An Efficient Multi-Keyword Semantic Based Search for Encrypted Cloud Data

[1] Dipika D. Chavan,[2] Dinesh M. Yadav,
[1]Post Graduate Student, [2]Director
[1]Computer Department,
[1]RSSOER, Pune, India

***Abstract* - Today, cloud computing technologies becomes more efficient and flexible with reduced cost and utilities of on-demand high quality applications and services, so internet usage strongly relies on cloud for privacy preserving and fast data retrieval. For customers, they would like to search the most relevant or related products or data, which is highly desirable in the "pay-per-use" cloud computing model. Sensitive data (such as photo albums, emails, tax information, financial records, etc.) is encrypted before outsourced to cloud. Though Searchable encryption scheme has been developed to retrieve the encrypted data, these schemes only support exact or fuzzy keyword search. Existing fuzzy search methods mainly estimate the similarity of keywords from the structure but the semantic relatedness is not considered. In secure semantic search through query keyword, semantic extension based on the co-occurrence probability of terms, the TFIDF scheme from information retrieval is used to calculate the similarity score of keywords between the documents and the user queries. To enhance the efficiency of the search method the extended keyword set is used with semantic words or natural language words for the keywords. This will eventually support data retrieval on querying semantic query. Even when user is unaware of exact or synonym of keywords of encrypted data, he can try searching it by its meaning in natural language.**

***Index Terms* - Cloud computing, Multi-keyword search, Semantic based search, TFIDF.**

I. INTRODUCTION

In recent years consumer centric cloud computing is a novel model for the enterprise-level IT organization that provides on-demand high quality applications and services from a shared group of computing resources. The Cloud Service Provider (CSP) has complete control over the outsourced data; it may possible that it can learn some additional information from that data therefore some problems like privacy of the data, arise in the circumstance. So, sensitive data have to encrypt before outsourcing to the cloud server. However the encrypted data makes the existing plaintext search approaches useless. The simple and awkward method is downloading all data and decrypt it locally is apparently impractical, because the cloud consumers need to search only the interested data rather all the data. Therefore it is necessary to discover an efficient and effective search service over encrypted outsourced data [1].

The existing search methods like ranked search, multi-keyword search enables the cloud customers to find the most relevant data rapidly. It also reduces the network traffic by directing the most relevant data to user request [1]. But In real search situateion it might be happen that user searches with the synonyms of the predefined keywords not the exact or fuzzy keywords, due to lack of the user's exact knowledge about the data. These approaches support only exact or fuzzy keyword search methods. That is there is no tolerance of synonym substitution and/or semantic variation which are the distinctive user searching behaviors happens very frequently. Therefore semantics based multi-keyword ranked search for encrypted cloud data remains a challenging problem [2].

To overwhelm this problem of effective search system the proposed system provides an efficient and flexible searchable scheme that provisions both multi-keyword ranked search and semantic based search. The Vector Space Model [1] is used to discourse multi-keyword search and result ranking. By using VSM, document index is built for each document i.e. each document is conveyed as vector where each dimension value is the Term Frequency (TF) weight of each equivalent keyword. Another vector is built in query phase. It has same dimension as that of document index and having dimension value as the Inverse Document Frequency (IDF) weight. Then cosine measure is used to compute the similarity between the document and the search query [2].To enhance the effectiveness of the search method we extend the keyword set with semantic words or natural language words for that each keywords. This will eventually support data retrieval on semantic query. Even if the user is unaware of the exact or synonyms of keywords of encrypted data, he/she can try to search it with its meaning in natural language. This makes the Semantic search more efficient and user need not to concern about the keyword generated for each particular word on the cloud by adapting this method data will be retrieved from the cloud in well secure manner and also cost can be minimized by employing this scheme into the structure [2].

II. PROBLEM DEFINITION

To develop a system which provides the semantic based search approaches over encrypted cloud data using semantic transformations and to provide the customers an efficient search, this gives the most relevant data according to the users query. The system provides higher search efficiency even when the user is unaware of the keyword substitution. Here the enhanced scheme is proposed for improving documental searches optimized for specific scenarios where user want to find a document but don't remember the exact words used, if plural or singular words were used or if a synonym was used. The system enables the data owners to control their outsourced data security and privacy, and verify security conditions, such as the integrity of data.

