International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Secured Access to Cloud Data through Encryption and Top-K Retrieval Using Multiple Keywords

Kavyashree J[1], Deepika N[2]

[1]New Horizon college of Engineering, Department of Computer Science & Engineering, Bangalore 560103, India

[2]New Horizon college of Engineering, Department of Computer Science & Engineering, Bangalore 560103, India

Abstract: ***Cloud computing is coming up as a promising platform for data outsourcing and high-quality data services. Since the sensitive information on cloud causes privacy problems, this paper relates to privacy issue of user data on cloud server when searching capability of the data is needed by the cloud user. The privacy of the data is achieved to some extent by encryption of the data. Searchable symmetric encryption (SSE) allows retrieval of encrypted data over cloud. When searching the data based on the user provided query, if the task of searching and ranking of the data as per query is fully given to cloud server the privacy will be breached and information is leaked. The server-side ranking based on order-preserving encryption (OPE) inevitably leaks data privacy. The paper aims to prevent the cloud server from getting the ranking information so that only user gets it. To remove the leakage, the paper proposes a double-round searchable encryption (DRSE) scheme that supports top-k multikeyword retrieval. In DRSE, vector space model and homomorphic encryption are employed. The vector space model helps to provide sufficient search accuracy, and the homomorphic encryption enables users to involve in the ranking while the majority of computing work is done on the server side by operations only on cipher text. As a result, information leakage can be eliminated and data security is guarded.***

Keywords: cloud, data privacy, ranking, homomorphic encryption, vector space.

1. Introduction

Cloud computing nowadays has become a necessary feasibility for data users to outsource their data. Since the privacy issues related to sensitive information like e-mails, health records and personal photos are increasing. Reports of data loss and privacy breaches in cloud computing systems appear regularly. The main threat on data privacy is in the cloud itself. When users outsource their private data onto the cloud, the cloud service providers are able to control and monitor the data and the communication between users and the cloud. So to ensure privacy, users usually encrypt the data before outsourcing it onto cloud, which brings great challenges to effective data utilization. But users still need to communicate with the cloud and allow the cloud to operate on the encrypted data, which causes leakage of sensitive information. In cloud computing there is an opportunity where data owners can share their outsourced data with a number of users, who might want to only retrieve the data files which they are interested in. One such way is through keyword-based retrieval. Keyword-based retrieval is a widely applied in plaintext scenarios, in which users retrieve relevant files in a file set based on keywords. But it would be a difficult task with cipher text scenario due to limited operations on encrypted data.

To improve feasibility and save on the expense in the cloud paradigm, it is preferred to get the retrieval result with the most relevant files that match users' interest instead of all the files, which indicates that the files should be ranked in the order of relevance by users' interest and only the files with the highest relevance are sent back to users.

A series of searchable symmetric encryption (SSE) schemes have been proposed to enable search on ciphertext. SSE schemes enable users to securely retrieve the ciphertext, but these schemes support only Boolean keyword search, i.e., 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. In order to improve security without sacrificing efficiency, schemes presented here show that they support top-k single keyword retrieval under various scenarios. Attempts are made to solve the problem of top-k multikeyword over encrypted cloud data. These schemes, however, suffer from two problems—Boolean representation and how to strike a balance between security and efficiency. In the former, files are ranked only by the number of retrieved keywords, which impairs search accuracy. In the latter, 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. The issue of secure multikeyword top-k retrieval over encrypted cloud data, thus, is: How to make the cloud do more work during the process of retrieval without information leakage.

In this paper, we solve the insecurity problem by proposing a double-round searchable encryption (DRSE) scheme. Contemporary techniques in the cryptography community and information retrieval (IR) community are used, including homomorphic encryption and vector space model. In the proposed scheme, the majority of computing work is done on the cloud while the user takes part in ranking, which guarantees top-k multikeyword retrieval over encrypted cloud data with high security and practical efficiency.

2. Retrieval of Encrypted Data from Cloud

Let's consider a cloud computing system which is hosting data service, as shown in Fig. 1, in which three entities are involved: cloud server, data owner, and data user. The cloud

server hosts data storage and retrieve services. Since data may contain sensitive information, the cloud servers cannot be fully trusted in protecting data. Hence outsourced files must be encrypted. It is not acceptable to have any kind of information leakage that would affect data privacy.

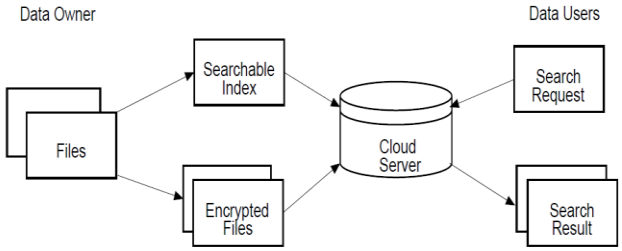

Figure 1: Retrieval of encrypted data from cloud- A scenario

The data owner has a collection of n files P= {f1, f2… fn} to outsource onto the cloud server in encrypted form and expects the cloud server to provide keyword retrieval service to data owner himself or other authorized users. In order to achieve this, the data owner needs to build a searchable index S from a collection of j keywords K = {k1, k2, ….., kj} extracted out of P, and then outsources both the encrypted index S' and encrypted files onto the cloud server.

