IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

NCRTET-17

National Conference on Recent Trends in Engineering & Technology

Government Polytechnic College Kasaragod, Kerala

Vol. 4, Special Issue 1

Multikeyword Search Supporting Classified Sub-Dictionary Over Encrypted Cloud Data

NAYANA VM
Student, LBS College Of Engineering Kasaragod, Kerala, India

Abstract: Cloud computing is a method for delivering information technology (IT) services in which resources are retrieved from the Internet through web-based tools and applications, as opposed to a direct connection to a server. Rather than keeping files on a proprietary hard drive or local storage device, cloud-based storage makes it possible to save them to a remote database. Cloud computing is a forth coming revolution in information technology (IT) industry because of its performance , accessibility, low cost and many other luxuries. It is an approach to maximize the capacity or step up capabilities vigorously without investing in new infrastructure, nurturing new personnel or licensing new software. It provides gigantic storage for data and faster computing to customers over the internet. It essentially shifts the database and application software to the large data centers, i.e., cloud, where management of data and services may not be completely trustworthy. To keep the outsourced data more secure ,the owner of the data will encrypt the data using AES encryption method. And owner provide the service to search and use these outsourced data by other users in the cloud. But it difficult perform the search on the encrypted outsourced data, for these First, I am introduce the relevance scores and preference factors upon keywords which enable the precise keyword search and personalized user experience. Second, I develop a practical and very efficient multi-keyword search scheme. The proposed scheme can support complicated logic search the mixed “AND”, “OR” and “NO” operations of keywords. Third, I further employ the classified sub-dictionaries technique to achieve better efficiency on index building, trapdoor generating and query. Through this I will achieve efficient search .

I. INTRODUCTION

Cloud computing is an casually using the real-time communication network and connect large number of computers that depict different types of computing concepts. Non-ambiguous technical or scientific description in cloud computing has not been accepted . In science, cloud computing is a one kind of the distributed computing network and capability to run a program on many related computers at the similar time. Cloud computing is called as a utility of the computing since it uses pay per use paradigm. In cloud computing, users can also right to use a variety of resources like storage, programs, and application development platforms. cloud computing is an emerging technology and it is also called as utility because client are used to store their data in the cloud server. In cloud server data can also be leaked to hackers therefore encrypted the data before sent to the cloud for data privacy.

With the prevalence of cloud services, more and more sensitive information are being centralized into the cloud servers, such as emails, personal health records, private videos and photos, company finance data, government documents, etc. To protect data privacy and combat unsolicited accesses, sensitive data has to be encrypted before outsourcing so as to provide end-to-end data confidentiality assurance in the cloud and beyond. However, data encryption makes effective data utilization a very challenging task given that there could be a large amount of outsourced data files. Besides, in Cloud Computing, data owners may share their outsourced data with a large number of users, who might want to only retrieve certain specific data files they are interested in during a given session. One of the most popular ways to do so is through keyword-based search. Such keyword search technique allows users to selectively retrieve files of interest and has been widely applied in plaintext search scenarios. Unfortunately, data encryption, which restricts user’s ability to perform keyword search and further demands the protection of keyword privacy, makes the traditional plaintext search methods fail for encrypted cloud data.In this work, the multi keyword search over encrypted cloud data is be achieved efficiently. The contributions of this work can be summarized in three aspects as follows: first 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. Second the “AND”, “OR” and

"NO" operations in the multi-keyword search for searchable encryption. Compared with schemes in [7], [5] and [6], the proposed scheme can achieve more comprehensive functionality and lower query complexity. Third 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.

II. RELATED WORK

The main issue to go the cloud by the owner is the storage and the sharing of the file. Storing and sharing of data is not an easy thing in the cloud since cloud is a third party and owner keeping his/her information on cloud . So it is important to take accurate solution for the sharing and storage of data. Many research are done on this, all start from the storing of the data as an encrypted data. Then next issue is about the searching on this encrypted data by the user. Many proposals are put forward for this,

1. Searchable Encryption

Song et al proposed a method for searchable encryption, in this paper all words in the document is individually encrypted using two- layered encryption construction [1].Then he construct the indexes for the data files using bloom filters. A bloom filter having unique words trapdoors is built for each file and kept on server. When the user need to search a word, he/she creates individual request for search by computing the trapdoor for the word and send to cloud server,. Then the cloud does process the request by checking the bloom filter containing the trapdoor of the query and corresponding file identifiers are returned. But it is not an at all for well.

