Middle-East Journal of Scientific Research 23 (6): 1053-1058, 2015
ISSN 1990-9233
© IDOSI Publications, 2015
DOI: 10.5829/idosi.mejsr.2015.23.06.22196

Efficient Data Storage and Retrieval in Cloud Environment Using Cuckoo Hashing and Latent Semantic Search

[1]Ashok George and [2]A. Sumathi

[1]Department of CSE, Anna University, India
[2]Department of ECE, Adhiyamaan College of Engineering,
Tamil Nadu, India

Abstract: In Cloud Computing, clients usually reduce their management cost by outsource their data to the cloud storage. To protect sensitive data on the cloud, Encryption on the outsourced data is a promising way, but it also introduce much difficulty to performing effective searches over encrypted information, which makes the search technology on plaintext unusable. In this paper, proposed a method latent semantic search over encrupted cloud data with cuckoo hashing technique. For the encrypted outsourced data, construct a searchable index by using cuckoo hashing technique. The basic idea is to use two hash function to provide each key two possible location in the hash table. The hash table is split into two tables of equal size. The two tables are indexd by using any one of the hash function. The search operation consists of the computation of the position corresponding to the queried word, which reduce the query response time.In Cloud, data user enter the query keyword,which is searched using the Latent semantic search technique, it return matching files including the latent semantic associated to the query keyword files by a mathematical technique called Singular value decomposition (SVD), which is used to reveal relationship between terms and documents and adopts reduced-dimension vector space to represent words and documents. Thus the relationship between terms is automatically capturedandby using the secret key the data user decrypt all the encrypted files. Thus, the evaluation of our proposed scheme reduces the query response timeandalso improves search efficiency.

Key words: Cloud Computing · Cuckoo Hashing · Latent Semantic Search (LSS) · Encryption · Indexing

INTRODUCTION

Due to massive growth of data, the Data owners tend to outsource their data in the cloud with the advantage of reducing cost and assuring availability. The outsourced data in the cloud may be under explored to the risk, so it is necessary for data owner to protect data from unsolicited access, which is achieved by utlizing any one of the proprietary encryption algorithm. The another important issue in cloud is efficient data retrievel, to overcome this problem several solutions have been proposed to allow the search of keywords over encrypted data which are not efficeint. In this paper, this problem is addressedby using "Cuckoo Hashing" where the data owner constructs a searchable index with all words listed in its files, cuckoo hashing done on the index file where it helps to assigning one word to a unique position in the index, easily to insert or delete a data in the index and to removes the collision in the index and improves the query efficiency by computation of the position corresponding to the query word. The Data Owner outsources the encrypted data with cuckoo hash index to the cloud server.

The another scheme utilized is"Latent Semantic Search"have been proposed to make data user easy to search, where in another search methods the exact matching keyword files will be retrieved. But in a large body of text the latent contextual-meaning have been exposed by using the method of LSS. It understands the meaning of the keyword entered by the data user and finds the relevant files by understanding the contents. In a set of documents looking at the keyword usage this concept have been done.

The data user enters the keyword to search over encrypted cloud data with cuckoo hashing index file, but not only it returns matching files including the latent semantically associated to the query keywordfiles. For example, Data user enter the keyword "motor" to

Corresponding Author: Ashok George, Department of CSE, Anna University, India.

search, the proposed scheme not only return the files where the "motor" keyword is associated but also the files including the term "bike". After receiving all the requested files from the cloud server, Data user decrypt all the encrypted files using same secrete key before accessing it.

This paper constructan efficient indexing and searching, which increasesthe speed of access over encrypted data from cloud server.The remaining section of the paper is organized as follows: Section 2 describes recent related works about Cuckoo hasing and Latent semantic analysis. Section 3 describe sytem design with needed diagrams and notations, The proposed scheme is described in Section 4. Simulation results and analysis of the proposed methodology are discussed in section 5. Finally, Section 6 renders the conclusions.

Related Work: In [1] proposed an approach to construct a practical history independent dynamic dictionary using the concept cuckoo hashing, the memory representation at any point in time yields no information on the specific sequence of insertions and deletions that led to its current content, other than the content itself. It prevents from leakage of information and each set of elements has a unique memory representation**.**

In [2] proposed an approachuse similarity measure of "coordinate matching" to capture the query keyword relevant files. To measure the score of each file the inner product similarity method is used. The search is flexible by supporting the exact multi-keyword ranked search.

