Secure Multi-Keyword Search over Encrypted Cloud Data

Parameshwar Rao D
Department of Computer Science and Engineering,
Jain Global Campus, Jain University, Jakkasandra Post,
Kanakapura Taluk, Ramanagara District, India

S. Balaji
Center for Emerging Technologies,
Jain Global Campus, Jain University, Jakkasandra Post,
Kanakapura Taluk, Ramanagara District, India

***Abstract* — Cloud computing is an emerging technology, providing great flexibility and cost savings. It has motivated data owners to outsource their data from local systems to commercial public cloud. But sensitive data has to be encrypted before outsourcing to cloud to secure such data. This requires the traditional method data utilization based on plain text keyword search to be augmented with additional feature of searching encrypted data. Considering the large number of data users and data in cloud, it is necessary for the search service to allow multi-keyword query and return the data in the order of their relevance to these keywords. However, retrieving all the files having queried keyword will be expensive in "pay as you use" cloud paradigm. In the proposed system, vector space model and homomorphic encryption are employed wherein, the vector space model helps to provide sufficient search accuracy and the homomorphic encryption enables cloud service providers to perform the search operation based on user's multi-keyword search query without the need to decrypt it and also enables users to involve in the ranking.**

Keywords — Cloud, Ranking, Homomorphic Encryption, Vector Space Model.

1. INTRODUCTION

Cloud computing is a promising service for data outsourcing and high quality data services. Its great flexibility and cost savings are motivating both the individual users and the enterprises to outsource their local complex data management system into the cloud. To protect data privacy and combat unsolicited accesses in the cloud, sensitive data such as e-mails, personal health records, photo albums, tax documents, financial transactions, etc. may have to be encrypted by data owners before outsourcing to the commercial public cloud. However, this obsoletes the traditional data utilization service based on plain text keyword search. Thus, in order to ensure privacy of the personal information over the cloud, data owner must encrypt the data before uploading it to the cloud. There is a problem faced by the users since the cloud service provider needs to perform the calculations on data in order to respond to the requests made by the user. User has to provide key to the cloud service provider to decrypt the data before executing the required calculations, this affects the confidentiality of data stored in the cloud by the data owner.

In cloud computing, data owner may share their outsourced data with a number of users who might want to retrieve the data files of their interest. One of the most popular ways to do so is through keyword-based retrieval. Keyword-based retrieval is a typical data service and widely applied in plain text scenarios, in which users retrieve relevant files in a file set based on keywords. This searchable encryption schemes are impractical for real world cloud computing scenarios because these systems are designed to handle either a single keyword search or a Boolean search. The main drawback of single keyword-based search is that, it is hard to express complex information needs. On the other hand, Boolean search uses the presence or absence of queried keywords to retrieve the matched documents [2] and it is very difficult for most of the users to control the number of retrieved documents. In contrast, Information Retrieval (IR) systems [8] utilize ranked-search model to rank all the retrieved documents according to some relevance criteria. Such a model provides a precise answer by retrieving only the top−k relevant documents from the whole document collection. Furthermore, for efficiency purposes, all the current searchable encryption schemes reveal the access pattern to the un-trusted cloud service providers.

In order to improve feasibility and save the cost, it is preferred to get the retrieval result with the most relevant files that match user's interest instead of all the files. This indicates that the files should be ranked in the order of relevance by user's interest and only the files with the highest relevance need to be sent back to the users.

Currently, files are ranked only by the number of retrieved keywords. This impairs search accuracy and security is implicitly compromised to tradeoff for efficiency, which is particularly undesirable in security-oriented applications. Preventing the cloud from involving in ranking and entrusting all the work to the user is a natural way to avoid information leakage. However, the limited computational power on the user side and the high communication overhead precludes information security [5].

The homomorphic encryption enables users to involve in the ranking while the majority of computing work is done on the server side by performing search operation only on cipher text. Along with sending the encrypted data

over cloud, the data owner may also send the searchable index. Searchable index is a collection of phrases and keywords, which facilitates fast and accurate information retrieval. Thus, storage of searchable index along with the encrypted data in the cloud optimizes speed and performance in finding relevant documents for a search query.

In the proposed approach, searchable index is built from the collection of files that needs to be stored over the cloud in order to facilitate the fast and accurate retrieval of data. Homomorphic encryption and vector space model that guarantees the retrieval of most relevant data by performing user's multi- keyword search operation over encrypted cloud data are employed.

2. RELATED WORK

Ning Cao and Cong Wang [1], establish a set of strict privacy requirements for a secure cloud data utilization system. Among various multi-keyword semantics, authors use the efficient similarity measure of "coordinate matching", i.e., as many matches as possible, to capture the relevance of data documents to the search query and further use "inner product similarity" to quantitatively evaluate such similarity measure. The drawback observed is that direct outsourcing the data vector or the query vector will violate the index privacy or the search privacy.

