Semantic Multi-Keyword Search Using Hashed Indexes on Encrypted Cloud Data Storage

Manu.Y.M¹, Shashikala.S.V²

¹P.G.Student, Department of Computer Science, BGSIT
Visveswaraya Technological University, Karnataka, India

²Professor and Head, Department of Computer Science
B.G.S Institute of Technology, Karnataka, India

ABSTRACT

Cloud storage is becoming popular and mandatory for more applications. Since data are stored in semi-trusted servers it is better idea to encrypt and store in cloud, but on encrypted data searching is difficult and it is a challenge. We proposed a system which provides both safety and efficient search process. Here the concept is system is creating keywords automatically from the uploaded files and weight age for each keyword is generated using Term Frequency Algorithm. After that keywords are converted into hash code and stored in cloud index for more safety. When users are searching with more than one keyword these keywords are hash coded and send to the server. Server has to refer the indexes with the received hash code and using Document Inverse Frequency Algorithm in will shortlist Top-K documents and send it to client system, where the documents are decrypted.

Keywords—Cloud Private Storage, Privacy-preservative criteria, Encrypted Cloud data, ranked search.

1. INTRODUCTION

Cloud computing is the emerging technology and it has long dreamed vision of computing, where cloud patrons can remotely store their data into the cloud. Patrons can enjoy the on-demand high-quality services from a shared puddle of configurable computing resources [2], [3]. Its economic savings and great flexibility are motivating both individual users and enterprises to outsource their local complex data into the cloud. Most sensitive data, for example, financial transaction details, health records of a patient, confidential e-mails, photo and video albums, tax related documents, and so on, these sensitive data have to be encrypted by data owners before outsourcing to the commercial public cloud storage [4]. Downloading all the data in the cloud and decrypting locally is clearly impracticable, because of huge amount of bandwidth cost in the cloud systems. Thus, exploring privacy conserving and most effective search service over encrypted cloud data is importance.

By considering the large number of on-demand data users and large volumes of data in the cloud storage, this problem is challenging and difficult to meet the requirements of performance, scalability and usability of the user. To meet the Effective data retrieval, the huge amount of documents demands the cloud server to perform result relevance ranking, instead of returning undifferentiated results of documents. Such ranked search system enables data users to find the most relevant data quickly, rather than sorting through every match in the content of data [5]. Unnecessary network traffic eliminated by Ranked search by sending back only the most relevant data, which is highly suitable in the cloud because in the cloud we have to pay what we use i.e. “pay-as-we-use”. For the Protection of privacy, such ranking operation should not leak any keyword i.e. related to information. On the other hand, to improve the accuracy of search result as well as to enhance the searching experience of user, it is also necessary for such ranking system to support multiple keywords search in the search request. Web search engines such as Google search, yahoo search and so on, users has to provide a set of keywords instead of only one keyword for search to retrieve the most relevant data in the cloud. Coordinate matching (i.e., as many matches as possible), it is an efficient similarity measure among such multi-keyword semantics. It has been widely used in the plaintext data retrieval community. But to apply it in the encrypted cloud data search system remains a very challenging task because of some privacy and security obstacles, including various strict requirements such as keyword privacy, index privacy and data privacy. In the literature survey, searchable encryption [15] is a helpful technique that treats encrypted data as documents and allows a cloud user to securely search through a single keyword and retrieve documents of interest from the cloud.
However, application of these approaches to the secure large scale cloud data utilization system would not be suitable and cannot oblige such high service-level requirements like usability of the system, searching experience of user, as well as easy discovery information. Even though few recent designs have been proposed to support Boolean keyword search [22], this helps to enhance the search flexibility for the users, but they are still not suitable to provide users with acceptable result ranking in the cloud data. To design an efficient encrypted data search mechanism that supports multi-keyword semantics without privacy breaks still remains a challenging problem. In this scenario, we describe and solve the problem of multi-keyword ranked search over encrypted cloud data while conserving privacy in the cloud computing.

Among different multi-keyword search request, as a user we select the efficient similarity measure of coordinate matching (i.e., as many matches as possible); it helps to capture the relevance of data documents to the search request. At the same time as constructing the index, document is associated with a binary vector as a sub index and search query is also described as a binary vector. Therefore similarity could be exactly measured by the inner product of the query vector with the data vector. However, index privacy or the search privacy will break by outsourcing the data vector or the query vector. Fulfill the challenge of supporting such multikeyword semantic without privacy ruptures, and in this paper we propose a basic idea for the MRS using secure inner product computation technique, and this is adapted from a secure k-nearest neighbor technique, and then give two significantly improved MRS schemes to achieve various privacy requirements.

