Implementation and evaluation of central keyword based semantic extension search scheme over encrypted outsourced data

Dipti D. Mehare[1], Prof. A. V. Deorankar[2]
[1]*P. G. Student, Department of Computer Science, Government college of Engineering, Amravati, Maharashtra, India.*
[2]*Assistant Professor, Department of Computer Science, Government college of Engineering, Amravati, Maharashtra, India.*

***Abstract*- In this paper, we design a central keyword semantic search based semantic extension scheme to improve the result delivery. Keywords may have a certain grammatical relationship among them which react the importance of keywords from the user's perspective intuitively. However, the existing search techniques regard the search keywords as independent and unrelated. For the first time, we take the relation among query keywords into consideration and design a keyword weighting algorithm to show the importance of the distinction among them. By introducing the keyword weight to the search protocol design, the search results will be more in line with the users demand. On top of this, we further design a novel central keyword semantic extension ranked scheme. By extending the central query keyword instead of all keywords, our scheme makes a good tradeoff between the search functionality and efficiency. To better express the relevance between queries and files, we further introduce the TF-IDF rule when building trapdoors and the index. Experiments on the real-world dataset show that our proposed schemes are efficient, effective and secure.**

I. INTRODUCTION

Cloud computing, the new term for the long dreamed vision of computing as a utility, enables convenient, on-demand network access to a centralized pool of configurable computing resource that can be rapidly deployed with great efficiency and minimal management overhead. The amazing advantages of Cloud Computing include: On-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing, and transference of risk. Thus, Cloud Computing could easily benefit its users in avoiding large capital outlays in the deployment and management of both software and hardware. Undoubtedly, Cloud Computing brings unprecedented paradigm shifting and benefits in the history of IT.As Cloud computing becomes increasingly popular; more people are inclined to outsource their data to the cloud, due to its exibility and unlimited resources. In addition, it can reduce local data maintenance costs and offer a convenient communication channel to share resources among the data owner and legitimate data users. Hence, a large amount of data, ranging from emails to personal health records etc. is increasingly outsourced to public clouds, such as Amazon Web Services, Microsoft Azure, Apple cloud and Google App Engine etc. However, storing data in the cloud also compromises data privacy, as the cloud servers are considered as semi-trusted or honest but curious. Thus, for privacy concerns, data owners must encrypt their (potentially) sensitive data before outsourcing. However, this renders traditional search schemes in the plaintext domain invalid. To enable effective searches over encrypted data, the data owner first builds an encrypted index based on the extracted keywords from data files and the corresponding index-based keyword matching algorithm, and then outsources both the encrypted data and the index structure to the cloud server. To search over the encrypted files, the cloud server integrates the trapdoors of keywords with the index information and finally returns the target files to the data users. There are three entities in our system: the data owner,

the data user and the cloud server. The data owner has a set of data files and wants to outsource it to the cloud. As these data files may contain sensitive information, the data owner encrypts the data before outsourcing, due to privacy concerns. To facilitate the efficient use of these encrypted data files, the data owner needs to build a searchable encrypted index based on the keyword set extracted from files. The index will then be outsourced to the cloud server along with the corresponding encrypted files. The data user is authorized by the data owner and searches the outsourced data files stored on the cloud via some input keywords. Based on these keywords, the user chooses the central keyword to extend, computes its trapdoor and sends it to the cloud server. The cloud server stores the encrypted data files and index, and also handles search requests from the data user. Upon receiving the trapdoor generated by an authorized user, the cloud server then searches over the index and returns the relevant files as the search results to the user.

