International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 3 (2015) pp. 7649-7658

http://www.ripublication.com

Privacy Preserving Remote Data Retrieval In Cloud Computing

NirmalaVairaganthan[1] and Shanmugalakshmi R[2]

[1]Research Scholar, Department of CSE, Government College of Technology, Coimbatore-641013, Tamil Nadu, India.
[2]Associate Professor, Department of CSE, Government College of Technology, Coimbatore-641013, Tamil Nadu, India.

ABSTRACT

The access of data stored in cloud computing by ensuring privacy of data becomes critical issue not only for the static data but also for the data which are retrieved for online applications. To make cloud to be best place to store and compute data, the data has to be encrypted. The method of searching and retrieving documents from cloud over encrypted data has to be achieved without loss of privacy. The searching mechanism must be very easy and options of data retrieval mechanism have to be supported by cloud servers. When the search keyword is given as input, the cloud server should retrieve most relevant documents matching the keyword and return them with learning anything about the keyword or the documents.

Keywords: Data retrieval, cloud security, searchable encryption

1. INTRODUCTION

The data owner initially preprocesses the documents and builds index before outsourcing. Both encrypted document and index is forwarded to the cloud server. Upon receiving the keyword from data owner the server tries to search the encrypted index and return the top ‘k’ encrypted documents that match the index, is given to the owner. The challenge in this search and retrieval process is the Known search keyword, document-keyword mapping, and document information. If the server knows any one of the above challenges, he may become curios and tries to take data.

The proposed technique tries to solve the multiple keyword searches in an encrypted data that are remotely stored. In the keyword semantics, the term frequency measure is used for document retrieval. In the index build stage the owner tries to associate the set of documents for keywords. The security and privacy issues for keyword search on encrypted data are given. An efficient multi-keyword search

scheme under various security considerations is provided. Finally the document ranking is done for the retrieved document. The implementation of the proposed scheme shows that the scheme is more secure and efficient than the existing schemes[2].

2. SYSTEM MODEL

The computing model consists of three entities the data owner, the data user and the cloud server [6].The data owner has a collection of documents to be outsourced in cloud in encrypted form. To allow computation over encrypted data the documents are initially preprocessed. The preprocessing step helps the data owner to identify and rank the documents according to the frequency of occurrence of keywords in a particular document. The documents are ranked based on similarity metrics between the keyword and the document.

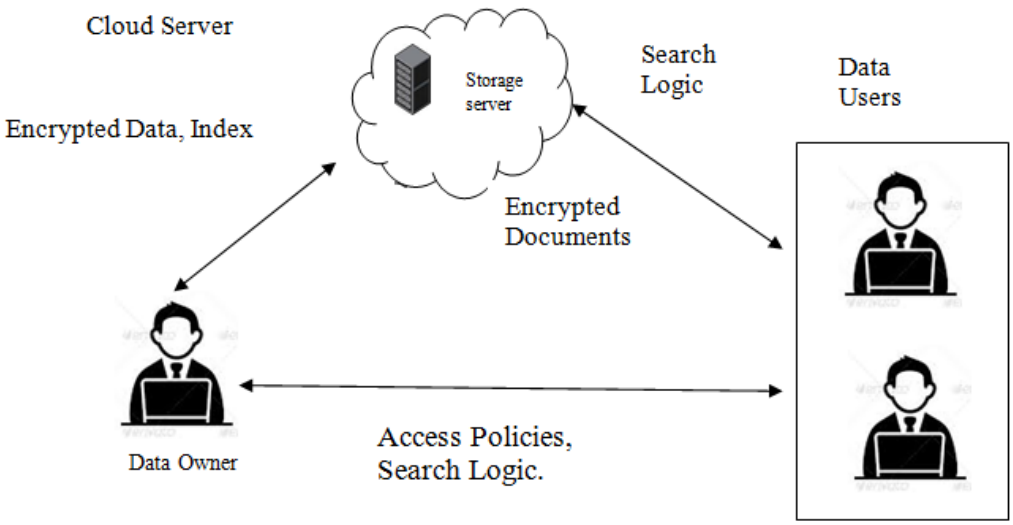

Figure1 System Model

Initially the similarity measures between each given terms for a given dataset are first identified. The associations between the terms are calculated using term and document frequency measure. Each term is linked with the relevant documents. The keyword is represented as a binary string. The encrypted document list that is associated with binary string is stored in an index.

