Fine-grained Multi-keyword Search Over Encrypted Cloud Data

S.Vinotha[1], M.P.Sujatha[2]
[1]PG Student, [2]Professor
[1 2] Department of CSE, [1 2] Sri Ramanujar Engineering College, TamilNadu, India

Abstract-- Data are stored in remote servers and can be shared for public access using cloud computing. Individuals when uploading data to shared remote servers usually encrypt their data from preventing the unauthorized users to access their sensitive information. This limits the information for searching the outsourced data. The above mentioned issue overcomes by developing the fine-grained multi-keyword search schemes over encrypted cloud data. First, to enable the precise keyword search and personalized user experience the relevance scores and preference factors upon keywords are derived. Second, a practical and efficient multi-keyword search scheme which supports complicated logic search using "AND", "OR" and "NO" operations over keywords. To achieve better efficiency on index building, trapdoor generating and query, the system employes classified sub-dictionaries technique. using sub-dictionaries techniques searching process can be made easier . The trapdoor algorithm used to generate secret key to the user to access information from multi-cloud. The multi-cloud response the search result to the user.

Keywords: *Multi-keyword,Trapdoor,Cloud computing, Searchable encryption.*

I. INTRODUCTION

The cloud computing 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. 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. The data encryption would significantly lower the usability of data due to the difficulty of searching over the encrypted data. Simply encrypting the data may still cause other security concerns. For instance, Google Search uses SSL (Secure Sockets Layer) to encrypt the connection between search user and Google server when private data, such as documents and emails, appear in the search results. However, if the search user clicks into another website from the search results page, that website may be able to identify the search terms that the user has used Issues, the searchable encryption has been recently developed as a fundamental approach to enable searching over encrypted cloud data, which proceeds the following operations.Firstly, the data owner needs to generate several keywords according to the outsourced data. These keywords are then encrypted and stored at the cloud server. When a search user needs to access the outsourced data, it can select some relevant keywords and send the cipher text of the selected keywords to the cloud server.

The cloud server then uses the cipher text to match the outsourced encrypted keywords, and lastly returns the matching results to the search user. To achieve the similar search efficiency and precision over encrypted data, a ranked keyword search scheme which considers the relevance scores of keywords. Propose a multi-keyword text search scheme which considers the relevance scores of keywords and utilizes a multidimensional tree technique to achieve efficient search query. Propose a multi-keyword retrieval scheme which uses fully homomorphic encryption to encrypt the index/trapdoor and guarantees high security. Propose a multi-keyword ranked search (MRSE),which applies coordinate machine as the keyword matching rule,i.e.,returndata with the most matching keywords. Although many search functionalities have been developed in previous literature towards precise and efficient searchable encryption, it is still difficult for searchable encryption to achieve the same user experience as that of the plaintext search, like Google search. This mainly attributes to following two issues.

Firstly, query with user preferences is very popular in the plaintext search. It enables personalized search and can more accurately represent user's requirements, but has not been thoroughly studied and supported in the encrypted data domain. Secondly, to further improve the user's experience on searching, an important and fundamental function into enable the multi-keyword search with the comprehensive logic operations, i.e., the "AND", "OR" and "NO" operations of keywords. This is fundamental for search users to prune the searching space and quickly identify the desired data. propose the coordinate matching search scheme(MRSE) which can be regarded as a searchable encryption scheme with "OR"operation.propose conjunctive keyword search scheme which can be regarded as a searchable encryption scheme with "AND" operation with the returned documents matching all keywords. However, most existing proposals can only enable search with single logic operation, rather than the mixture of multiple logic operation son keywords, which motivates our work.

In this work, we address above two issues by developing two Fine-grained Multi-keyword Search (FMS) schemes over encrypted cloud data. Our original contributions can be summarized in three aspects as follows:

We introduce the relevance scores and the preference factors of keywords for searchable encryption. The relevance scores of keywords can enable more precise returned results, and the preference factors of keywords represent the importance of keywords in the search keyword set specified

by search users and correspondingly enables personalized search to cater to specific user preferences. It thus further improves the search functionalities and user experience.