2. Fuzzy keyword search

Jin Li et al proposed that , while preserving keyword privacy they find and resolves the problem of effective fuzzy keyword search on encrypted data stored over cloud [2]. This search method improves system usability by sending back the exact matched result, when user input query search matches absolutely with the previously defined keywords or else it will return the similar documents based on the semantic relevance of the keyword requested by end users. This paper used distance method to measure keyword similarity. The result of their work is mainly two things first one is the when the user has input search query exactly matching the pre-stored keywords, the server produces the search output in terms of the files containing those keywords. And the second one is , if there are any mismatches exists in types or format in the searching query words, the similar possible output based on pre-specified semantics will be presented by server to the clients.

3. Privacy-preserving multi-keyword text search

Wang et al propose a privacy-preserving multi-keyword text search (MTS)[3] method with similarity based ranking to address the problem of efficient search in encrypted cloud data. They created vector space model to ensure the grouping of similar documents, it reduces the complexity in accessing the relevant files in faster manner and use term frequency measure to develop the search keyword indexes. Cosine similarity measure is used to facilitate the search results ranking accuracy. And this is one of the better methods to check similarity among the files. The indexes of the keyword are formulated in tree based structure so it will improve the searching efficiency, that is that gives faster access to the documents. They adapted different multi-dimensional algorithm to increase the search efficiency than the linear search.

4. Multi-keyword Ranked Search method

Ren et al defines and solves the problem of multi-keyword ranked search over encrypted cloud data, and establish a variety of privacy requirements [4]. It also uses a binary tree for the structuring the index of the keyword. So it will help the easy retrieval of the data that the user is searching. But here also the searching efficiency is not reached the peak.In my paper it gave importance to both the searching efficiency and the easy retrieval of the data.

III. SYSTEM MODEL, THREAT MODEL AND SECURITY REQUIREMENTS

1. SYSTEM MODEL:

The systems have three components,

•*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 client:* An inquiry client inquiries the outsourced documents from the cloud server with taking after three stages. To start with, the search client gets both the secrete key and symmetric key from the information owner. Second, as indicated by the search keywords, the pursuit client utilizes the secrete key to produce trapdoor and sends it to the cloud server. Last, she gets the coordinating archive gathering from the cloud server and decrypt them with the symmetric key.

B) THREAT MODEL AND SECURITY REQUIREMENTS

In the threat model, the cloud server is assumed to be "honest-but-curious", which is the same as most related works on secure cloud data search [5], [6], [7]. Specifically, the cloud server honestly follows the designated protocol specification. However, the cloud server could be "curious" to infer and analyze data (including index) in its storage and message flows received during the protocol so as to learn additional information. I consider two threat models depending on the information available to the cloud server, which are also used in [5], [7].

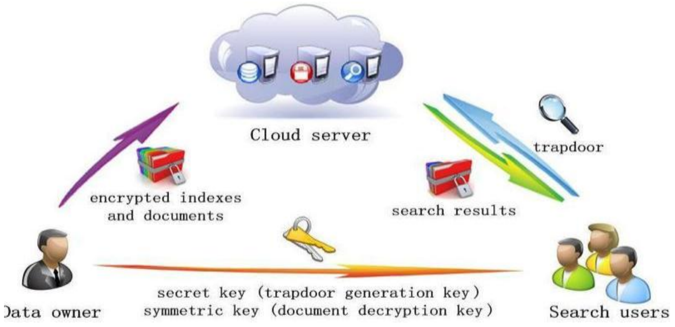

Fig1:system model

Known Cipher text Model*:* The cloud server can just know encrypted document collection C and file collection I, which are both outsourced from the data owner.

Known Background Model: The cloud server can have more knowledge than what can be accessed in the known cipher text model, for example, the connection relationship of trapdoors and the related factual of other data, i.e., the cloud server can have the measurable data from a known practically identical dataset which bears the comparable nature to the focusing on dataset.Similar to Refs. [7], [5], I assume search users are trusted entities, and they share the same symmetric key and secret key. Search users have pre-existing mutual trust with the data owner. For ease of illustration, I do not consider the secure distribution of the symmetric key and the secret key between the data owner and search users, it can be achieved through regular authentication and secure channel establishment protocols based on the prior security context shared between search users and the data owner [8]. In addition, to make our presentations more focused, I 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 new documents, updating existing documents, and deleting existing documents, are separated issues

C) SECURITY REQUIREMENTS

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 the authorized search users.

Privacy protection of index and trapdoor*:* The index and the trapdoor are created based on the documents' 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, it may learn the major subject of a document, even the content of a short document therefore, the content of index and trapdoor cannot be identified by the cloud server.