3. DESIGN GOALS

1. Without having the prior knowledge on the encrypted data, cloud server has to search all the encrypted cloud data to retrieve documents matching the query.
2. Large communication overhead results for searching presence or absence of a keyword that is highly complex in the cloud environment.

4. PRELIMINARIES

Vector Space Model.[9] The representation of a set of documents as vectors in a common vector space is known as the vector space model and is fundamental to a host of information retrieval operations ranging from scoring documents on a query, document classification and document clustering. The most popular similarity measure is the cosine coefficient, which measures the angle between the document vector and the keyword query vector. The vector space model procedure can be divided in to three stages. The first stage is the document indexing where content bearing terms are extracted from the document text. The second stage is the weighting of the indexed terms to enhance retrieval of document relevant to the user. The last stage ranks the document with respect to the query according to a similarity measure.

Term Frequency. The term frequency for a query keyword q in document d is defined as the number of times that q occurs in d.[9]

Document Frequency (df). [9] Document frequency df, is a measure of the informativeness of a given query keyword q in a given document collection.

Term Frequency-Inverse Document Frequency (td-idf). It is a numerical statistics to show how essential a word is to a given document collection. The value increases along with the number of times the word appears in the document.

Similarity Coefficients. [9] The similarity in vector space models is determined by using associative coefficients based on the inner product of the document vector and query vector, where matched keywords indicates similarity. The inner product is usually normalized. The most popular similarity measure is the cosine coefficient, which measures the angle between the document vector and the given keyword.

$$similarity\ score = \cos(d.q) = \frac{d.q}{|d|.|q|} = \frac{\sum_{i=1}^{n} d_i.q_i}{\sqrt{\sum_{i=1}^{n}(d_i)^2 X \sqrt{\sum_{i=1}^{n}(q_i)^2}}} \quad (1)$$

Where$\cos(d,q)$ is the cosine similarity of document d and query vector q.

5. DATA RETRIEVAL PROCESS

5.1 Preprocess

The documents have to be preprocessed before outsourcing. The preprocessing step involves various stages in document processing. Initially the most valuable keywords are identified. The identified keyword will be mapped with the documents in the document collection. In the proposed method, similarity between the term and the document is found using simple cosine similarity. Both the document and the keyword are represented as a vector coefficient, and the inner product is estimated.

5.2 Query Keyword-Document matching score

If the query keyword does not occur in the document, score would be 0.The more frequent the query keyword in the document, the higher the score. The score value lies in the range of [0,1]. The procedure for calculating score value is given in Algorithm 1.

Algorithm 1*Score* (Query Keyword q, document d)

D = set of documents retrieved for query *keyword q*
scores[n] =0
len[n]
for each query keyword *q*
do calculate $W_c$and fetch index for q
for each pair (d,$tf_{q,d}$) in index
do scores[d]+= $w_{q,d}$x $w_{q,c}$
for each d
do scores[d] = scores[d]/len[n]
Return top k list of scores[]

5.3 Build Index

The unique codes which identify the keyword and associated document list are maintained in the Index I. The index and encrypted set of document are placed in the cloud. Finally, the access control mechanism is employed to manage decryption capabilities given to users and the data collection can be updated in terms of inserting new documents, updating existing documents, and deleting existing documents. The index is built based on the frequency of occurrence of a particular search word in a relevant document. The data user can send queries to retrieve the relevant document based on the authorization code it has received from the owner.

Each keyword is associated with the unique value that leads to the document retrieval. Based on the score, the encrypted keyword is mapped to the collection of documents. The documents are arranged based on the score value.

