International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 09 | Sep 2018 www.irjet.net p-ISSN: 2395-0072

An Efficient Ranked Multi-Keyword Search for Multiple Data Owners Over Encrypted Cloud Data: Survey

Roshni Rajendran[1], Vani V Prakash[2]

[1]*M. Tech. Student, Computer Science and Engineering, Sree Buddha College of Engineering, Kerala, India.*
[2]*Assistant Professor, Computer Science and Engineering, Sree Buddha College of Engineering, Kerala, India.*

Abstract - *Many people are using the cloud storage for storing their large amount of data. Not only by individuals many companies, industrialists are also using the cloud storage. Day-by-day the amount of people using the cloud storage is increasing due to its easiness of use. The data that has been stored in cloud may contain some secret documents also. Thus a secured storage and a secured data retrieval is necessary. Many searchable algorithms for cloud is existing. But less of them provide proper protection for the data that is stored. To increase confidentiality in the case of multiple data owners a tree-based ranked multi-keyword search scheme can be used. By considering a large amount of data in the cloud, the TF-IDF model is used to develop a multi-keyword search and return the top search results. The cloud server also uses a depth first search algorithm to find the corresponding file from the cloud.*

Key Words: **Index, TF-IDF, CBF, Paillier cryptosystem, BIDS, DFS, CSP, CPABE, PSED.**

1. INTRODUCTION

In cloud storage the data is stored in logical pools as digital data. In multi-owner scenario, the same data will contain several owners. A main server will be there to handle the entire data. The cloud may contain multiple servers may be reside in multiple locations. The main server or the cloud storage providers will be responsible for protection and handling of the stored data. The cloud users will buy or lease the storage capacity from these cloud storage providers. Cloud storage enables distributed and scalable network access to the digital data. A problem that has to be faced in cloud storage is the secured search over the encrypted data.

The most challenging task in cloud storage is secured search on encrypted cloud data. There are various search schemes are existing. But they results either in system overhead or sometimes those methods will be really hard to implement over large data sets. To prevent the unauthenticated access the data will be stored in cloud as in the encrypted form.

To provide an efficient search, a tree based multi keyword search scheme is constructed[1]. The words that are seemed as keywords for a document are identified and an index is formed. All the indexes such formed are then merged into one. For each search requests a depth first search is used to identify the corresponding data file of the user. The TF-IDF model is used to return the top results. A depth first search is used to perform efficient search.

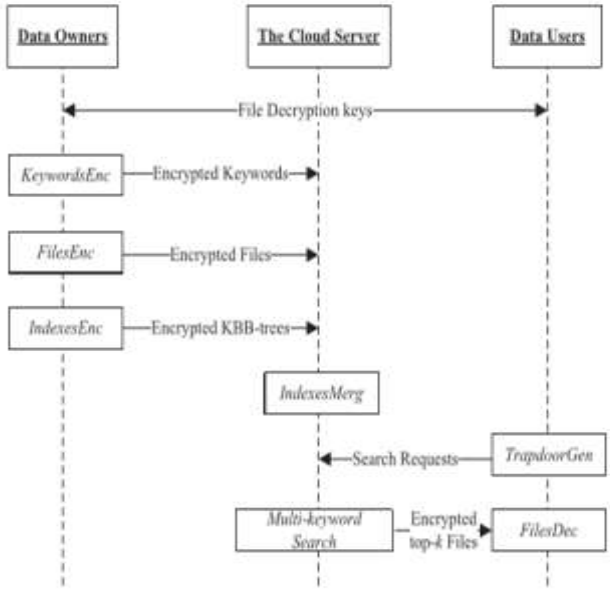

Fig -1: Tree based search scheme

If a user wishes to retrieve only those documents which contains certain words, then the user has to define any kind of mapping of words to that documents using the corresponding keywords. For the proper retrieval of data, the user has to be define the mapping or any method to the cloud storage initially. The method must be work without loss of data confidentiality.

A user can read and write data over internet through the allotted space in a cloud. This file sharing can be done from any location. Since every operations are done in the server, a proper backup and recovery system is essential.

But several security risks are existing if the data storage has been done without proper security measures. Many third party providers are existing now a days. Those are differ in their security measures that they have taken. There can be several documents that have more than one owners. But the entire part of the data will not visible to all of the owners due to security measures. An example for this is the health record system of certain patients in a health care sector. The patients such as the data users have to access the documents regarding their health conditions. For that they have to access top data files from different data owners.

In personal health record system, data user such as a patient should have the ability to access their top data files about a specific case from different data owners. These data owners

may be health monitors, hospitals, doctors etc. Similarly, the employees in an enterprise should have the ability to search data files by the other employees. The multiple data owners top-k query, whereby the cloud server can merge multiple data indexes encrypted with different keys and efficiently support top-k query.