We realize the "AND", "OR" and "NO" operations in thematic-keyword search for searchable encryption. Compared with the proposed scheme can achieve more comprehensive functionalityand lower query complexity.

We employ the classified sub-dictionaries technique to enhance the efficiency of the above two schemes. Extensive experiments demonstrate that the enhanced schemes can achieve better efficiency in terms of index building, trapdoor generating and query in the comparison.

II. SYSTEM ARCHITECTURE,MODELSAND SECURITY REQUIREMENTS

A. System Architecture

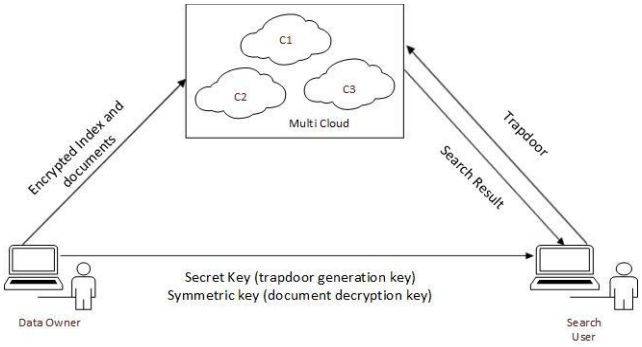


Fig. 1. System Architecture

System Architecture diagram describes the overall process between the data owner and search user. The data owner provides two kind of keys such as "Secret Key" and "Symmetric Key" to the requested user. Requested user who want to access the cloud and download the required documents from multi-cloud.

The authorized user can access the cloud through the "Secret Key" with the required document "keyword". Data owner search based on the "Trapdoor method" search on multi-cloud index, after getting the data from the cloud, data owner decrypt using "Symmetric Key" and sent it back to the search user.

B. System Model

Shown in Fig.1, we consider a system consists of entities.

Data owner: The data owner outsources her data tithe 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 frothed cloud server and decrypts them with the symmetric key.

C. SECURITY REQUIREMENTS

The information available to the cloud server know as hypertext model and background model. we assume search users are trustedentities,and they share the same symmetric key and secretkey.Search users have pre-existing mutual trust with the data owner. we do not consider following issues,including the access control problem on managing decryption capabilities given to users and the data collection's updating problem on inserting newdocuments,updating existing documents and deleting existingdocuments,areseparated issues. Based on the above threat model, we define the security requirements as follows:

Confidentiality of documents: The outsourced documents provided by the data owner are stored in the cloud server. If they match the search keywords, they are sent to the search user. Due to the privacy of documents, they should not be identifiable except by the data owner and theauthorized search users.

Privacy protection of index and trapdoor: The index and the trapdoor are created based on the document keywords and the search keywords,respectively.If the cloud server identifies the content of index or trapdoor, and further deduces any association between keywords and encrypted documents.Therefore,the content of index and trapdoor cannot be identified by the cloud server.

Unlink ability of trapdoor: 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.

III. PRELIMINARIES

In this section, we define the notation and review the securing computation and relevance score, which will serve as the basis of the proposed schemes.

A. Notation

F-the document collection to be outsourced, denoted asa set of N documents= (F1; F2; · ·· ; FN).C-the encrypted document collection according to F,denoted as a set of N documents=(C1;C2; · ·· ;CN).

FID-the identity collection of encrypted documents C,denotedasFID= (FID1; FID2; · ·· ; FIDN).W-the keyword dictionary, including m keywords, denoted as W = (w1;w2; · ·· ;wm).

I-the index stored in the cloud server, which is built from the keywords of each document, denoted as I =(I1; I2; · ·· ; IN).fW-the query keyword set generated by a search user, which is a subset of W.TfW—the trapdoor for keyword set fW.FID—the identity collection of documents returned tote search user.FMS(CS)—the abbreviation of FMS and FMSCS.