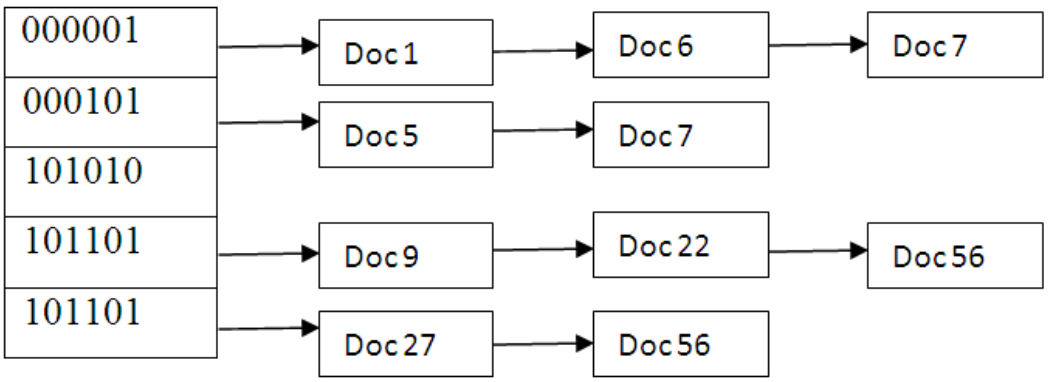

Figure2Index

5.4 Search Strategy

The data owner generates binary vector B[b1,b2,….b12] for each keyword taken from the dataset. The first two bits (b1, b2) specify the position of starting of the keyword. The consecutive 6 bits identifies the keyword and remaining 4 bits acts as padding bits that helps to mask the keyword. The padding bits either can be consecutive four bits or it can be a scrambled one but the six bits of the keyword come all together always.

0	0	1	0	1	0	1	0	0	1	0	1

Figure3Keyword Representation

6. PRIVACY PRESERVING DATA RETRIEVAL (PPDR)

6.1 System Architecture

The proposed system is shown in figure 4. We assume that the system is semi trustable. The data owner initially preprocesses the file and encrypts them. The Index is built for the set of keywords. Then the indexes along with encrypted documents are sent to the cloud server. The index is built based on secret key and the search strategy logic.

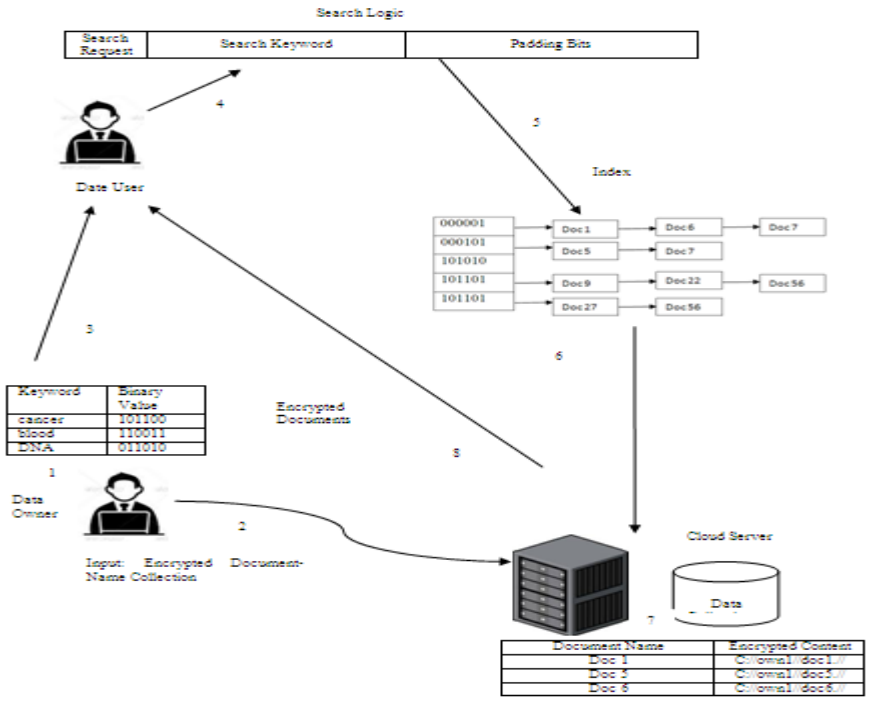

Figure4 System Architecture

6.2 PPDR Framework

KeyGen. The algorithm takes the security parameter λ ε 'Zp' and outputs secret key shk.

BuildIndex(D,K,shk). The keyword Kϵ {0,1}*, secret key shk , Document collection D are given as input and algorithm outputs an index that maps K to D.

