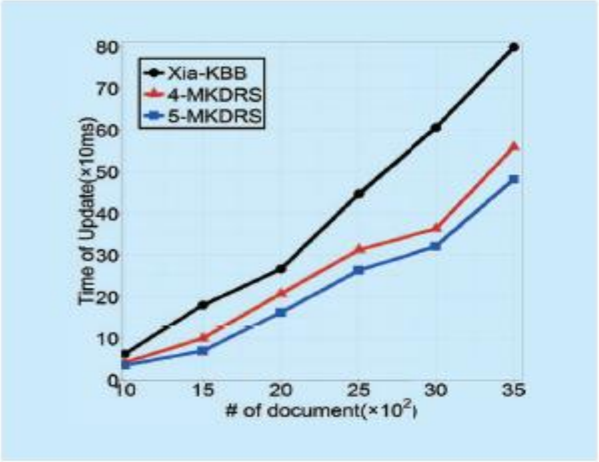

Figure 4. Update time for document collection

4. Performance Analysis:

We implement the proposed scheme using C++ on a Windows 7 operation system with Pairing-Based Library to simulate the cost of cryptographic operations. The document set we test on is the collection of Request for Comments (RFC). The keyword dictionary is extracted from document collection according to traditional criterion [12].

The experiments consist of the performance of index construction, search and update processes. Our scheme is tested respectively by setting the branching degree of keyword tree to 4 -MKDRS) and (5-MKDRS). We compare our scheme with the recent work of Xia at.al. [7] Which also achieves secure and dynamic ranked search? The main difference between their scheme and ours is that they employ a Keyword Balanced Binary (KBB) tree as index structure and a matrix multiplication for inner-product encryption which cannot be proved secure in a rigorous cryptographic analysis [30]. To guarantee the same security level in the evaluation, we set the underlying encryption methods to be the same.

5. Conclusion:

In this paper, we propose a new secure multi-keyword search scheme is supporting both result ranking and dynamic document updating. We construct a keyword tree as an index to enable efficient search and update operations our design of index also enjoys better I/O efficiency, which is more suitable for a large-scale cloud data scenario. Furthermore, the similarity-clustering technique can be implemented to boost the search process. We prove that our scheme rigorously is L-secure and search pattern is better protected. Experiments over real-world data demonstrate the proper performance of our scheme.

References:

[1]Jingbo Yan, Yuqing Zhang, Xuefeng Liu," Secure multi-keyword search supporting dynamic update and ranked retrieval," Communication Technology (ICT)., Volume: 13, Issue: 10.

[2] SONG D, WAGNER D, PERRIG A. Practical techniques for searches on encrypted data[C]// Proceedings of IEEE Symposium on Security and Privacy, 2000: 44-45.

[3] BOSCH C, HARTEL P, JONKER W, et al. A Survey of Provably Secure Searchable Encryption [J].ACM Computing Surveys, 2015, 47(2): 1-51.

[4] GOLLE P, STADDON J, and WATERS B. Secure Conjunctive Keyword Search over Encrypted Data[C]// Proceedings of Applied Cryptography and Network Security (ACNS), June 8-11, 2004:31-45.

[5] SUN Wenhai, WANG Bing, CAO Ning, et al. Verifiable Privacy-preserving Multi-keyword Text Search in the Cloud Supporting Similarity-based Ranking [J]. IEEE Transactions on Parallel and Distributed Systems, 2013, 25(11):71-82.

[6] CAO Ning, WANG Cong, LI Ming, 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.

[7] XIA Zhihua, WANG Xinhui, SUN Xingming, et al.A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data [J].IEEE Transactions on Parallel and Distributed Systems, 2016, 27(2): 340-352.

[8] KAMARA S, PAPAMANTHOU C, ROEDER T. Dynamic searchable symmetric encryption[C]//Proceedings of the 2012 ACM conference on Computer and communications security (CCS),2012: 965-976.

[9] CASH D, JAEGER J, JARECKI S. Dynamic Searchable Encryption in Very-Large Databases: Date Structures and Implementation[C]// Network & Distributed System Security Symposium (NDSS),February 23-26, 2014.

[10] KAMARA S, PAPAMANTHOU C. Parallel and dynamic, searchable symmetric encryption[C]//Proceedings of Financial Cryptography and Data Security (FC), April 1-5, 2013: 258-274.

[11] BISHOP A, JAIN A, KOWALCZYK L. Function-Hiding Inner Product Encryption[C]//Proceedings of Advances in Cryptology—ASIAic

[12] A.Raghavendra Praveen Kumar, K.Tarakesh,U.Veeresh," A Secure and Dynamic Multi Keyword Ranked Search Scheme over encrypted." International Journal of Computer Engineering In Research Trends., vol.2, no.12, pp. 1137-1141, 2015.

[13] Mr. M. VEERABRAHMA CHARY, Mrs.N.SUJATHA," A Novel Additive Multi-Keyword Search for Multiple Data Owners in Cloud Computing ." International Journal of Computer Engineering In Research Trends., vol.3, no.6, pp. 308-313, 2016.

[14] G.Lucy, D.Jaya Narayana Reddy, R.Sandeep Kumar," Enabling Fine-grained Multi-keyword Search Supporting Classified Sub-dictionaries over Encrypted Cloud Data." International Journal of Computer Engineering In Research Trends., vol.2, no.12, pp. 919-923, 2015.

[15] G.Dileep Kumar, A.Sreenivasa Rao," Privacy-Preserving Public Auditing using TPA for Secure Searchable Cloud Storage data." International Journal of Computer Engineering In Research Trends., vol.2, no.11, pp. 767-770, 2015.