Secure Multi-Keyword Search Over Encrypted Outsourced Data

Anukrishna P. R[†] **, Dr Vince Paul**[††]

[†,††]Department of Computer Engineering, Sahrdaya college of engineering & technology, Kerala, India

Summary

In current era exabytes of data are getting generated in the world wide. As cloud computing provide us efficient storage most of the users are ready to outsource the data to the cloud. As a result encryption the data and uploading to cloud has become very common nowadays. We present method to search in encrypted data. This is attained by combining vector space model and widely used TF*IDF model. Index tree is generated for all documents and index is encrypted using KNN algorithm and utilizes greedy depth first search to provide efficient multi-keyword ranked search. Performance is analyzed to demonstrate the efficiency of this schema

Key Words:

Searchable encryption, multi-keyword ranked search, dynamic update, cloud computing

1. Introduction

Nowadays data is created constantly, and at an ever-increasing rate, all these and more create new data must be stored somewhere for some purpose. From 2010 amount of data is getting generated has become huge as everything and everyone is leaving a digital footprint. Cloud computing provides an attractive data storage and interactive pattern with advantages, including on-demand self-services, location individualistic resource pooling and omnipresent network access. These attracting features, both users and enterprises are inspired to outsource their data to the cloud, instead of getting software and hardware to manage the data. When an organization store hosts applications or data on cloud, it prevent the ability to have physical access to the servers hosting its information. However cloud service provider claims strong protection, security and privacy are the major issues in cloud computing. The best way for data confidentiality is encrypting the data before storing it in the cloud, which become a challenging task. In recent years many cryptographic algorithms is proposed for cipher text schemas. But existing keyword based information retrieval algorithms are used on pain text that cannot be applied on encrypted documents. Downloading all the data from the cloud and decrypting is practically impossible.

General purpose solutions to address above issues are fully-homomorphic encryption or oblivious RAMs. In fully homomorphic encryption is simple, given ciphertexts that encrypt π1,..., πt, it should allow anyone to output a ciphertext that encrypts f(π1,..., πt) for any specific function f, as long as that function can be efficiently computed. No information about π1,..., πt or f(π1,..., πt), or any intermediate plaintext values, should leak; the inputs, output and intermediate values are always encrypted. An Oblivious RAM (ORAM) simulator is a compiler that transforms algorithms in a way that the resulting algorithms preserve the input-output behaviour of the original algorithm. And the transformed algorithm's memory access pattern is independent of the memory access pattern of the original algorithm. These methods need massive operation and high computational cost for cloud server and user. Special purpose solutions to searchable encryption schema are more efficient in terms of functionality and security. When encrypted data is outsourced, searchable encryption schema allows the user to search on ciphertext domain. Many works have been proposed to attain this various search functionality such as single keyword search, similarity search, multi-keyword boolean search, ranked search, multi-keyword ranked search, etc. In current scenario muti-keyword search is more useful for its effectiveness. Dynamic muti-keyword rank search has proposed with automatic update and delete operation. Most of these schemas has not supported efficient muti-keyword search.

Our contributions include (1.) Efficient and secure muti-keyword rank search using tree based schema with automatic update and deletion operations. That helps to reduce the search complexity. (2.) To provide multi-keyword ranked search, vector space model with the commonly-used "term frequency * inverse document frequency"(TF*IDF) model are joined in the index construction and query generation. In order to obtain high search efficiency, we proposes a "Greedy depth-first Search (GDFS)" algorithm based on this index tree. (3.) KNN algorithm is utilized to encrypt Index and query vectors.

2. Related Work

Searchable encryption schemes empower the clients to store the encrypted data to the cloud and execute keyword search over cipher text domain. Searchable encryption schemes can be constructed either by using public key based cryptography or by symmetric key based cryptography.

Manuscript received November 5, 2016
Manuscript revised November 20, 2016

2.1 Single Keyword Searchable Encrytion

Song et al [2] first introduced the method of searchable encryption. In this each word in the document has encrypted independenly. This is a symmetric searchable encryption (SSE) scheme, in which scanning is from whole data collection word by word. As a result the search time of their scheme is linear and searching cost is high. Goh et al [3] defined a secure index structure and formulate a security model for index known as semantic security to avoid adaptive chosen keyword attack (ind-cka). They proposes z-idx for efficient ind-cka secure index construction using pseudo-random functions and bloom filters. The search time of this scheme is O(n), where n is the cardinality of the document collection.