In contrast to the single-user scenario, developing an efficient scheme for multiple data owners becomes is a challenge. To implement privacy preservation and efficient searches, a tree-based index structure for each data owner's encrypted data can be built. For a specific query condition, data users need to generate a trapdoor for each data owner, and the cloud should also search each index. This is obviously inefficient, due to the linear relationship of the number of trapdoors and data owners. A simple way to overcome this limitation is to let each data owner utilize the same key to encrypt their data files. Nevertheless, any one of the owners being compromised may lead to a system crash.

2. LITERATURE SURVEY

Various methods are used for searching of data files in cloud for multi-owner scenario. Some of them are discussed below.

2.1 Practical Techniques for Searches on Encrypted Data

Dawn Xiaodong et al. [2] proposed a method for searching without any loss of data confidentiality. If a mobile user wants to retrieve the documents containing a particular keyword from the mail storage server with limited bandwidth. The problem is the server has to know about the content of the documents. So the problem is to support the search queries without revealing all the data.

The servers must be trusted and must not reveal the data without proper authorization. The untrusted server leads to undesirable security and privacy risks in applications.

The untrusted server must not learn anything about the plaintext rather than the ciphertext. So that the untrusted server cannot search for a word without the user's authorization by using the techniques of controlled searching. The user can ask the untrusted server to search for a secret word without revealing the word to the server by supporting hidden queries. The untrusted server learns nothing more than the search result about the plaintext by supporting query isolation.

First the problem of searching on encrypted data is defined. Assume user A has a set of documents and stores them on an untrusted server S. For example, A could be a mobile user who stores her email messages on an untrusted mail server. Because S is untrusted, A wishes to encrypt her documents and only store the ciphertext on S. Each document can be divided up into 'words'. Each 'word' can be any token such as a word or a sentence. The user A may have only a low-bandwidth network connection to the server S, he/she wishes to only retrieve the documents which contain the word W. In order to achieve this goal, we need to design a scheme so that after performing certain computations over the ciphertext.

Server S can determine with some probability whether each document contains the word W without learning anything else.

There seem to be two types of approaches. One possibility is to build up an index that, for each word W of interest, lists the documents that containing W. Another method is to perform a sequential scan without an index. The use of an index is that it is faster than the sequential scan when the documents are large. But the index will increase overhead due to storing and updating of index. So the use of index is more suitable for the read-only data. At first a scheme for searching on encrypted data without an index is analyzed.

In all schemes, by allowing server S to search for a word W we effectively disclose to him a list of potential locations where W might occur. If we allow S to search for too many words, he may be able to use statistical techniques to start learning important information about the documents. One possible defense is to decrease m(so that false matches are more prevalent and thus server's information about the plaintext is 'noisy'), but we have not analyzed the cost effectiveness of this tradeoff in any detail.

A better defense is for user A to periodically change the key, re-encrypt all the documents under the new key, and reorder the ciphertext according to some pseudorandom permutation (known to A but not to server). This will help prevent server S from learning correlations or other statistical information over time. This technique may also be helpful if A wants to hide from S the places where the searched word occurs in the documents of interest.

In all the schemes, we must trust server S to return all the search results. If S holds out on us and returns only some (but not all) of the search results, A will have no way to detect this. An assumption is made that server S does not misbehave in this way. Even when this type of attack is present, it is possible to combine this scheme with hash tree techniques to ensure the integrity of the data and detect such attacks.

The remote searching on encrypted data using an untrusted server is considered here. This techniques have a number of crucial advantages: they are provably secure; they support controlled and hidden search and query isolation; they are simple and fast . More specifically, for a document of length n, the encryption and search algorithms only need O(n) stream cipher and block cipher operations and they introduce almost no space and communication overhead.

This scheme considers every documents which contain the same keyword. So there is a chance to return the unwanted documents also only because of that keyword is present. This scheme is also very flexible, and it can easily be extended to support more advanced search queries.

2.2 Secure Index for Resource-Constraint Mobile Devices in Cloud Computing

Hanbing Yao et al. [3] proposed a secure index based on counting Bloom filter (CBF) for ranked multiple keywords search. Nowadays more organizations and users are outsourcing their data into cloud server. In order to protect

data privacy, the sensitive data have to be encrypted, which increases the heavy computational overhead and brings great challenges to resource-constraint devices. In this scheme, several algorithms are designed to maintain and lookup CBF, while a pruning algorithm is used to delete the repeated items for saving the space.

The problem of secure ranked search over encrypted data in the cloud server is discussed here. In the proposed scheme, counting Bloom filter is used to generate the secure index for ranked multiple keywords search. Moreover, several algorithms are designed to maintain and lookup CBF and a pruning algorithm is used to delete the repeat items for saving the space. The Paillier cryptosystem is employed to encrypt relevance scores. It ensures that even the same relevance scores will be encrypted into different bits, which can help to resist statistical analyses. The major computing work in rank is done by the cloud server on the encrypted relevance scores, which make the resource constraint mobile devices can easily search over encrypted data.