Unlinkability 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. 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.In cloud computing, secure analysis on outsourced encrypted data is a major topic. As a often used query for online applications, secure k-nearest neighbors (k-NN) computation on encrypted cloud data has inward much notice, and several solutions for it have been put forward. on the other hand, most existing schemes assume the query users are fully trusted and all query users share the total key which is used to encrypt and decrypt data holder's outsourced data. It is constitutionally not realistic in lots of real-world applications.Here propose a novel secure and efficient scheme for k-NN query on encrypted cloud data in which the key of data owner to encrypt and decrypt outsourced data will not be completely reveal to any query user. so, the scheme can efficiently support the secure k-NN query on encrypted cloud data even when query users are not reliable enough.

D) RELEVANCE SCORE

The relevance score between a keyword and a document represents the frequency that the keyword appears in the document. 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 the frequency of a given keyword in a document and IDF (inverse document frequency) represents the importance of keyword within the whole document collection. Without loss of generality, I select a widely used expression in Ref. [9] to evaluate the relevance score as

$$\sum_{w \in W} 1 \div (|F_j|).(1 + Inf_{j,w}).In(1 + N \div f_w)$$

Where $f_{j,w}$ denotes the TF of keyword w in document F_j, f_w denotes the number of documents contain keyword w,N denotes the number of documents in the collection , and $|f_j|$ denotes the length of F_j , obtained by counting the number of indexed keywords.

IV. PROPOSED SCHEMES

In this section, first propose a variant of the secure kNN computation scheme, which serves as the basic framework of our schemes. Furthermore, I 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, present a variant of the secure kNN computation scheme to achieve the searchable encryption property.

Initialization

The data owner randomly generates the secret key K =(S, M_1, M_2), where S is a m+1) 1)-dimensional binary vector, M_1 and M_2 are two(m+1) χ (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.

Index Building

The data owner first utilizes symmetric encryption algorithm (e.g., AES) to encrypt the document collection (F_1, F_2, . . . , F_N) with the symmetric key sk [24], the encrypted document collection are denoted as C_j(j = 1, 2, . . . , N). Then the data owner generates an m-dimensional binary vector P according to C_j(j = 1, 2, . . . , N), where each bit P ½i& indicates whether the encrypted document contains the keyword w_i, i.e., P [i] = 1 indicates yes and P [i] = 0indicates no. Then she extends P to a (m+1) dimensional vector P ʻ, where P ʻ[m + 1] = 1. The data owner uses vector S to split P ʻ into two (m+1)-dimensional vectors (p_a, p_b), where the vector S functions as a splitting indicator. Namely, if S[i] = ‘(i = 1, 2, . . . , m +1), p_a[i] and p_b [i] are both set as P ‘[i], if S[i] = 1(i = 1, 2, . . . , m+ 1), the value of P ʻ[i] will be randomly split into p_a[i] and p_b[i] (P ʻ[i] = p_a[i] + p_b[i]). Then, the index of encrypted document C_j can be calculated as I_j = (p_aM_1, p_bM_2). Finally, the data owner sends C_j||FID_j|| I_j (j = 1, 2, . . . , N) to the cloud server.

Trapdoor Generating

The search user firstly generates the keyword set W for searching. Then, she creates a m-dimensional binary vector Q according to W. value of Q[i] is the the weight of search keywords,. With the weight of keywords, it can also implement some operations like “OR”, “AND” and “NO” in the Google Search to the searchable encryption. Assume that the keyword sets corresponding to the “OR”, “AND” and “NO” operations are (w'_1, w'_2, . . . , w'_{l1}), (w''_1, w''_2, . . . , w''_{l2}) and (w'''_1, w'''_2, . . . , w'''_{l3}), respectively. Denote “OR”, “AND” and “NO” operations by ∧,∨, and -, respectively. Thus the matching rule can be represented as (w'_1 ∧ w'_2 ∧….w'_{l1})^ (w''_1 ∨ w''_2 ∨,….w''_{l2}) ^ (-w'''^1 - w'''_2 … - :w'''_{l1}). For “OR” operation, the search user chooses a super increasing sequence (a_1 > , a_2, . . . ,> a_{l1})($\sum_{i=1}^{j-1} ak$<a_j(j=2,…l_1)) to achieve searching with keyword weight.To enable searchable encryption with “AND” and “NO” operations, the search user chooses a sequence (b1, b2, . . . ,bl2 , c1, c2, . . . , cl3), where similar to above $\sum_{i=1}^{j-1} ak$<b_h(h=1,2,…l_2) and $\sum_{k=1}^{l1} ak$+$\sum_{h=1}^{l2} ah$ < c_i(i=1,2,….l_3).assume (w'_1, w'_2, . . . , w'_{l1}) are ordered by ascending importance , then accorting to the search keyword set (w'_1 ∧ w'_2 ∧….w'_{l1})^ (w''_1 ∨ w''_2 ∨,….w''_{l2}) ^ (-w'''^1 - w'''_2 … - :w'''_{l1}), the corresponding value in Q are set as ((a_1 , a_2 ,….a_{l1} ,b_1 , b_2,….b_{l2} ,-c_1 ,- c_2 … - :c_{l1}).Other values in Q are set as 0.In the Query phase , For a document F_j,if the corresponding R_j > 0, claims that F_j can satisfy the above matching rule.Then extend Q to a (m+1) dimensional binary vector Q’.where Q”=Q’.r and then split Q” into two (m+1)-dimensional vectors (q_a, q_b). q_a[i] and q_b [i] are both set as Q ‘[i], if S[i] = 1(i = 1, 2, . . . , m+ 1), the value of Q ʻ[i] will be randomly split into q_a[i] and q_b[i] (Q ʻ[i] = q_a[i] + q_b[i]).thus the trapdoor T_w can be generated as T_w=(M_1^{-1}.q_a,$M_2^{-1}q_b$).Then send T_w to the cloud.

