Enhancing Security And Efficiency In Cloud Server Based On MRSE

[1]S.KALAIMATHI, [2] Mrs. K.BHUVANESHWARI M.Sc[SW]

[1]Pursuing M.E, Dept of CSE

[2]Assistant Professor, Department of Computer Science and Engineering

Abstract- Data owners are motivated to outsource their complex data management systems from local sites to the commercial public cloud for great flexibility and economic savings. But for protecting data privacy, sensitive data have to be encrypted before outsourcing, which obsoletes traditional data utilization based on plaintext keyword search. We present a secure and efficient multi-keyword ranked search scheme over encrypted data, Specifically, the vector space model and the broadly utilized TF×IDF model are joined as a part of the record development and question era, which meanwhile supports flexible dynamic operations. The vector space model helps to provide sufficient search accuracy, and the AES encryption enables users to involve in the ranking while the majority of computing work is done on the server side by operations only on cipher text. As a result, information leakage can be eliminated and data security is ensured. The user can decrypt the encrypted documents with the shared secret key. To further propose a "Greedy Search (GDFS)" algorithm for accurate ranking calculations.

Keywords—Multi-keyword ranked search, Searchable encryption, Dynamic updates, Cloud computing, Secret Key.

I. INTRODUCTION

Cloud computing is a conversational phrase used to express a variety of dissimilar types of computing ideas that occupy large number of computers that are connected through a real-time communication network i.e. Internet. In science, cloud computing is the capability to run a program on many linked computers at the same time. The fame of the term can be recognized to its use in advertising to sell hosted services in the sense of application service provisioning that run client server software on a remote location. Cloud computing relies on sharing of resources to attain consistency and financial system alike to a utility (like the electricity grid) over a network. The cloud also centers on maximize the effectiveness of the shared resources. Cloud resources are typically not only shared by multiple users but as well as dynamically re-allocated as per demand. This can perform for assigning resources to users in dissimilar time zones. For example, a cloud computing service which serves American users during American business timings with a specific application (e.g. email) while the same resources are getting reallocated and serve Indian users during Indian business timings with another application (e.g. web server).

Despite of the various advantages of cloud services, outsourcing sensitive information (such as e-mails, personal health records, company finance data, government documents, etc.) to remote servers brings privacy concerns. The cloud service providers (CSPs) that keep the data for users may access users' sensitive

information without authorization. So, it can lead security problem.

II. RELATED WORK

Due to different cryptography Primitives, searchable encryption schemes can be constructed using public key based cryptography. or symmetric key based cryptography. searchable encryption schemes enable the client to store the encrypted data to cloud and execute keyword search over encrypted domain. Proposed the first symmetric searchable encryption (SSE) scheme, and the search time of their scheme is linear to the size of the data collection. Goh [8] proposed formal security definitions for SSE and designed a scheme based on Bloom filter. The search time of Goh's scheme is O (n), where n is the cardinality of the document collection. proposed two schemes (SSE-1 and SSE-2) which achieve the optimal search time. Their SSE-1 scheme is secure against chosen-keyword attacks (CKA1) and SSE-2 is secure against adaptive chosen keyword attacks (CKA2). These early works are single keyword Boolean search schemes, which are very simple in terms of functionality. Afterward, abundant works have been proposed under different threat models to achieve various search functionality, such as single keyword search, similarity, multi-keyword Boolean search, ranked search, and multi keyword ranked search etc. Multi-keyword Boolean search allows the users to input multiple query keywords to request suitable documents.

Among these works, conjunctive keyword search schemes only return the documents that contain all of the query keywords. Disjunctive keyword search schemes return all of the documents that contain a subset of the query keywords. Predicate search schemes are proposed to support both conjunctive and disjunctive search. All these multi keyword search schemes retrieve search results based on the existence of keywords, which cannot provide acceptable result ranking functionality. Ranked search can enable quick search of the most relevant data. Sending back only the top-k most relevant documents can effectively decrease network traffic. Some early works have realized the ranked search using order-preserving techniques, but they are designed only for single keyword search.

The first privacy-preserving multi-keyword ranked search scheme, in which documents and queries are represented as vectors of dictionary size. With the "coordinate matching", the documents are ranked according to the number of matched query keywords. However, scheme does not consider the importance of the different keywords, and thus is not accurate enough. In addition, the search efficiency of the scheme is linear with the cardinality of document collection. In [29], Zhang et al. proposed a scheme to deal with secure multi-keyword ranked search in a multi-owner model. In this scheme, different data owners use different secret keys to encrypt their documents and keywords while authorized data users can query without knowing keys of these different data owners. The authors proposed an "Additive Order Preserving Function" to retrieve the most relevant search results. However, these works don't support dynamic operations.