The Paillier cryptosystem is used to encrypt relevance scores. It will make sure that the same relevance scores are encrypted into different bits. So this can resist the statistical analyses on the ciphertext of the relevance scores. Moreover, the Paillier cryptosystem supports the homomorphic addition of ciphertext without the knowledge of the private key, the major computing work in ranking could be moved from user side to the cloud server side. Therefore, this scheme can effectively use in resource-constraint mobile devices such as 5G mobile terminals.

2.3 An Efficient and Compact Indexing Scheme for Large-scale Data Store

Peng Lu et al.[4] proposed that the large amount of data in the Cloud can be managed by the bitmap based indexing scheme(BIDS). To speed up query processing, an effective mechanism is to build indexes on attributes are used in query predicates. But conventional indexing schemes fail to provide a scalable service. The size of these indexes are proportional to the data size, so it is not space efficient to build many indexes. As such, it becomes more crucial to develop effective index to provide efficient search in the cloud.

A compact bitmap indexing scheme is used for construct index for a large-scale data store. To reduce the index cost, a query efficient partial indexing technique is adopted, which dynamically refreshes the index to handle updates and process queries. This indexing approach is used to maximize the number of indexed attributes, so that a wider range of queries, including range and join queries, can be efficiently supported. This indexing scheme is light-weight. Also the compactness allows to maintain the bitmap indexes in memory so that performance overhead of index can be minimized.

BIDS index is storage efficient and easy to maintain, which makes it more scalable. It is built on top of the underlying DFS and cached in the distributed memory. BIDS adopts bit-sliced encoding and pre-sorting to ensure compactness. To further reduce the index size, the index is dynamically tuned based on the query patterns. BIDS based query processing is also used to efficiently handle the queries.

2.4 Preferred Keyword Search over Encrypted Data in Cloud Computing

Zhirong Shen et al.[5] discuss about the problem of preferred keyword search over encrypted data (PSED). The scale of massive files in the cloud requires flexible search query to retrieve accurate search results without receiving the unneeded files. On the other hand, given the large amount of users in cloud environment, different users may find different things relevant when searching because of different preferences, indicating the necessity of preferred search support to cope with user's various preferences. Thus, exploring a flexible search service with preferred search support over encrypted data is extremely meaningful in cloud environment. Sensitive data are usually stored in encrypted form to protect data confidentiality in cloud utilization, by making traditional search service on plaintext inapplicable. Thus, enabling keyword search over encrypted data becomes very important.

Many data users with various search preferences becomes necessary to support preferred keyword search and output the data files in the order of the user's preference. The larger preference generally means the higher priority order. Since keywords and their frequencies are practical tools to characterize the file content and their significance, the relevance of a file to a query can be divided into many sub-relevance to represent the correlation of the file to keywords in the query. The product of the preference and the keyword weight to server as this sub-relevance, and take the accumulated sub-relevance to act as the relevance score of the file to the query.

By using the appearance frequency of each keyword to serve as its weight the keyword searching is done. A preference value for each user is also analyzed. This preference value is expressed by using Lagrange polynomial. Each of the keyword weights are represented by using vectors. Then the preference polynomial into vector format their inner products are calculated to find the relevance measure between data files and a query.

PSED focuses on preferred search over multiple fields, which aims to locate the accurate matching files and rank them according to the calculated scores. Due to these much of calculations, this method produces overhead to the server.

2.5 An Efficient File Hierarchy Attribute-Based Encryption Scheme in Cloud Computing

Shulan Wang et al.[6] proposed an efficient file hierarchy attribute-based encryption scheme in cloud computing. This encryption technology can solve the challenging problem of secure data sharing in cloud computing. The shared data files generally have the characteristic of multilevel hierarchy, particularly in the areas like healthcare, military etc. However, the hierarchy structure of shared files has been done using Ciphertext-policy attribute-based encryption(CPABE).

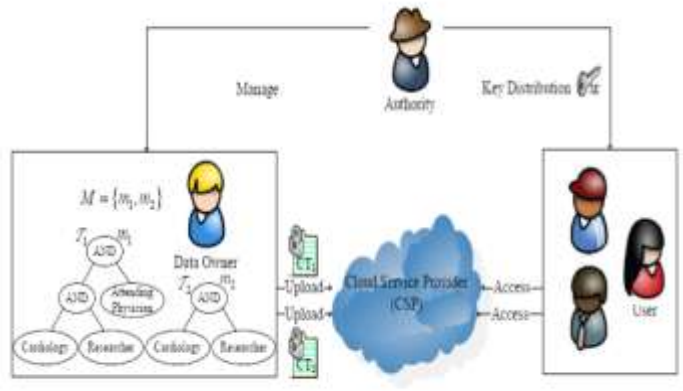

Fig -2: File sharing in cloud

The data files in multiple levels are integrated into a single access structure. That is data files of different data users in a group can be integrated into one. Then the hierarchical files are encrypted using the integrated one. The components of ciphertext that related to attributes can be shared by the files. So, the ciphertext storage and time cost of encryptions are saved. As the number of files increasing, the advantages of this scheme become more and more noticeable.

