International Journal of communication and computer Technologies, ISSN: 2278-9723

Available at http://www.ijccts.org

Cloud Data Retrieval for Multi related keyword based on Clustering Technology

V.Jayasree [1], M.Nithya [2], S.Prabaharan [3]

Dept of Computer Science and Engineering[1,2,3],

1. PG Student - VMKVEC, 2. Professor - VMKVEC, 3. Asst. Professor – VMKVEC

Received: 15-07-2012, **Revised:** 02-09-2012, **Accepted:** 09-10-2012, **Published online:** 16-12-2012

Abstract

Cloud computing is a rapidly growing technology highly preferred by small and medium business where resources such as storage media, platform and applications shared over the Internet is used by multiple users in an organization. The common pay-as-you-go subscription model of Cloud computing makes it a successful technology, on the other hand, prerequisite conditions to impose confidentiality on cloud data and scalability requirement increases the complexity of cloud computing. Traditional algorithms provide either Boolean or ranked search result for the given single keyword does not retrieves accurate data on end-user's interest in searching Cloud data with multiple related keywords during the same transaction. In this paper, an efficient clustering technique is used to retrieve encrypted cloud data for multiple related keywords. Inclusion of clustering technique to group related keywords together retrieves efficient and accurate cloud data. The proposed system ranks cloud data based on end user feedback on top of existing ranking algorithms which simply relies on keyword occurrence in a document increases the accuracy of data retrieved.

Index terms – Cloud Computing, Clustering, Secured Ranked multiple related keyword search, Keyword ranked search, confidential Data.

1. INTRODUCTION

In cloud computing, small businesses can expand or shrink resources and/or services as business needs change. The common pay-as-you-go subscription model is designed to let small and medium business easily add or remove services and typical organization will only pay based on usage. The sensitive data have to be encrypted prior to outsourcing for data privacy and combating unsolicited accesses. However, data encryption makes effective data utilization a very challenging task given that there could be a large amount of outsourced data files. Besides, in Cloud Computing, data owners may share their outsourced data with a large number of users, who might want to only retrieve certain specific data files they are interested in during a given session. One of the most popular ways to do so is through keyword-based search. Such keyword search technique allows users to selectively retrieve files of interest and has been widely applied in plain text search scenarios. Unfortunately, data encryption, which restricts the user's ability to perform keyword search and further demands the protection of keyword privacy, makes the traditional plain text search methods fail for encrypted cloud data.

Ranked search can also elegantly eliminate unnecessary network traffic by sending back only the most relevant data, which is highly desirable in the "pay-as-you use" cloud paradigm. For privacy rotation, such ranking operation, however, should not leak any keyword related information. On the other hand, to improve search result accuracy as well as to enhance the user searching experience, it is also crucial for such ranking system to support multiple keywords search, as single keyword search often yields far too coarse result. As a common practice

indicated by today's web search engines (e.g., Google search), users may tend to provide a set of keywords instead of only one as the indicator of their search interest to retrieve the most relevant data. Searchable encryption schemes usually build up an index for each keyword of interest and associate the index with the files that contain the keyword. By integrating the trapdoors of keywords within the index information, effective keyword search can be realized while both file content and keyword privacy are well-preserved. Although allowing for performing searches securely and effectively, the existing searchable encryption techniques do not suit for cloud computing scenario since they support only exact keyword search.

The aim of this paper is to achieve an efficient system where any authorized user can perform a search on a remote database with multiple keywords, without revealing neither the keywords he searches for nor the contents of the documents he retrieves. Our proposed system differs from the previous works which assume that only the data owner queries the database. In contrast to previous works, our proposal facilitates that a group of users can query the database provided that they possess trapdoors for search terms that authorize the users to include them in their queries. Moreover, our proposed system is able to perform multiple keyword search in a single query and ranks the results so the user can retrieve only the top matches using score dynamics.

1.1 The Basic Scheme

In this section, we start from the review of existing searchable symmetric encryption schemes and provide the definitions and framework for our proposed ranked searchable symmetric encryption, which here helps in implementing the score dynamics of the documents that's been available in the cloud server. The cloud sever consists of indexed data using the key words of the documents in an efficient manner, which could be able to retrieve in a highly efficient manner for the users in the ranked order retrieval process.