III. PROBLEM STATEMENT

In the Existing system, a general approach to protect the data Confidentiality is to encrypt the data before outsourcing. Existing system using keyword based information retrieval technique. This keyword-based information retrieval, which are widely used on the plaintext data, cannot be directly applied on the encrypted data. Downloading all the data from the cloud and decrypt locally is obviously impractical. So overhead will be increased on both cloud server and data owner. And also

these works don't support the dynamic operations such that insertion and deletion of documents. There is no security measures added in this system even if the users has login credential.

Disadvantages:

- Huge cost in terms of data usability.
- Existing System methods not practical due to their high computational overhead for both the cloud sever and user.
- Existing system cannot support the dynamic operations like insertion an deletion of documents in encrypted cloud data.

IV. SYSTEM MODEL

A. PROPOSED SYSTEM:

In order to improve search efficiency, we propose a secure tree-based search scheme over the encrypted cloud data, which supports multi-keyword ranked search and dynamic operation.

Our contributions are summarized as follows:

1) We design a searchable encryption scheme that supports both the accurate multi-keyword ranked search and flexible dynamic operation on document collection.
2) Due to the special structure of our tree-based index, the search complexity of the proposed scheme is fundamentally kept to logarithmic. And in practice, the proposed scheme can achieve higher search efficiency by executing our "Greedy Depth-first Search" algorithm. Moreover, parallel search can be flexibly performed to further reduce the time cost of search process.

In this technique the following are the different things which we have to implement

1. Data Owner
2. Cloud Server
3. Data User

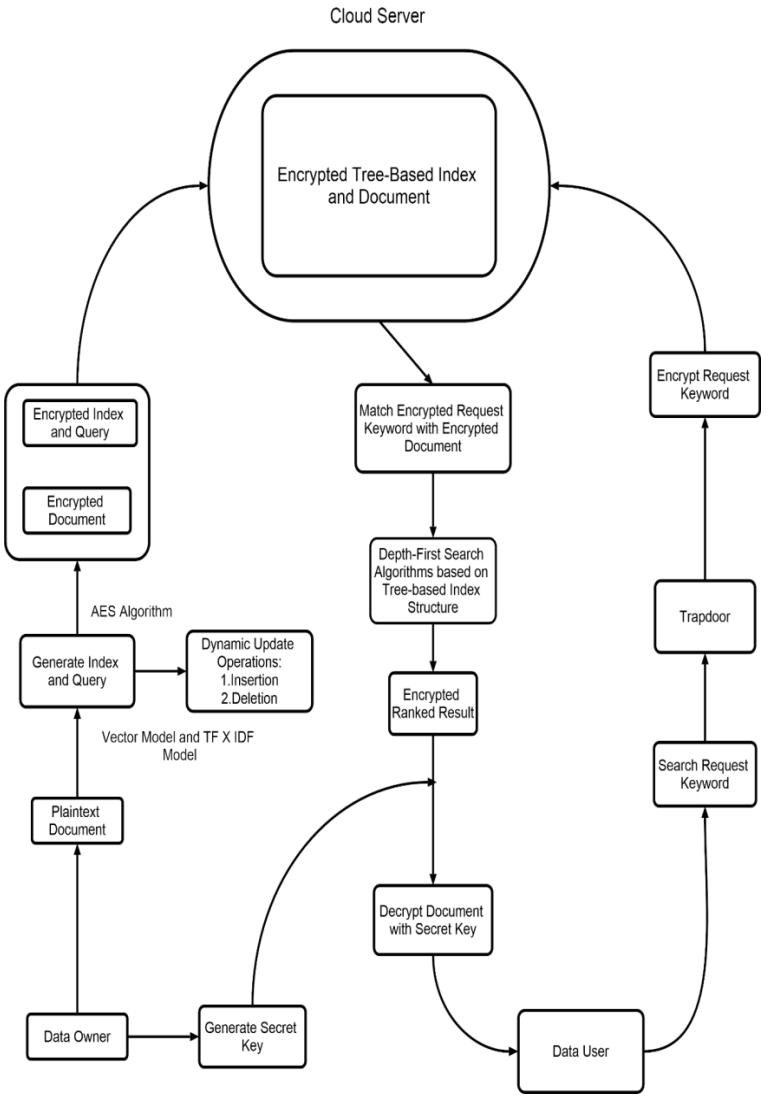

Fig.1. Architecture diagram of the MRSE Implementation.