Peng lu, Jiadi Yu, Xin Dong [2], introduce Two Round Searchable Encryption (TRSE) which preserves privacy of data retrieved but at higher communication overhead which has direct impact on efficiency.

Ayad Ibrahim, Hai Jin, Ali A. Yassin, Deqing Zou [3], propose a scheme that uses two distinct cloud servers, one for storing the secure index, while the other is used to store the encrypted document collection. Such a new setting prevents leaking the search result, i.e. the document identifiers, to the adversary cloud servers. The drawback is that utilizing two cloud servers is expensive which makes it impractical to use.

Bharath K, Samanthula and Wei Jiang [4], propose an efficient method for converting an encrypted integer z into encryptions of the individual bits of z and security primitive to construct a new protocol for secure evaluation of range queries in the cloud computing environment. Also, authors employ Privacy-Preserving Range Query (PPRQ) protocol which protects the confidentiality of the data and input query but reveals data access patterns.

Jiadi Yu, Peng Lu, Yanmin Zhu and Guangtao Xue [5], formulates the privacy issue from the viewpoint of similarity relevance and scheme robustness. It is observed that server-side ranking based on Order-Preserving Encryption (OPE) inevitably leaks data privacy. Data updates like adding or deleting files lead to a new challenge to the searchable encryption scheme.

Maha Tebaa, Said El Hajji, Abdellatif El Ghazi [6], propose a method to perform the operation on encrypted data without decrypting it and show that the same result as well when the calculations were carried out on the raw data but the efficiency is a tradeoff.

3. PROPOSED SYSTEM

The proposed approach retrieves the data using multi-keyword search over encrypted cloud data. Ranking is left to the user side in order to achieve data privacy.

In this approach, data owner uploads both encrypted files and the searchable index to the cloud server. As shown in Figure 1, data user sends a query consisting of multi-keywords to the cloud service provider. These queries will be processed by the cloud service provider by computing the scores from the encrypted index stored on the cloud and then cloud service provider will return the encrypted scores of files to the user. Next, the user decrypts the scores and picks up the top-k highest scoring file identifiers and request to the cloud server. Finally, the user gets the search result from the cloud server.

In order to reduce the computational burden on the user side, computing work is done at the cloud service side. It is essential to choose an encryption scheme that guarantees the operability and security on the server side.

Homomorphic encryption allows only specific types of computations to be carried out on the corresponding cipher text. Though the original fully homomorphic encryption property has such a fine property, which employs ideal lattices over a polynomial ring [8] it is complicated and inefficient for practical utilization.

In the fully Homomorphic Encryption Over the Integers (FHEI) scheme [7], the approximate integer Greatest Common Divisor (GCD) is incorporated to provide sufficient security. The cipher text resulted from the encryption of data will be large in size. In order to reduce the size of cipher text and the communication overhead, the original FHEI scheme should be modified to be more flexible in order to ensure the correctness of the decryption. Fortunately, as a result of employing the vector space model in top-k retrieval, only addition and multiplication operations over integers are necessary to compute the relevance scores from the encrypted searchable index. In the proposed approach, the original homomorphism in a full form is reduced to a simplified form that only supports integer operations, which achieves better efficiency than the full form. To further reduce the communication overhead, an ElGamal encryption is employed that result in generation of cipher text with smaller size.

Figure 1: Retrieval of Encrypted Cloud Data

Thus, the proposed system ensures the secure multi-keyword search over encrypted cloud data with improved efficiency.

4. ALGORITHMS USED

The algorithms that are used in the proposed system are as follows:

Algorithm 1: ElGamal

It is an asymmetric algorithm. That is, it makes use of two keys- (i) public key to encrypt the files and (ii) secret key to decrypt the files.

The steps followed in this algorithm are as follows:

ElGamal Key Generation Protocol:

The key generation protocol for the ElGamal algorithm is as follows:

Step 1: Start
Step 2: Create a random prime number, p. This number is the ElGamal "modulus".
Step 3: Select two other random numbers, g and x, both of which are less than p; these numbers do not have to be prime.
Step 4: Compute y, where $y = g^x \bmod p$. The public key is p, g and y. The private key is x.
Step 5: Stop

ElGamal Encryption Function:

The ElGamal encryption function is as follows:

Step 1: Start
Step 2: Break the plaintext into blocks based on the length of the public key modulus.
Step 3: Process each plaintext block, m:
a. Choose a random number, k, relatively prime to p - 1 (i.e., k and p - 1 have no common factors).
b. Compute $a = g^k \bmod p$.
c. Compute $b = (y^k m) \bmod p$.
d. Concatenate a and b to form an encrypted data block.
e. Concatenate the encrypted data blocks to form the cipher text.
Step 4: Stop

ElGamal Decryption Function:

The ElGamal decryption function is as follows:

Step 1: Start
Step 2: Break the cipher text into blocks that are twice the length of the key modulus
Step 3: Process each cipher text block:
a. Break the cipher text block in half to form a and b
b. Use the private key to compute the decrypted block m where

$$m = \frac{b}{a^x} \bmod p$$

c. Concatenate the decrypted blocks to form the plaintext.
Step 4: Stop

Algorithm 2: TopkSelect (source, k)

This algorithm is used to retrieve only the top-k ranked file list from the cloud server which is the result of the search operation according to the data user's multi-keyword search query.

