INTERNATIONAL JOURNAL OF EMINENT ENGINEERING TECHNOLOGIES

Volume 5, Issue 1 JULY 2017

A SECURE AND DYNAMIC MULTI-KEYWORD RANKED SEARCH SCHEME OVER ENCRYPTED CLOUD DATA

[1] MOHAMMED KHALEEL AHMED, M.TECH, ASSOCIATE PROFESSOR. DEPT OF C S E, ,

[2] MD SABAIR, PG SCHOLAR IN C S E,

[3] DR G S S RAO, PROFESSOR & HOD, DEPARTMENT OF C S E

[1]ahmedkhaleelmohammed@gmail.com, [2] profgssrao@gmail.com, [3] sabair0509@gmail.com,

[1, 2, 3] NAWAB SHAH ALAM KHAN COLLEGE OF ENGINEERING AND TECHNOLOGY.

Abstract: Nowadays, more and more people are motivated to outsource their local data to public cloud servers for great convenience and reduced costs in data management. But in consideration of privacy issues, sensitive data should be encrypted before outsourcing, which obsoletes traditional data utilization like keyword-based document retrieval. In this paper, we present a secure and efficient multi-keyword ranked search scheme over encrypted data, which additionally supports dynamic update operations like deletion and insertion of documents. Specifically, we construct an index tree based on vector space model to provide multi-keyword search, which meanwhile supports flexible update operations. Besides, cosine similarity measure is utilized to support accurate ranking for search result. To improve search efficiency, we further propose a search algorithm based on "Greedy Depth-first Traverse Strategy". Moreover, to protect the search privacy, we propose a secure scheme to meet various privacy requirements in the known ciphertext threat model. Experiments on the real-word dataset show the effectiveness and efficiency of proposed scheme.

Keywords: Encrypted cloud data, Multi-keyword ranked search, Dynamic update.

I. INTRODUCTION

Recent years, cloud computing enjoys great reputation in data management due to its outstanding capability in computing, storage and various applications. Through cloud services, people could enjoy convenient, on-demand network access to a shared pool of configurable computing resources with great efficiency and minimal economic overhead [1]. Despite of the various advantages offered by cloud services, transfer of sensitive information (such as e-mails, company finance data, and government documents, etc) to semi-trusted cloud server brings concerns about privacy issues. For instance, the cloud server may leak information to unauthorized entities or even be hacked, which puts the outsourced data at risk. Traditionally, sensitive data should be encrypted by data owners before outsourcing, which, however, obsoletes traditional data utilization service like keyword-based information retrieval. To enable traditional utilization on encrypted data, searchable encryption techniques [2-14], especially those based on symmetric key cryptography [4-14], are proposed for efficient keyword search over encrypted data. By now, many symmetric searchable encryption (SSE) schemes have been developed as an attempt for enriching the search flexibility, like single-keyword search [4-10] and multi-keyword search [11-14]. To enhance the accuracy of search result, multi-keyword search is more suitable for real world than single-keyword search. Among those multi-keyword search works, many have realized the conjunctive keywordsearch, subnet search, or range search [11, 12], but they don't support accurate ranked search. In plaintext information retrieval (IR) community, there are many state-of-the-art technologies for multi-keyword ranked search, for instance, cosine measure in the vector space model [15].

To provide ranked search functionality on encrypted data, Cao et al. [13] proposed a privacy-preserving multi-keyword ranked search scheme. With "coordinate matching", search result is ranked according to the number of matched keywords, which is not accurate enough. And their search complexity is linear with the number of documents in dataset. Then, in [14], Sun et al. proposed a multi-keyword

search scheme that supports similarity-based ranking. They constructed a searchable index tree based on vector space model and adopted cosine measure together with "term frequency (TF) × inverse document frequency (IDF)" weight to provide accurate ranking. Finally, their search algorithm achieves better-than-linear search efficiency. However, most of the above schemes work for static data, and can not efficiently handle dynamic collections (i.e., document collections that must be updated). Based on the inverted index structure proposed in [8], Kamara et al., [16] constructed an encrypted inverted index that can handle dynamic data efficiently. In their index structure, same documents in different index entries are linked by pointers. Additionally, two arrays are constructed for tracing documents for a specific keyword and tracing keywords for a specific document, by which insertion or deletion of a document could be realized. But, their scheme is very complex and difficult to implement. Subsequently, as an improvement, Kamara et al., [17] introduced a new tree-based scheme that can simply handle dynamic data. In their scheme, every document has a corresponding index, and all these indexes are merged to a hierarchical index tree. Thus, update operations could be easily implemented due to the inherent feature of tree

structure. Besides, the search scheme can achieve sub-linear search time through parallel execution. However, this scheme is designed for single-keyword search and doesn't take the accurate result ranking into account.

