© 2018 IJSRSET | Volume 4 | Issue 1 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section : Engineering and Technology

Privacy Preserving Multi Keyword Search Using Cosine Similarity in Cloud

Jincy Easow, Prof Jisha P Abraham
Department of Computer Science, Mar Athanasius College of Engineering, Kothamangalam, Kerala, India

ABSTRACT

Cloud Computing is a mature model of IT infrastructure that provides on demand high quality applications and services from a shared pool of computing resources. It becomes an increasingly popular for data owners to outsource their data to public cloud servers and also allowing data users to retrieve this data. For privacy concerns, secure searches over encrypted cloud data has motivated several research works under the single owner model. However, most cloud servers in practice do not just serve one owner instead, they support multiple owners to share the benefits brought by cloud computing. The proposed scheme is to deal with multiple data owners to store their sensitive information securely in cloud and to allow data users to retrieve data through multiple keywords. To enable cloud servers to perform secure search without knowing the actual data of both keywords and outsourced data, systematically construct a novel secure search protocol using cosine similarity approach.

Keywords : Cloud Computing, Cosine Similarity, Vector Space Model, Secure Search.

I. INTRODUCTION

Cloud computing is an information technology paradigm that enables ubiquitous access to shared pools of configurable system resources and higher-level services that can be rapidly provisioned with minimal management effort, often over the Internet. Cloud computing relies on sharing of resources to achieve coherence and economy of scale, similar to a utility. Third-party clouds enable organizations to focus on their core businesses instead of expending resources on computer infrastructure and maintenance. Cloud providers typically use a "pay-as-you-go" model, which can lead to unexpected operating expenses if administrators are not familiarized with cloud-pricing models. For enterprise users, data growth is tremendous with online business transactions. The demand for outsourcing data storage, and management has increased dramatically. Business data is vital to companies, they can get reliable, available, fault tolerance and performance from cloud service providers, but they cannot take the risk that service provider to scan data. Enterprise users have to consider data confidentiality and safeguard the data from unauthorized access and analysis, even browsing from service provider. If enterprise users were willing to outsource data storage to cloud, they would prefer flexible but secure data storage and management services.

As data itself and its access pattern are the two major aspects of privacy issues concerned by cloud users, the outsourced data on the cloud must leak as little information as possible. But here consider the application scenario where a group of users share data through an untrusted data storage server. The server can store and search on encrypted data it hosting for this group of users. It uses indexed and encrypted keywords to serve searching on encrypted data. And also require the search can be executed without leaking access pattern, such as the frequency of a query have to be protected against man-in-the middle attacker, advantages for storage server to analyse and so on.

Despite the tremendous advantages, privacy concern is one of the primary hurdles that prevents the widespread adoption of the cloud by potential users, especially if their sensitive data are to be outsourced to and computed in the cloud. Examples may include financial and medical records, government confidential files and social network profiles. Cloud service providers (CSPs) usually enforce user's data security through mechanisms like firewalls and virtualization. However, these mechanisms do not protect user's privacy from the CSP itself since the CSP possess full control of the system hardware and lower levels of software stack. There may exist disgruntled, profiteered, or curious employees that can access user's sensitive information for unauthorized purposes. Although encryption before data outsourcing can preserves data privacy of user's against the CSP, it also makes the effective data utilization, such as search over encrypted data, a very challenging task. To protect data confidentiality and eliminate the access pattern leakage issue, especially to protect against passive attack from eavesdropping.

The organization of this document is as follows. In Section 2 (Literature Survey), I'll give a critical appraisal of the previous work published in the literature pertaining to the topic of the investigation. In Section 3 (Methodology), present specifies the features and methodology used for the project. Discussed in Section 4(Conclusion) a conclusion is the last part of something, its end or result.

II. LITERATURE SURVEY

Anuradha Meharwade[1] proposed keyword search over encrypted cloud data. The system consist of three entities like data owners, data users and cloud service provider. Data owner upload the files containing sensitive information to the cloud service provider. Cloud server provides the search service for the authorized user over encrypted data. This scheme returns the search results according to the relevance of the file with respect to the keyword. The first file has more word related information than other files. Searchable index is used for the information retrieval in this scheme, which stores a list of mappings from keywords to the list of files containing keywords. For ranked search purposes, by determining which files are most relevant by assigning a numerical score. Once the index is built, encrypt the file and upload both index and the encrypted file to the server. When user sends a request to the CSP to retrieve files containing information relevant to the keyword. Upon receiving the request from the user, server firstly search on the searchable index and get the file id of the relevant files, then send them to the requested user in the rank order. Decryption algorithm is used for file decryptions and which is downloaded at the user end.

