American Journal of Applied Sciences 11 (6): 906-911, 2014
ISSN: 1546-9239

doi:10.3844/ajassp.2014.906.911 Published Online 11 (6) 2014 (http://www.thescipub.com/ajas.toc)

EFFICIENT RANKED AND SECURE FILE RETRIEVAL IN CLOUD COMPUTING

Jospin Jeya, J. and E. Kannan

Vel Tech Technical University, Avadi, Chennai, India

Received 2013-12-30; Revised 2014-01-02; Accepted 2014-04-03

ABSTRACT

Cloud computing facilitates extremely scalable services that can be consumed over internet. An important aspect of the cloud services is that user data are stored remotely in unknown machines in which users do not posses or manage. Since the data's are stored remotely, we have to keep in mind that sensitive cloud data have to be encrypted before they are outsourced to the commercial public cloud, which makes efficient data utilization service. Searchable encryption file retrieval technique allows users to securely search over encrypted data through search word. Ranking the files based on relevance scores greatly enhances system usability by making it possible relevance ranking instead of sending unwanted results and further ensures the file retrieval accuracy. In this study, we are developing an automated system for both named and un-named documents based on the clustering algorithms. We implement the ranking and searching algorithm to retrieve top k files. We also provides the mapping and encryption algorithm to protect the information. The resulting design is able to provide efficient ranking which will reduce the search time drastically and reduce the communication overhead. The mapping and encryption algorithms protect document against an outside attackers and prevent an untrusted cloud data provider from learning data.

Keywords: Index Terms-Ranked Search, File Retrieval, Search Word, Cloud Computing

1. INTRODUCTION

Cloud computing is a new computing prototype that is built on virtualization, parallel and distributed computing, utility computing and service oriented architecture. For the past many years, cloud computing has emerged as one of the most important and used technology in the IT industry and academic world. The benefits of cloud computing include reduction in costs and in capital expenditures, increased operational efficiencies, scalability, flexibility, immediate time to market and many more. The various cloud computing models are Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). It is a subscription-based service, where in one can obtain networked storage space and computer resources. According to the different needs of the user, one can subscribe different types of cloud. Clouds are classified as (Armbrust *et al*., 2009; Boneh *et al*., 2004; Chang and Mitzenmacher, 2005; CSA, 2009; Curtmola *et al*., 2006; Goh, 2003; Krebs, 2009).

1.1. Public Cloud

A public cloud can be used by any user with an internet connection and access to the cloud space.

1.2. Private Cloud

A private cloud is established for a specific group or organization and access to this cloud is limited just to at group or organization alone.

1.3. Community Cloud

A community cloud is shared between two or more organizations which have common cloud requirements.

1.4. Hybrid Cloud

A hybrid cloud is fundamentally a combination of at least two clouds, where the clouds included are a blend of public, private, or community.

Corresponding Author: Jospin Jeya, Research Scholar, Vel Tech Technical University, Avadi, Chennai, India

As Cloud Computing becomes common, more and more sensitive information are being centralized into the cloud, such as e-mails, personal health records, banking information, company finance data, government documents, etc. The fact that data owners and cloud server are no longer in the same trusted domain may put the outsourced unencrypted data at risk. To mention a few, some of the risks are that the cloud server may leak data information to unauthorized entities or sometimes even the cloud server might be hacked. Therefore, to prevent data privacy and to combat unsolicited accesses, it is necessary that sensitive data have to be encrypted prior to outsourcing. However, such data encryptions render effective data utilization a very challenging task due to the basic reason that there could be a large amount of outsourced data files. Not only that, many a times, in Cloud Computing, data owners may share their complete outsourced data with a large number of users, who actually might want to retrieve only certain specific data files that they are interested in during a given session (Ming *et al*., 2011; Murugesan, 2011; Huang *et al*., 2013; Song *et al*., 2000; Wang *et al*., 2010; 2012; Witten *et al*., 1999; Yu *et al*., 2013).

One of the most commonly used methods to do is through search word based search. This technique permits the user to selectively retrieve files of importance and has been widely used in plaintext search scenarios. Unfortunately, as data encryption restricts user's ability to perform search word search and also stress the security of search word privacy, it makes the conventional plaintext method fail. When this method is straightly implemented in huge collaborative data outsourcing cloud environment, they may suffer from the following drawbacks. On the one hand for each search request, users without pre knowledge of the file have to go every retrieved file, in they have to identify ones that are most identical to their importance. Because of this, huge amount of post processing overhead is there. On the other hand, invariably sending back all files leads to large unnecessary network traffic. Secure and efficient file retrieval method greatly enhances system usability by sending the corresponding file in a ranked order regarding to some scores (Search word frequency).