Data owner - has a collection of documents that he wants to outsource to the cloud server in encrypted form while still keeping the capability to search on them for effective utilization. In our scheme, the data owner first builds a secure searchable tree index

I from document collection F and then generates an encrypted document collection C for F . Afterwards, the data owner outsources the encrypted collection C and the secure index to the cloud server, and securely distributes the key information of trapdoor generation (including keyword IDF values) and document decryption to the authorized data users.

Data users - are authorized ones to access the documents of data owner. With t query keywords, the authorized user can generate a trapdoor TD according to search control mechanisms to fetch k encrypted documents from cloud server. Then, the data user can decrypt the documents with the shared secret key.

Cloud server - stores the encrypted document collection C and the encrypted searchable tree index I for data owner. Upon receiving the trapdoor TD from the data user, the cloud server executes search over the index tree I, and finally returns the corresponding collection of top-k ranked encrypted documents. Besides, upon receiving the update information from the data owner, the server needs to update the index I and document collection C according to the received information.

B. VECTOR SPACE MODEL

The vector space model and the widely-used "term frequency (TF) X inverse document frequency (IDF)" model are combined in the index construction and query generation to deliver multi-keyword ranked search.

TF- the term frequency is the number of times a given term (keyword) appears within a document.

IDF-the inverse document frequency is obtained through dividing the cardinality of document collection by the number of documents containing the keyword.

In the vector space model, each document is denoted by a vector, whose elements are the normalized TF values of keywords in this document. Each query is also denoted as a vector Q, whose elements are the normalized IDF values of query keywords in the document collection. Naturally, the lengths of both the TF vector and the IDF vector are equal to the total number of keywords, and the dot product of the TF vector Du and the IDF vector Q can be calculated to quantify the relevance between the query and corresponding document.

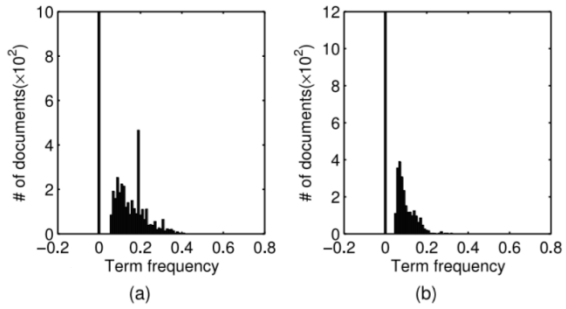

Fig.2. Distribution of term frequency (TF) for (a) keyword "subnet", and (b) keyword "host".

The relevance Score function is defined as:

$$\text{RScore}(D,Q) = \sum \text{TF X IDF}$$

RScore(D,Q) – The function to calculate the relevance score for query vector Q and index vector D stored in u.

C. ALGORITHMS AND TECHNIQUES

In this system, there are following algorithms can be used to improve search efficiency as well as in order to enhance the security on encrypted cloud data.

- Index Construction Algorithm
- Greedy Depth-first Search (GDFS) Algorithm
- AES algorithm

1.Index Construction Algorithm

Outstanding to the special structure of our tree-based index, the proposed search scheme can flexibly attain sub-linear search time and deal with the deletion and insertion of documents.

Input: the document collection F= $\{f_1, f_{2,\ldots\ldots} f_n\}$ with FID.

Step 1: Initialization:

For input document set F, which is imposed by the ordering of the identifiers id =(1,2, ... ,n), build a red-black tree T on top of id (1,2, ... ,n). The node's data structure is defined as the same as that in KRB tree: $\{D_u, id, v, z\}$.

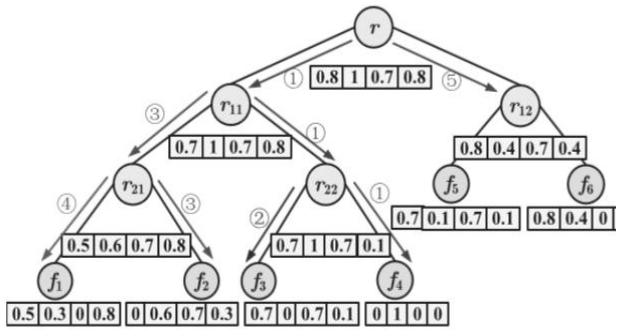

Fig.3. Tree-based index with document collection

Step 2: Add data to all nodes:

- If u is a leaf node, $D_u[i] = tf_{id,i}$ if and only if the document pointed by f_{id} contains keyword w_i ($tf_{id,i}$ is the normalized TF (D) value of keyword in document f_{id}), otherwise $D_u[i] = 0$.
  Node u = { ID,D,P_1,P_2,FID }
- Can get the biggest normalized TF value of keyword w_i to those documents in the subtree rooted by u , which can be utilized to calculate the maximum possible similarity score in these documents.