III. RELATED WORK

Zhangjie Fu, Xingming Sun, Nigel Linge and Lu Zhou offers an effective methodology that solves the problem of multi-keyword ranking search for encrypted cloud data which supports synonym queries. To discourse multi-keyword search and result ranking, Vector Space Model (VSM) is used to construct document index, that is , each document is conveyed as a vector and having dimension value as the Term Frequency (TF) weight for its equivalent keyword. Another vectoris created in the query phase. This vector is having the same dimension as that of document index and dimension value for this is the Inverse Document Frequency (IDF) weight. Then cosine measure can be used to calculate similarity of one document to the search query [1].

J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, uses the Fuzzy keyword search method that improves system usability by sending back only the matching files having exact match of the predefined keywords or the nearest possible matching files based on keyword similarity semantics measures, when *exact* match fails. They use the edit distance to compute keywords similarity and develop an advanced technique on constructing fuzzy keyword sets, which greatly reduces the storage and representation overheads [3].

C. Wang, N. Cao, J. Li, K. Ren, and W. Lou put forward the Ranked search that enhances system usability by sending back only the matching files to the customer in a ranked order according to relevance criteria as keyword frequency. It gives a straightforward yet ideal construction of ranked keyword search under the state-of-the-art searchable symmetric encryption (SSE) security definition, and demonstrates its inefficiency. To achieve more practical performance, they offer a definition for ranked searchable symmetric encryption, and give an efficient design by properly exploiting the existing cryptographic primitive and order-preserving symmetric encryption (OPSE) [4].

N. Cao, C. Wang, M. Li, K. Ren, and W. Lou describes a system that resolves the problem of privacy-preserving in multi-keyword ranked search for encrypted cloud data (MRSE)[5], and establish a set of strict privacy requirements for a secure cloud data utilization system. With various multi-keyword semantics, they use the efficient principle of "coordinate matching", i.e., as many matches as possible, to extract the similarity score between search query and data documents, and further use "inner product similarity" to quantitatively formalize such principle for similarity measurement [2].

W. Sun, B. Wang, M. Li, W. Lou, and Y. T. Hou present multi-keyword text search (MTS) scheme[6] using similarity-based ranking to solve the problem of privacy. To further improve the search privacy, they offer two secure index schemes to convene the strict privacy requirements with strong threat models. In particular, to support multi-keyword queries and search result ranking functionalities, they proposes to build the search index based on the vector space model [2].

IV. SYSTEM WORKFLOW

A. System Overview

The following Fig.1 shows the proposed system's architecture.

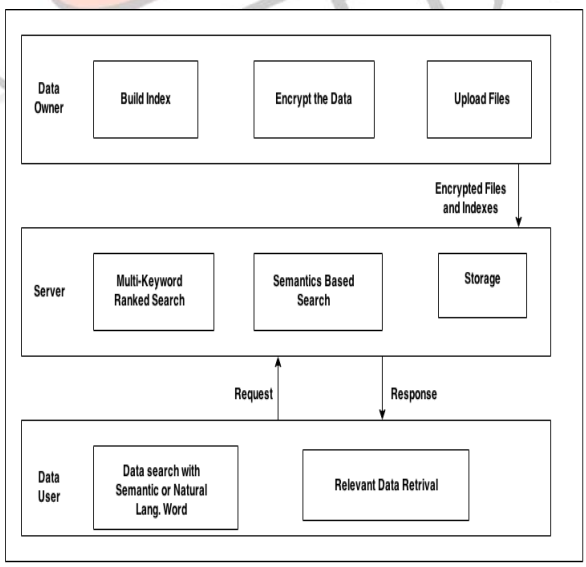


Figure1. System Architecture

The system consists of three entities as data owner, cloud server and data user. The data owner is either individual or an organization having a set of documents as document collection, then for each document a index is built. This index is expressed as

a vector with dimension value term frequency; another vector is also generated in query phase with same dimension and dimension value as inverted document frequency. To protect these documents data owner encrypts them along with the indexes and then uploaded on the cloud server.