In [3] proposed an approachit present "Dictionary based fuzzy set Construction", in which each keyword is corresponding with much less fuzzy keywords. The index size and storage is reduced and communication overheads are great improvements.

In [4] proposed an approachto combines inverted index with order-preserving symmetric encryption (OPSE). By using the numerical relevance scores which can be calculated by TF×IDF, the order of retrieved files is determined. It saves communication overhead and enhances system usability. This solution only supports single keyword ranked search.

In [5] proposed an approach to build a private trie-traverse searching index by employing wildcard-based fuzzy set. It returns the corresponding files by the condition, if the predetermined set value is greater than the the edit distance between retrieval keywords and ones from the fuzzy sets in the searching phase. This method support format inconsistence and tolerance of minor typos, but do not support semantic fuzzy search.

In [6] proposed an approach and it use the nearest neighbour search technique to find Euclidean distance between two vectors to improve the search accuracy of the ranked search.

In [7] proposed an approachfor a word search protocol and it builds by using the techniques keyed hash functions and cuckoo hashing for the outsourced data and by using these techniques a searchable index is constructed and In the cloud server, the information about the data is not revealed by the word search queries and it is guaranted by using the private information retrieval of short information. And it includes the delegation and the revocation capabilities by using the Attribute based encryption and Oblivious pseudo random function

In [8] proposed an approachwere multi keyword ranked search using latent semantic analysis(LSA). LSA returns not only the matching files but also including the terms latent semantic associated with the query keyword with the secure search functionality by employing the scheme "k-nearest neighbour(k-NN)"

In [9] proposed an approach to locate the optimal centroids of the clusters by using the Cuckoo Search Clustering Algorithm based on Levy Flight by using the bench mark data set and the solution obtained is well in web document clustering.

In [10] proposed an approachto combine a tree-based index structure and various adaption methods for multi-dimensional (MD) algorithm to improve search efficiency. Two secure index schemes namely cipher text model and background model are used for strict privacy requirements under strong threat models.

In [11] proposed an approach for automatic categorization of web pages by improving the Cuckoo Search Clustering Algorithm based on Levy Flight. The solution is well performed by obtaining the relavent information in the web pages.

Three key features like key confidentiality, key sharing and key authenticationplays a vital role for secure data sharing in the cloud environment [12]. These three key features are obtained by implementing the Dynamic Group Key Protocol which relies on the Key Generation Center (KCG). This KCG is used to achieve the anonymous access control over the cloud environment.

In military environment the mobile nodes suffers with the periodic network connectivity and the Disruption-tolerant network (DTN) technology provides the communication service [13] in between the wireless devices. Thus for secure communication Ciphertext-Policy attribute- based encryption (CP-ABE) technique is utilized to encrypt the communication message.

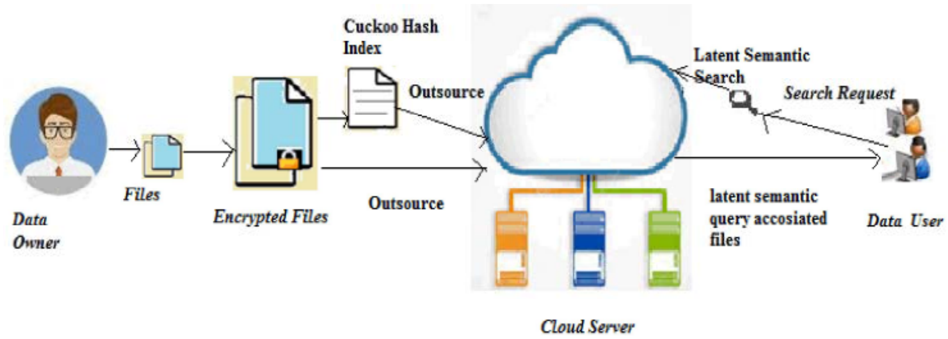

Fig. 1: Architecture of Cuckoo Hashing with Latent Semantic Search Over Encrypted Cloud Data