Cash et al [4] recently design and implement an efficient method. Few method lacks ranking mechanism that makes the users to take a long time to select what they want when massive documents contain the query keyword. To attain rank mechanism order-preserving techniques are utilized. Wang et al [5] developed encrypted invert index to achieve secure ranked keyword search over the encrypted data. In the keyword search phase, the cloud server calculates the relevance score between documents and the query. In this approach, relevant documents are ranked according to their relevance score and outputs the op-k results. Boneh et al [6] designed searchable encryption in public key settings. In search phase anyone can use public key to write to the data stored on server but only authorized data users owning private key can search.

2.2 Multiple Keyword Searchable Encryption

Multi-keyword boolean search allows the users to input multiple query keywords to return suitable documents. This approach uses conjunctive keyword search or disjunctive keyword search. Conjunctive keyword search schemes return the documents that contain all of the query keywords. Disjunctive keyword search schemes return all the documents that contain a subset of the query keywords. Predicate search schemes support both conjunctive and disjunctive search. All these multi-keyword search schemes retrieve search results based on the existence of keywords, which cannot provide ranking functionality.

Cao et al. [7] introduces the first privacy-preserving multi-keyword ranked search scheme. It designed the documents and queries as vectors of dictionary size. The documents are ranked according to the number of matched query keywords based on coordinate matching. However, this scheme does not consider the priority of the different keywords, and thus is not accurate enough. Also the search efficiency of this scheme is linear with the cardinality of document collection. Sun et al. [8] proposed a secure multi-keyword search scheme that supports similarity-based ranking. To provide ranking, a searchable index tree based on vector space model and cosine measure together with TF*IDF are utilize to provide ranking results. Sun et al. search algorithm attains better-than-linear search efficiency but results in precision loss.

Orencik et al. [9] proposed a secure multi-keyword search method which constructed local sensitive hash (LSH) functions to cluster the similar documents. The LSH algorithm is suitable for similar search schema but cannot provide exact ranking. In [10], Zhang et al. designed 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 and authorized data users can query without knowing keys of these different data owners. The authors proposed an Additive Order Preserving Function to return the most relevant search results. Drawback of these works is it doesn’t support dynamic operations.

Practically, most of the data owner may need to update the document collection after uploading the collection to the cloud server. Thus, the secure encryption schemes are expected to support the insertion and deletion of the documents. There are many dynamic searchable encryption schemes. Song et al. [2] proposed a method which support update operation by considering each document as a sequence of fixed length words, and is individually indexed. However this work has low efficiency. Goh [3] proposed dynamic operations by updating of a Bloom filter along with the corresponding document. However, this scheme has linear search time and suffers from false positives. Kamara and Papamanthou [11] proposed a new search scheme to handle dynamic update on document data stored in leaf nodes based on tree-based index.In [12], Cash et al. presented a data structure T-set for keyword/identity. Then, a document can be represented by a sequence of independent T-Sets. Based on this structure, Cash et al. [13] proposed different dynamic searchable encryption scheme. In their construction, newly added tuples are stored and deleted tuples are recorded. The final search result is achieved through excluding tuples in the deleted list from the ones retrieved from original and newly added tuples.

3. The System and the Threat Models

Entities involved in the system model are data owner, data user and cloud server, as illustrated in fig1.

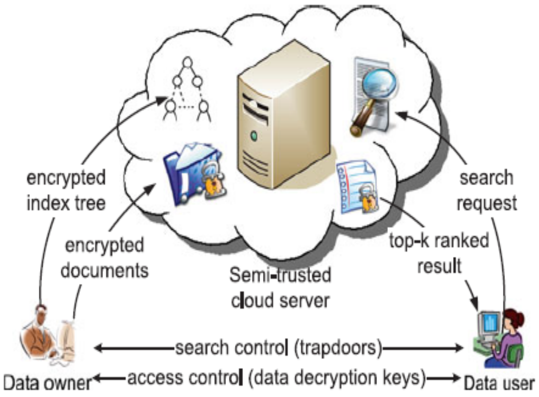

Fig -1 Architecture of ranked search over encrypted cloud data.