Cloud server calculate the relevance score and apply the multi-keyword ranked search and semantic based search on it then store them on server to serve the data users. The data users are the customers who want to find most relevant data to their queries. The only authenticated users can search their queries on this system. The cloud server returns most relevant data to the user's query.

B. Algorithms

Algorithm 1: Key Generation

Key Generation algorithm is running on the data owner side, it generates the secret key PKi upon getting input of some security parameter using AES algorithm.
Data: We have received the request from client to insert file (F). Data Owner now wants to validate F format.
Result: Private Key (PKi) [key is used for encryption and decryption mechanism]
Begin
1. result = ValidateFormat(F);
ValidateFormat = (.html , .text , .java , .xml , .xsd , .js , .sql , .data , .log,....)
2. If the format is valid then client want to secure the data using AES Algorithm. CipherData(Fi) = AES(Fi);
After all the response has been generated PKi stores into internal db of SS(security service).The main Idea behind to hide PKi is provide security to CipherText(Bi) , So no one else can used the key and try to decrypt the block.

Algorithm 2 : FileUpload

FileUpload(F, PKi) → Y'. Here FileUpload takes as input the secret key K and F and outputs a tuple Y',
Data: System need to store CipherText(Bi) to the CSP along with Token (TiBi). System used the Referral Data integrity algorithm to associated the Bi → TiBi
Result: The data is store on the cloud into vault server with 2 level of indexing. Below is the indexing structure.
Index →→ DbServer →→ VaultServer
Begin
1. Along with the CipherText(Bi) and Token(TiBi) system start the generation of the metadata. Meta Data Contains Fields. -logged User Info (U-info) -File Name (Fname)
2. Once all the Data ciphterText (BiFi) has been process successfully, Data is send to the CSP for storage.
3. Various keyword sets are creates in database for each wordtype like synonyms, definitions and antonyms.

Algorithm 3 : DataRetrive

DataRet(PKi , Y') → F. The user uses the Private key Pki to retrieve or download the File F from protected answer Y' returned by cloud upon getting input of the secret key PKi and the answer Y' of PKi.
Data: User has received the request for retrieval of File (F) from CSP database.
Result: The file is downloaded to the client end.
Begin
1. System send the request to CSP (CSP - Metadata) to validate the request is valid or not i.e. F is present onto CSP or not. You will use the same indexing technique.
2. If F is available, System will retrieve the (F-Metadata) present on CSP , Then the input is given to the security Algorithm. i.e Fi= AES(DecryptData(Fi));

Algorithm 4 : FileSearch

This algorithm is used to find the most relevant files and these files are arranged in sorted order that decides the relevance of the files to the user's query words.
Data: User sends his request (keywords) to the CSP to find the relevant files.
Result: The ranked and relevant files are returned to the user's request
Begin
1. User sends the request to CSP in the form of keywords.
2. CSP checks the database to find the files containing the user's query keyword.
3. Different keyword sets are checked by the webCrawler function to extract the keywords.
HtmlDocument = WebCrawler(httpAddress, Word , ReleatedwordType);
Ksim / Kdef / Kanto = HtmlParser(HtmlDocument);
4. Ranking is done based on the total no of word counts in file. count = GetWordCount(Fi , K);
storeCount(Count , Fi , K);
where K is the Keyword database

5. Relevant files are returned to the user.

C. Mathematical Model :

Let the Proposed system can be expressed as
MRSS = {*F*, K, I, O, SS } Where

- F - The Plaintext File Collection
- K - Extended Keyword set with semantic Words
- I - Input keywords given by user
- O- Output set containing no. of relevant documents
- SS - Searchable Scheme

The searchable Scheme SS = (KeyGeneration, FileUpload, DataRetrive, Search) is secure.

