International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 20 Issue 3 – MARCH 2016.

FINE-GRAINED MULTI-KEYWORD SEARCH OVER ENCRYPTED CLOUD DATA

N. Azarudheen[*1], S.D. Bharanidharan[#2] , S. Kavimugilan[$3] and Mrs. Subhashini[&4]

[*]*B.E, Computer Science and Engineering, Dhanalakshmi College Of Engineering, Chennai, India*
[#]*B.E, Computer Science and Engineering, Dhanalakshmi College Of Engineering, Chennai, India*
[$]*B.E, Computer Science and Engineering, Dhanalakshmi College Of Engineering, Chennai, India*
[&]*B.E, M.E, Assistant professor, Computer Science and Engineering, Dhanalakshmi College Of Engineering, Chennai, India*

***Abstract*--- Using cloud computing, individuals can store their data on remote servers an allow data access to public users through the cloud servers. As the outsourced data are likely to contain sensitive privacy information, they are typically encrypted before uploaded to the cloud. This, however, significantly limits the usability of outsourced data due to the difficulty of searching over the encrypted data. In this paper, we address this issue by developing the fine-grained multi-keyword search schemes over encrypted cloud data.It investigated on the fine-grained multi keyword search (FMS) issue over encrypted cloud data in two steps. The FMS I creates index to the encrypted data which will be stored in the cloud by the data owner. The FMS II achieves secure and efficient search using the search key. The extensibility of the file set and the multi-user cloud environments. Towards this direction, we have made some preliminary results on the extensibility and the multiuser cloud environments. Another interesting topic is to develop the highly scalable searchable encryption to enable efficient search on large practical databases.Lastly, we analyse the security of the proposed schemes in terms of confidentiality of documents, privacy protection of index and trapdoor, and unlinkability of trapdoor. Through extensive experiments using the real-world dataset, we validate the performance of the proposed schemes. Both the security analysis and experimental results demonstrate that the proposed schemes can achieve the same security level comparing to the existing ones and better performance in terms of functionality, query complexity and efficiency.**

***Index Terms*—Searchable encryption, Multi-keyword, Fine-grained, Cloud computing.**

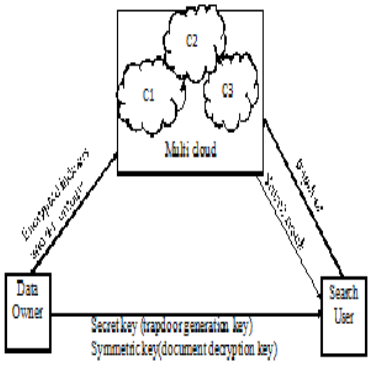
I. INTRODUCTION

The cloud computing treats computing as a utility and leases out the computing and storage capacities to the public individuals.The individual can remotely store her data on the cloud server, namely data outsourcing, and then make the cloud data open for public access through the cloud server.
This represents a more scalable, low-cost and stable way for public data access because of the scalability and high efficiency of cloud servers, and therefore is favorable to small enterprises.

Note that the outsourced data may contain sensitive privacy information. It is often necessary to encrypt the private data before transmitting the data to the cloud servers.

II. SYSTEM ARCHITECTURE

SYSTEM DESIGN



III. SYSTEM MODEL

As we consider a system consists of three entities.

• **Data owner**:
The data owner outsources her data to the cloud for convenient and reliable data access to the corresponding search users. To protect the data privacy, the data owner encrypts the original data through symmetric encryption. To improve the search efficiency, the data owner generates some keywords for each outsourced document. The corresponding index is then created according to the keywords and a secret key. After that, the data owner sends the encrypted documents

and the corresponding indexes to the cloud, and sends the symmetric key and secret key to search users.

• Cloud server:

The cloud server is an intermediate entity which stores the encrypted documents and corresponding indexes that are received from the data owner, and provides data access and search services to search users.

When a search user sends a keyword trapdoor to the cloud server, it would return a collection of matching documents based on certain operations.

• Search user:

A search user queries the outsourced documents from the cloud server with following three steps.

First, the search user receives both the secret key and symmetric key from the data owner. Second, according to the search keywords, the search user uses the secret key to generate trapdoor and sends it to the cloud server.

Last, she receives the matching document collection from the cloud server and decrypts them with the symmetric key.

IV. EXISTING SYSTEM

• The cloud computing, individuals can store their data on remote servers and allow data access to public users through the cloud servers.

• As the outsourced data are likely to contain sensitive privacy information, they are typically encrypted before uploaded to the cloud.

• This, however, significantly limits the usability of outsourced data due to the difficulty of searching over the encrypted data.

• It is **Single user cloud environments**.

Disadvantage:

• It is Single user cloud Environment.

• It is significantly limits the usability of outsourced data due to the difficulty of searching over the encrypted data.

