Preserving Privacy of Cloud Data Using Homomorphic Encryption

Parameshwar Rao D
Department of Computer Science and Engineering,
Jain Global Campus, Jain University, Jakkasandra Post,
Kanakapura Taluk, Ramanagara District, India
parme147@gmail.com

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

***Abstract*—Cloud computing is a technology, which enables user to make use of the computing power and storage resources provided by cloud service providers. However, the presence of users' personal data over the cloud such as emails, bank account details, health records, personal photos may cause privacy issues. Data encryption ensures privacy of users' data to some extent but it is compromised during the retrieval of data. Since, it requires decryption of the data by cloud service providers in order to search for a data among a huge collection of encrypted data that is stored over cloud. 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 users' multi keyword search query without need to decrypt it and enables users to involve in the ranking. The majority of computing work is done on the server side by performing search operation on cipher text itself to eliminate information leakage and to ensure privacy of data.**

***Keywords*—cloud, data privacy, ranking, homomorphic encryption, vector space model.**

1. Introduction

Cloud computing is an emerging technology and promising pattern for data outsourcing and high quality data services. Cloud computing has attracted a lot of research and development effort in the past few years. However, privacy concerns arise whenever the sensitive data is outsourced to cloud. In order to ensure privacy of the personal information over the cloud, data owner must encrypt the data before uploading it to the cloud service provider. But 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 must provide the key to the server to decrypt the data before executing the calculations required, which might affect the confidentiality of data stored in cloud. Homomorphic encryption can be employed which enable the cloud service provider to perform the operations on encrypted data itself.

Furthermore, 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.

Basically, keyword based retrieval of data can be performed with one of the three search operations: (i) Boolean keyword search that support only keyword search operation on the basis of Boolean itself, that is, whether a keyword exists in a file or not, without considering the difference of relevance with the queried keyword; (ii) single keyword search that supports the search operation to be carried out based on a single keyword query and (iii) multi keyword search that enables searching of the data based on the multiple keyword queries. However, keyword based retrieval turns out to be a difficult task in cipher text scenario due to limited operations possible on the encrypted data.

A numerous forms of Searchable Symmetric Encryption (SSE) schemes had been used to enable search on cipher text. However, these SSE schemes enable users to securely retrieve the cipher text; but, these schemes supported only Boolean keyword search, that is, whether a keyword exists in a file or not, without considering the difference of relevance with the queried keyword of these files in the result [2].

In order to improve feasibility and save on the expense in the cloud, 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.

In the former, files were ranked only by the number of retrieved keywords, which 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 computational 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, to facilitate 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 requirement 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 system aims at preserving privacy and retrieval of data using multi keyword search over encrypted cloud data. To achieve data privacy, ranking is left to the user side.

In the proposed scheme, the data owner uploads both encrypted files and the searchable index on to the cloud server. As shown in Figure 1, when the cloud server receives a query consisting of multi keywords, it computes the scores from the encrypted index stored on the cloud and then returns the encrypted scores of files to the data user. Next, the data user decrypts the scores and picks up the top-k highest scoring file identifiers to request to the cloud server. Then, the data user gets the search result from the cloud server.

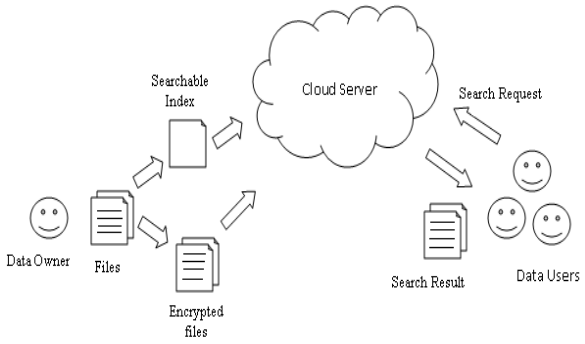

Figure 1: Retrieval of Encrypted Cloud Data

To reduce the computational burden on the user side, computing work should be done at the cloud service side. It is necessary to choose an encryption scheme that guarantees the operability and security at the same time on the server side. Homomorphic encryption allows specific types of computations to be carried out on the corresponding cipher text. The result is that the cipher text of the result is obtained by performing the same operations performed on the plain text. That is, homomorphic encryption allows computation of cipher text without knowing anything about the plain text to get the correct encrypted result. Although it has such a fine property, the original fully homomorphic encryption scheme, which employs ideal lattices over a polynomial ring [8], is too 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 used to provide sufficient security. The cipher text resulted from the encryption of data will be large in size. To reduce the size of cipher text and the communication overhead, the original FHEI scheme must 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 to top-k retrieval, only addition and multiplication operations over integers are needed to compute the relevance scores from the encrypted searchable index. Therefore, the original homomorphism in a full form is reduced to a simplified form in the proposed system that only supports integer operations, which allows better efficiency than the full form.

4. Algorithms Used

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

Algorithm 1: 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 2: 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: begin 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 3: 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.

6. CONCLUSION

The proposed work focuses on addressing the privacy of cloud data by allowing the cloud server to perform the search operation over the encrypted data without decrypting them and enabling users to involve in the ranking. The majority of computing work is carried out by server side by performing operations on cipher text.

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