K = (Ksim + Kdef + Kanto)
Ksim = { Ksim1 , Ksim2, Ksim3 , Ksim4 ,.....}
Kdef = { Kdef1 , Kdef2 , Kdef3 , Kdef4,.....}
Kanto = { Kanto1 , Kanto2 , Kanto3 , Kanto4,.....}
F = { F1,F2,F3, ...}

V. METHODOLOGY

A. Multi-Keyword Ranked Search:

The existing systems like exact or fuzzy keyword search, supports only single keyword search. These schemes doesn't retrieve the relevant data to users query therefore multi-keyword ranked search ranked search over encrypted cloud data remains a very challenging problem[1].To solve this problem, an efficient and flexible searchable technique is developed that supports multi-keyword ranked search. To deal with multi-keyword search and result ranking, Vector Space Model (VSM) [6] is used for building document index. In that each document is considered as a vector and each vector's dimension value is Term Frequency (TF) weight of its corresponding keyword. Another vector is also generated in the query phase where the vector has the dimension similar with document index and its dimension value is the Inverse Document Frequency (IDF) weight. Then cosine measure is used to calculate the similarity of one document to the user's search query [2].

To enhance the search efficiency, an index tree structure is generated which is a balance binary tree. Index tree is built with the document index vectors. Therefore the relevant documents can be searched by traversing tree [2].

B. Semantic Based Search:

When the user is searching the data on cloud server it might be possible that the user is unaware of the exact words or synonyms of the words which he/she want to search, i.e. there is no tolerance of synonym or semantic substitution which are the common user searching behaviors and that happens very frequently. To resolve this problem semantic based search scheme [8] is used. For improving the search for information on web it is necessary that search systems can understand what the user wants, so they are able to answer objectively [2].

The Semantic Web proposed to simplify the meaning of word by annotating them with metadata. Then by associating metadata to words, semantic searches can be significantly improved as compared to traditional keyword searches. Semantic search[8] allow users to use the natural language word to express what he wants to find. Here the enhanced scheme is proposed for improving documental searches optimized for specific scenarios where user want to search a document but doesn't remember the exact words used, if plural or singular words were used or if a synonym was used or if antonyms used. The method takes into consideration: 1) the number of direct words of the search queries that are in the document; 2) the number of word variation (plural/singular or different verbs conjugation) of the search query that are in the document; 3) the number of synonyms and also antonyms of the words in the search query that are in the document[2].

The proposed Semantic Search Scheme makes the search process more efficient. It could return not only the exactly matched files, but also the files including the terms semantically related to the query keyword. The concept of co-occurrence probability of terms is used to get the semantic relationship of keywords in the dataset. It offers appropriate semantic distance between terms to accomplish the query keyword extension[2]. To guarantee the security and efficiency, the data is encrypted before outsourced to cloud, and provides security to datasets, indexes and keywords also.

VI. RESULT AND DISCUSSION

A. Metadata Construction :

The time cost of each entry directly depends on the number of keywords in the file and the overall efficiency is also related to the number of the files in the document collection. Table 1 lists the metadata construction performance. For the reason to eliminate the difference of various file set construction choices, both the metadata size and construction time listed are the average value.

Table 1: File Metadata Construction Overhead

No. of Documents	Per file metadata size (KB)	Per file metadata build time
1000	0.18	0.28
2000	0.20	0.30

B. Search Efficiency:

The search process includes query expansion, calculating the total relevance score and ranking the result in descending order. Compared to the original ranked search, this approach introduces the keyword extension cost, and the calculation cost of final relevance score. So the size of semantically extended keywords set is a factor to the query efficiency.

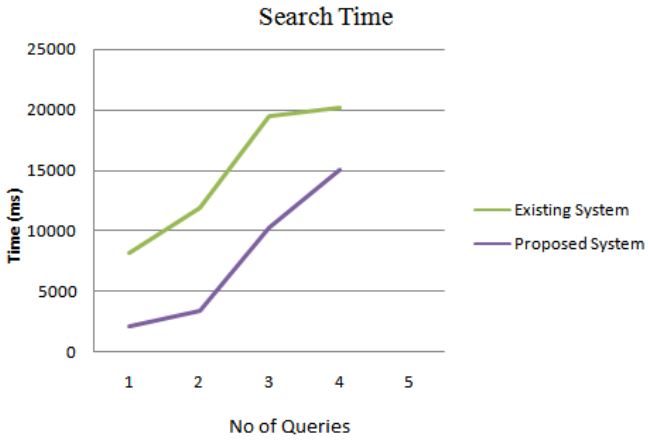


Figure 2. Time required to search documents in existing and proposed system.