System Design

Architecture for Data Access Over Encrypted Cloud Data: The framework for proposed scheme is shown in the Figure 1. As shown in the figure, a cloud data system consist of three entities, the Data Owner, the cloud Server and Data user. The Data Owner has a collection of File (F) and Data Owner creates the File Identifier (f_{id}) to identify the each file uniquelyand a set of different keywords ω_1, ω_2,…., ω_n is extracted from the data collection File and the list L_ω of distict keywords is constructed from the File(F). The Data Owner encrypts the File (F) by using any proprietary encryption algorithm and the Index (I) created by using the Cuckoo Hashing Technique, where it helps to assigning one word to a unique position in the index, easily to insert or delete a data in the index and to removes the collision in the index and improves the query efficiency by computation of the position corresponding to the query word.TheData Owner upload both the cuckoo hashing Index (I) and the encrypted data collection (F) to the Cloud Server.

The Latent Semantic Search is used for searching purpose, which adopts the concept of singular value decomposition method and to extract the files it computes thequery-documentvectorcosine similarities. The document coordinates for each file is computed using Latent semantic Index and the Index File (IF) is constructed with document coordinates with the file identifier (f_{id}). When the Data User enters the t_u keywords to search over the cloud server. The keyword matched file is indexed by using the cuckoo hashing index after the Latent Semantic Search is applied to extract the matching files in the File (F) and latent semantic associated with the query keyword files in the File (F) and returns the extracted files to the data user. By using the secret key the data user decrypts the file.

Notations and Preliminaries: The notations used in this paper are listed below:

O - Data Owner who outsource their file to the Cloud Server

F - Data Owner has a collection of plain text documents F = $\{f_1,f_2,f_3….f_n\}$

f_{id} - UniqueFile Identifier for each file associated with F

L_ω - A set of different keywords L_ω= (ω_1, ω_2, ….,ω_n) is extracted from the data collection F

I - The Data Owner construct index table by using the scheme Cuckoo Hashing Index for the encrypted data

t_u - The t keywords entered by the data user

H_{mac} - Message attribute code

K_{mac} - Master key

K_{enc} - Secret key

Cuckoo Hashing Analysis: Cuckoo Hashing is a hashing technique for solving hash collision with worst case constant deletion and look-up time.Thestorage requirement is minimized and the insertion time is amortized constant. It is easy to implement and efficient both in practice and theory with rare difference.

In Cuckoo hashing two hash Tables T_1 and T_2 of the same size are used and indexing done by using the two hash function h_1 and h_2 respectively.

To insert an element x in Table (T_1) by using the hash function h_1. If some other element y stored in that location, y must be expelled (“cuckoo” hashing). We insert y in other valid location Table (T_2) by using the hash function (h_2). If another element z is occupied on that position the cuckoo hashing continued like this until it find an vacant position and the process finishes, or it cannot find a vacant position and this procedure can stop and rehash with new hash functions(table size is increased) and it shows that insert process take constant time (on average).

To search an element x in Cuckoo hashing[8], it can exist at any one of the location, but it cannot exist on two locations. The element x can be in Table (T_1) at position $h_1(x)$ or in Table (T_2) at position $h_2(x)$). In constant time we can check on both locations.

To delete an element x, the two possible locations are checked and it can delete if the element is find.

Latent Semantic Search: For discovering the latent semantic relationship, The proposed scheme uses the concept Latent Semantic Analysis (LSA). The semantic structure between the words and the files, adopts the concept of Singular Value Decomposition (SVD) [6].

The term-document matrix consist of rows and each row represent the data vector for each file

$$A' = (A'[1] \ldots\ldots\ldots A'[m] = \begin{bmatrix} tf_{t,1} & \cdots & tf_{t,m} \\ \vdots & \ddots & \vdots \\ tf_{n,1} & \cdots & tf_{n,m} \end{bmatrix} \quad (1)$$

$$A'_{nm} = X'_{nt} \times S'_{tt} \times Y'_{tm} \quad (2)$$

where A' is the decomposed into the product of three other matrices. X' and Y' is the orthonormal column and S' is the diagonal. Order the size of singular values in S', the first rank k largest may be kept and set zero to all other remaining smaller ones.

$$A_{nm} = X'_{nk} \times S'_{kk} \times Y'_{km} = A' \quad (3)$$

where matrix A is the product of resulting matrices and is the matrix of rank k.