Our contribution can be described as follows:

- We give the problem statement of efficient ranked and secure file retrieval in cloud computing over encrypted data
- We provide the clustering algorithm to cluster the related documents, which reduces the time to calculate the relevance score for each search word
- Binary search tree algorithm is used for the index storage, which reduces the search time and modifications can be easily done
- Experimental results shows the secure and efficiency of the proposed solution

The remaining part of the paper is summarized as follows: Section 2 describes the problem statement, section 3 provides the clustering and build index algorithm, section 4 describes the file performance of the efficient ranked and secure file retrieval in cloud computing.

2. PROBLEM STATEMENT

We take into consideration an encrypted file hosting service involves three different units, as shown in **Fig. 1**: Data owner, data user and cloud server. Data owner has a collection of n data file C = (F1, F2,….Fn) which he desires to outsource on the cloud server in an encrypted form. The data owner will first build searchable index I before outsourcing. The unique search words W = (w1, w2,…wn) are identified from the file collection C. The searchable index and the file collection C are stored on the cloud server in an encrypted form. If the authorized user wishes to retrieve file with the particular search word w, the authorized users prepares and send a search request. The search request contains the encrypted search word of the keyword to the cloud server. The cloud server receives the search request and uses the search Index algorithm to return the corresponding set of files to the authorized user. The search result should be returned based on some ranked scores, to reduce the communication overhead .The cloud server learn nothing since the search word and the files are encrypted. Bandwidth can be reduced by sending an optional value k along with the request by the user. The cloud server then sends only the top-k most related files to the authorized user for the interested search word.

We are opting MD5 encryption algorithm for authentication which is bit more complex when compared to the traditional algorithms in storing the data but it is low level of hacking because of 32 bit encryption. Binary Search algorithm is more traditional but it does not meet the efficiency than Binary search Tree. Hence We Suggest for Binary search tree searching algorithm. Binary search Tree indexing and storing of data provides a peak level performance in searching times. Hence, our Design Goals supports 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.

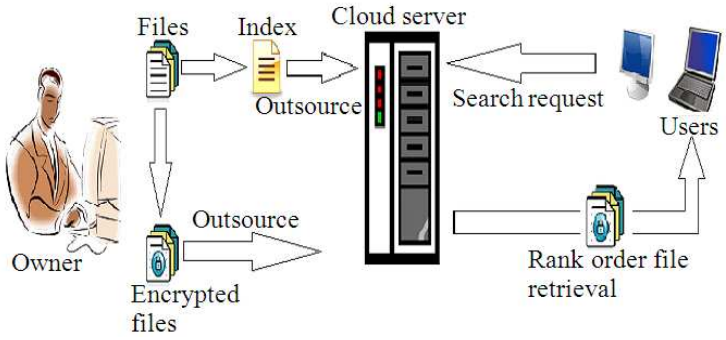

Fig. 1. Architecture for secure file retrieval in cloud computing

Thus, 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.

2.1. Basic Scheme

A secure ranked file retrieval scheme is inclusive of four algorithms (Key generation, encryption algorithm, build index, search word). A secure ranked file retrieval scheme can be implanted by these four algorithms in two steps, setup and retrieval.

2.1.1. Setup

The owner executes the key generation algorithm and gets the public and secret parameter. Then the owner generates the secure searchable index using Build Index algorithm from the distinct words identified from the collection of file C. The data owner then encrypts the file collection C and gives it to the cloud server along with the secure searchable index. The data owner then sends the necessary secret keys and the search word list to a group of authorized users in a secure channel.

2.1.2. Retrieval

The authorized user generates the secure request for the interested search word and gives it to the cloud server. After receiving the request, the cloud server uses the search word algorithm to get the matched file IDs. If the match occurs, the cloud server retrieves the file IDs of the most relevant files to a search word instead of retrieving all the documents relevant to a search word. The ranked order retrieved files then sent back to the user. In this study we concentrate on single word search. Since the score is always same for the given search word, the result is always accurate.

3. K-MEANS CLUSTERING ALGORITHM

The K-means clustering algorithm is the efficient algorithm for large data sets. This algorithm divide the set of objects, according to their attributes or features into K-clusters. K is the user defined constant. The main aim is to identify k centroids, one for every cluster. The centroid of a cluster is identified in such a way that it is strongly connected to all objects in the cluster.

3.1. K-means Algorithm

1. Decide K number of clusters.
2. Decide K objects arbitrarily as the initial cluster center.
3. Repeat
   - 3.1 Allocate each object to their nearby cluster.
   - 3.2 Compute new clusters.
4. Until
   - 4.1 No changes in the cluster centers or
   - 4.2 No object changes its cluster.