Before giving the main result, to first start with a straightforward yet an ideal scheme, where the security of our ranked searchable encryption is the same as previous SSE schemes, i.e., the user gets the ranked results without letting cloud server learn any additional information more than the access pattern and search pattern. However, this is achieved with the tradeoff of efficiency: namely, either should the user wait for two round-trip time for each search request, or he may even lose the capability to perform top-k retrieval, resulting the unnecessary communication overhead. .

1.2 Traditional System:

Ranked search greatly enhances system usability by enabling search result relevance ranking instead of sending undifferentiated results, and further ensures the file retrieval accuracy. Ranked Searchable encryption allows data owners to outsource their data in an encrypted manner while maintaining the selectively search capability over the encrypted data. In the existing system, the documents and data are stored in a secure cloud storage whereas the documents were scanned with the keywords and an index of information were stored for future searching options. The existing system has used symmetric encryption algorithm to store the data securely.

Fig no:-1 Architecture for search over encrypted cloud data.

Design Goals

To enable ranked searchable symmetric encryption for effective utilization of outsourced and encrypted cloud

data under the aforementioned model, our system design should achieve the following security and performance guarantee. Specifically, we have the following goals:

1) Ranked keyword search: to explore different mechanisms for designing effective ranked search schemes based on the existing searchable encryption framework;

2) Security guarantees: to prevent the clouding server from learning the plaintext of either the data files or the searched keywords, and achieve the "as-strong-as-possible" security strength compared to existing searchable encryption schemes;

3) Efficiency: above goals should be achieved with minimum communication and computation overhead.

Disadvantages:

The secure searchable encryption scheme does not perform any function when new updates in files or when any modifications are performed.

The relevance score algorithm is not updated frequently when there are some modifications in the owner files.

2 RELATED WORK

Traditional searchable encryption has been widely studied as a cryptographic primitive, with a focus on security definition formalizations and efficiency improvements. So they first introduced the notion of searchable encryption. They proposed a scheme in the symmetric key setting, where each word in the file is encrypted independently under a special two-layered encryption construction. To further enhance search efficiency, a per-keyword-based approach was proposed, where a single encrypted hash table index is built for the entire file collection, with each entry consisting of the trapdoor of a keyword and an encrypted set of related file identifiers. Searchable encryption has also been considered in the public-key setting. Then the first public-key-based searchable encryption scheme construction, with the public key can write to the data stored on the server but only authorized users with the private key can search. As an attempt to enrich query predicates, conjunctive keyword search over encrypted data.

These include the following

(a) Secure searchable encryption scheme does not perform any functions when new updates in files or when any modifications are performed.

(b) The relevance score algorithm is not updated frequently when there are some modifications in the owner files.

2.1 Contributions:

In Cloud Computing, an outsourced file collection might not only be accessed but also updated frequently for various application purposes. Hence, supporting the score dynamics in the searchable index for a secure storage engine which is reflected from the corresponding file collection updates, is thus of practical importance. In our system, we consider score dynamics as adding newly encrypted scores for newly created files, or modifying old encrypted scores for modification of existing files in the file collection. Symmetric key encryption doesn't have major scope in security perspective that's why we are opting MD5 encryption algorithm which is bit more complex, when compared to the traditional algorithms in storing the data. B-Tree indexing and storing of data provides a peak level performance in searching times.

3 PROPOSED SYSTEM

In this paper, we solve the problem of supporting efficient ranked keyword search for achieving effective utilization of remotely stored encrypted data in Cloud Computing. This is done by developing an efficient clustering algorithm to group the 'related' keywords together. One-to-many order preserving technique protects the score information.

3.1 Overall description:

The scenario of the score dynamics mechanism is based on the one-to-one order preserving mapping. An efficient clustering algorithm is used to retrieve encrypted cloud data for multiple related keywords. The multiple related keywords are clustered together and ranked, the information is stored in the index which results in accurate search result when the user searches database with multiple related keywords in the same transaction. The proposed system also ranks cloud data based on end user feedback on top of existing ranking algorithms(which relies on keyword occurrence increases the accuracy of data retrieved).

