International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 12 | Dec 2019 www.irjet.net p-ISSN: 2395-0072

Data Mining - Secure Keyword Manager

Mrs. Nalini S. Jagtap[1], Ms. Rachana Mudholkar[2], Mrs. Pratiksha Shevatekar[3]

[1]Asst. Prof., Dept. of Computer Engineering, Dr D. Y. Patil Inst. of Eng., Mgmt. and Research, Maharashtra, India.
[2]Asst. Prof., Dept. of Computer Engineering, Dr D. Y. Patil Inst. of Eng., Mgmt. and Research, Maharashtra, India.
[3]HOD., Dept. of Computer Engineering, Dr D. Y. Patil Inst. of Eng., Mgmt. and Research, Maharashtra, India.

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 cipher text threat model.*

Key Words: Data Mining, Keyword, Database, Security, Network, Cloud, AI

1. INTRODUCTION

We propose a scheme, termed Efficient keyword retrieval, in which each user can choose the keyword of his own keyword to determine the percentage of matched linked keyword to be returned. The basic idea of keyword matching is to construct a privacy preserving mask matrix that allows the cloud to filter out a certain percentage of matched files before returning to the ADL. This is not a trivial work, since the cloud needs to correctly filter out files according to the keyword of queries without knowing anything about user privacy. Focusing on different design goals, we provide two extensions: the first extension emphasizes simplicity by requiring the least amount of modifications from the keyword scheme, and the second extension emphasizes privacy by leaking the least amount of information to the cloud.

2. LITERATURE SURVEY:

Searchable encryption schemes enable the clients to store the encrypted data to the cloud and execute keyword search over cipher text domain. Due to different cryptography primitives, searchable encryption schemes can be constructed using public key based cryptography or symmetric key based cryptography. Song et al. 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 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. Curtmola et al. 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 model search functionality, such as single keyword search, similarity search, multi-keyword boolean search, ranked search, and multi-keyword ranked search [1], [2], [3], [4], etc. Multi-keyword boolean search allows the users to input multiple query keywords to request suitable documents. Among these works, conjunctive keyword search schemes [1], [2], [5] only return the documents that contain all of the query keywords. Disjunctive keyword search schemes [4], [5] return all of the documents that contain a subset of the query keywords. Predicate search schemes [3], [4], [5] 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. but they are designed only for single keyword search. Cao et al. [3] realized 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, Cao et al.'s 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. Sun et al. [1] presented a secure multi-keyword search scheme that supports similarity-based ranking. The authors constructed a searchable index tree based on vector space model and adopted cosine measure together with TF×IDF to provide ranking results. Sun et al.'s search algorithm achieves better-than-linear search efficiency but results in precision loss. Orencik et al. [5] proposed a secure multi-keyword search method which utilized local sensitive hash (LSH)functions to cluster the similar documents. The LSH algorithm is suitable for similar search but cannot provide exact ranking. In [3], 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. Practically, the data owner may need to update the document collection after he upload the collection to the cloud server. Thus, the SE schemes are expected to support the insertion and deletion of the documents. There are also several dynamic searchable encryption schemes. In the work of Song et al., the each document is considered as a sequence of fixed length words, and is individually indexed. This scheme supports straightforward update operations but with low efficiency. Goh proposed a scheme to generate a sub-index (Bloom filter) for every document based on keywords. Then the dynamic operations can be easily realized through updating of a Bloom filter along with the corresponding document. However, Goh's scheme has linear search time and suffers from false positives. In 2012, Kamara et al. constructed an encrypted inverted index that can handle dynamic data efficiently. But, this scheme is very complex to implement. Subsequently, as an improvement, Kamara et al. proposed a new search scheme based on tree based index, which can handle dynamic update on document data stored in leaf nodes. However, their scheme is designed only for single keyword Boolean search. In, Cash et al. presented a data structure for keyword/identity tuple named "TSet". Then, a document can be represented by a series of independent T-Sets. Based on this structure, Cash et al. proposed a dynamic searchable encryption scheme. In their construction, newly added tuples are stored in another database in the cloud, and deleted tuples are recorded in a revocation list. The final search result is achieved through efficient information retrieval query using aggregation and distribution layer.

3. ALGORITHM:

Step 1:

Each user runs the Query Gen algorithm to send a query to the ADL, where the user query consists of the chosen keywords and the query rank.