Wenhai Sun[2] proposed a privacy preserved multi-keyword text search over encrypted data. The proposed system model system consist of three entities: data owner, data user and the cloud server. In this system, the data owner outsources huge collection of documents in the encrypted form together with an encrypted searchable index tree. Search control mechanism proposed in this paper is by broadcast encryption through which data user obtain the encrypted search query. Upon the receipt of query, cloud server starts searching over the index tree and return corresponding set of encrypted documents, which will be ranked by frequency based similarity measures. Accurate multi-keyword ranked search is achieved by adopting cosine measure to evaluate similarity score. Final similarity score for the document obtained by summing up the scores from each level. Based on these similarity score, the cloud server determines the relevance of the document to the query. Then send back the most relevant document back to the user.

Ruixuan Li[3] proposed multi keyword ranked query search over encrypted cloud data. The system has three roles: data owner, data consumer and cloud service provider. The data owner identifies the sensitive files, those are encrypted using any standard symmetric encryption algorithm. It also specifies the keyword set for make a keyword dictionary to be used for queries by data consumers. For each file, an index vector is generated based on the keywords present in it. Index vectors are also encrypted and then joined together to form an index file. After the encryption to be performed, both the encrypted files and the index files are uploaded to the cloud. For effective search query, data owner needs to define a secret key, elements in the secret key are randomly generated. So only the data consumer can decrypt it. Query is executed as that firstly data consumer send a request containing set of searching keywords to the data owner. Data owner generate the trapdoor for the request using the secret key and then send it to the data consumer who send the request. Finally, data consumer forward the trapdoor to the cloud server. Cloud server performs the matching operation based on the trapdoor is conducted and a set of encrypted files are identified. After receiving the encrypted files from the cloud server, data consumer again send the request to the data owner in order to obtain the secret key for decrypts those received files.

III. METHODOLOGY

The Cloud computing is an increasingly popular due to benefits provided to the users, which can be accessed from anywhere, anytime and through any device. Despite of many benefits, most of the users are reluctant to outsource their sensitive data into the cloud medium. In order to protect the data privacy, the data owners encrypted their sensitive data before outsourcing, which helps to protect the data from the cloud service provider itself. The proposed method perform secure and easiest way for the data storage and data retrieval by the users. There are four entities are present here: Data Owner, Cloud Server, Administrative Server and the Data Users. Data owner uses the symmetric encryption algorithm AES is used to encrypt the data before outsourcing to the cloud. So data owner encrypt the sensitive information and send it to the cloud server. And also data owner extract the keywords from the file, encrypt it and send them to the administration server. Upon receiving the keywords from the data owner, administration server perform symmetric pre-encryption and randomization to prevent statistical analysis by an eavesdropper on the encrypted data.

The proposed system make use of the techniques of information retrieval community i.e. vector space model. Data owner has certain files containing sensitive data, due to data privacy they are uploaded to the cloud server after encryption over these files. So the files are encrypted using any standard symmetric encryption algorithms. Before outsource the encrypted files to the cloud, data owner has to extract certain relevant keywords from the files and send to the administration server. In proposed system to ensure security and efficiency most of the work is done by cloud. Term frequency-inverse document frequency weighing scheme is used to assign score for each document and cosine similarity to find the similarity. Cloud server then returns the calculated scores to the data user and data user send back the top scores to the cloud server. Thus there is a two round communication between cloud serer and data user.

Cloud provides suitable platform for sharing documents as well as resources. Usually to prevent leakage of sensitive information, the uploaded documents are encrypted. Retrieval of interested document is done with the help of a secure index. Data owners itself build an index for each documents he uploads into the cloud. The index is also encrypted. Here Apache’s Lucene API is used to extract keyword. Apache Lucene is a high performance full featured text extractor which extracts the keywords from the encrypted uploaded document. The cloud server stores the encrypted documents and corresponding encrypted index. Cloud users who want to retrieve the documents of his interest in relevant order issues query, which is also encrypted by symmetric encryption algorithm. Cloud server on receiving the query extract the key terms and stem it. To find the documents satisfying all the keywords in the query, both the documents and the query is represented as vector. Each dimension of the vector corresponds to the presence and absence of the keyword. The presence of a term in the vector is denoted by 1 otherwise 0. Term frequency-inverse document frequency is used to calculate the score of each term and cosine similarity to rank the documents. After ranking top-k documents scores matching the user's interest is returned back to the data user.