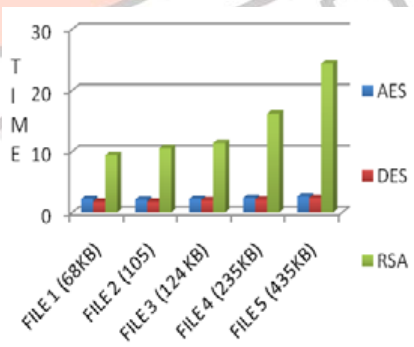
C. Security Analysis:

The following table 2 shows the security algorithm performance analysis on the basis of time required to encrypt and decrypt given input file size.

Table 2: Security algorithm performance analysis

Input File Size in (KB)	DES	AES	RSA
68	1.8	2.2	9.4
105	1.8	2.1	10.5
124	2.0	2.2	11.4
235	2.1	2.4	16.2
435	2.4	2.6	24.4

Figure 3 Shows Graph of comparison of security algorithms. The x -axis shows input image size in KB and y-axis shows the process time in seconds. The following figure shows the comparative graph which compares the performance of AES, DES and RSA algorithms on the basis of process time required to encrypt and decrypt provided input file size



.

Figure 3: Comparison of security algorithms

VII. CONCLUSION

The proposed Semantic Search Scheme makes the Search process more efficient. It could return not only the exactly matched files, but also the files including the terms semantically related to the query keyword. The concept of co-occurrence probability of terms is used to get the semantic relationship of keywords in the dataset. To guarantee the security and efficiency, the data is encrypted before outsourced to cloud, and provides security to datasets, indexes and keywords also. The overall performance evaluation of this scheme includes the cost of metadata construction, the time necessary to build index as well as the efficiency of search.

REFERENCES

[1] Zhangjie Fu, Xingming Sun, Nigel Linge and Lu Zhou, "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, February 2014.

[2] Dipika Chavan, Dinesh Yadav, "Achieving Efficiency of Encrypted Cloud Data with Synonym Based Search and Multi-Keyword Ranked Search" International Journal of Science and Research (IJSR) volume 4,issue 1,January 2015
[3] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," Proceedings of IEEE INFOCOM'10 Mini-Conference, San Diego, CA, USA, pp. 1-5, Mar. 2010.
[4] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," Proceedings of IEEE 30th International Conference on Distributed Computing Systems (ICDCS), pp. 253-262, 2010.
[5] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," Proceedings of IEEE INFOCOM 2011, pp. 829-837, 2011.
[6] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, and Y. T. Hou, "Privacy preserving multi-keyword text search in the cloud supporting similarity based ranking," ASIACCS 2013, Hangzhou, China, May 2013, pp. 71-82, 2013.
[7] C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data," IEEE Trans. Parallel and Distributed Systems, vol. 23, no. 8, pp. 1467- 1479, Aug. 2012.
[8] Sara Paiva,"A Fuzzy Algorithm for Optimizing Semantic Documental Searches", International Conference on Project Management / HCIST 2013.
[9] S. Kamara, and K. Lauter, "Cryptographic cloud storage," FC 2010 Workshops, LNCS 6054, PP. 136-149, Jan. 2010.
[10] C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data," IEEE Trans. Parallel and Distributed Systems, vol. 23, no. 8, pp. 1467- 1479, Aug. 2012.

Author's profiles

Dipika Chavan received Diploma degree in Computer Engineering from I.O.P.E. Lonere, Maharashtra and B.E. in Computer Engineering from RMCET, Ratnagiri, Maharashtra in 2007 and 2010 respectively. She is now pursuing M.E. in Computer Engineering at RSSOER, Pune, Maharashtra. Her area of interest is Cloud Computing, Data Mining.

Dinesh Yadav is Director of Rajarshi Shahu College of Engineering and Research, JSPM NTC Pune,India. He obtained Bachelor of Engineering, Masters of Engg. and PhD in Electronics and Telecommunication Engineering. His area of interest is Image Processing, Digital Signal Processing and Networking.