Proposed Methodology: In the proposed framework, The Data owner wants to upload a large file F to cloud server S with the advantage of cost and assuring availability. Before outsourcing data, the data owner encrypts file F by using any proprietary encryption algorithm and owner builds the index for the distinct words in the file.

The Data Owner creates unique file identifier (f_{id}) for each File (F). The encryption is done by semantically secure encryption C=Enc ($k_{enc,}$ F) where k_{enc} is the secret key and F is the file.

The Message attribute code MACs $L_H=\{h_1, h_2,\ldots,h_n\}$ is buildusing the equation 4

$$h_i = H_{mac}(k_{mac}, \omega_i \parallel f_{id}) \quad (4)$$

where k_{mac} is master key, ω_i is the distinct words in the List $L_\omega = \{\omega_1,\omega_2,\ldots\ldots \omega_n\}$ and f_{id} is the file identifier. L_H is split into two tables with equal size T_1 and T_2 and build two hash functions H_1 and H_2

Index I is build using the T_1, T_2, H_1 and H_2 and now the Data Owner Outsource the Encrypted Files (F) with the Index (I).

By using the latent semantic search the document vectorisfind for each file and a separate Index File (IF)consisting of file identifier and the document vector coordinates of each file is maintained.

Running Queries: When the data user search for a keywordt_w.The cloud server called theToken algorithm to computes the MAC $h=H_{mac}(k_{mac}, \omega\parallel F_{id})$. Then the algorithm query runs to compute the h1(h)=(x,y) and h2(h)=(x1,y1) then potential position of h in the tableT1 or T2 is find. Then the matching keyword files are retrieved after indexing.

For the files retrieved the index file IF is mapped using the file Identifier (f_{id}) and retrieving the document vectors coordinates for the mapped files. The queries are formed into pseudo-documents to form a matrix q. The new query vector coordinates in the reduced 2-dimensional space is find by using the equation 5.

$$q = q^T X_k S_k^{-1} \quad (5)$$

where q is a querey matrix, X is orthonormal column of rank k and S is the diagonal S of rank k.

The query-document cosine similarities is computed by the formula

$$sim(q,d) = \frac{q * d}{|q||d|} \quad (6)$$

where q is the query vector find by equation 5 and the d is the document vector find by equation 3

The documents will be ranked in descending order by using the scores computed by the equation 6.

Performance Analysis

Simulation Setup*:* The 7 secter benchmark data set is used to find the F-Measure for our proposed scheme.

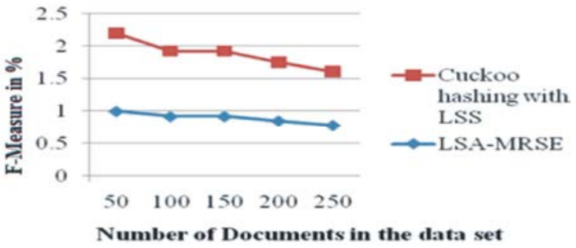

Fig. 2: F-Measure for Cuckoo hashing with Latent semantic search

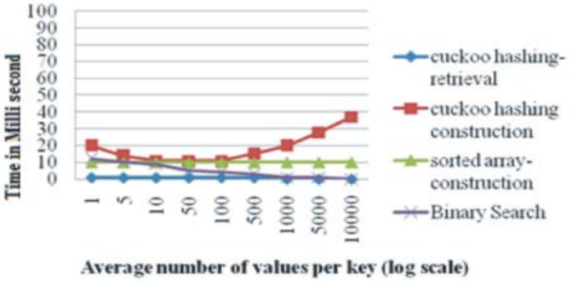

Fig. 3: Construction and retrieval times of cuckoo hashing and sorted array

In the data set, 300 webpages are randomly selected to index the related document by using cuckoo hashing Index and after by using the cosine similarity value the latent semantically associated documents are retrieved by using our proposed scheme and the results are showed in the Figure 2.

