Volume: 2, Issue: 5, 162-167
May 2015
www.allsubjectjournal.com
e-ISSN: 2349-4182
p-ISSN: 2349-5979
Impact Factor: 3.762

Sarulatha.M
PG Scholar, Dept of CSE
Sri Krishna College of Technology Coimbatore, India

Anantha Prabha.P
Assistant Professor, Dept of CSE, Sri Krishna College of Technology, Coimbatore, India

Correspondence:
Sarulatha.M
PG Scholar, Dept of CSE
Sri Krishna College of Technology Coimbatore, India

Secure semantic based search over cloud

Sarulatha.M, Anantha Prabha.P

Abstract
Data are outsourced to the cloud for ease of access. Data is encrypted before outsourcing to the cloud to guarantee the security. To achieve efficient data utilization, searching over encrypted cloud data has a great challenge. In the existing system, MKQE(Multi-keyword query scheme) retrieves only keyword related files MKQE entirely depends on the submitted query keyword and didn't consider the semantics of keyword..Searching efficiency is improved by semantic based search. Semantic based search is an important and necessary functionality for information retrieval in cloud storage services. This scheme returns the exactly matched files and also semantically related files. Meta data file is created for each file. Cloud server builds the index and dictionary to a keyword. With less communication user can retrieve more files from the cloud server. Files are returned on the basis of the relevance score.

Keywords: Cloud computing, semantic search

1. Introduction
Cloud computing is the long dreamed vision of computing as service, where cloud customers can vaguely store their data into the cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources. Because of the cloud services, flexibility and economic savings are motivating both individuals and enterprises to outsource their local complex data management system into the cloud. With the popularity of cloud services, such as Microsoft Azure, Apple iCloud, Google AppEngine, more and more companies are planning to move their data onto the cloud. The straightforward solution to protect data privacy is to encrypt sensitive data before being outsourced. Typically, a user retrieves files of interest to him/her via keyword search instead of retrieving back all the files. This keyword based search technique has been widely used in our daily life, e.g. Google plaintext keyword search. Though the technologies are invalid after the keywords are encrypted. One of the most popular ways to do so is through keyword-based search. This kind of keyword search technique allows users to selectively retrieve files of interest and has been widely applied in plaintext search scenarios [5]. To enhance the search flexibility and usability, some research is done on fuzzy keyword search [14-18]. These solutions support tolerance of minor typos and format inconsistencies, such as, search for "billion" by carelessly typed as "bilion", or "datamining" by typed as "data-mining". This schemes mainly take the structure of terms into consideration and use edit distance to evaluate the similarity. In the literature, searchable encryption [5]–[13] is a helpful technique that treats encrypted data as documents and allows a user to securely search through a single keyword and retrieve documents of interest. They didn't consider the terms semantically related to query keyword, so many related files are omitted. In this paper, we propose a similar search solution based on semantic based search supports the similarity ranking. Semantic based expansion similar search returns the exactly matched files and the files including the terms semantically related to the query keyword.

2. Problem Definition
Cloud computing has emerging as a promising pattern for data outsourcing and high-quality data services. Such concerns of sensitive information on cloud potentially cause privacy problems. Data encryption protects the data over the cloud. In existing system Multi keyword query encryption (MKQE), consumer retrieves the files related to the keyword. To improve the searching efficiency, semantic based search mechanism is used. It retrieves the files related to the keyword and the semantically related files. Two-round searchable encryption (TRSE) scheme employing the fully homomorphic encryption, which fulfills the security requirements of multi keyword top-k retrieval over the encrypted cloud data. With semantic based search the user are able to retrieve files related to the keyword and semantically related files.

We consider the system model involving three different entities: data owner, data user and cloud server, as illustrated in Figure 1. Data owner uploads a collection of n text files F ={F1, F2, F3,…, Fn} in encrypted form C, together with the encrypted metadata set, to the cloud server. Note that, a corresponding file metadata is constructed for each file. Each file in the collection is encrypted with common symmetric encryption algorithm, e.g. AES.