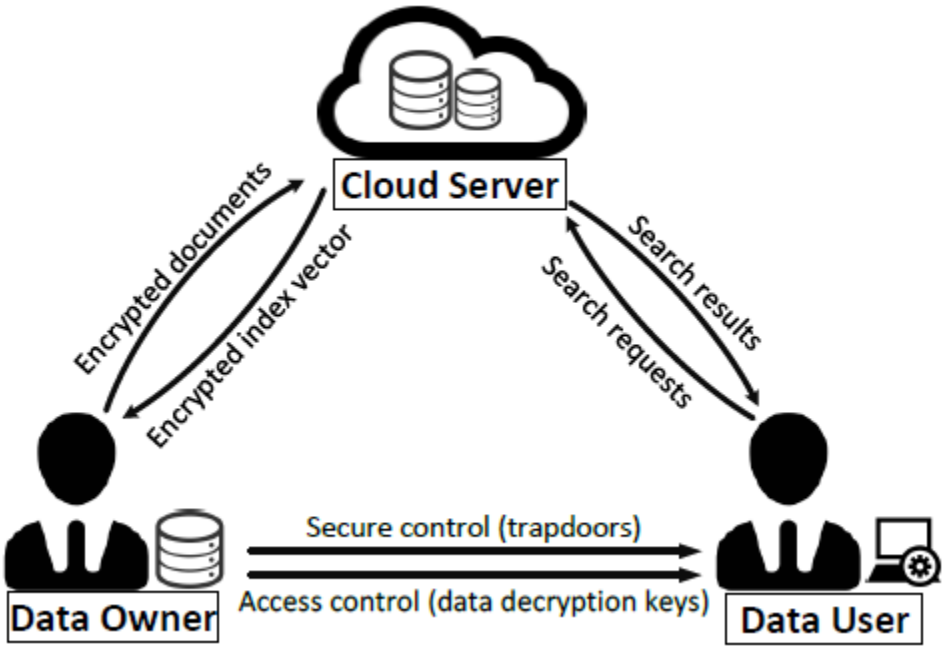

Fig No. 1-The System Model

II. SYSTEM IMPLEMENTAION

- **Objectives:**
  - To develop a cloud based search engine application.
  - To implement Modified AES algorithm for document security.
  - To improve keywords matching using separate XML for document keywords.
  - To implement caesar encryption algorithm to encrypt keywords of documents.
  - To implement quick keywords extraction method to extract keywords from document.
  - To improve searching performance of the system.

- **Scope of Problem:**

In existing system, when Data owner outsources his document on cloud, the document will get encrypted using AES and stored on cloud. After encryption when any other user wants to search that document, he will specify search query and trapdoor key. The search query will get processed to extract keywords. The keywords will be searched in every encrypted documents within the specified scope(document group as per the trapdoor key) to calculate document wise keywords weight(frequency).Keywords weight according search result re-ranking will be done.

Finally search result will be delivered to user. If user wants to download document, he have to specify secrete key. If secrete key is verified, documents will be decrypted and delivered it to user.

- **Implementation:**

In proposed system, when Data owner outsources his document on cloud, system will read the complete document and apply NLP on sentences to fetch keywords from the document. The keywords fetched from the document will be saved in XML file. Then the document will be encrypted using Advanced AES encryption algorithm and stored on cloud. The Advanced AES algorithm is a user defined algorithm, which divides the document into two parts .The two parts will get reversed and encrypted separately using different keys and AES algorithm. The separately encrypted parts will be merged and stored on cloud. After encryption when any other user wants to search that document, he will specify search query and trapdoor key. The search query will get processed to extract keywords. Synonyms for extracted keywords will be searched over www using any API. Keywords set will improved with available synonyms. The keywords will be searched in XML file rather than document which reduces search time. Keywords weights already saved in XML so there is no need to calculate the weight every time. Keywords weight according search result re-ranking will be done. Finally search result will be delivered to user. If user wants to download document, he have to specify secrete key. If secrete key is verified, documents will be decrypted and delivered it to user.