Performance and Security Analysis: The Performance of the proposed method is compared with the LSA-MRSE scheme

F-Measure: Precision is the fraction of retrieved document that are relevant

$$Precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|} \tag{7}$$

Recall is the fraction of relevant documents that are retrieved.

$$recall = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|} \tag{8}$$

F-measure that combines precision and recall is the harmonic mean of precision and reca

$$F = \frac{2 \times precision \times recall}{precision + recall} \tag{9}$$

In F-measure, The proposed scheme achieves score higher than the original LSA-MRSE. Since the original scheme employ to retrieve the relevant files with less time efficiency when compared to the proposed scheme. The query efficiency is achieved by cuckoo hashing and retrieve the most relevant file by Latent semantic search and it yields time efficiency with search efficiency. Figure. 2 shows that scheme achieves remarkable result.

Figure. 3 shows that cuckoo hashing construction takes time more to compute the hash functions and store the key value in anyone of the table. For retrieval process the worst cases lookup is O(1), where the key values are searched in any one of the hash table and it make constant time in retrieval process. In many other hash table algorithm the worst-case bound on the time to do a lookup is not constant which is contrast to cuckoo hashing.

CONCLUSION

In this paper, Cuckoo hashing with Latent Semantic Search scheme over encrypted cloud data is proposed. By using the cuckoo hashing index the Index (I) is constructed which improves the query efficiency and the encrypted file with Index (I) is outsorcedconsidering the security and privacy. In addition,searching made by Latent Semantic Search over the indexed File(IF) to retrieve the most relaventfile, which improves the search efficiency with least possible time.The proposed scheme is to construct efficient indexing and increase speed of access over encrypted data from cloud server.

REFERENCES

1. Mony Naor, Gil Segav and Udi Weider, 2008. History – Independent Cuckoo Hashing, 35th International Colloquium on Automata, Languages and Programming (ICALP).
2. Cong Wang, Ning Cao, Jin Li, Kui Ren and Wenjing Lou, 2011. Secure ranked keyword search over encrypted cloud data, Distributed Computing Systems (ICDCS), IEEE.
3. Chang Liu, Liehuang Zhu, Longyijia Li and Yuan Tan, 2011. Fuzzy Keyword Search on Encrypted Cloud Storage Data with small index, CCIS, IEEE.
4. Ning Cao, Cong Wang, Ming Li and Kui Ren, 2011. Privacy-preserving multi-keyword ranked search over encrypted cloud data,INFOCOM, IEEE.

5. Wang, C., Kui Ren, Shucheng Yu and K.M.R. Urs, 2012. Achieving usable and privacy-assured similarity search over outsourced cloud data, INFOCOM, IEEE.
6. Deepa, P.L., S. Vinoth Kumar and S. Karthik, 2013. Improving Search Efficiency of Encrypted Cloud, International Journal of Scientific & Engineering Research (IJSER).
7. Elkhiyaoui Kaoutar, Onen Melek and Molva Refik, 2013. Privacy Preserving Delegated Word Search in the Cloud, EURICOM.
8. Li Chen, Zhihua Xia, Xingming Sun and Jin Wang, 2013. An Efficient and Privacy-Preserving Semantic Multi-Keyword Ranked Search over Encrypted Cloud Data, Advanced Science And Technology Letters.
9. Moe Moe Zaw and EiEi Mon, 2013. Web Document Clustering Using Cuckoo Search Clustering Algorithm based on Levy Flight, IJIAS.
10. Wenhai Sun, Bing Wang, Ning Cao, Ming Li, Wenjing Lou, Y. Thomas Hou and Hui Li, 2013. Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking, ACM SIGSAC.
11. Moe Moe Zaw and Ei Eimon, 2014. A Text-based Web Document Clustering System by using Improved Cuckoo Search Clustering system based on Levy Flight, ICAET.
12. Dharani, R. and M. Narmatha, 2014. Secured Data Sharing with Traceability in Cloud Environment, International Journal of Inventions in Computer Science and Engineering, 1(8).
13. Bhuvaneshwaran, B. and A. Vijay, 2015. Distribution of Secured Data Retrieval using Efficient Tolerant Military Network, Journal of Recent Research in Engineering and Technology, 2(2).