Fig. 1. Architecture of the search over encrypted cloud data.

The main contributions of this paper are summarized as follows. First, we look into the problem of multi-keyword ranked search over encrypted cloud data and establish a set of strict privacy requirements for such a secure cloud data. Second, we introduce two MRS schemes based on the similarity measure of co-ordinate matching while meeting different privacy requirements in two different threat models. Third we investigate some further enhancements of our ranked search mechanism to support more search semantics. Finally, thorough analysis investigating privacy and efficiency guarantees of the proposed schemes is presented here; real-world data set experiments show the proposed schemes certainly introduce low overhead on computation.

Rest part of this paper is organized as follows: In Section 2 we introduce the system model, our design goals, and the preliminary. Section 3 describes the MRS framework and privacy requirements. Section 4, which describes the proposed schemes. Section 5 presents simulation results. And finally conclude the paper in Section 6.

2. PROBLEM FORMULATION

2.1 System Model

Cloud data hosting service consist of three different entities, as shown in Fig. 1: They are 1.Data owner, 2.Data user, and 3.cloud server. The data owner has a collection of data documents i.e. F these data documents to be outsourced to the cloud server in the encrypted form i.e. C by using encryption technique. To enable the searching capability over C for effective utilization of data, before outsourcing, the data owner should initially build an encrypted searchable index i.e. I from F (index from data documents), and then outsource both the index I and the encrypted document collection C to the cloud server.

To search the document collection for T given keywords i.e. search request, corresponding trapdoor T is acquired by an authorized user. By receiving T from a data user i.e. search request, and the cloud server is responsible to search the index I and return the corresponding set of encrypted documents from the cloud server. In this case to improve the document retrieval accuracy from the server, search result should be ranked by the cloud server on the basis of some ranking conventionality. And, to reduce the communication cost in the cloud system, data user may send an optional number k along with the trapdoor T so that the cloud server only sends back top-k documents that are most relevant to the search query T. Lastly, Decryption capabilities are given to users by the access control mechanism. Data collection can be updated by inserting new documents to the data collection, updating existing documents in the data collection, also deleting existing documents in the cloud server.
2.2 Design Goals

Our system design should concurrently achieve security and performance guarantees as shown below.

→ Privacy-conserving: To prevent the cloud server from additional information from the data set and the index I and to achieve requirements for the privacy.

→ Multi-keyword ranked search among encrypted cloud data: Design search schemes which allow multi-keyword query and provide result similarity ranking for effective data retrieval, instead of returning undifferentiated results from the server.

→ Efficiency: Above two goals on privacy and functionality should be achieved with low computation overhead and communication.

2.3 Preliminary on Coordinate Matching

In this scenario co-ordinate matching [6] is in-between similarity measure which uses the number of query keywords appearing in the document to quantify the relevance of that document to the query request. Only when users know the exact subset of the data set to be retrieved, for these Boolean queries perform well with the exact search requirement specified by the user in the search request. As we know in cloud computing, practically it is difficult, because huge amount of outsourced data from the data user. So, more flexible to users to specify a list of keywords indicating their interest and retrieve the most relevant documents with a rank order.

3. FRAMEWORK AND PRIVACY REQUIREMENTS FOR MRS

Here, we define the framework of multi-keyword ranked search over encrypted cloud data (MRS) and establish various strict system wise privacy requirements for such a secure data utilization system in the cloud.

3.1 MRS Framework

For easy presentation, operations on the data documents are not shown in the framework since the data owner could easily employ the traditional symmetric key cryptography to encrypt and then outsource data. With focus on the query and index. Neither the search control nor the access control is within the scope of this framework. Although the former is to regulate how trapdoors are acquired by authorized users.

3.2 Privacy Requirements for MRS

In this paper, we investigate and establish a set of strict privacy requirements specifically for the MRS framework.

As for the data privacy, the data owner can resort to the traditional symmetric key cryptography to encrypt the data before outsourcing to the cloud. Next with respect to the index outsourcing to the cloud. Next with respect to the index privacy for the data. Cloud server deduces any association between keywords and encrypted documents, this learn the major subject of a document, even the content of a short document. Because of this, the searchable index should be constructed to prevent the cloud server from performing such kind of association attack. Among various search privacy requirements are involved in the query procedure are more difficult and complex to confrontation as shown below.