The score value calculation of the keywords are explained as follows. To weight the relevance of a document with respect to a keyword the simplest way is scoring. For scoring different ranking models exist, of these TF-IDF weighing scheme is the most common method. Term frequency and Inverse document frequency are the two attributes of this scheme. How many times a particular keyword occurs in a document are defined by Term Frequency (T_f) whereas in how many document a particular keyword exists is defined by Document frequency (D_f). The inverse document frequency is calculated as follows:

$$Id_f = \log \left[\frac{N}{D_f} \right]$$

Where N is the number of files

By using tf-idf weighting scheme, a term t from file f is given a score as:

$$\text{Tf} - IDf_{t,f} = Tf_{t,f} * IDf_t$$

A combination of the Vector Space Model and the Boolean model is used in Lucene Scoring. Vector space model determines how many times a query keyword appears in a document with respect to the number of times the keyword appears in all the documents in the collection. Boolean model on the other hand narrows down the documents that need to be scored based on the use of boolean logic.

A. Vector Space Model

To find the similarity of queried keyword with the existing documents, vector space model is used. In vector space model both documents and query is represented in the form of vector. For example consider the following two texts:

Text 1: Rahul loves me more than Kiran loves me

Text 2: Vivek likes me more than Rahul loves me

The keywords that occur in this are: me Rahul loves Kiran than more likes Vivek

Then count the number of times each keyword exists in the text as:

me	2	2
Rahul	1	1
likes	0	1
loves	2	1
Vivek	0	1
Kiran	1	0
than	1	1
more	1	1

The two vectors that correspond to the text are:

a: [2, 1, 0, 2, 0, 1, 1, 1]

b: [2, 1, 1, 1, 1, 0, 1, 1]

Here an 8 dimensional vector is formed. Cosine similarity is further used to find the similarity.

B. Cosine Similarity

To rank the documents different ranking techniques are available. Here to retrieve the top-K documents matching the user's interest, cosine similarity is used. An example for calculating cosine similarity is given below:

Consider a small collection of documents, consisting the following three documents:

document1: "nice shirt"

document2: "very nice smell"

document3: "los angel times"

Step1: Calculating Term frequency

For all the documents, then calculate the T_f scores for all the terms in C. Assign the score 1 if the keyword appear in that particular document, otherwise assign 0:

Table 3.1 Term Frequency

	angles	los	nice	shirt	smell	times	very
document 1	0	0	1	1	0	0	0

document 2	0	0	1	0	1	0	1
document 3	1	1	0	0	0	1	0

Step 2: Inverse Document Frequency

The total number of documents is N=3. Therefore, the idf values for the terms are:

angles $log_2(3/1) = 1.584$

los $log_2(3/1) = 1.584$

nice $log_2(3/2) = 0.584$

shirt $log_2(3/1) = 1.584$

smell $log_2(3/1) = 1.584$

times $log_2(3/1) = 1.584$

very $log_2(3/1) = 1.584$

Step 3: TF x IDF then multiply the tf scores by the idf values of each term, obtaining the following matrix of documents-by-terms:

Table 3.2 TF-IDF Matrix

	angles	los	nice	shirt	smell	times	very
Doc.1	0	0	0.584	1.584	0	0	0
Doc.2	0	0	0.584	0	1.584	0	1.584
Doc.3	1.584	1.584	0	0	0	1.584	0

Step 4: Vector Space Model and Cosine Similarity

Let the query given by the user be: "very very times", calculate the tf-idf vector for the query, and compute the score of each document in C relative to this query, using the cosine similarity. When computing the tf-idf values for the query terms, divide the frequency by the maximum frequency (2) and multiply with the idf values. Using the formula given below we can find out the similarity between any two documents.