Searchlogic(k). The interested keywords are given as input and this algorithm generates a corresponding searchlogic.

Querylogic(SL,q,I). When the server receives request (SL,q), the server searches index I using the search logic SL and returns top k documents sorted in the order of query keyword.

6.3 The PPDR Scheme

Initialize.To make an efficient keyword search we use the inner product similarity metrics [5]. The keyword collection KC[k1,k2,..kn] is a set of collected keywords that appear in the selected dataset. Each keyword ki is associated with the binary query vector Q[m] ϵ {0,1}.The search strategy S consists of multiple query keyword S[q1,q2,q3] along with the padding bits.

When the data user wants to access the data from the documents, he contacts the data owner initially. The data owner sends the search strategy logic. The user embeds the keyword in the search logic and forwards it to the cloud server. The cloud server returns the top k results as requested by the user. The user on receiving the encrypted documents contacts the data owner for decryption key of the returned documents. On receiving the key he can make use of the documents retrieved.

Index I(D,K). The existing scheme uses indexing mechanism for each document and hash function is used to associate the keyword query to the set of defined integers. The integer associated with the keyword is secretly shared to the authorized users. If the server learns the association then the system can be intruded for the smaller size of data. In the proposed scheme, only the data owner can design the search strategy with the secret key. The usage of secret key eliminates the guessing attack.

The index I is initially built for the set of keywords K {k1,k2,…km} that are associated with the set of documents D. Each Keyword is represented in {0,1}* bits. Each of the indexed keyword is mapped to the relevant documents. Finally, the bitwise product of indices of all keyword in the document D provides the Index I for the keyword K. The index is the l-bit sequence of binary strings of keyword that maps the matched documents for the given query keyword. Each document vector is multiplied with the random matrix A[5].

Search Logic SL (q).The search logic for any given keyword is obtained by any of the two following methods

Exact keyword match. Given the set of multiple search keywords as input, the binary vector Q is generated. The r bit in query represents a keyword k. Additional padding bits are added in the logic to mask the keyword. Initial m bit represent the starting position of a keywordand the continuous r bit from the starting bit is the keyword and remaining bits are padding bits. The inverse of a matrix A^{-1} is multiplied with the query vector.

Semantic keyword match.The search logic also builds the semantically similar terms using wordnet [10]. For all the identified index keywords the semantic terms are identified and are managed by the data owner in his search logic. The owner during preprocessing itself would find the term-term association and maintains in the searchlogic. When the user enters the keyword and if the keyword is not present, the semantic keyword search begins. If Query keyword matches index keyword, query keyword is replaced by the indexed keyword and request will be forwarded to the cloud server.

QueryLogic(SL,I). The cloud server performs match over the received search logic and index given by the data owner. The similarity score matches the keyword with the collection of document and the top 'k' documents which having highest score is considered. The encrypted documents of top 'k' results are returned to the data user. The similarity score is measure using the inner product of the document and query vector.

$$I.SL = \{A.\ D_i\}\ .\ \{A^{-1}.Q\} = D_i.Q \qquad (2)$$

7. PERFORMANCE ANALYSIS

The performance of PPDR scheme is analyzed based on efficiency, privacy and Precision.

Efficiency. The number of keyword appearing in the document is $D_i.Q$ .The final result based on the similarity score will be accurate. The proposed system tries to provide both search accuracy and privacy. The binary representation along with padding bits hide the keyword from the server and the similarity coefficient measurement tries to provide the best top P results as requested by the data user. The PPDR scheme is highly efficient than the existing schemes for multiple keyword retrieval.

Privacy. The data privacy is achieved through standard symmetric key encryption technique. The search logic privacy achieved through different places of keyword positioning. The keyword is highly masked and will not occupy same position when each time querying. It avoids search logic relating problem.

Precision. The precision depends on the number of relevant documents returned after

the query is executed. The proposed system has high precision since the mapping between the keyword and the documents are initially processed and the semantic of the query terms is also estimated which never misses the document return.