In this paper, we propose a secure dynamic multi-keyword ranked search scheme over encrypted cloud data, which supports top- k retrieval and dynamic updates on dataset. Specifically, we adopt the vector space model to provide multi-keyword queries, and cosine measure together with TF×IDF weight is utilized to achieve accurate ranked results.

To improve the search efficiency, we construct a tree-based index structure and propose a top- k ranked search algorithm over this index which has logarithmic search time. Besides, benefiting from the index tree structure, update on documents is available in our scheme. The proposed dynamic multi-keyword ranked search scheme (DMRS) is secure under the known cipher text model. Our contributions are summarized as follows:

1. Our proposed search scheme achieves multi-keyword ranked search over encrypted data with high efficiency and search result accuracy.

2. We propose a secure DMRS scheme which meets privacy requirements in the known cipher text model.

3. Benefiting from tree-based index structure, our search scheme supports dynamic update operation (like deletion and insertion) on documents, which caters to real-world needs and is superior to most current static schemes.

The reminder of this paper is organized as follows. In SectionII, we give a brief introduction to the system model and threat model, our design goals, and the preliminary.
SectionIII describes the DMRS search scheme in detail, and simulation results and performance analysis are presented in the section that follows.

II. PROBLEM FORMULATION

A. The System and Threat Model

The system model in this paper involves three different entities: the data owner, the data user, and the cloud server, as illustrated in Figure 1. The data owner outsources his local data to the cloud server, including a collection of encrypted documents C generated from F , and an encrypted searchable index tree I . Through access control mechanism [18], which is a separate issue out of the scope of this paper, a user could be authorized to access the data of data owner. Then, with t query keywords, the authorized user would generate a corresponding trapdoor T for the search request through search control mechanisms. Upon receiving the trapdoor T , the cloud server executes similarity search over the index tree I and finally returns the corresponding set of ranked encrypted documents. Moreover, user could choose the number of retrieved documents through submitting parameter k , which means the cloud server could terminate searching process while the top- k documents have been retrieved. This would simultaneously reduce the communication overhead and satisfy data users' requirements. Finally, the authorized user decrypts these received documents. The cloud server in this paper is considered as "honest-but-curious". Specifically, the cloud server honestly and correctly executes instructions according to the designated protocol. Meanwhile, it is "curious" to infer and analyze received data (including index and message flows) in its storage, which helps it acquire additional information.

1. Known ciphertext threat model: In this model, the cloud server only knows theoutsourced data from the data owner (including encrypted document set C and the searchable index tree I), and the encrypted query (trapdoor)*T* submitted by authorized user

Fig.1. Architecture Dynamic Multi-keyword Ranked Search Scheme

B. Design Goals

To enable efficient, secure and dynamic multi-keyword ranked search over outsourced encrypted cloud data under the aforementioned models, our system design should simultaneously achieve the following design goals.

1. Dynamic Multi-Keyword Ranked Search: To design a search scheme over encrypteddata which provides not only effective multi-keyword query and accurate result ranking, but also dynamic update on document collections.

2. Search Efficiency: Our search scheme aims to achieve better practical search efficiencythan linear search [13] by exploring a tree-based index structure and an efficient search algorithm.

3. Privacy-preserving: To prevent the cloud server from learning additional informationfrom the dataset, the index tree, and the queries. The specific search privacy requirements are summarized as follows, *1) Index Confidentiality and Query Confidentiality*: the underlying plaintext information (including keywords in the index and query, keywords' TF values stored in the index, and IDF values of query keywords) should be protected from cloud server; *2) Trapdoor Unlinkability*: the cloud server should not be able to determine whether twoencrypted queries (trapdoors) are generated from the same search request; *3) KeywordPrivacy*: the cloud server could not identify the specific keyword in query, index or dataset.

C. Notations and Preliminaries

- F – The plaintext document collection, denoted as a set of n documents F = (f1, f2, ... , fn).
- C – The encrypted document collection for F, denoted as C = (c1, c2, cn).
- W – the dictionary, i.e., the keyword set consisting of m keywords, denoted as W = (w1 , w2 , ... , wm).
- I – the searchable encrypted index tree.
- T – the unencrypted form of index tree I , which is stored in data owner side.
- Du–the index vector stored in node u of index tree.
- Q –the query vector generated from search request.
- I u–the encrypted form of Du.
- T –the encrypted form of Q, which is always called the trapdoor for the search request.