Query

With the index I_j(j=1,2,..N) and trapdoor T_W , the cloud server calculates the query results as
R_j=I_j.T_W=r.(P.Q)
If R_j>0, the corresponding document identity FID_j will be returned.

Sub-Dictionary

Classify the total dictionary to many sub-dictionaries such as common sub-dictionary, computer science sub-dictionary, mathematics sub-dictionary and physics sub-dictionary, etc. Here the data owner should first choose corresponding sub-dictionaries. Then her own dictionary can be combined as {f_1||$Subdic_1$||f_2|| $Subdic_2$|| }, where $Subdic_i$ represents all keywords contained in corresponding sub-dictionary and f_i is filling factor with random length which will be 0 string in the index, the filling factor is used to confuse length of the data owner’s own dictionary and relative positions of sub-dictionaries. Then, the data owner and search user will use this dictionary to generate the index and trapdoor, respectively. Note that in an dictionary, two professional sub-dictionaries can even contain a same keyword, but only the first appeared key-word will be used to generate index and trapdoor, another will be set to 0 in the vector. And the secret key K will be formed as (S,M_1, M_2, $|f_1|$, D_{ID1} , $|f_2|$, D_{ID2} , . . .), where D_{IDi} represents the identity of sub-dictionary and $|f_i|$ is the length of f_i.

Dictionary updating:In the searchable encryption schemes with dictionary, dictionary update is a challenge problem because it may cause to update massive indexes outsourced to the cloud server. Here when it needs to change the sub-dictionaries or add new sub-dictionaries, only the data owners who use the corresponding sub-dictionaries need to update their indexes, most other data owners do not need to do any update operations. Such dictionary update operations are particularly lightweight.

V. CONCLUSION

In this propose schemes which not only support multi-keyword search over encrypted data, but also 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, And my proposal is efficient in terms of index building, trapdoor generating and query.

REFERENCE

[1]. 1 Jarecki, S., Jutla, C., Krawczyk, H., Rosu, M., and Steiner, M.(2013, November). Outsourced symmetric private information retrieval. In Proceedings of the 2013 ACM SIGSAC conference on Computer and communications security (pp. 875-888). ACM.

[2]. 2 Jin Li, Qian Wang, Cong Wang†, Ning Cao, Kui Ren and Wenjing

[3]. Lou, " Fuzzy keyword search over encrypted data in cloud computing", Proceedings IEEE, 2010 - ieeexplore.ieee.org.

[4]. 3 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," Hangzhou, China, 2013, pp. 71-82.

[5]. 4 N. Cao, C. Wang, M. Li,K. Ren, andW. J. Lou, "Privacy- Preserving Multi-keyword Ranked Search over Encrypted Cloud Data," in Proc. IEEE INFOCOM, Shanghai, China, 2011,pp. 829-837.

[6]. 5 W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li, "Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 11, pp. 3025–3035, Nov. 2014.

[7]. 6 J. Yu, P. Lu, Y. Zhu, G. Xue, and M. Li, "Towards secure multi-keyword top-k retrieval over encrypted cloud data," IEEE Trans. Dependable Secure Comput., vol. 10, no. 4, pp. 239–250, Jun. 2013.

[8]. 7 N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 1, pp. 222–233, Jan. 2014.

[9]. 8 D. Stinson, Cryptography: theory and practice. CRC press, 2006.

[10]. 9 J. Zobel and A. Moffat, "Exploring the similarity space," in Proc. ACM SIGIR Forum, vol. 32, no. 1, 1998, pp. 18–34.