Keyword privacy: Users usually prefer to keep their search from being exposed to others like the cloud server in the cloud environment, most important concern is to hide what they are searching. On the other hand the trapdoor can be generated in a cryptographic way to protect the query keywords in the search request. Therefore the cloud server do some statistical analysis over the search result to make an estimate. The cloud server knows some background information of the data set.

Trapdoor unlinkability: The trapdoor generation function should be a randomized one instead of being deterministic. To infer the relationships of any given trapdoors by cloud sever is not possible; it determines whether the two trapdoors are formed by the same search request. Or else, deterministic trapdoor generation would give the cloud server advantage to gather frequencies of different search requests; this may break the earlier keyword privacy requirement. Next is the access pattern. Within the ranked search result, the access pattern is the sequence of search results where every search result is a set of documents with rank order. Particularly, the search result for the query keyword set W is denoted as FW consisting of the id list of all documents ranked by their relevance to W. Then the access pattern is denoted as which are the results of sequential searches. On the other hand few searchable encryptions, access pattern is not focused here for the efficiency concerns.

4. PRIVACY CONSERVING AND EFFICIENT MRS

The representative privacy guarantee in the related literature of this paper, like searchable encryption scheme, in this the server should learn nothing but search results only. By using this general privacy
description phenomenon, for the first time we explore and establish a set of strict privacy requirements specifically for the MRS framework scheme. For the data privacy, the data owner can resort to the traditional symmetric key cryptography to encrypt the data before outsourcing to the cloud, this effectively prevent the cloud server from prying into the data which is outsourced. In case of index privacy, when the cloud server deduces any association between keywords and encrypted documents, this gives the major subject of a document, because of this; the searchable index should be constructed to prevent the cloud server from performing such kind of association attack.

4.1 MRS_I: Privacy-Conserving Scheme in Known Cipher text Model

The adapted secure inner product computation scheme is not good enough for our MRS design. The major reason is that the only randomness involved is the scale factor \( r \) in the trapdoor generation, which does not provide sufficient non determinacy in the overall scheme as required by the trapdoor unlinkability requirement as well as the privacy requirement for keyword. To facilitate a more advanced design for the MRS, we now provide our MRS_I scheme as follows.

4.1.1 MRS_I Scheme

In our more advanced design, instead of simply removing the extended dimension in the query vector. To preserve this dimension extending operation assign a new random number \( t \) to the extended dimension. This type of newly added randomness is expected to increase the difficulty for the cloud server to learn the relationship among the received trapdoors in the vector, and keyword requirements for privacy and randomness should also be carefully calibrated in the search result to obfuscate the document frequency and diminish the chances for keywords reidentification. While introducing some randomness in the final similarity score is an effective way toward what we expect in this scheme. Particularly, unlike the randomness in the query vector which is involved, then we have to insert a fake keyword into each data vector and random value is assigned for the same. Individual vector \( D_i \) is extended to \( (n+2) \) dimension instead of \( (n+1) \), where a random variable representing the dummy keyword is stored in the extended dimension.

4.2 MRS_I Scheme

The privacy leakage shown above is caused by the fixed value of random variable. For the elimination of such fixed property in any particular document in the cloud, many fake keywords instead of only one should be inserted into every data vector. Each vector is extended to \( (n+U+1) \) dimension instead of \( (n+2) \) dimension, and \( U \) is the number of fake keywords inserted.

4.3 MRS_I_TF

In the ranking principle coordinate matching (as many matches as possible), and the keyword presence in the document or the query is shown as 1 in the data vector or the query vector i.e. search query. In fact, there are more factors which could make impact on the usability of the search. Example, if one keyword appears in most documents in the data set of the cloud, the importance of this keyword in the query is less than other keywords which appears in fewer documents of the cloud. User may prefer this to the other document which contains the query keyword in only one location.

To capture these information in the search process list, we can use the TF*IDF weighting rule within the vector space model to calculate the similarity, where TF (or term frequency) is the number of times a given term or keyword (we will use them interchangeably hereafter) appears within a file (to measure the importance of the term within the particular file), and IDF (or inverse document frequency) is obtained by dividing the number of files in the whole collection by the number of files containing the term within file. In several hundred variations of the TF *IDF weighting scheme in the search process, there is no single combination of them outperforms any of the others unanimously. While maintaining generality, select an example formula that is commonly used and widely seen in the literature for the relevance calculation. To calculate the relevance score as shown in (1) on the server side:

\[
\text{Score}(F_i, Q) = \frac{1}{|F_i|} \sum_{w_j \in W} (1 + \ln f_{i,j}) \cdot \ln \left( 1 + \frac{n}{f_j} \right) .
\]  

5. PERFORMANCE ANALYSIS

In this section, we demonstrate a thorough experimental evaluation of the proposed technique on a real-world data set: the Enron Email Data Set. We randomly select different number of e-mails to build data set to demonstrate performance analysis.