3. Related works

Dongsheng Wang, Shaojing Fu et al. [3] proposed sFKS over encrypted cloud data. Fuzzy keyword search over support top-k ranked similarity search and then extend it to first realize the special-string-first matching rule, which is very useful for sFKS schemes to support exploratory search or uncertain search. To construct secure searching indexes, extract a fingerprint vector from each original keyword. The fingerprint vectors extract can be used to evaluate the similarity between different strings. The features of secure kNN encryption make it a feasible encryption algorithm. Data owners first collect the original data and set a symmetric key for data encryption. Data owner builds a secure inverted index structure by associating each secure index with certain file identities used to locate corresponding data files. The data owner sends the data set to the cloud and distributes the secret keys to each authorized data user. The data user submits the query trapdoor to the cloud server. The searching time is mainly spent in finding the candidate secure indexes in the cloud side. It is not compatible with other symbols or languages. It does not support more matching rules and fuzzy multi-keyword search functionalities.

Ning Cao, Cong Wang et al. [4] proposed multi-keyword ranked search over encrypted cloud data Multi-keyword Ranked Search Encryption (MRSE)**.** The large number of data users and documents in the cloud, it is necessary to allow multiple keywords in the search request and return documents in the order of their relevance to these keywords The data owner outputs a symmetric key. Owner builds a searchable index which is encrypted by the symmetric key and then outsourced to the cloud server. After the index construction, the document collection can be independently encrypted and outsourced. With keywords as input, it generates a corresponding trapdoor. When the cloud server receives a query request as it performs the ranked search on the index with the help of trapdoor and finally returns the ranked id list of top-*k* documents sorted by their similarity. Documents are returned in the order of their relevance to the keywords. **"**Inner product similarity" is used evaluate similarity measure. It does not support multi keyword semantic over encrypted data.

Qin Liu, Guojun Wang, et al [5] propose a practical privacy-preserving ranked keyword search scheme based on PIR that allows multi-keyword queries with ranking capability Private Information Retrieval (PIR), provides useful cryptographic tools to hide the queried search terms and the data retrieved from the database while returning most relevant documents to the user. This scheme increases the security of the keyword search scheme. The frame work of the proposed scheme is that The data owner collects the information in the database and lacks the means to maintain the database. The data owner creates a search index for each document. The search index is created using a secret key based trapdoor generation function where the secret keys1 are only known by the data owner. Then, the data owner uploads these search index to the server together with the encrypted documents. the trapdoor information ,the user generates the query and submits it to the server and receives metadata for the matched documents in a rank ordered . the user interacts with the data owner in order to decrypt the documents and ge t the corresponding plaintext and the data owner does not learn the documents that it is assisting to decrypt. This do not involve any security or privacy-preserving techniques.

Zhangjie Fu, Xingming Sun et al. [7] proposed synonym-based search and similarity ranked search. The data owner generates the secret key and picks a random key. The secret key includes a randomly generated vector and two invertible matrices. The data owner calls procedure buildindex. The encrypted searchable index tree is generated. The encrypted query vector is sent to the cloud server. The vector space model is adopted combined with cosine measure, which is popular in information retrieval field, to evaluate the similarity between search request and document. The performance of the proposed schemes is analyzed with search efficiency and search accuracy by the experiment on real-world dataset. The results show that the proposed solution is very efficient and effective in supporting synonym-based searching. The performance of search scheme is evaluated as the number of documents increases.This does not support syntactically related files.