3.2 Authentication function

Authentication function describes the interface between the user and system and the admin provided the type of authentication. The user is allowed to create his testimonial to login into the system. An admin needs to approve the users created and login approval the user will be allowed to access the application. Authentication is provided by encrypting the user name and password, this protects sensitive information from unauthorized users.

3.3 Document uploading process

In this process, the users are allowed to upload their documents. While uploading, the user is allowed to provide the access nature of the document. The documents will be archived with the help of rich streaming APIs which integrates the SQL Server Database Engine with an NTFS file system by storing varbinary (max) binary large object (BLOB) data as files on the file system. In addition, the user will be given the option of revoking the access at any time. Based on the provided accessibility, the documents will be accessed by the other users. Internally, the access details will be logged in the Access log file.

3.4 Document parse

In this process, the uploaded document is parsed by using the document parser interface. A mechanism with the top-down keyword parsing technique reads the complete document with specified keywords. In addition it matches the version of the content type definition that is used by a list or document library.

3.5 Document History view

The major advantage of this is related to maintaining the history of the documents. All the files that are uploaded are stored in the library. Various versions of the documents were maintained by the cloud database. The data owner will be provided an option of taking back the previous version documents. This is one of the major jargons specified in our proposed system.

3.6 Clustering algorithm

Clustering is an important application area for many fields including data mining, statistical data analysis, compression, vector quantization, and other business applications. Clustering has been formulated in various ways in the machine learning, pattern recognition, optimization and statistics literature. The fundamental clustering problem is grouping together (clustering) similar data items.

During the search process, the user has always desired to input multiple related keywords of his interest rather than a single keyword. Basically any document deal with single concept in brief and the interrelated sub-topics. Grouping the related topics together and forming cluster helps customers to get the desired document of their interest.

The most general approach is to view clustering as a density estimation problem. We assume that in addition to the observed variables for each data item, there is a hidden, unobserved variable indicating the "cluster membership". The data are assumed to arrive from a mixture model with hidden cluster identifiers. In general, a mixture model M having K clusters Ci, i=1,…,K, assigns a probability to a data point x:

$$\Pr(x \mid M) = \sum_{i=1}^{K} W_i \cdot \Pr(x \mid Ci, M)$$

where Wi are the mixture weights. The problem is estimating the parameters of the individual Ci, assuming that the number of clusters K is known. The clustering optimization problem is that of finding parameters of the individual Ci which maximize the likelihood of the database given the mixture model. For general assumptions about the distributions for each of the K clusters, the EM algorithm is a popular technique for estimating the parameters.

3.7 User feedback on Search result

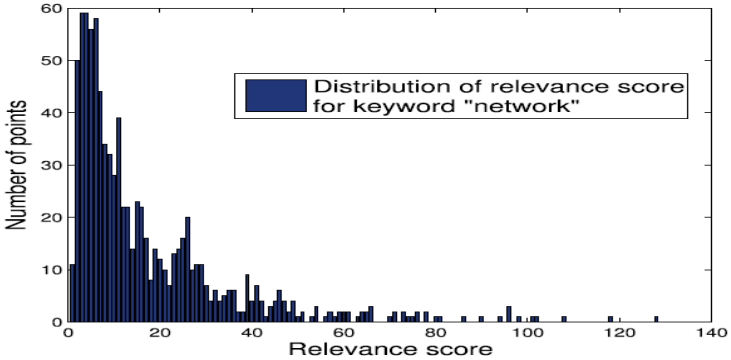
It is always advisable and it makes sense to get user feedback on the search result as the user manually reads the document 'rates' the document based on the accuracy of retrieval for the given multiple related keyword. The system displays (1) Not relevant (2) Relevant (3) Most relevant 'radio buttons' and facilitates the user to rate the documents retrieved for the particular keywords.