Step 2:

Given users' queries, the ADL runs the Matrix Construct algorithm (Alg. 2) to send a mask matrix to the cloud. The mask matrix M is a d-row and r-column matrix, where dis the number of keywords in the dictionary, and r is the highest rank of queries. The mask matrix M can be constructed as follows: For each keyword w, the ADL first sets w's rank with l, the highest query rank choosing this keyword. Then, for the row corresponding to keyword w, the ADL sets the first r−l columns to 1 and the last l columns to 0. The example mask matrix is shown in Fig. 5-(a). Note that, the reason for setting the first r−l columns, rather than random r−l columns, to 1 is to ensure that, given any two files with rank l, the probability of the product of the columns corresponding to file keywords being 0 is l/r.

Step 3:

Based on the mask matrix, the cloud runs the File Filter algorithm (Alg. 3) to filter out a certain percentage of matched files and returns a union buffer to the ADL. The process is as follows: For each file Fj, the cloud first multiplies the k-th columns that correspond to Fj's keywords in the mask matrix to obtain cj, where k=j MODr. The example columns chosen for each file. Then, the cloud powers the file content to cj to obtain ej and maps (ci, ei) to many entries of a union buffer as the Ostrovsky scheme. Here, cj denotes the occurrence of ranked keywords in file Fj. Thus, cj will be larger than 0, and file Fj will be returned only when l+k≤r, where k=j MODr.

Step 4:

The ADL runs the Result Divide algorithm to distribute files to each user. The ADL first recovers all files that match user queries as the File Recover algorithm in the Ostrovsky scheme. Then, the ADL distributes appropriate files to each user based on the user queries. To make sure that the ADL distributes files correctly, we can require the cloud to attach file keywords with the file content. Thus, the ADL can find out all of the files that match each user's query by executing keyword searches.

4. SYSTEM ARCHITECTURE:

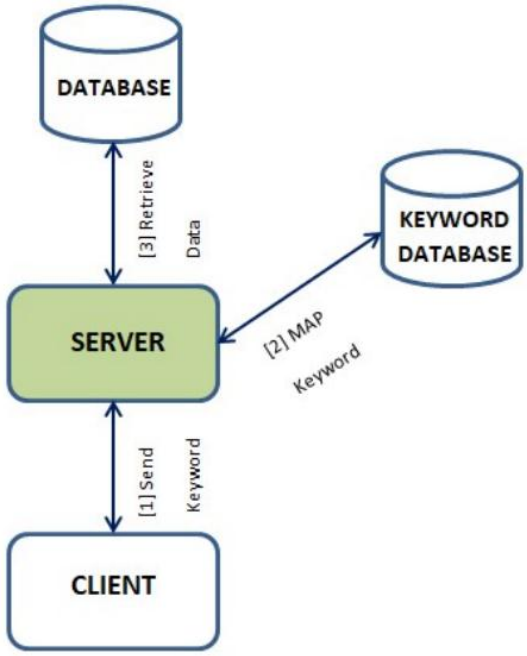

Fig -1: System Architecture

Cloud computing as an emerging technology is expected to reshape information technology processes in the near future.

Due to the overwhelming merits of cloud computing, e.g., cost-effectiveness, flexibility and scalability, more and more organizations choose to outsource their data for sharing in the cloud. As a typical cloud application, an organization subscribes the cloud services and authorizes its staff to share files in the cloud. Each file is described by a set of keywords, and the staff, as authorized users, can retrieve files of their interests by querying the cloud with certain keywords. In such an environment, how to protect user privacy from the cloud, which is a third party outside the security boundary of the organization, becomes a key problem.

5. SOFTWARE AND HARDWARE:

- Hardware
    - o Intel i5 processor
    - o 4 GB ram
    - o 500 GB HDD
- Software
    - o JDK 8
    - o NetBeans IDE
    - o MSSQL Server 2008 R2

6. ADVANTAGES

- ✓ Data Security
- ✓ Central Management
- ✓ Communication Security

7. CONCLUSION

We proposed keyword matching schemes based on an ADL to provide differential keyword services while protecting user privacy. By using our schemes, a user can retrieve different percentages of matched keywords by specifying public keyword of different ranks. By further reducing the communication cost incurred on the cloud, the ORQI schemes make the private searching technique more applicable to a cost-efficient cloud environment. However, in the ORQI schemes, we simply determine the rank of each file by the highest rank of queries it matches. For our future work, we will try to design a flexible ranking mechanism for the ORQI schemes.

REFERENCES

[1] K.Ren,C.Wang,Q.Wangetal.,"Security challenges for the public cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69–73, 2012. M. Young, the Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[2] C . Gentry, "A fully homomorphic encryption scheme," Ph. D. dissertation, Stanford University, 2009 .

[3] S. Kamara and K. Lauter, "Cryptographic cloud storage," in Financial Cryptography and Data Security. Springer, 2010, pp. 136–149.

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

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