1. **Vector space model and ranking function:** Vector space model [15] is widely used inplaintext information retrieval, which efficiently represents documents with multi-dimensional vectors. And in this paper, we use the cosine measure together with TF×IDF rule to provide accurate ranking. Here, term frequency (TF) is simply the number of times a given term or keyword appears within a document, and inverse document frequency (IDF) is obtained through dividing the number of documents in the whole collection by the number of documents containing the term. For document vector, each element represents the TF value of

corresponding keyword in this document, and for query vector, each element represents the IDF value of corresponding keyword in the dataset. To quantify the similarity of each document vector and the query vector, the deviation of angles (*i.e.*, cosine values) between the two vectors is calculated. There are various similarity evaluation functions for cosine measure, and we choose the one in [15]. Setd_i as the document vector of document f_i and$q$ asthe query vector, the similarity score (cosine value) of the document vector, the similarity score (cosine value) of the document and query is calculated as follows: documents that contain term t , N denotes the total number of documents in the collection,and d_i , q are the Euclid lengths of vectord_i and$q$, functioning as the normalization factors.Then the value of cosine measure can be denoted as the inner product of d_i and q 's unit vectors.

2. **Keyword red-black tree-based search algorithm:** The red-black tree [19] is a type ofself-balancing binary search tree which supports efficient search and update operations. Adapted from red-black tree, the keyword red-black (KRB) tree proposed in [17] is a dynamic data structure that can efficiently answer multi-keyword queries. The data structureof node u in KRB tree could be simply defined as $\{D_u, id, v, z\}$, where v is u 's left child and z is u 's right child. Note that only the leaf node has the real value of id that points to a specific document. D_u is the m -bit binary index vector of node u , and $D_u[i]$ represents keyword w_i ($i = 1, \ldots, m$). If u is a leaf node, $D_u[i] = 1$ if and only if the document pointed by $u \rightarrow id$ (i.e. f_{id}) contains keyword w_i , otherwise, $D_u[i] = 0$; if u is an internal node, $D_u[i] = D_v[i]$ *or* $D_z[i]$, i.e. $D_u$is computed by the bitwise Boolean OR operation of children'index vectors. Then, we can say that if $D_u[i] = 1$ for internal node u , there is at least one pathfrom u to some leaf storing identifier id that f_{id} contains w_i . In this paper, we modify the KRB tree and name it MKRB (Modified KRB) tree. The detailed construction procedure will be illustrated in Section 3, which is denoted as *buildIndexT ree*(F)

III. DMRS SCHEME

A. Tree-based Index Construction

In this part, we present a detailed description of our index tree construction. The procedure is presented in Table 1, and an example of our index tree is shown in Figure 2. Note that the index tree T built here is plaintext.

Table 1. The Process of Index Tree Construction

buildIndexT ree(F

)1. Initialization:

For input document set $F=(f_1, f_2, \ldots, f_n)$, which is imposed by the ordering of the identifiers $id=(1,2,\ldots,n)$, build a red-black tree T on top of $id(1,2,\ldots,n)$. The node's data structure is defined as the same as that in KRB tree: $\{D_u, id, v, z\}$

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 $u \rightarrow id$ (i.e. f_{id}) contains keyword w_i ($tf_{id,i}$ is The normalized TF value of keyword i in document f_{id}), otherwise, $D_u[i] = 0$.
- If u is an internal node, with left child v and right child z , D_u is computed recursively as follows:

$$D_u[i] = \max(D_v[i], D_z[i]), i = (1, 2, \ldots, m) \quad (2)$$

Specifically, $D_u[i]$ stores the biggest normalized TF value of w_i among its child nodes. If $D_u[i] \neq 0$, then there is at least one path from u to some leaf that stores the identifier id , such that f_{id} contains w_i . Meanwhile, we 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. For example, through node r_{11} in Figure 2, we know that the biggest normalized TF value of the second keyword among $\{f_1, f_2, f_3, f_4\}$ is 1.

3. Output plaintext index tree T .

B. Search Algorithm

The search process of our DMRS scheme starts from theroot node with a recursive procedure upon the tree in a special depth-first manner, which is called as "Greedy Depth-first Traverse Strategy". Specifically, if the node's similarity score is less than or equal to the minimum similarity score of the currently Selected top- k documents, search process returns to the parent node, otherwise, it goes down to examine the child node. The similarity score of each node u is calculated as Formula (1), i.e., the inner product of query vector Q and data vector Du . This procedure is executed recursively until the objects with top- k scores are selected. The search can be done very efficiently, since only part of the index tree is visited due to the relatively accurate maximum score prediction.

Algorithm 1 shows the process of our proposed search scheme. The following notations are used in the pseudo code for our tree-based search algorithm:

SimScore(Du ,Q) – function of similarity score calculation for unit query vectorQ and unit index vector Du , which is equal toCos (Du ,Q) .

• T op -k _List –the list of the top- k documents'identifies
which are arranged in descendingorder according to
 documents' similarity scores.
• kth _score –the current lowest score of top- k documents
inT op -k _List , and is set as 0

when the size ofT op -k _List is less than k .

• hchild –child node with higher similarity score.

• lchild –child node with lower similarity score.