Entire experiment system is implemented by C language on a Linux Server with Intel Xeon Processor. Performance of our technique is evaluated regarding the efficiency of four proposed MRS schemes, as well as the tradeoff between search precision and privacy.
5.1 Precision and Privacy

As presented in Section 4, fake keywords are inserted into each data vector and some of them are selected in every query processing. Hence the similarity scores of documents will be not exactly correct. On the other hand, the cloud server returns top-k documents based on similarity scores of data vectors to query vector i.e. request, few real top-k relevant documents for the query may be omitted. Reason for this either their original similarity scores are decreased or the similarity scores of some documents out of the real top-k are increased, because of fake keywords. User will evaluate the purity of the k documents retrieved, we describe a measure as precision \( P_k = k'/k \) where \( k' \) is number of real top-k documents that are returned by the cloud server.

result but lower rank privacy guarantee, at the meanwhile large results in higher rank privacy guarantee but lower precision. In other words, our scheme provides a balance parameter for data users to satisfy their different requirements on precision and rank privacy.

5.2 Efficiency Index Construction

To build a searchable subindex \( I_i \) for each document \( F_i \), the first step is to map the keyword set extracted from the document \( F_i \) to a data vector, followed by encrypting every data vector \( D_i \). Mapping time cost or encrypting depends directly on the dimensionality of data vector which is determined by the size of the dictionary. Time cost of building the whole index is also related to the number of subindex which is equal to the number of documents in the data set.

Fig. 4. Building index time cost. (a)Size of data set is different with the same dictionary, \( n=4000 \). Fig. 4 (b) for the same data set with different size.
Fig. 4(a) shows that, given the same dictionary where \( W \) is equal to 4000. And building the whole index time cost is nearly linear with the size of data set since the time cost of building each sub-index is static. Above Fig i.e. Fig. 4(b) shows that the number of keywords indexed in the dictionary determines the time cost of building a sub-index. But, as shown in both figures, such additional computation in the TF *IDF weighting rule is insignificant considering much more computation are caused by the splitting process and multiplication of matrix process. Regardless building index time is not a negligible overhead for the data owner in the cloud. In this case the size of sub-index is absolutely linear with the dimensionality of data vector which is determined by the number of keywords in the dictionary. The sizes of sub-index are very close in the two MRS schemes because of trivial differences in the dimensionality of data vector.

6. CONCLUSION AND FUTURE WORKS

In this research work, we establish a variety of privacy requirements and for the first time we are defining the problem of multi-keyword ranked search over encrypted cloud data in the distributed storage medium. Among various multi-keywords, select the efficient similarity measure of co-ordinate matching and we use inner product similarity measure. Here we introduce a basic idea of MRS using secure inner product computation scheme. By using analysis investigating privacy and efficiency guarantees of proposed schemes, real-world data set experiments show our proposed schemes introduce low overhead on computation. Our future work, focus on checking the integrity of the rank order in the search result assuming the cloud server is untrusted.

7. ACKNOWLEDGMENTS

This research is a combined effort of me and our guide of BGS Institute of Technology. I would like to gratefully acknowledge my organization, staff, and our guide for their valuable suggestion and support.

REFERENCES


About Authors:

**Manu.Y.M** received the B.E degree in Computer Science from Visvesvaraya Technological University Karnataka in 2011, and currently he is a post graduate student pursuing M.Tech in Computer Science and Engineering from B.G.S Institute of Technology under Visvesvaraya Technological University Karnataka. He has presented 5 papers in National level conferences and 2 papers in International conferences. His main research interests include Computer Networks, cloud computing and big data. He is currently doing his research project on cloud security in data storage.

**Shashikala.S.V** received her B.E degree in Computer Science from Mysore University, Karnataka in 1990 and M. Tech in Computer Science and Engineering from Visvesvaraya Technological University in 2005; she is currently working towards her PhD degree in the area of big data over a multipath TCP. She is a Professor and Head in the department of computer science at B.G.S Institute of Technology; B.G.Nagar, Mandya and having totally 22 years of teaching experience. She has presented 4 papers in National level conferences and also presented 3 papers in International conferences, she has published 3 papers in International Journals. Her main research interests include computer networks, big data and data mining.