Advanced AES encryption algorithm:

1. **File Encryption: Steps**
   - Upload file groupwise.
   - Read file contents and find keywords from the file using NLP algorithm Store document details and keywords in an XML file using Caesar algorithm.
   - Read bytes of the file into a byte array.
   - Seperate the file into two parts.
   - Generate two keys for seperated parts of length 16 bytes array
     key1=PKey1_(current_year)@(5 digit random number),
     key2=PKey2_(current_year)@(5 digit random number)
   - Encrypt each part by its individual key using AES algorithm. Find length of the encrypted parts, enc_len1, enc_len2, respectively.
   - Merge the encrypted array s into a single byte array and reverse the newly merged array.
   - Upload the reversed array into a single file.
2. **File Decryption algorithm: steps**
   - Enter search keyword and trapdoor key into search engine panel.
   - Check whether the encryption key is correct and user requesting has permission to access the group.
   - Generate document details and decrypted file like decrypted keys, encrypted parts size , document name etc.
   - Read the decrypted file bytes and separate it into two parts.
   - Decrypt the separated parts by using their individual keys using AES algorithm.
   - Merge the decrypted parts into a single byte array and upload the decrypted array.
   - Download file.

III. EXPERIMENTAL EVALUATION

In this section, we present the experimental evaluation of the proposed scheme. We investigated the success of our scheme in the context of searching of keywords. To perform our evaluation, we use result of existing and proposed scheme. We enter central keyword for searching data and trapdoor key is entered according to group. Synonym for central keyword will search in API. Then data related to that all keyword will deliver to user. But Data will search for that keyword only that group. Result will show how many documents are present in existing and proposed database for that keyword.

Example:

Supposed we enter “Encryption” as a central keyword and trapdoor key for any group. Then evaluation will gives result for that keyword that how

many documents are present in existing and proposed scheme. Graphical representation is given below.

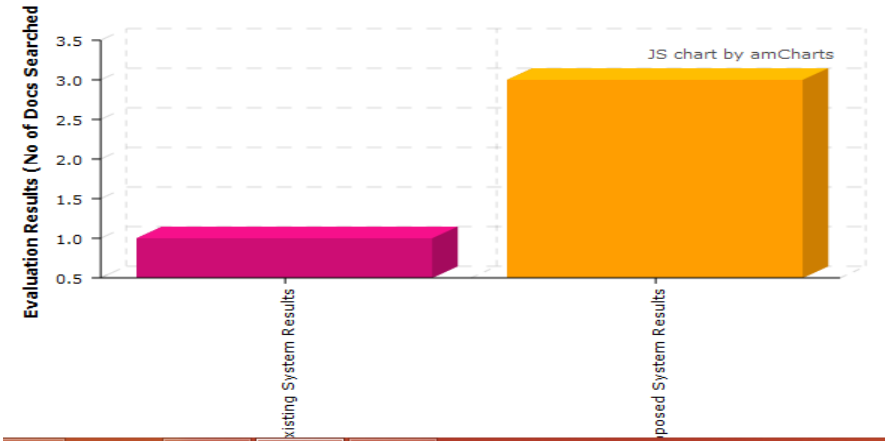

Fig No. 2- Evaluation Report

IV. CONCLUSIONS

In this report, for the first time, took the relationship among the query keywords into consideration and designed a keyword weighting algorithm based on the relations. Also designed a central keyword semantic extension scheme according to the keyword weights. By choosing the central keyword instead of not all the keywords to extend, our scheme achieves a tradeoff between functionality and efficiency. To express the relevance between the query and the files better, introduced the TF-IDF rule when building the trapdoor and index. By storing the IDF value in the dictionary, this scheme can support updates for adding new files. Moreover, this scheme can support efficient additions of the keyword collection.

REFERENCES

[1]. Zhangjie Fu, Xinle Wu, Qian Wang, Member "Enabling Central Keyword-based Semantic Extension Search over Encrypted Outsourced Data" IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY.

[2]. C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on. IEEE, 2010,
pp. 253–262.

[3]. J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in INFOCOM, 2010 Proceedings IEEE. IEEE, 2010, pp. 1–5.

[4]. W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li, "Privacy preserving multi-keyword text search in the cloud supporting similarity based ranking," in Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security. ACM, 2013, pp. 71–82.

[5]. N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multikeyword ranked search over encrypted cloud data," IEEE Transactions on parallel and distributed systems, vol. 25, no. 1, pp. 222–233, 2014.