V. PROPOSED SYSTEM

• The extensibility of the file set and the **multi-user cloud environments**. Towards this direction, we have made some preliminary results on the extensibility and the multiuser cloud environments.

• Another interesting topic is to develop the **highly scalable searchable encryption** to **enable efficient search on large practical databases**.

Advantage:

• It used Multi-user cloud Envirnments.

• Highly scalable searchable encryption

• Time efficient.

VI. PROPOSED SCHEMES

In this section, we firstly propose a variant of the secure kNN computation scheme, which serves as the basic framework of our schemes. Furthermore, we describe two variants of our basic framework and the corresponding functionalities of them in detail.

Basic Framework

The secure kNN computation scheme uses Euclidean distance to select k nearest database records. In this section, we present a variant of the secure kNN computation scheme to achieve the searchable encryption property.

1 .Initialization

The data owner randomly generates the secret key that is $K = (S; M1; M2)$, where S is a $(m+1)$-dimensional binary vector, $M1$ and $M2$ are two $(m + 1) \times (m + 1)$ invertible matrices, respectively, and m is the number of keywords in W. Then the data owner sends $(K; sk)$ to search users through a secure channel, where sk is the symmetric key used to encrypt documents outsourced to the cloud server.

.2 Index building

The data owner firstly utilizes symmetric encryption algorithm (e.g., AES) to encrypt the document collection $(F1; F2; \cdot ; FN)$ with the symmetric key sk [23], the encrypted document collection are denoted as $Cj(j = 1; 2; \cdot\cdot ;N)$.

Then the data owner generates an m-dimensional binary vector P according to $Cj(j = 1; 2; \cdot\cdot\cdot ;N)$, where each bit $P[i]$ indicates whether the encrypted document contains the keyword wi, i.e., $P[i] = 1$ indicates yes and $P[i] = 0$ indicates no.

Then she extends P to a $(m + 1)$-dimensional vector P', where $P'[m + 1] = 1$. The data owner uses vector S to split P' into two $(m + 1)$-dimensional vectors $(pa; pb)$, where the vector S functions as a splitting indicator.

Namely, if $S[i] = 0(i = 1; 2; \cdot ; m + 1)$, $pa[i]$ and $pb[i]$ are both set as $P'[i]$; if $S[i] = 1(i = 1; 2; \cdot ; m + 1)$, the value of $P'[i]$ will be randomly split into $pa[i]$ and $pb[i]$ $(P'[i] = pa[i]+pb[i])$. Then, the index of encrypted document Cj can be calculated as $Ij = (paM1; pbM2)$. Finally, the data owner sends $Cj \,||FIDj\, ||Ij$ $(j = 1; 2; \cdot\cdot\cdot ;N)$ to the cloud server.

3 Trapdoor generating

The search user firstly generates the keyword set fW for searching. Then, she creates a m dimensional binary vector Q according to fW, where $Q[i]$ indicates whether the i-th keyword of dictionary wi is in fW, i.e., $Q[i] = 1$ indicates yes and $Q[i] = 0$ indicates no. Further, the search user extends Q to a $(m + 1)$-dimensional vector Q', where $Q'[m + 1] = -s$ (the value of $-s$ will be defined in the following schemes in

detail). Next, the search user chooses a random number $r > 0$ to generate $Q'' = r \cdot Q'$.
Then she splits Q'' into two $(m + 1)$ vectors $(qa; qb)$: if $S[i] = 0(i = 1; 2; \cdots ; m + 1)$, the value of $Q''[i]$ will be randomly split into $qa[i]$ and $qb[i]$; if $S[i] = 1(i = 1; 2; \cdots ; m + 1)$, $qa[i]$ and $qb[i]$ are both set as $Q''[i]$. Thus, the search trapdoor TfW can be generated as $(M^{-1}1\ qa; M^{-1}2\ qb)$. Then the search user sends TfW to the cloud server.

VII. CAMELLIA ALGORITHM

All symmetric block cipher algorithms share common characteristics and variables, including mode, key size, weak keys. The following sections contain descriptions of the relevant characteristics of Camellia.

Mode

NIST has defined five modes of operation for AES and other FIPS-approved ciphers [SP800-38a]: CBC (Cipher Block Chaining), ECB (Electronic Codebook), CFB (Cipher Feedback), OFB (Output Feedback),and CTR (Counter). The CBC mode is well defined and well understood for symmetric ciphers, and it is currently required for all other ESP ciphers. This document specifies the use of the Camellia cipher in CBC mode within ESP.

Key Size

Camellia supports three key sizes: 128 bits, 192 bits, and 256 bits. The default key size is 128 bits, and all implementations must support this key size. Implementations may also support key sizes of 192 bits and 256 bits. Camellia uses a different number of rounds for each of the defined key sizes. When a 128-bit key is used, implementations must use 18 rounds. When a 192-bit key is used, implementations must use 24 rounds.