4. DOCUMENT RANKING

One of the most commonly used word weighting schemes TF-IDF assigns weights to each word using word frequency and inverse document frequency (idf). The word frequency of the word denotes the no of times the given word appears in a document. The inverse document frequency is the total no of documents in a collection containing the word with respect to the total no of documents in a collection.

$$\text{Score}\left(q,d_i\right)=\sum_{t\in q} t_f - idf_{t,i}$$

$$\text{Score}\left(q,d_i\right)=tf_{it}.\log N / N_t$$

Where:

tf, idf_{ti} = The word weight of the word in the document
d_i, tf_i = The word frequency of the word in the document d_i
N = The total number of documents in a collection
N_t = He number of documents in a collection containing the word.

4.1. Document Ranking Algorithm

This Algorithm takes one keyword at a time. For each keyword, it computes the partial rank/score for each document in the collection C. The final score/rank of each document in the collection C is computed when all the keywords are processed

Require: Query q and document collection C.
1: Score [N] = 0:0
2: For all term term t in query q do
3: for all document di in collection C do
4: $w_{it} = tf_{it}$: LogN/N_t
5: Score[i] = Score[i] + w_{it}
6: end for
7: End for
8: R = SORT(C; Score [])
9: return R

5. INVERTED INDEX

The index structure is used to store the search word. The size of the index is reduced by using various IR techniques. Then binary search tree technique is used to store and search the search word.

5.1. Build Index (K,C)

1. Initialization
   Examine C and take out the unique words W from each cluster.
2. Build index list
   For 1≤i≤ |f(wi)| in each cluster
   a) Identify the score for the file fij
3. Secure the index list

Where:

C = Collection of files to be outsourced.
W = The distinct keywords extracted from file collection C.
|f(wi)| = The no of files containing the keyword wi.
f(wi) = The set of identifiers of files containing the keyword wi.

Table 1 shows the one example of index structure posted by the owner for a search word.

Table 1. Structure of index for a search word

Search word	w_i (represented as unique no)			
File ID	F_{i1}	F_{i2}	…	F_{in}
Score	4.75	12.6		2.5

5.2. Search Word Mapping Algorithm

This algorithm takes a search word w contains n number of characters c1,c2,c3…cn ,each character is represented by one byte an d returns number in the range of 0-255.

An table T of 256 random bytes permutation of the values 0-255 is used in this algorithm.

5.3. Algorithm

1. h [0] = 0;
2. For each character in the search word
   a. Compute h[i] = T [h[i-1] XOR c[i]]
3. End loop.
4. Return the value of h[n].

6. ENCRYPTION ALGORITHM

The algorithm used to encrypt the file is symmetric algorithm. This algorithm is used to encrypt the file using the key, which will produce the cipher text. The cipher text is actually transmitted to the cloud server. An encryption algorithm also gives the decryption algorithm, which will give the information to receiver how to decrypt the plaintext. The key generation algorithm is used to produce the key that the owner and the receiver need to share. We have analyzed the performance of various encryption algorithm which is suitable for cloud computing.

6.1. Authentication

The authentication between the owner and the user is done with MD5 message-digest algorithm. The MD5 message-digest algorithm is a cryptographic hash function that makes a 128-bit (16-byte) hash value.MD5 processes a variable-length message into a fixed-length output of 128 bits. The input message is broken up into chunks of 512-bit blocks (sixteen 32-bit words); the message is padded so that its length is divisible by 512. The padding works as follows: First a single bit, 1, is appended to the end of the message. This is followed by as many zeros as are required to bring the length of the message up to 64 bits less than a multiple of 512. The remaining bits are filled up with 64 bits representing the length of the original message, modulo 264.

7. INDEX SEARCH

The search word is mapped to a unique number using the mapping algorithm. The index with a search word, file ids with relevance score is sent to the cloud server. Binary search tree data structure is used by the cloud server to store the index.

For the required search word, the user sent a request with a mapped keyword and k. Upon receiving the request the server uses binary search tree algorithm to locate the corresponding match. If the match occurs it retrieves top k file ids and sent the corresponding encrypted file.

8. RESULT AND PERFORMANCE ANALYSIS

The scores for each file for a particular search word is stored in addition to the file ID. The cloud server can only use the score which will not disclose any useful information. Due to the power of encryption algorithm, the file is protected well. Since the keyword is also mapped to unique number the cloud provider has no knowledge about the data.

8.1. Construction of Index

In the existing system the index is constructed by scanning all the files. As we have used the clustering algorithm to group the related documents, the time taken to calculate the relative score for each search word is drastically reduced. i.e., we need not scan all the files to calculate the relevance score for each search word. Since the search word is represented as the number binary search tree algorithm is used to store the search word list in the server and the search time to identify the search word in the index is reduced.