4. EFFICIENT RANKED SEARCHABLE SYMMETRIC ENCRYPTION SCHEME

The above straightforward approach demonstrates the core problem that causes the inefficiency of ranked searchable encryption. That is how to let server quickly perform the ranking without actually knowing the relevance scores. To effectively support ranked search over an encrypted file collection, we have now resorted to the newly developed cryptographic primitive—order preserving symmetric encryption to achieve more practical performance. Note that by resorting to OPSE, our security guarantee of RSSE is inherently weakened compared to SSE, as we now let the server know the relevant order.

4.1 Using Order Preserving Symmetric Encryption

The OPSE is a deterministic encryption scheme where the numerical ordering of the plaintexts gets preserved by the encryption function. That gives the first cryptographic study of OPSE primitive and provides a construction that is provably secure under the security framework of pseudorandom function or pseudorandom permutation. Namely, considering that any order-preserving function. An OPSE is then said to be secure if and only if an adversary has to perform a brute force search over all the possible combinations of M out of N to break the encryption scheme. If the security level is chosen to be 80 bits, then it is suggested to choose M ¼ N=2 > 80 so that the total number of combinations will be greater than 280. Their construction is based on an uncovered relationship between a random order-preserving function (which meets the above security notion) and the hypergeometric probability distribution, which will later be denoted as HGD. That refers readers to for more details about OPSE and its security definition. At the first glance, by changing the relevance score encryption from the standard indistinguishable symmetric encryption scheme to this OPSE, it seems to follow directly that efficient relevance score ranking can be achieved just like in the plaintext domain. However, as pointed out earlier, the OPSE is a deterministic encryption scheme. This inherent deterministic property, if not treated appropriately, will still leak a lot of information as any deterministic encryption scheme will do. One such information leakage is the plaintext distribution. Take Fig. 2, for example, which shows a skewed relevance score distribution of keyword



"network," sampled from 1,000 files of our test collection. For easy exposition, we encode the actual score into 128 levels in domain from 1 to 128. Due to the deterministic property, if we use OPSE directly over these sampled relevance scores, the resulting ciphertext shall share exactly the same distribution as the relevance score in Fig. 3. On the other hand, previous research works have

shown that the score distribution can be seen as keyword specific.

Actually, it can be considered as the most simplified version of searchable symmetric encryption that satisfies the nonadaptive security system.The relevance score calculation in the following presentation. Its definition is as follows:

$$score(Q, Fd) = \sum_{t \in Q} \frac{1}{|Fd|} \cdot (1 + \ln fd, t) \cdot \ln(1 + \frac{N}{ft})$$

Here, Q denotes the searched keywords; Fd,t denotes the TF of term t in file Fd; ft denotes the number of files that contain term t; N denotes the total number of files in the collection; and jFdj is the length of file Fd, obtained by counting the number of indexed terms, functioning as the normalization factor.

4.2 Searching in Cloud Repositories

The similar framework of previously proposed searchable symmetric encryption schemes and adapt the framework for our ranked searchable encryption system. A ranked searchable encryption scheme consists of four algorithms (KeyGen, BuildIndex, TrapdoorGen, SearchIndex). Our ranked searchable encryption system can be constructed from these four algorithms in two phases, Setup and Retrieval:

Setup. The data owner initializes the public and secret parameters of the system by executing Key-Gen, and pre-processes the data file collection C by using BuildIndex to generate the searchable index of the unique words extracted from C. The owner then encrypts the data file collection C, and publishes the index including the keyword frequency-based relevance scores in some encrypted form, together with the encrypted collection C to the Cloud. As part of Setup phase, the data owner also needs to distribute the necessary secret parameters (in our case, the trapdoor generation key) to a group of authorized users by employing off-the-shelf public key cryptography or more efficient primitive such as broadcast encryption.

Retrieval. The user uses TrapdoorGen to generate a secure trapdoor corresponding to his interested keyword, and submits it to the cloud server. Upon receiving the trapdoor, the cloud server will derive a list of matched file IDs and their corresponding encrypted relevance scores by searching the index via SearchIndex. The matched files should be sent back in a ranked sequence based on the relevance scores. However, the server should learn nothing or little beyond the order of the relevance scores.