4. Proposed scheme

Secure Semantic based Search Scheme

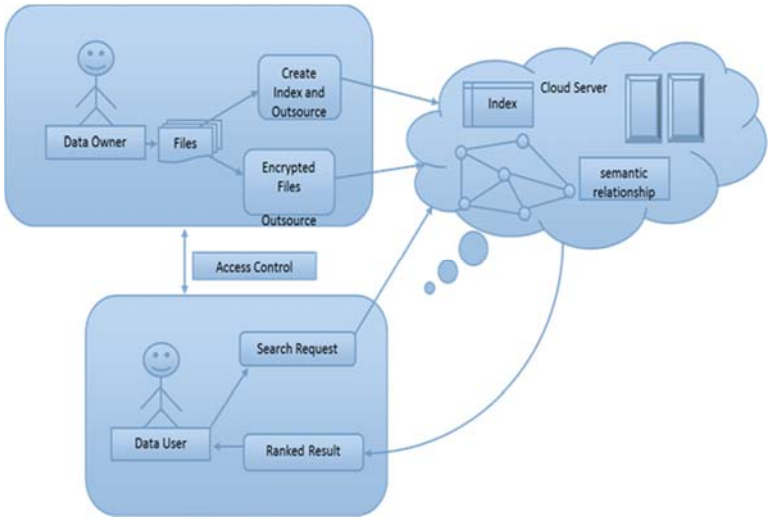

Fig 1: Overall Design

Figure 3.1 represents the block diagram of this system, in which data owner collects the files and create index to the files. The indexed files are outsourced to the cloud server. Files and index are encrypted before they are outsourced to the cloud. Data owner creates the dictionary to the keywords and it is stored in the cloud. Data user request the keyword and retrieves the files related to the keyword and semantically related files.

The proposed system consists of the following modules

- Registering with Cloud and encryption of files
- Index file creation and Dictionary creation
- Retrieval of files using semantic based search

5. Registering with cloud and encryption of files

Registering in to cloud

Aspose free cloud is purchased to use the cloud storage. The user initially register the details to the cloud and then the user can use the cloud. Once the user register the cloud it can be used when they needed. The user can login to the cloud when the user need of cloud. User can retrieve files with the keyword. When the user login to the cloud then the user can search data over the cloud, upload a file over the cloud. The owner can view the list of users who uses the cloud. The data are encrypted before it is outsourced to the cloud. The encrypted data is searched over the searchable encryption method of TRSE.

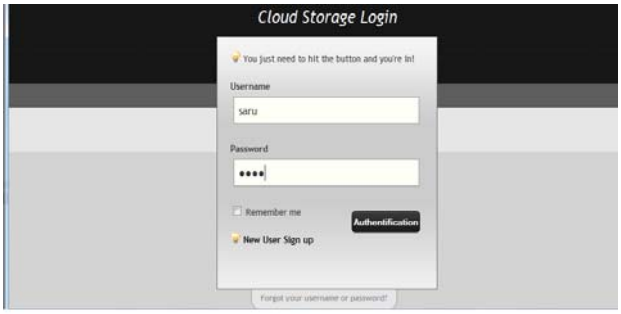

Fig 2: Registering in to cloud

Encryption of files

TRSE scheme employing the fully homomorphic encryption, which fulfills the security requirements of multi keyword top-k retrieval over the encrypted cloud data. The files are encrypted by homomorphic encryption scheme. It allows specific types of computations to be carried out on the cipher text and to obtain an encrypted result which When decrypted matches the result of operations performed on the plaintext. Before it is sent, the data in the cloud is encrypted with homomorphic encryption and operations are executed in the encrypted data and the results are decrypted, it is same as the operations performed on the original data. Homomorphic encryption cryptosystem provides security and data confidentiality. In homomorphic encryption, there is just one operation on the plaintext that has a corresponding operation on the cipher text. For example, plain RSA has that property. Suppose cipher text c1 is the encryption under a public key pk of plaintext m1, and c2 is the encryption under the same key of m2.

$$c1=enc(pk,m1) \quad (1)$$

$$c2=enc(pk,m2) \quad (2)$$

where c1 and c2 are cipher text, m1 and m2 are plain text, pk is public key.

Then multiplying the cipher texts results in something which, when decrypted, is identical to the result of multiplying the two plain texts. If sk is the decryption key corresponding to pk,

$$m1 \times m2=dec(sk, c1 \times c2 \quad (3)$$

An algorithm is homomorphic if it is made up of addition, subtraction, multiplication functions. In the proposed system, we use multiplication while encrypting the data [2].sk is the secret key.

This algorithm consists of the following steps:

- Key generation
- Encryption
- Decryption

The data owner encrypts files and keywords using this algorithm and stores it in cloud storage. When the user requires file to be retrieved, user can request from the cloud. The cloud perform computations on the encrypted data without knowing anything of the files and the keywords. It will send back the results to the user. The data user can decrypt it.