Data owner outsource the encrypted documents to the cloud server. In our scheme, the data owner initially creates a secure searchable tree index I from document collection F, and then generates an encrypted document collection C for F. Then data owner outsource the encrypted document collection C and tree index I to the cloud server and provides the key information of trapdoor generation and document decryption details to authorized data users. Data owner generates update information and send that to the server.

Data users are those have authorization to access the documents of data owner. Let t be the query keywords, the authorized user can generate a trapdoor TD according to search mechanisms to fetch k encrypted documents from cloud server. Data user can decrypt the documents with shared secret key.

Cloud server stores the encrypted document collection C and the encrypted searchable tree index I for data owner. After receiving the trapdoor TD from the data user, the cloud server searches the index tree I, and finally returns the corresponding collection of top-k ranked encrypted documents. If the data owner updates the document collection , according to the update information from the data owner, the server needs to update the index I and document collection C.

The cloud server in the proposed scheme is considered as “honest-but-curious”, which is employed in most of the works on cloud. Two threat model proposed by Cao et al. cloud server has adopted in this.

Known ciphertext model:- In this model, the cloud server only knows the encrypted document collectionC, the searchable index tree I, and the search trapdoor TD submitted by the authorized user. That is to say, the cloud server can execute ciphertext-only attack(COA) in this model.

Known background model:- In model is equipped with more knowledge such as term frequency statistics. Example for the distribution of term frequency is given below.

4. Design Goals

Index confidentiality and query confidentiality:- The paintext information, including keywords in the index and query, Term Frequency values of keywords stored in the index, and Inverse document frequency values of query keywords, should be protected from cloud server;

Trapdoor unlinkability:- The cloud server should not determine if two encrypted queries are generated from the same search request;

Keyword privacy:- The cloud server should not be able to identify the specific keyword in query, index or document collection by considering the statistical information like term frequency. Note that our proposed scheme is not designed to protect access pattern, i.e., the sequence of returned documents.

5. The Proposal Schemas

In this section unencrypted dynamic multi-keyword ranked search (UDMRS) scheme which is constructed on the basis of vector space model and KBB tree. Based on the UDMRS scheme, two secure search schemes are constructed against two threat models, respectively as proposed by Xia[1]

5.1 Index Construction of UDMRS Scheme

During index construction tree node is getting generated for all the documents these nodes are leaf of the index tree. Internal nodes of the tree are generating based on leaf node.

CurrentNodeSet – It is the set of currently processing nodes which have no parents.

TempNodeSet—The set of the recently generated nodes.

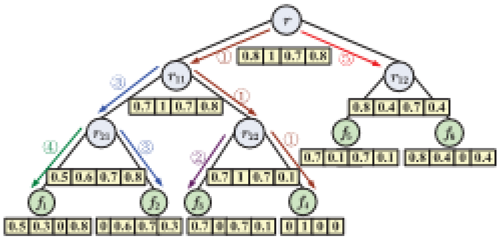

Fig. 3. An example of the tree-based index with the document collection $\mathcal{F} = \{f_i | i = 1, \ldots, 6\}$ and cardinality of the dictionary $m = 4$. In the construction process of the tree index, we first generate leaf nodes from the documents. Then, the internal tree nodes are generated based on the leaf nodes. This figure also shows an example of search process, in which the query vector Q is equal to $(0, 0.92, 0, 0.38)$. In this example, we set the parameter $k = 3$ with the meaning that three documents will be returned to the user. According to the search algorithm, the search starts with the root node, and reaches the first leaf node f_4 through r_{11} and r_{22}. The relevance score of f_4 to the query is 0.92. After that, the leaf nodes f_3 and f_2 are successively reached with the relevance scores 0.038 and 0.67. Next, the leaf node f_1 is reached with score 0.58 and replace f_3 in $RList$. Finally, the algorithm will try to search subtree rooted by r_{12}, and find that there are no reasonable results in this subtree because the relevance score of r_{12} is 0.52, which is smaller than the smallest relevance score in $RList$. [1]