The steps followed in this algorithm are as follows:
Step 1: Start
Step 2: Set topk = 0; topkid = 0;
Step3: Begin loop for all item ∈ source do
Step 4: INSERT (topk, (item, itemindex))
Step 5: end for loop.
Step 6: Begin loop for all tuple ∈ topk do
Step 7: topkid.append(tuple[1])
Step 8: end for loop
Step 9: return topkid
Step 10: Stop

Algorithm 3: Insert (topk, (item, itemindex))

This algorithm is used to insert/store the keywords, in order to build a searchable index. Searchable index is a collection of keywords that facilitates fast and accurate retrieval of data.

The steps followed in this algorithm are as follows:
Step 1: Start
Step 2: Condition check if length (topk) < k then
Insert (item, item index) into topk in non-decreasing order of item
Else if condition fails then continue
Step 3: Bein loop for all element ∈ topk do
Step 4: if item < element [0] then
Continue
Step 5: else if condition fails then
Step 6: Discard topk [0], insert (item, item index) into topk in non decreasing order of item
Step 7: end if condition
Step 8: end for loop
Step 9: end if condition
Step 10: Stop

Algorithm 4: Porter Stemmer

Porter Stemmer is one of the algorithms that are used in the information retrieval to reduce the size index files. A single stem typically corresponds to several full terms by storing stems instead of terms compression factors are achieved.

The steps followed in this algorithm are as follows:
Step 1: Start
Step 2: Gets rid of plurals and –ed or –ing suffixes.
Step 3: Turns terminal y to i when there is another vowel in the stem.
Step 4: Maps double suffixes to single ones: –ization, -ational, etc.
Step 5: Deals with suffixes –full, -ness, etc.
Step 6: Takes off –ant, -ence, etc.
Step 7: Removes a final –e.
Step 8: Stop.

5. Security Analysis

The foremost thing to be analyzed in the proposed system is that cloud server should not be able to know the

content, either of the data files, searchable index or the search keyword queries. Secondly, the cloud server should not be able to know the similarity relevance of terms or files so that the proposed system is highly robust.

The proposed system is able to conceal the access pattern and search pattern to be hidden from the cloud server; that is, if suppose the same keyword "t" is requested in two different queries as REQ1 and REQ2. Then, it forms the corresponding query vector say T1 and T2. After that, REQ1 and REQ2 are encrypted into two different cipher texts. Thus, same keywords in different queries are independent to each other, which mean that the keywords retrieved are hidden; thus, the access pattern and search pattern are secure.

In the proposed approach, an ElGamal encryption algorithm is used, it reduces the size of cipher text resulted by encrypting the plain text. As the size of cipher text is reduced, the communication overhead between the cloud service provider and the data user will also be reduced which improves the efficiency.

6. CONCLUSION

The proposed work ensures the secure multi-keyword search and top-k data retrieval over encrypted cloud data. The majority of computing work is carried out by server side by performing operations on cipher text which reduces computation overhead on data user side.

REFERENCES

[1] Ning Cao, Cong Wang, Ming Li, Kui Ren and Wenjing Lou, "Privacy-Preserving Multi-keyword Ranked Search over Encrypted Cloud Data", IEEE: 2011

[2] Peng Lu; Jiadi Yu; Xin Dong; Guangtao Xue; Minglu Li "Privacy-Aware Multi-Keyword Top-k Search over Untrust Data Cloud", 18th International Conference on Parallel and Distributed Systems (ICPADS), pp.252 – 259, 2012

[3] Ayad Ibrahim, Hai Jin, Ali A. Yassin, Deqing Zou, "Secure Rank-ordered Search of Multi-keyword Trapdoor over Encrypted Cloud Data", published at Asia-Pacific Services Computing Conference (APSCC), 2012

[4] Bharath K, Samanthula and Wei Jiang, "Efficient Privacy-Preserving Range Queries over Encrypted Data in Cloud Computing", IEEE: 2013

[5] Jiadi Yu, Peng Lu, Yanmin Zhu and Guangtao Xue, "Toward Secure Multi keyword Top-k Retrieval over Encrypted Cloud Data" IEEE: 2013

[6] Maha Tebaa, Said El Hajji, Abdellatif El Ghazi, "Homomorphic Encryption method applied to Cloud Computing", IEEE: 2012

[7] M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully Homomorphic Encryption over the Integers," Proc. 29th Ann. International Conference, Theory and Applications of Cryptographic Techniques, H. Gilbert, pp. 24-43, 2010

[8] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," Proc. 41st Ann. ACM Symp, Theory of computing (STOC), pp. 169-178,2009