8. EXPERIMENT ANALYSIS

8.1 Dataset

The Oshumed [7] dataset contains associated documents of over 106 queries. The Ohsumed test collection is a subset of the MEDLINE database, which is a bibliographic database of important, peer-reviewed medical literature maintained by the National Library of Medicine. A query keyword is related to a medical search. The dataset consists of around 16,140 query-document pairs with relevance judgments.

8.2 Experimental Setup

The proposed technique is evaluated over the Oshumed data set. The system is implemented by java language over Luceneframework on a windows machine with Intel Xeon Processor with 4 GB RAM. Queries over search logic were issued to retrieve documents from the Oshumed document collection [7]. The queries are represented in the form of binary vectors and then the proposed methods have been applied to return top 10 documents.

8.3 Experimental Analysis

The proposed system is tested over Oracle cloud. The Index is built by reading the contents of the Oshumed Data set. Only unique words are identified and mapped to the document list and index is built and encrypted. The encrypted index along with the document collection is placed in cloud.The results plotted shows Index build time while preprocessing and it is compared with the existing system. Figure5 shows the time taken for building index in proposed system is less when compared to the existing system[5]. Figure6 shows the search logic build time during retrieval process.

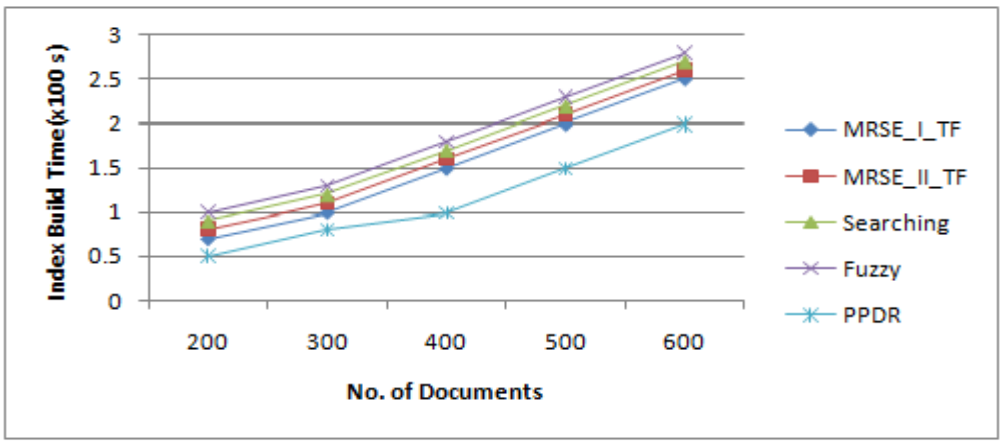

Figure 5 Index building time during preprocess

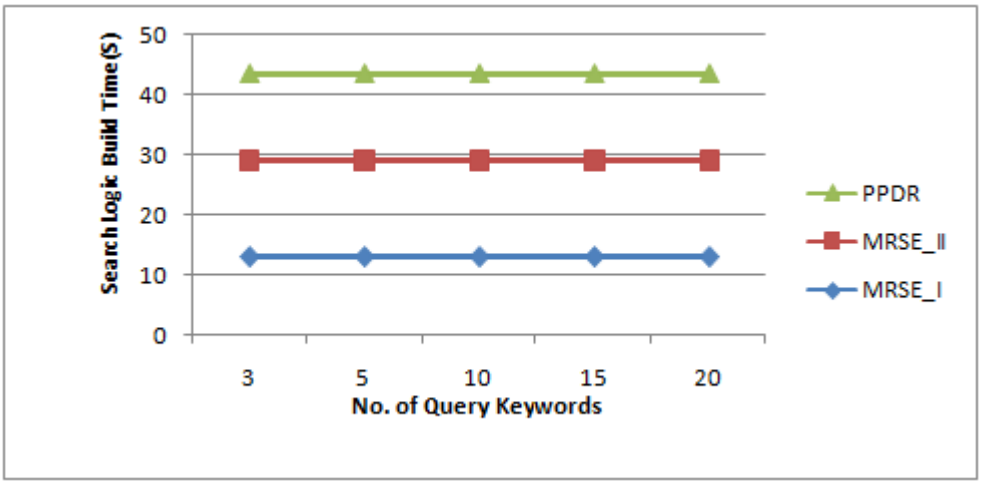

Figure 6 Search Logic build time during retrieval

9. RELATED WORK

The search started with a single keyword. The encrypted search index along with the documents is stored in the remote server. The trapdoor which carries the secret key lets the index to open and retrieve the documents for the given query keyword. The advancement of the search leads to the combination of keywords using conjunctive or disjunctive policies emphasized over the keywords. In conjunctive policy search, if all the keywords given in the search query matches the index then the documents are returned if not no result is returned and in disjunctive one, if either one of the keyword matches then the result is returned. The variation of the search techniques are proposed, which provides option for searching the documents for multiple keywords at the same time. The documents matching all of the keywords are returned.

[9] employs Boolean expressions for searching and retrieving documents. The Boolean expression includes conjunction, disjunction and negation operations. The Gram-Schmidt orthogonalization is applied over the keyword vector. The inner product is computed between the query and the document and the results are returned.

[4] explores the order preserving mapping technique is used for searching the encrypted data. Encrypted files have to be processed after searching and it may leads to collision. The IDF factor is considered for calculating score value,in which there is chance of information leakage.