The data user is authorized to process multikeyword retrieval over the outsourced data. The computing power on the user side is limited, which means that operations on the user side should be simplified. The authorized data user at first generates a query Q = { (k1', k2',….kr') | ki' € K, 1≤ S ≤ r ≤ j }. For privacy consideration, which keywords the data user has searched must be concealed. Thus, the data user encrypts the query and sends it to the cloud server that returns the relevant files to the data user. Afterward, the data user can decrypt and make use of the files.

3. Proposed scheme

Existing SSE schemes implement server-side ranking based on OPE to improve the efficiency of retrieval over encrypted cloud data. However, server-side ranking based on OPE violates the privacy of sensitive information, which is considered unacceptable in the security-oriented third party efficiency. To achieve data privacy, ranking has to be left to the user side. Traditional user-side schemes, however, load heavy computational cloud computing scenario, i.e., security cannot be tradeoff for burden and high communication overhead on the user side, due to the interaction between the server and the user including searchable index return and ranking score calculation. Thus, the user-side ranking schemes are challenged by practical use. A more server-siding scheme might be a better solution to privacy issues.

We propose a new searchable encryption scheme, in which novel technologies in cryptography community and IR community are employed, including homomorphic encryption and the vector space model. In the proposed scheme, the data owner encrypts the searchable index with homomorphic encryption. When the cloud server receives a query consisting of multikeywords, it computes the scores from the encrypted index stored on cloud and then returns the encrypted scores of files to the data user. Next, the data user decrypts the scores and picks out the top-k highest scoring files' identifiers to request to the cloud server. The retrieval takes a double-round communication between the cloud server and the data user. We, thus, name the scheme the DRSE scheme, in which ranking is done at the user side while scoring calculation is done at the server side.

3.1 Practical Homomorphic Encryption Scheme

To reduce the computational burden on the user side, computing work should be at the server side, so we need an encryption scheme to guarantee the operability and security at the same time on server side. Homomorphic encryption allows specific types of computations to be carried out on the corresponding ciphertext. The result is the ciphertext of the result of the same operations performed on the plaintext. That is, homomorphic encryption allows computation of ciphertext without knowing anything about the plaintext 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 is too complicated and inefficient for practical utilization. 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, we can reduce the original homomorphism in a full form to a simplified form that only supports integer operations, which allows more efficiency than the full form does.

On the basis of homomorphism property, the encryption scheme can be described as four stages:

KeyGen, Encrypt, Evaluate and Decrypt.
KeyGen(): The pair of keys i.e. Secretkeys(SK) and Publickeys(PK) are generated for the purpose of encryption.
Encrypt(PK,m): Encrypt the message using primary key.
Evaluate(c1; c2; . . . ; ct). Apply the binary addition and multiplication to the ciphertext ci, perform all necessary operations.
*Decrypt( ):*decrypt the resulting vector n download the necessary files using Secret key.

3.2 Framework of DRSE

The framework of DRSE includes four algorithms:

Setup, IndexBuild, TrapdoorGen, ScoreCalculate and Rank.
Setup(). The data owner generates the secret key and public keys for the homomorphic encryption scheme
IndexBuild(). The data owner builds the secure searchable index from the file collection P. Technologies from IR community like stemming are employed to build searchable index S from P, and then S is encrypted into S', output the secure searchable index S'.
TrapdoorGen(). The data user generates secure trapdoor from his request Q. Vector Tw is built from user's multikeyword request Q and then encrypted into secure trapdoor Tw ' with public key from PK, output the secure trapdoor Tw'.
ScoreCalculate**().** When receives secure trapdoor Tw', the cloud server computes the scores of each files in S' with Tw' and returns the encrypted result vector R back to the data user.

Rank. The data user decrypts the vector R with secret key SK and then requests and gets the files with top-k scores.

The whole framework can be divided into two phases: Initialization and Retrieval.

The Initialization phase includes Setup and IndexBuild.

The Setup stage involves the secure initialization, while the IndexBuild stage involves operations on plaintext. For security concerns, the vast majority of work should only be done by the data owner.

The details of the Initialization phase are as follows:
Initialization Phase:

1) The data owner calls KeyGen() to generate the secret key SK and public key set PK for the homomorphic encryption scheme. Then the data owner assigns SK to the authorized data users.
2) The data owner extracts the collection of j keywords, K= {k1, k2,…… ,kj}and their TF and IDF values out of the collection of n files,
   P= {f1,f2,……,fn}.
3) The data owner encrypts the searchable index S to secure searchable index S'
4) The data owner encrypts P = {f1,f2,……,fn} into P'= {f1',f2',……,fn' } and then outsources P' and S' to the cloud server.