Weak Keys

At the time of writing this document, there are no known weak key for Camellia.

Performance

Performance figures of Camellia are available at [Camellia-Web].This web site also includes performance comparison with the AES cipher and other AES finalists. The NESSIE project [NESSIE] has reported performance of Optimized Implementations independently.

VIII. MODULES DESCRIPTION

Data owner

The data owner outsources her data to the cloud for convenient and reliable data access to the corresponding search users. To protect the data privacy, the data owner encrypts the original data through symmetric encryption. To improve the search efficiency, the data owner generates some keywords for each outsourced document. The corresponding index is then created according to the keywords and a secret key. After that, the data owner sends the encrypted documents and the corresponding indexes to the cloud, and sends the symmetric key and secret key to search users.

Cloud server

The cloud server is an intermediate entity which stores the encrypted documents and corresponding indexes that are received from the data owner, and provides data access and search services to search users. When a search user sends a keyword trapdoor to the cloud server, it would return a collection of matching documents based on certain operations.

Trapdoor generation

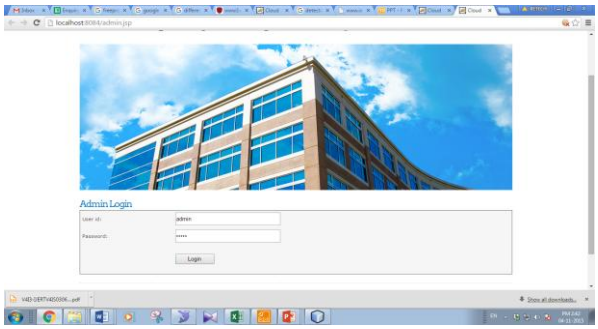
The documents stored in the cloud server may be searched many times. The cloud server should not be able to learn any keyword information according to the trapdoors, e.g., to determine two trapdoors which are originated from the same keywords. Otherwise, the cloud server can deduce relationship of trapdoors, and threaten to the privacy of keywords. Hence the trapdoor generation function should be randomized, rather than deterministic. Even in case that two search keyword sets are the same, the trapdoors should be different.

Search user

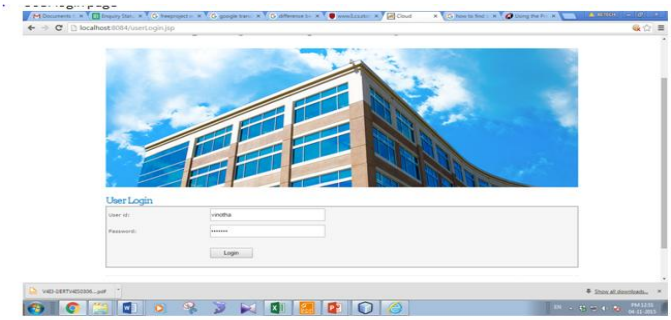
A search user queries the outsourced documents from the cloud server with following three steps. First, the search user receives both the secret key and symmetric key from the data owner. Second, according to the search keywords, the search user uses the secret key to generate trapdoor and sends it to the cloud server. Last, she receives the matching document collection from the cloud server and decrypts them with the symmetric key.