In [5] coordinate matching technique is used to find the similarity between given multiple keyword queries and the document collection. The KNN secure technique is adopted where the dimension of the document and query vector is extended which helps to mask the details of the keyword and the document. Instead of returning the exact matched documents the nearest matching documents are returned. The precision of the system is low when compared to the privacy. There is no option for searching Boolean or single keyword searches.

10. CONCLUSION

In this paper, multiple keywords are given as a search query in retrieving encrypted

documents from the cloud server. The similarity measure for the given query and the documents are estimated using the cosine similarity. The bit representation and different positioning of the query word in the search logic helps to preserve the privacy of the words and also avoids the information leakage to the cloud server. The number of keywords in the search query does not affect the performance of the system. The overall computation and communication overhead of the proposed work is very minimal when compared to the existing multiple keyword searches.

ACKNOWLEDGEMENT

The author(s) wish to thank ministry of Human Resource Development (MHRD), Government of India for funding the research under TEQIP scheme and also extend the thanks to Government College of Technology, Coimbatore for providing Infrastructure facility to carry out the research.

REFERENCES

[1] http://en.wikipedia.org/wiki/Cloud_computing.

[2] CengizÖrencik, ErkaySavaş, "Efficient and secure ranked multi-keyword search on encrypted cloud data", in proceedings of 2012 Joint EDBT/ICDT Workshops, ACM New York, NY,2012.

[3] Bernardo Ferreira, Henrique Domingos, "Search Private data in a cloud encrypted domain" ,in Proceedings of the 10th Conference on Open Research Areas in Information Retrieval, 2013 .

[4] Cong Wang, Ning Cao, Jin Li, KuiRen, Wenjing Lou, "Secure Ranked Keyword Search over Encrypted Cloud Data", In Proceedings of the 2010 IEEE 30th International Conference on Distributed Computing Systems,IEEE Computer Society Washington, DC, 2010.

[5] Ning Cao, Cong Wang, Li, Ming , KuiRen, Wenjing Lou, "Privacy-Preserving Multi-keyword Ranked Search over Encrypted Cloud Data" , IEEE Transaction on Parallel and Distributed systems,25(1), 2014.

[6] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data" ,IEEE Transaction on Parallel and Distributed systems,23(8), 2012.

[7] Hersh WR, Buckley C, Leone TJ, Hickam DH, "OHSUMED: An interactive retrieval evaluation and new large test collection for research", in Proceedings of the 17th Annual international ACM SIGIR Conference on Research and development in information retrieval, Springer-Verlag,NY,1994.

[8] http://cogsys.imm.dtu.dk/thor/projects/multimedia/textmining/node5.html

[9] TarikMoataz, AbdullatifShikfa, "Boolean Symmetric Searchable Encryption", in Proceedings of the 8th ACM SIGSAC symposium on Information computer and communications security NY,2013.

[10] http://wordnet.princeton.edu/

[11] J.Li, Qian Wang , Cong Wang and Nin Cao,'Fuzzy Keyword Search over Encrypted data in Cloud Computing', IEEE proceedings INFOCOM, pp. 1-5, 2010.