The Retrieval phase involves TrapdoorGen, ScoreCalculate, and Rank, in which the data user and the cloud server are involved. As a result of the limited computing power on the user side, the computing work should be left to server side as much as possible. Meanwhile, the confidentiality privacy of sensitive information cannot be violated. So the ranking should be left to the user side while the cloud server still does most of the work without learning any sensitive Information.

The details of the Retrieval phase are as follows:
Retrieval Phase:

1) The data user generates a set of keywords
   Q = {w1',w2'…….,wr' } to search, and then the query vector Tw ={ m1, m2,……,mj } is generated. After that, Tw is encrypted into trapdoor Tw' and then the user sends Tw' to the cloud server.
2) For each file vector, the cloud server computes and returns the result vector R' to the data user.
3) The data user decrypts R' into R .Then TOPKSELECT(R,k) is invoked to get the top-k highest-scoring files' identifiers and sends it to the cloud server.
4) The cloud server returns the encrypted k files {fi1, fi2,……,fik } to the data user.

As a result of the limited computing power on the user side, we are mostly concerned about the complexity of ranking. Since the decryption of R can be accomplished in *O(n)* time, the only function that could influence the time complexity of ranking is the top-k select algorithm, i.e., TOPKSELECT algorithm. The details of TOPKSELECT algorithm are shown in Fig. 2a. Since the complexity of the INCLUDE algorithm is *O(k)* as illustrated in Fig. 2b, the overall complexity of TOPKSELECT algorithm is *O(nk*). Note that k, which denotes the number of files that are most relevant to the user's interest, is generally very small compared to the total number of files. In case of large value of k, the complexity of the TOPKSELECT algorithm can be easily reduced to O (n log k) by introducing a fixed-size min-heap.

TOPKSELECT (Source, K)

Input:
List Source to be selected
Number K

Initialization:
Set topk=0; topkid=0;

Iteration:
for all element € source do
INCLUDE(topk,(element,elementindex))
end for
for all pair € topk do
topkid.append(pair[1])
end for

(a)

INCLUDE (topk, (element, elementindex))

Input:
List topk to store the topk scoring element
pair (element, elementindex)

Iteration:
if len (topk)<k then
include (element, elementindex) into topk in ascending order of element
else
for all element € topk do
if element< item [0] then
Continue
else
discard topk [0], include (element, elementindex) into topk in ascending order of element
end if
end for
end if

(b)

Figure 2: (a) Algorithm TOPKSELECT (b) Algorithm INCLUDE

4. Conclusion

Secured access to cloud data through encryption and top-k retrieval using multiple keywords is an approach which provides the security with third party cloud server through encrypted data and allows computation to be performed on encrypted data through homomorphic encryption and hence scoring is done by cloud server while the ranking by the user, thus by reducing the burden on the user side and hence enables multiple keywords based search of only relevant files for multiple authorized user. Hence data privacy is guaranteed.

Since binary addition and multiplication operations are involved in DRSE scheme. The size of the cipher text doubles after multiplication, so to reduce the communication overhead, modular reduction can be applied in future.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and M. Zaharia, "A View of Cloud Computing," Comm. ACM, vol. 53, no. 4, pp. 50-58, 2010.
[2] M. Arrington, "Gmail Disaster: Reports of Mass Email Deletions",http://www.techcrunch.com/2006/12/28/gmail-disasterreportsof- mass-email-deletions/, Dec. 2006.
[3] Amazon.com, "Amazon s3 Availability Event: July 20, 2008," http://status.aws.amazon.com/s3-20080720.html, 2008.
[4] RAWA News, "Massive Information Leak Shakes Washington over Afghan War," http://www.rawa.org/temp/runews/2010/08/20/massive-information-leak-shakes-washington- overafghan-war.html, 2010.
[5] AHN, "Romney Hits Obama for Security InformationLeakage,"http://gantdaily.com/2012/07/25/romney-hits-obama-forsecurity-information-leakage/, 2012.
[6] Cloud Security Alliance, "Top Threats to Cloud Computing,"http://www.cloudsecurity alliance.org, 2010.
[7] C. Leslie, "NSA Has Massive Database of Americans' Phone Calls", http://usatoday30.usatoday.com/news/washington/
[8] 2006-05-10/, 2013.
[9] R. Curtmola, J.A. Garay, S. Kamara, and R. Ostrovsky, "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions", Proc. ACM 13th Conf. Computer and Comm. Security, (CCS), 2006.
[10] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure Ranked Keyword Search over Encrypted Cloud Data," Proc. IEEE 30th Int'l Conf. Distributed Computing Systems (ICDCS), 2010.
[11] S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, "Zerber+r: Top-k Retrieval from a Confidential Index," Proc. 12th Int'l Conf. Extending Database Technology: Advances in Database Technology (EDBT), 2009.