Algorithm 1 Proposed Search Algorithm on index treeI

```
procedure GreedyDFS( IndexTreeNodeu) if
(uis not a leaf node) then
    if (SimScore(D_u,Q) >k^th_score) then
        GreedyDFS(u ® hchild) ;
        GreedyDFS(u ® lchild) ;
    else return;
    end if
else Update(u);
end if
end procedure

procedure Update( IndexTreeNodel)
if (the size of Top -k _Listis less thank) then
    add l ® id toTop -k _List ;
else
    if (SimScore(D_l,Q) >k^th_score) then
        delete the identifier with the score as k^th_score fromTop -k _List and
        insertl ® id ;
        refresh k^th_score ;
    else return;
    end if
end if
end procedure
```

example of our search scheme. The arrows with numbers indicate the search process for top-3 documents while query Q = (0, 0.92, 0, 0.38) (note that the index vectors of leaf nodes have been normalized): the arrows with

 indicate the first search round and finally obtain document f_4 with similarity score 0.92; the arrows with

② indicate the search for second document and obtain document f_2 with score 0.67; and the arrow with

③ indicates the search for third document and finally retrieves document f_1 with score 0.58 as result. The cross means this path can not lead to reasonable results. For example, the path from r to r_{12} is rejected as the similarity score of r_{12} is less than the minimum similarity score of $\{f_4 , f_2 , f_1\}$, *i.e.*, 0.58.

IV. CONCLUSION

In this paper, we propose an efficient multi-keyword ranked search scheme over encrypted cloud data, which supports dynamic update operations. Among various multi-keyword semantics, we choose the popular one, *i.e.*, vector space model to present the relevance between documents and keywords. And cosine similarity measure is used to quantitatively evaluate the similarity between outsourced documents and query keywords, and furthermore achieve accurate ranked search results. With respect to search efficiency and update operations, we design a tree-based index and propose an efficient search algorithm. Moreover, in terms of privacy-preserving, we adopt a secure scheme in the known ciphertext threat model and successfully satisfy the privacy requirements. Eventually, experiments on the real- world dataset demonstrate the effectiveness and efficiency of our DMRS scheme. In the future, we will concentrate on designing more efficient search algorithm and secure scheme in enhanced threat model.

VI. REFERENCES

[1]K. Ren, C. Wang and Q. Wang, "Security challenges for the public cloud", Internet Computing, IEEE, vol. 16, no. 1, (2012), pp. 69-73.

[2]D. Boneh, E. Kushilevitz, R. Ostrovsky and W. E. Skeith III, "Public key encryption that allows PIR queries", Advances in Cryptology-CRYPTO 2007, Springer, (2007), pp. 50-67.

[3]D. Boneh, G. Di Crescenzo, R. Ostrovsky and G. Persiano, "Public key encryption with keyword search", Advances in Cryptology-Eurocrypt 2004, (2004), pp. 506-522.

[4]P. Van Liesdonk, S. Sedghi, J. Doumen, P. Hartel and W. Jonker, "Computationally efficient searchable symmetric encryption", Secure Data Management, Springer, (2010), pp. 87-100.

[5]M. Bellare, A. Boldyreva and A. O'Neill, "Deterministic and efficiently searchable encryption", Advances in Cryptology-CRYPTO 2007, Springer, (2007), pp. 535-552.

[6]D. X. Song, D. Wagner and A. Perrig, "Practical techniques for searches on encrypted data, Security and Privacy, 2000. S&P 2000", Proceedings. 2000 IEEE Symposium on, (2000), pp. 44-55.

[7]Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data", Applied Cryptography and Network Security, (2005), pp. 442-455.

[8]R. Curtmola, J. Garay, S. Kamara and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions", Proceedings of the 13th ACM conference on Computer and communications security, (2006), pp. 79-88.

[9]S. Zerr, D. Olmedilla, W. Nejdl and W. Siberski, "Zerber+Top-k retrieval from a confidential index", Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, (2009), pp. 439-449.

[10]W. Cong, C. Ning, R. Kui and L. Wenjing, "Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data", Parallel and Distributed Systems, IEEE Transactions, vol. 23, no. 8, (2012), pp. 1467-1479.

[11]D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data", In Theory of cryptography, Springer, (2007), pp. 535-554.

[12]J. Katz, A. Sahai and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products", Advances in Cryptology–EUROCRYPT 2008, Springer, (2008), pp. 146-162.

MR MOHAMMED KHALEEL AHMED M.Tech in computing science from Nawab shah alam khan college of engineering and technology,He is currently a Associate Professor in Nawab shah alam khan college of engineering and technology,Hyderabad.

MDSABAIR Currently Pursuing His M.Tech in Computer Science and Technology at the College Of Nawab Shah Alam Khan College Of Engineering and Technology, Hyderabad.