B. Secure kNN Computation

propose a secure k-nearest neighbor (kNN) scheme which can confidentially encrypt two vectors and compute Euclidean distance of them.Firstly,the secret key (S;M1;M2)should be generated. The binary vector S is a splitting indicator to split plaintext vector into two random vectors, whichcan confuse the value of plaintext vector.AndM1 and M2 are used to encrypt the split vectors.

C. Relevance Score

The relevance score between a keyword and a document represents the frequency that the keyword appears in thedocument.It can be used in searchable encryption for returning ranked results. A prevalent metric for evaluating the relevance score is TF × IDF, where TF (term frequency) represents (inverse document frequency) represents the importance of

Keyword within the whole document collection. Without loss of generality, we select a widely used expression into evaluate the relevance score as

$$Score(\widetilde{\mathcal{W}}, F_j) = \sum_{w \in \widetilde{\mathcal{W}}} \frac{1}{|F_j|} \cdot (1 + \ln f_{j,w}) \cdot \ln(1 + \frac{N}{f_w}) \quad (1)$$

Where fj; w denotes the TF of keyword w in document Fj;fw denotes the number of documents contain keyword w; N denotes the number of documents in the collection; and |Fj |denotes the length of Fj, obtained by counting the number of indexed keywords.

IV. 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.

A. Basic Framework

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

1) Initialization

The data owner randomly generates the secret key K =(S;M1;M2)

whereS is a (m+1)-dimensional binary vector,M1 and M2 are two (m+ 1) × (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 skis 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; · ·· ; FN) with the symmetric key sk[23], the encrypted document collection are denoted as Cj(j = 1; 2; · ·· ;N).

Then the data owner generates an dimensional binary vector P according toCj(j =1; 2; · ·· ;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] = 0indicates no. Then she extends P to a (m+ 1)-dimensional vector P,whereP′[m + 1] = 1. The data owner usesvector 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; · ·· ;m + 1), pa[i] and pb[i]are both set as P′[i]; if S[i] = 1(i= 1; 2; · ·· ;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 documentCjcan be calculated as Ij= (paM1; pbM2). Finally, the data owner sends Cj ||FIDj ||Ij(j = 1; 2; · ·· ;N) to the cloud rapdoorgenerating.The search user firstly generates the keyword set fWforsearching. Then, she creates am-dimensional binary vector According to fW, where Q[i] indicates whether the i-th keyword of dictionary wiis in fW, i.e., Q[i] = 1 indicates yes andQ[i] = 0 indicates no. Further, the search user extends Q toa (m + 1)-dimensional vector Q′, whereQ′[m + 1] =−s(the value of −s will be defined in the following schemes in detail).Next, the search user chooses a random numberr>0 to generate Q′′ = r · Q′. Then she splits Q′′ into two (m + 1) vectors (qa; qb): if S[i] = 0(i= 1; 2; · ·· ;m + 1),

the value of Q″[i] will be randomly split into qa[i] and qb[i];if S[i] = 1(i= 1; 2; · ·· ;m + 1), qa[i] and qb[i] are both set as Q″[i].

3) Query

The index Ij(j = 1; 2; · ·· ;N) and trapdoor T the

cloud server calculates the query result as

$$R_j = I_j \cdot T_{\widetilde{W}} = (p_a M_1, p_b M_2) \cdot (M_1^{-1} q_a, M_2^{-1} q_b)$$
$$= p_a \cdot q_a + p_b \cdot q_b = P' \cdot Q''$$
$$= rP' \cdot Q' = r \cdot (P \cdot Q - s)$$

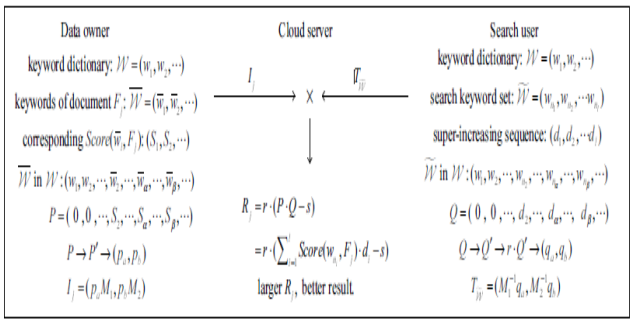
If Rj>0, the corresponding document identity FIDj willbe returned.