Home page

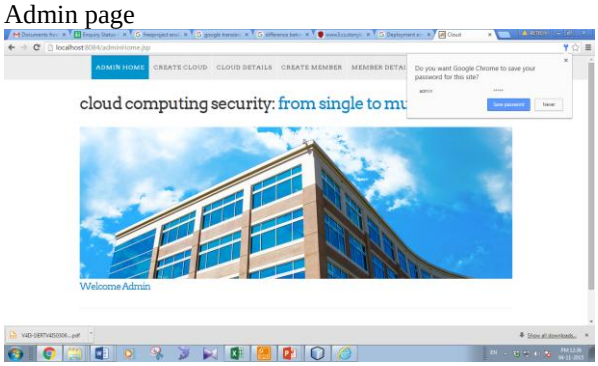
Admin login



User login page

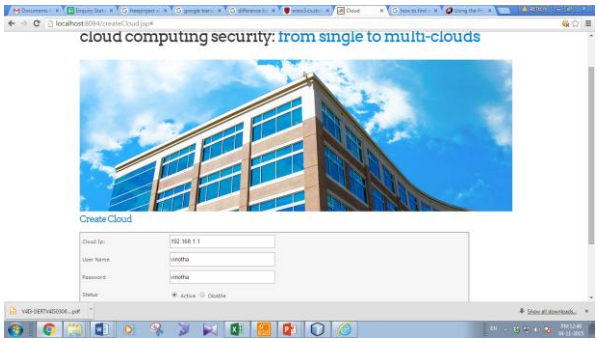


Admin page

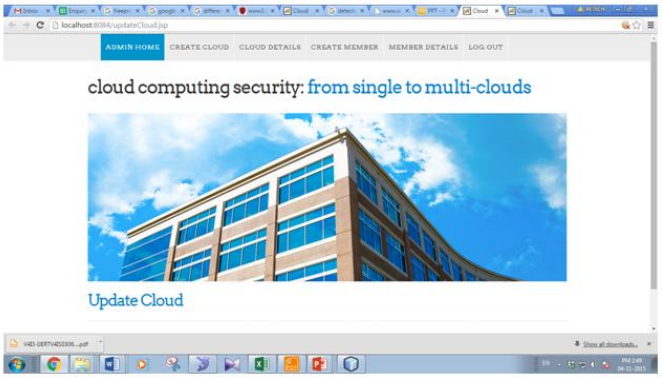


Login success page

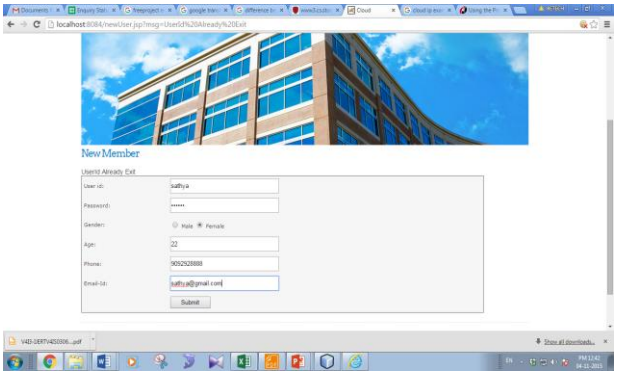
Create cloud



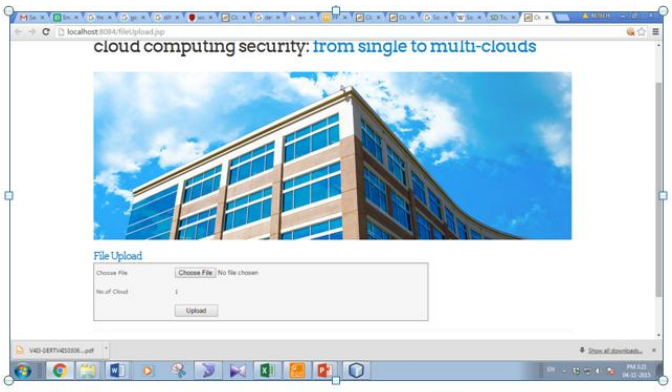
Cloud update page



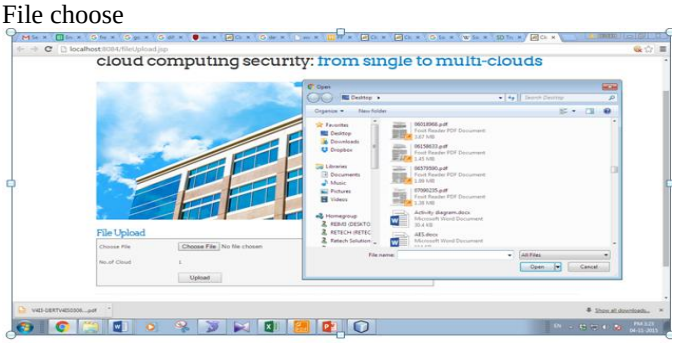
User Registration page



File upload page



File choose



IX. CONCLUSION

The extensibility of the file set and the multi-user cloud environments. Towards this direction, it have made some preliminary results on the extensibility and the multiuser cloud environments. Another interesting topic is to develop the highly scalable searchable encryption to enable efficient search on large practical databases. FMSCS(fine-grained multi keyword search classified sub-dictionaries) to improve efficiency. It use AES algorithm and indexing method for the purpose of secure and easy to access the file in multi cloud Environment.

REFERENCE

[1] 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.

[2] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in Applied Cryptography and Network Security. Springer, 2004, pp. 31–45.

[3] H. Liang, L. X. Cai, D. Huang, X. Shen, and D. Peng, "An smdp based service model for interdomain resource allocation in mobile cloud networks," IEEE Transactions on Vehicular Technology, vol. 61, no. 5, pp. 2222–2232, 2012.

[4] H. Li, D. Liu, Y. Dai, T. H. Luan, and X. Shen, "Enabling efficient multi-keyword ranked search over encrypted cloud data through blind storage," IEEE Transactions on Emerging Topics in Computing, 2014, DOI10.1109/TETC.2014.2371239.

[5] C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data," IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 8, pp. 1467–1479, 2012.

[6] W. K. Wong, D. W.-l. Cheung, B. Kao, and N. Mamoulis, "Secure knn computation on encrypted databases," in Proceedings of SIGMO International Conference on Management of data. ACM, 2009, pp.139–152.