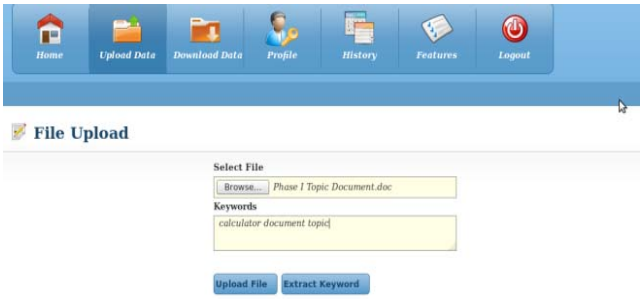

Fig 3: Uploading files to cloud

6. Index file creation and Dictionary Creation

Index Creation

Data owner creates the index. Owner decides the keyword and the keyword is set as the index to a file. It can also be automatically generated by cloud. Index is created as a list of mappings which correspond to each keyword. The list for a particular keyword contains details such as:

- File ids of the files which has the particular keyword
- Term frequency for each file which denotes the number of times the keyword has occurred in the file.
- Length of each file
- Relevance score for each file
- Number of files that has the particular keyword

Term frequency, length of the file, number of files for the keyword are used to calculate the relevance score for each file.

The index and the files are stored in encrypted form in the cloud. Whenever user searches for a word, the request is sent to the cloud, which searches over the index and sends the entire mapping that is created for the word to the user. The user has the overhead to decrypt and request to retrieve the most relevant files based on the relevance score. When user searches for a file, the keyword is sent to the cloud, which searches the index, finds out the most relevant files and requests cloud for the files to be retrieved and sent to user thereby ensuring data confidentiality in cloud [8].

Dictionary Creation

Synonyms are words with the same or similar meanings. In order to improve the accuracy of search results, the keywords extracted from outsourced text documents need to be extended by common synonyms, as cloud customers searching input might be the synonyms of the predefined keywords, not the exact or fuzzy matching keywords due to the possible synonym substitution and lack of exact knowledge about the data. The synonyms of predefined keywords differ greatly from fuzzy matching keywords in spelling.

WordNet is a large lexical database of English. Nouns, verbs, adjective and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. The resulting network of meaningfully related words and concepts can be navigated with the browser. WordNet is freely and publicly available for download. WordNet's structure makes it a useful tool for computational linguistics and natural language processing.

WordNet superficially resembles a thesaurus, in that it groups words together based on their meanings. However, there are some important distinctions. WordNet interlinks not just word forms, strings of letters, but specific senses of words. As a result, words that are found in close proximity to one another in the network are semantically disambiguated. WordNet labels the semantic relations among words, whereas the groupings of words in a thesaurus does not follow any explicit pattern other than meaning similarity.

7. Retrieval of files using Semantic Based Search

The user generates a secure trapdoor of keyword using TrapdoorGen, and submits it to the cloud. Upon receiving the query trapdoor, the cloud first automatically expands the query keyword [9].Then the server searches the index through SearchIndex, and eventually sends back the matched files in a ranked sequence according to the total relevance scores. Details are as follows.

The user generates a trapdoor $T_{w=}\pi_x(w)$ for an interested keyword w, by calling TrapdoorGen(w). Upon receiving the trapdoor T_w, the server first expands the query keyword to obtain the extensional query trapdoor.

$$T_w' = \{\pi_x(w), \pi_x(w_i')\}, \forall\, w_i' \in S_w. \quad (4)$$

By calling Search Index, the server locates the matching entries of the index via $\pi_x(w)$ and $\pi_x(w_i')$ which include the file identifiers and the associated order-preserved encrypted relevance scores.

The server then computes the total relevance score of each file. In the end, the server sends back the matched files in a ranked sequence, or sends top-k most relevant files if the user provides the optional value k.