B. FMS I

In the Framework, P is a m-dimensional binary vector, and each bit P[i] indicates whether the encrypted document contains the keyword wi.In the FMS I, the data owner first calculates the relevance score between the keyword wi and document Fj. The relevance score can be calculated as follows:

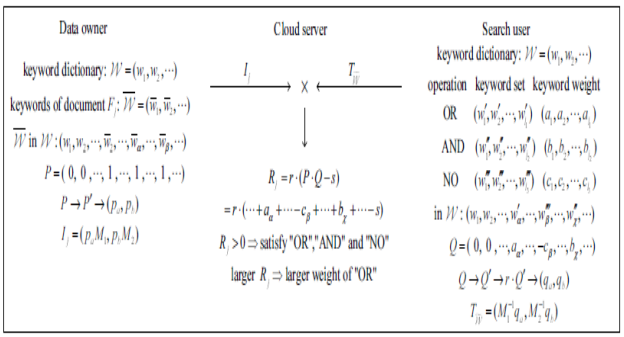
$$Score(w_i, F_j) = \frac{1}{|F_j|} \cdot (1 + \ln f_{j,w_i}) \cdot \ln(1 + \frac{N}{f_{w_i}}) \qquad (3)$$

where fj;widenotes the TF of keyword wiin document Fj;fwidenotes the number of documents contain keyword wi; N denotes the number of documents in the collection; and |Fj |denotes the length of Fj, obtained by counting the number ofindexed keywords. Then the data owner replaces the value of P[i] with the corresponding relevance score. On the other hand, we also consider the preference factors of keywords. The preference factors of keywords indicate the importance of keywords in the search keyword set personalized defined by the searchuser.For a search user, he may pay more attention to the preference factors of keywords defined by him than the relevance scores of the keywords. Thus, our goal is that if a document has a keyword with larger preference factor than other documents, it should have a higher priority inthe returned FID; and for two documents, if their largest preference factor keywords are the same, the document with higher relevance score of the keyword is the better matching result.



C. FMS II

In the FMS II, we do not change the vector P in the Basic Framework, but replace the value of Q[i] by the weight of search keywords, as shown in Fig. 3. With the weight of keywords, we can also implement some operations like "OR","AND" and "NO" in the Google Search to the searchable encryption.



V. PERFORMANCE EVALUATION

We evaluate the performance of the proposed schemes using simulations, and compare the performance with that of existing proposals.

A. Functionality

We compare FMS(CS) I and FMS(CS) II, respectively. MRSE [6] can achieve multi-keyword search and coordinate matching using secure kNN computation scheme. Considers the relevance scores of keywords. Compared with the other schemes, our FMS(CS) I considers both the relevance scores and the preference factors of keywords.

That if the search user sets all relevance scores and preference factors of keywords as the same, the FMS(CS)I degrades to MRSE and the coordinate matching can be achieved. And in the FMS(CS) II,if the search user sets all preference factors of "OR" operation keywords as the same,

the FMS(CS) II can also achieve the coordinate matching of "OR" operation keywords. Particularly, the FMS(CS) I achieves some fine-grained operations of keyword search,i.e., "AND", "OR" and "NO" operations in Google Search, which are definitely practical and significantly enhance the functionalities of encrypted keyword search.

B. Query Complexity

In the FMS(CS) II, we can implement "OR", "AND"and"NO" operations by defining appropriate weights of keywords, this scheme provides a more fine-grained. If the keywords to perform "OR", "AND"and"NO"operationsare $(w1', w2', \ldots, wl1')$, $(w1'', w2'', \ldots, wl2'')$ and$(w1''', w2''', \ldots, wl3''')$respectively.Our FMS(CS) II cancomplete the search with only they would complete the search through the following steps:

The "OR" operation of l1 keywords, they need only one query Query$(w1', w2', \ldots, wl1')$to return a collectionof documents with the most matching keywords (i.e.,coordinate

matching), which can be denoted as X =Query(w1′, w2′, … , wl1′)