The **Table 2** shows the index construction performance for a set of 500 files.

8.2. Efficiency of Search

Since we are using binary search tree data structure, the server need not scan every index for each search word. So, the overall search time is greatly reduced. The file ids are arranged based on the relevance score; it is very easy to retrieve top k file ids.

Table 2. Index construction performance

No of files	Per search word list size	Construction time
500	10.4 KB	2.7s

9. CONCLUSION

In this study we provide the efficient solution for retrieving remotely stored encrypted file in large scale file collections. We investigate techniques to rank order the files and retrieve most relevant files from an encrypted collection based on the user request. The projected method keeps the confidentiality of the request as well as the retrieved file. This scheme provides less communication overhead since only the top k relevant files are retrieved. This results in significant amount of bandwidth being saved. Since the file is encrypted and the search word is mapped the cloud server has no knowledge about the data.

Securing communication links, combating traffic analysis and developing a secure auditing protocol to verify whether the data are correctly stored in the cloud, will be our future work.

10. REFERENCES

Armbrust, M., A. Fox, R. Griffith, A.D. Joseph and R.H. Katz *et al*., 2009. Above the clouds: A berkeley view of cloud computing. University of California, Berkeley.

Boneh, D., G.D. Crescenzo, R. Ostrovsky and G. Persiano, 2004. Public key encryption with keyword search. Proceedings of the International Conference on Advances in Cryptology, May 2-6, Interlaken, Switzerland, pp: 506-522. DOI: 10.1007/978-3-540-24676-3_30

Chang, Y.C. and M. Mitzenmacher, 2005. Privacy preserving keyword searches on remote encrypted data. Proceedings of the 3rd International Conference on Applied Cryptography and Network Security, Jun. 7-10, New York, NY, USA, pp: 442-455. DOI: 10.1007/11496137_30

CSA, 2009. Security guidance for critical areas of focus in cloud computing. Cloud Security Alliance.

Curtmola, R., J. Garay, S. Kamara and R. Ostrovsky, 2006. Searchable symmetric encryption: Improved definitions and efficient constructions. Proceedings of the 13th ACM Conference on Computer and communications security, Oct. 30-Nov. 03, ACM Press, New York, NY, USA, pp: 79-88. DOI: 10.1145/1180405.1180417

Goh, E.J., 2003. Secure indexes. Cryptology ePrint Archive.

Krebs, B., 2009. Payment processor breach may be largest ever. Washington Post.

Ming, L., S. Yu, N. Cao and W. Lou, 2011. Authorized private keyword search over encrypted data in cloud computing. Proceedings of the 31st International Conference on Distributed Computing System, Jun. 20-24, IEEE Xplore Press, Minneapolis, MN., pp: 383-392. DOI: 10.1109/ICDCS.2011.55

Murugesan, K., 2011. Cluster-based term weighting and document ranking models. MSc Thesis, University of Kentucky.

Huang, R., G. Yu, Z. Wang, J. Zhang and L. Shi, 2013. Dirichlet process mixture model for document clustering with feature partition. IEEE Trans. Knowl. Data Eng., 25: 1748-1759. 10.1109/TKDE.2012.27

Song, D.X., D. Wagner and A. Perrig, 2000. Practical techniques for searches on encrypted data. Proceedings of the IEEE Symposium on Security and Privacy, May 14-17, IEEE Xplore Press, Washington, DC., pp: 44-55. DOI: 10.1109/SECPRI.2000.848445

Wang, C., N. Cao, J. Li, K. Ren and W. Lou, 2010. Secure ranked keyword search over encrypted cloud data. Proceedings of the IEEE 30th International Conference Distributed Computing Systems, Jun. 21-25, IEEE Xplore Press, Genoa, Italy, pp: 253-262. DOI: 10.1109/ICDCS.2010.34

Wang, C., N. Cao, K. Ren and W. Lou, 2012. Enabling secure and efficient ranked keyword search over outsourced cloud data. IEEE Trans. Parallel Distrib. Syst., 23: 1467-1479. DOI: 10.1109/TPDS.2011.282

Witten, I.H., A. Moffat and T.C. Bell, 1999. Managing Gigabytes: Compressing and Indexing Documents and Images. 1st Edn., Morgan Kaufmann, San Francisco, ISBN-10: 1558605703, pp: 519.

Yu, J., P. Lu, Y. Zhu and G. Xue, 2013. Toward secure multikeyword top-k retrieval over encrypted cloud data. IEEE Trans. Dependable Secure Comput., 10: 239-250. DOI: 10.1109/TDSC.2013.9