4.3 Security Analysis for One-to-Many Mapping

Our one-to-many order-preserving mapping is adapted from the original OPSE, by introducing the file ID as the additional seed in the final ciphertext chosen process. Since such adaptation only functions in the final ciphertext selection process, it has nothing to do with the randomized plaintext-to-bucket mapping process in the original OPSE. In other words, the only effect of introducing file ID as the new seed is to make multiple plain text duplicates m's no longer deterministically mapped to the same ciphertext c, but instead mapped to multiple random values within the assigned bucket in range R. This helps flatten the ciphertext distribution to some extent after mapping. However, such a generic adaptation alone only works well when the number of plaintext duplicates are not large. In case there are many duplicates of plaintext m, its corresponding ciphertext distribution after mapping may still exhibit certain skewness or peaky feature of the plaintext distribution, due to the relative small size of assigned bucket selected from range R.

This is why we propose to appropriately enlarge R. Note that in the original OPSE, size R is determined just to ensure the number of different combinations between D and R is larger than 280. But from a practical perspective, properly enlarging R in our one-to-many case further aims to ensure the low duplicates (with high probability) on the ciphertext range after mapping. This inherently increases the difficulty for adversary to tell precisely which points in the range R belong to the same score in the domain D, making the order-preserving mapping as strong as possible. Note that one disadvantage of our scheme, compared to the

original OPSE, is that fixing the range size R requires preknowledge on the percentage of maximum duplicates among all the plaintexts (i.e., max=_ in (3)). However, such extra requirement can be easily met in our scenario when Building the searchable index.

5. Concluding Remarks

In this paper, initially we discussed the secured and ranked keyword search technique. Later the necessity of forming clusters to group 'related' keyword was described. Then, the power of clustering in providing accurate search result for the given multiple related keywords was highlighted. The knowledge of user utilized to rank the document derives a final efficient and accurate proposed system. By thorough analysis, we showed that our proposed solution is secure, scalable and accurate.

References

[1] 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 '10), 2010.

[2] P. Mell and T. Grance, "Draft Nist Working Definition ofCloudComputing,"http://csrc.nist.gov/groups/SNS/cloudcomputing/index.html, Jan. 2010.

[3] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M.Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," Technical Report UCB-EECS-2009-28, Univ. of California, Berkeley, Feb. 2009.

[4] Cloud Security Alliance "Security Guidance for Critical Areas of Focus in Cloud Computing," http://www.cloudsecurityalliance.org, 2009.

[5] Z. Slocum, "Your Google Docs: Soon in Search Results?" http://news.cnet.com/8301-17939_109-10357137-2.html, 2009.

[6] B. Krebs, "Payment Processor Breach May Be Largest Ever,"
http://voices.washingtonpost.com/securityfix/2009/01/payment_processor_breach_may_b.html, Jan. 2009.

[7] I.H. Witten, A. Moffat, and T.C. Bell, Managing Gigabytes: Compressing and Indexing Documents and Images. Morgan Kaufmann, May 1999.

[8] D. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," Proc. IEEE Symp. Security and Privacy, 2000.

[9] E.-J. Goh, "Secure Indexes," Technical Report 2003/216, Cryptology ePrint Archive, http://eprint.iacr.org/, 2003.

[10] D. Boneh, G.D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public Key Encryption with Keyword Search," Proc. Int'l Conf. Advances in Cryptology (EUROCRYP '04), 2004.

[11] Y.-C. Chang and M. Mitzenmacher, "Privacy Preserving Keyword Searches on Remote Encrypted Data," Proc. Int'l Conf. Applied Cryptography and Network Security (ACNS '05), 2005.

[12] R. Curtmola, J.A. Garay, S. Kamara, and R. Ostrovsky, "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions," Proc. ACM Conf. Computer and Comm. Security (CCS '06), 2006.

[13] A. Singhal, "Modern Information Retrieval: A Brief Overview," IEEE Data Eng. Bull., vol. 24, no. 4, pp. 35-43, 2001.