The "AND" operation of l2 keywords cannot generate a query for multiple keywords to achieve the "AND" operation. Therefore, after costingl2 queries Query(w′′)=(i = 1,2 … , l2) they can do the "AND" operation, and the corresponding document set can be denoted as Y=Query(w1″)∩Query(w2″)∩⋯∩Query(wl″2)

The "NO" operation of l3 keywords, they need l3queriesQuery (w′′′i)(i= 1; 2; · ·· ; l3), firstly. Then, thedocument set of the "NO" operation can be denoted as

Z=Query (w1‴)∩Query(w2‴)∩⋯∩Query(wl‴3)

Finally, the document collection achieved "OR", "AND"and "NO" operations can be represented as X∩Y∩Z.

VI. RELATED WORK

There are mainly two types of searchable encryption in literature, Searchable Public-key Encryption (SPE) and Searchable Symmetric Encryption (SSE).

A. SPE

SPE is first proposed by supports single keyword search on encrypted data but the computation overhead is heavy. In the framework of SPE, propose conjunctive, subset, and range queries on encrypted data. propose a conjunctive keyword scheme which supports multi-keyword search. an efficient public key encryption with conjunctive subset keywords search. However, these conjunctive keywords schemes can only return the results which match all the keywords simultaneously, and cannot rank the returned results.

B. SSE

The concept of SSE is first developed the ranked keyword search scheme, which considers the relevance score of a keyword. However, the above schemes cannot efficiently support multi-keyword search which is widely used to provide the better experience to the search user.

This approach can return the ranked results of searching according to the number of matching keywords. Within this framework, they leverage an efficient index to further improve the search efficiency, and adopt the blind storage system to conceal access pattern of the search user an authorized and ranked multikeywordsearch scheme (ARMS) over encrypted cloud data by leveraging the cipher text policy attribute-based encryption(CP-ABE) and SSE techniques. Security analysis demonstrates that the proposed ARMS scheme can achieve collusion resistance.

In this paper, we propose FMS(CS) schemes which not only support multi-keyword search over encrypted data, butalso achieve the fine-grained keyword search with the function

to investigate the relevance scores and the preference factors of keywords and, more importantly, the logical rule of keywords. In addition, with the classified sub-dictionaries, our proposals efficient in terms of index building, trapdoor generating and query.

VII. CONCLUSION AND FUTURE WORK

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.

REFERENCES

[1]. C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure KNN Computation on Encrypted Database," in Proceedings of ICDCS. IEEE 2010.

[2]. WenhaiSun,BingWang,NingLi,WenjingLou,Y.ThomasHou,Fellowand Hui Li," VerifiablePrivacy-Preserving Multi-KeywordText Search in the Cloud SupportingSimilarity-Based Ranking", IEEE Transactions on paprallelDistrbuteSystem,November 2014.

[3]. T. Jung, X. Mao, X. Li, S.-J. Tang, W. Gong, and L. Zhang, "Privacypreserving data aggregation without secure channel: multivariate polynomial evaluation," in Proceedings of INFOCOM. IEEE,Aprl 2013.

[4]. Q. Shen, X. Liang, X. Shen, X. Lin, and H. Luo, "Exploiting geodistributed clouds for e-health monitoring system with minimum service delay and privacy preservation," IEEE Journal of Biomedical and Health Informatics,March2014.

[5]. Y. Yang, H. Li, W. Liu, H. Yang, and M. Wen, "Secure dynamic searchable symmetric encryption with constant document update cost,"inProceedings of GLOBCOM. IEEE, 2014, to appear.

[6]. 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

[7]. [7] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proceedings of S&P. IEEE, 2000, pp. 44–55.

[8]. R. Li, Z. Xu, W. Kang, K. C. Yow, and C.-Z. Xu, "Efficient multikeywordranked query over encrypted data in cloud computing," FutureGeneration Computer Systems, vol. 30, pp. 179–190, 2014.

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