In cloud computing server accepts the user files and creates some parameters. The one who manages the cloud servers and provides multiple services for client is the Cloud Service Provider (CSP). A data owner can encrypt the data files and upload the generated ciphertext to CSP. A user can downloads and decrypts the ciphertext from CSP. These shared files must have hierarchical structure.

That is many hierarchy subgroups or a group of files may be located at different access levels. If the files in the same hierarchical structure can be encrypted by using integrated access structure, then the storage cost of ciphertext and time cost of encryption could be saved.

The hierarchical files are encrypted with an integrated access structure and the ciphertext components related to attributes could be shared by the files. The main advantage of this method is that users can decrypt all authorization files by computing secret key once.

2.6 Forward Secure Searchable Symmetric Encryption

Muhammad Saqib Niaz we et al.[7] proposed a forward secure searchable symmetric encryption. One of the important security threats in cloud is the data outsourcing to a third party. An unauthorized access is one of the security threat to the outsourced data. It can be avoided by encrypting the data before outsourcing. However, encrypting data before outsourcing renders it unsearchable to the data owner.

Searchable encryption schemes are developed to specifically search on encrypted data. A dynamic searchable encryption is the one that allows the data owner to add or delete a file after data outsourcing. Dynamic searchable encryption schemes are vulnerable to two specific security threats that are not applicable to the static searchable encryption schemes namely forward privacy and backward privacy. Forward privacy requires that the addition of a file should not reveal the presence of a previously searched keyword. Backward privacy requires that a search should not return the file identifier of a previously deleted file.

A dynamic searchable scheme that guarantees forward privacy is constructed. It only uses the symmetric key algorithms hence reducing the requirements for storage and processing power on the client side. Furthermore, this method is space reclaiming. After the deletion of a file, the redundant data nodes are also deleted from the secure index in the subsequent searches. Because of this space reclaiming capability of the scheme, the scheme is also partially backward private.

3. CONCLUSION

Various methods are used to make index and do searching in the encrypted text etc. But in a multiple data owner model which is considered for analyzing about the data sharing in cloud computing an efficient ranked multi-keyword search scheme over encrypted data is done. The index trees for each data files are merged into one. The searching is done using a DFS algorithm. That is a secure search protocol that allows different data owners to encrypt the files and indexes with different keys are used. Then, a tree-based index structure for each data owner allows the cloud server to merge encrypted indexes without knowing any information. This tree based search scheme is more efficient in keyword mapping that other existing methods.

REFERENCES

[1] T. Peng, Y. Lin, X. Yao and W. Zhang, "An Efficient Ranked Multi-Keyword Search for Multiple Data Owners Over Encrypted Cloud Data," *in* IEEE Access, *vol. 6, pp. 21924-*21933, 2018.

[2] Dawn Xiaoding Song, D. Wagner and A. Perrig, "Practical techniques for searches on encrypted data," *Proceeding* 2000 IEEE Symposium on Security and Privacy. S&P 2000, Berkeley, CA, USA, 2000, pp. 44-55.'

[3] H. Yao, N. Xing, J. Zhou and Z. Xia, "Secure Index for Resource-Constraint Mobile Devices in Cloud Computing," in IEEE Access, vol. 4, pp. 9119-9128, 2016.

[4] P. Lu, S. Wu, L. Shou and K. Tan, "An efficient and compact indexing scheme for large-scale data *store,"* 2013 IEEE 29th International Conference on Data Engineering (ICDE), Brisbane, *QLD,* 2013, pp. 326-337.

[5] Z. Shen, J. Shu and W. Xue, "Preferred keyword search over encrypted data in cloud computing," 2013 IEEE/ACM 21st International Symposium on Quality of Service *(*IWQoS*)*, Montreal, QC, 2013, pp. 1-6.

[6] S. Wang, J. Zhou, J. K. Liu, J. Yu, J. Chen and W. Xie, "An Efficient File Hierarchy Attribute-Based Encryption Scheme in Cloud Computing," in IEEE Transactions on

Information Forensics and Security, vol. 11, no. 6, pp. 1265-1277, June 2016.

[7] M. S. Niaz and G. Saake, "Forward secure searchable symmetric encryption," 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST), Cambridge, 2017, pp. 49-54.