Output: plaintext index tree T.

2. Greedy Depth-first Search (GDFS) Algorithm

Input : The index tree
Output : RList
if (u is not a leaf node) then
if(RScore> kth Score) then
GreedyDFS(child node with higher similarity score)
GreedyDFS (child node with lower similarity score)
else return;
end if
else if(RScore>kth Score)
Delete the element with smallest relevance score from RList
Insert a new element (RScore,u.FID)and sort elements of RList,
end if
end procedure
RScore=∑ TF X IDF
Kth Score – The smallest relevance score in current RList.

We construct a result list denoted as RList, whose element is defined as (RScore, FID). Here, the RScore is the relevance score of the document FID to the query, which is calculated according to Formula. The RList stores the k accessed documents with the largest relevance scores to the query. The elements of the list are ranked in descending order according to the RScore, and will be updated timely during the search process.

3. AES Algorithm

The secure AES (Advanced Encryption Standard) algorithm is used to encrypt the index and query vectors, and temporarily ensure correct significance score control between encrypted index and query vectors. The search control mechanisms to fetch encrypted

documents from cloud server and produced to data user. Then, the data user can decrypt the documents with the shared secret key. Secret key will be send to user at time user want to decrypt the document.

V. CONCLUSION AND FUTURE WORK

In this scheme can propose an efficient Multi-Keyword ranked search scheme over encrypted cloud data, which supports dynamic update operations. The various Multi-keywords, it select the vector space model to present the relevance between documents and keywords. Evaluate the similarity between outsourced documents and query keywords, and also achieved accurate ranked search results. With respect to search efficiency and update operations, we design a tree-based index and propose an efficient Greedy Depth-first Search (GDFS) algorithm. Furthermore, in terms of secure, the implemented a AES algorithm and successfully satisfy the security requirements. With this approach, achieved safe and search effectiveness.

In this schemes will focus on increase the scalability and performance. In that, will try to improve the searchable encryption scheme. The future work would concentrate on using Elliptic Curve Cryptography (ECC) encryption technique for better performance. Further, we intend to analyze the behavior of our proposed system for multiuser environment.

REFERENCES

[1] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-Preserving Multi-Keyword Ranked Search overEncrypted Cloud Data," Proc. IEEE INFOCOM, pp. 829-837, Apr, 2011.

[2] L.M. Vaquero, L. Rodero-Merino, J.Caceres, and M.Lindner, "A Break in the Clouds: Towards a Cloud Definition," ACM SIGCOMM Comput. Commun. Rev.,vol. 39, no. 1, pp. 50-55, 2009.

[3] N. Cao, S. Yu, Z. Yang, W. Lou, and Y. Hou, "LT Codes-Based Secure and Reliable Cloud StorageService," Proc. IEEE INFOCOM, pp. 693-701, 2012.

[4] S. Kamara and K. Lauter, "Cryptographic Cloud Storage," Proc. 14th Int'l Conf. Financial Cryptograpy and Data Security, Jan. 2010.

[5] A. Singhal, "Modern Information Retrieval: A BriefOverview," IEEE Data Eng. Bull., vol. 24, no. 4, pp. 35-43, Mar. 2006.

[6] I.H. Witten, A. Moffat, and T.C. Bell, Managing Gigabytes: Compressing and Indexing Documents and Images. Morgan Kaufmann Publishing, May 2009.

[7] D. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," Proc. IEEE Symp. Security and Privacy, 2000.

[8] E.J. Goh, "Secure Indexes," Cryptology ePrint Archive, http:// eprint.iacr.org/2003/216. 2003.

[9] Y.C. Chang and M. Mitzenmacher, "Privacy Preserving Keyword Searches on Remote Encrypted Data," Proc. Third Int'l Conf.Applied Cryptography and Network Security, 2013.

[10] R. Curtmola, J.A. Garay, S. Kamara, and R. Ostrovsky, "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions," Proc. 13thACM Conf.Computer and Comm. Security (CCS '06), 2010.