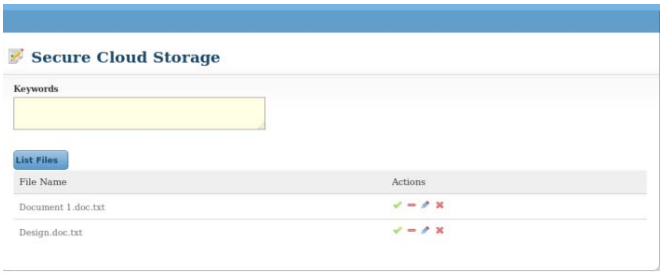

Fig 4: List of files for download

Once the documents are stored and indexed, the important function is to rank them. Numeric score is calculated for each file. The ranking functions are based on the TF X IDF rule, where TF stands for Term frequency which represents the number of times a keyword is present in a file and IDF stands for Inverse Document Frequency which is defined as the ratio of number of file containing the word to the total number of files present in the server. The large number of data users and documents in the cloud, it is necessary to allow multi keyword in the search query and return documents in the order of their relevancy with the queried keywords. Scoring is a natural way to weight the relevance. Based on the relevance score, files are ranked in either ascending or descending. TF-IDF weighting is used to score and rank files. Among these schemes involves two attributes-term frequency and inverse document frequency.

The TF-IDF weighting involves two attributes:

- Term frequency
- Inverse document frequency

Term frequency denotes the number of occurrences of term t in file f.

Document frequency refers to the number of files that contains term t, and the inverse document frequency N is defined as: IDFT1 ⁄4 log DFT, where N denotes the total number of files.

Rank function: In information retrieval, a ranking function is usually used to evaluate relevant scores of matching files to a request. Among lots of ranking functions, the "*TF*×*IDF*" rule [6] is most widely used, where *TF* (term frequency) denotes the occurrence of the term appearing in the document, and *IDF* (inverse document frequency) is often obtained by dividing the total number of documents by the number of files containing the term. That means, *TF* represents the importance of the term in the document and *IDF* indicates the importance or degree of distinction in the whole document collection. Each document is corresponding to an index vector *Dd* that stores normalized *TF* weight, and the query vector *Q* stores normalized *IDF* weight. Each dimension of *d D* or *Q* is related to a keyword in *W*, and the order is same with that in *W*, that is, *D* [*i*] *d* is corresponding to keyword *i w* in *W*. The similarity evaluation function [15] is

employed for cosine measure. The notations used in similarity evaluation function are showed as follows:

- $f_{d,j}$, the TF of keyword *j w* within the document *d*;
- f_j , the number of documents containing the keyword w_j ;
- *M*, the total number of documents in the document collection;
- *N*, the total number of keywords in the keyword dictionary;
- $w_{d,j}$, the TF weight computed from $f_{d,j}$;
- $w_{q,j}$, the IDF weight computed from *N* and f_j;

The definition of the similarity function is as follows:

$$SC(Q, D_d) = \frac{\sum_{j=1}^{N} w_{q,j} \cdot w_{d,j}}{\sqrt{\sum_{j=1}^{N} (w_{q,j})^2} \cdot \sqrt{\sum_{j=1}^{N} (w_{d,j})^2}} \quad (1)$$

Where the vector *Q* and D_q are both unit vectors.

8. Performance Analysis

Performance is analyzed by the number of files retrieved through the semantic based search over cloud. More files are retrieved from the cloud using the proposed scheme.