5.2 Search Process of UDMRS Scheme

The search process is based on the “Greedy Depth-first Search” algorithm. We build a result list represented as RList, whose element is defined as <RScore,FID> , where Rscore is the relevant score for the document FID. Rscore is the dot product of encrypted form of Q and index vector stored in tree node u.

```
Algorithm
Step 1: if the node u is not a leaf node then
Step 2: if RScore(Du;Q) > kthscore then
Step 3: GDFS(u:hchild);
step 4: GDFS(u:lchild);
5: else
6: return
7: end if
8: else
9: if RScore(Du;Q) > kthscore then
10: Delete the element with the smallest relevance score from RList;
11: Insert a new element ⍰ RScore(Du;Q); u:FID ⍰ and sort all the elements of RList;
12: end if
  13: return
14: end if
```

6. Performance Analysis

Experimental analysis has been done to check the search precision on different privacy level. The search precision of scheme is influenced by the dummy keywords in EDMRS scheme. Precision’ is defined as Pk=k’/k, where k’ is the number of real top-k documents in the retrieved k documents. If a smaller standard deviation σ is set for the random variable Σεv, the secure scheme is supposed to obtain higher precision. The results are shown in Fig. 2(a). In the EDMRS scheme, phantom terms are added to the index vector to obscure the relevance score calculation; as a result the cloud server cannot identify keywords by analysing the TF distributions of special keywords. . The larger rank privacy denotes the higher security of the scheme, which is illustrated in Fig.2 (b).

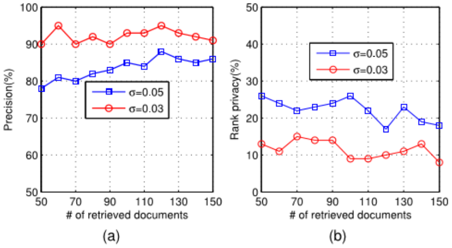

Fig -2 Precision

7. Conclusion

In recent years, many efforts have been directed towards the design of efficient mechanisms for searching over encrypted data. Many of these methods only supported single keyword search and others simply offered conjunctive or disjunctive searches for multi-keyword queries. From experimental analysis shown that proposed schema provide us an efficient and secure multi-keyword rank search.

References

[1] Zhihua Xia, Xinhui Wang, Xingming Sun, and Qian Wang, “A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data” , in IEEE Transcations , 2016

[2] D. X. D. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data", in Proc. S P 2000, Berkeley, CA , 2000.

[3] E.-J. Goh, Secure Indexes, IACR Cryptology ePrint Archive, IEEE TKDE , 2003.

[4] Cash, D., Jaeger, J., Jarecki, S., Jutla, C., Krawczyk, H., Rosu, M. C., and Steiner, M. “Dynamic searchable encryption in very large databases: Data structures and implementation”, In Proc. of NDSS , 2014.

[5] C. Wang, N. Cao, J. Li, K. Ren, and W. J. Lou, “Secure Ranked Keyword Search over Encrypted Cloud Data”, in Proc. ICDCS, Genova, Italy , 2010.

[6] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search”, in Proc. Eurocrypt, Interlaken, Switzerland , 2004.

[7] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, “Privacy-preserving multi-keyword ranked search over encrypted cloud data”, in Proc. IEEE Infocom, 2011.

[8] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li, “Privacy-preserving multikeyword text search in the cloud supporting similarity-based ranking”, in Proc. 8th ACM SIGSAC Symp. Inf., Comput. Commun. secur., 2013.

[9] C. Orencik, M. Kantarcioglu, and E. Savas, “A practical and secure multi-keyword search method over encrypted cloud data”, in Proc. IEEE 6th Int. Conf. Cloud Comput., 2013.

[10] W. Zhang, S. Xiao, Y. Lin, T. Zhou, and S. Zhou, “Secure ranked multi-keyword search for multiple data owners in cloud computing”, in Dependable Syst. Networks (DSN), IEEE 44th Annu.IEEE/IFIP Int. Conf., 2013.

[11] S. Kamara and C. Papamanthou, “Parallel and dynamic searchable symmetric encryption”, in Proc. Financ. Cryptography Data Secur., 2013.

[12] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Rosu, and M. Steiner, “Highly-scalable searchable symmetric encryption with support for boolean queries”, in Proc. Adv. Cryptol., 2013.

[13] D. Cash, J. Jaeger, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Rosu, and M. Steiner “Searchable encryption in very large databases: Data structures and implementation”, in Proc. Netw. Distrib. Syst. Security Symp., 2014.