Cosine Similarity (d1, d2) = Dot product(d1, d2) / ||d1|| * ||d2||

Dot product (d, d2) = d1[0] * d2[0] + d1[1] * d2[1] * … * d1[n] * d2[n]

||d1|| = square root(d1[0]2 + d1[1]2 + ... + d1[n]2)

||d2|| = square root(d2[0]2 + d2[1]2 + ... + d2[n]2)

The query entered by the user can also be represented as a vector q [0 0 (2/2)*0.584 0 (1/2)*0.584=0.292 0] Calculate the length of each document and of the query: Length of d1 = sqrt(0.584^2+0.584^2+0.584^2)=1.011 Length of d2 = sqrt(0.584^2+1.584^2+0.584^2)=1.786 Length of d3 = sqrt(1.584^2+1.584^2+0.584^2)=2.316 Length of q = sqrt(0.584^2+0.292^2)=0.652 Then the similarity values are: cosSim(d1,q) = (0*0+0*0+0.584*0.584+0*0+0.584*0.292+0.584*0) / (1.011*0.652) = 0.776 cosSim(d2,q) = (0*0+0*0+0.584*0.584+1.584*0+0*0.292+0.584*0) / (1.786*0.652) = 0.292 cosSim(d3,q) = (1.584*0+1.584*0+0*0.584+0*0+0.584*0.292+0*0) / (2.316*0.652) = 0.112

According to the similarity values, the final order in which the documents are presented as result to the query will be: d1, d2, d3.

Proposed Scheme

Cloud provides a platform to store huge amount of data containing sensitive information. Data owner upload the documents in an encrypted form into the cloud server. Before upload documents into the cloud, the data owner extract keywords using Apache Lucene and send it to the administration server. Administration server encrypt the keywords using any symmetric encryption algorithm and upload them to the cloud server. When the data user send a request in the form of keywords to the administration server. Administration server firstly authenticate the data user then encrypt the request and forward the trapdoor to the cloud server. The remaining operation like searching is done by the cloud server, which is performed by using the cosine similarity. Hence obtain the matching indexes from the stored keywords. If a matching is found, the cloud server get the id of the matched document and then send back the encrypted file to the requested data user. Again the data user send a request to the data owner to obtain the secret key used to encrypt that particular file. Upon receive the secret key from the data owner, data user can download the file and use it.

IV.CONCLUSION

Cloud computing is increasingly popular area, it become common to access global storage space over the Internet. The main problem of storing data in a trusted third party is regarding security. Even data is encrypted before outsourcing; effective information retrieval is a big challenge. To overcome these challenges, proposed a scheme which enables data owners to upload encrypted data files into global storage and allow several authorized users to perform operations such as uploading, search, share and retrieval over them. Instead of all the files, proposed method enable users to obtain the result with the most relevant files that match users' requirement. Documents of highest relevance are only sending back to the user.

V. REFERENCES

[1]. Cong Wang, Ning Cao, Jin Li, Kui Ren and Wenjing Lou, "Secure Ranked Keyword Search over Encrypted Cloud Data", IEEE Distrib. Comput. Syst., June 2010.

[2]. Anuradha Meharwade, G. A. Patil, " Efficient Keyword Search Over Encrypted Cloud data", International Conference on Information Security and Privacy, Dec. 2015.

[3]. Quin Liu, Guojun Wang and Jie Wuz, "Secure and Privacy Preserving Keyword Search on Encrypted Cloud Data", ELSEVIER Journel of Network and Computer Application, March 2011.

[4]. Wenhai Sun, Bing Wang, and Ning Cao, "Privacy Preserving Muli-Keyword Text Search in Cloud Supporting Similarity based Ranking", ACM Symposium on Information and Computer security, May 2013.

[5]. Zhang Xu, W kang, Rin Li, K Yow and C Xu, "Efficient Multi-keyword Ranked Query on Encrypted Cloud Data", IEEE Int. Conf. Parallel Distribution System, December 2012.

[6]. C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," IEEE Trans. Comput., Feb 2013

[7]. Jiadi Yu, Member, IEEE, Peng Lu, Yanmin Zhu, Member, IEEE, Guangtao Xue, Member, IEEE Computer Society, and Minglu Li, Toward Secure Multikeyword Top-k Retrieval over Encrypted Cloud Data IEEE Transactions On Dependable And Secure Computing, Vol. 10, No. 4, July/August 2013

[8]. Sun-Ho Lee and Im-Yeong Lee,"Secure Index Management Scheme on Cloud Storage Environment" ,International Journal of Security and Its Applications Vol. 6, No. 3, July, 2012

[9]. Craig Gentry" Fully Homomorphic Encryption Using Ideal Lattices" In the 41st ACM Symposium on Theory of Computing (STOC), 2009.

[10]. Sun-Ho Lee and Im-Yeong Lee,"Secure Index Management Scheme on Cloud Storage Environment" ,International Journal of Security and Its Applications Vol. 6, No. 3, July, 2012