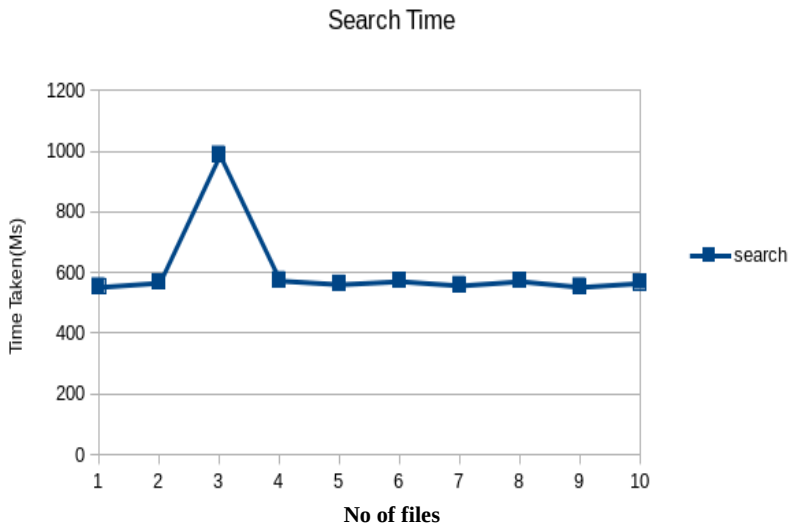

Fig.5 Search time

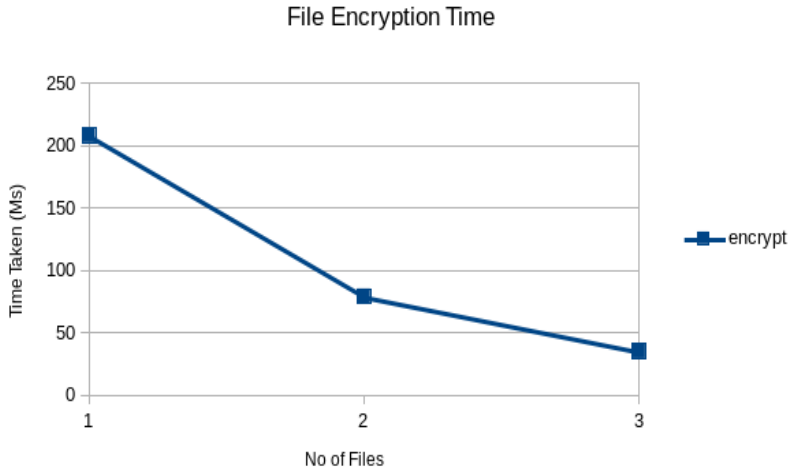

Fig 6: Encryption time

9. Conclusion

Semantic Based Search retrieve the files related to the keyword and the semantic files related to the keyword. Search efficiency is improved by semantic based search with minimum communication and computation overhead. TRSE is used for encryption. TRSE scheme employing the fully homomorphic encryption, which fulfills the security requirements of multi keyword top-k retrieval over the encrypted cloud data. The datasecurity is the main issue in the cloud. The data are encrypted before it is outsourced to the cloud. Semantic based search will use more secure encryption primitives in the future for reducing the complexity and for providing more secure storage.

10. References

1. Ankatha Samuyelu Raja Vasanthi A (2012), ‘ Secured Multi-keyword Ranked Search over Encrypted Cloud Data ‘, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 10, pp.115-119.
2. Cong Wang, Ning Cao, Jin Li, Kui Ren, and Wenjing Lou (2010), ‘Secure Ranked Keyword Search over Encrypted Cloud Data’, International Conference on Distributed Computing Systems, pp.253-262.
3. Dongsheng Wang, Shaojing Fu, and Ming Xu (2013), ‘Privacy-preserving Fuzzy Keyword Search Scheme over Encrypted Cloud Data’, IEEE International Conference on Cloud Computing Technology and Science, pp.663-670.
4. Ning Cao, Cong Wang, Ming Li, Kui Ren, and Wenjing Lou (2011), ‘Privacy-Preserving Multi-keyword Ranked Search over Encrypted Cloud Data’, IEEE INFOCOM, pp.829-837.
5. Qin Liu, Guojun Wang, Jie Wu (2012), ‘Secure and privacy preserving keyword searching for cloud storage services’, Journal of Network and Computer Applications 35, pp.927–933.
6. Qin Liu, Guojun Wang, and Jie Wu (2009), ‘An Efficient Privacy Preserving Keyword Search Scheme in Cloud Computing’, International Conference on Computational Science and Engineering, pp.715-719.
7. Zhangjie Fu, Xingming Sun, Nigel Linge, Lu Zhou (2014), ‘Achieving Effective Cloud Search Services: Multi-keyword Ranked Search over Encrypted Cloud Data Supporting Synonym Query’, IEEE Transactions on Consumer Electronics, Vol. 60, No. 1, pp.164-172.

8. Zhihua Xia, Yanling Zhu, Xingming Sun and Lihong Chen (2014), 'Secure semantic expansion based search over encrypted cloud data supporting similarity ranking', Journal of Cloud Computing: Advances, Systems and Applications, pp.1-11.
9. Zhihua Xia, Li Chen, 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 Vol.31, pp.284-289.