A Searchable Ciphertext Retrieval Method Based on Counting Bloom Filter over Cloud Encrypted Data

Kuang YueJuan, Li Yong, Li Ping

***Abstract*—As we all known that it is difficult to retrieve the ciphertext on cloud environment, that the existing searchable encryption schemes have low efficiency in time, and that the file retrieval index does not support updating and the retrieval results cannot be sorted by accuracy. In view of these problems, in this paper, we firstly construct the index of document retrieval based on counting bloom filter, and map the keyword hash in the file set to the counting bloom filter index vector so that the retrieval of ciphertext by keyword is achieved. And, at the same time, it supports the dynamic update on retrieval index of the file. Secondly, the sorting of the retrieval results according to the relevancy degree cannot be achieved owing to the counting bloom filter itself does not have the semantic function. We introduce the key words frequency matrix and TF-IDF model to calculate the relevancy scores of key words in order to sort the search results according to the relevance scores. Finally, we prove the security, updatability, sorting capability and time efficiency of the method mentioned in this paper from the aspects of theoretical and experimental performance analysis.**

***Index Terms*—Searchable Ciphertext Retrieval, Counting Bloom Filter, Term Frequency-inverse Document Frequency Model, Relevance Score**

I. INTRODUCTION

WITH the increasing popularization of cloud computing technology, more and more users choose to outsource local data to cloud server storage to obtain high-quality and low-cost application services. However, in recent years, the continuous occur of information security events, especially the emergence of 2016 global "blackmail" virus event, has caused Internet users to worry about the security of data privacy in the cloud computing environment. Therefore, it is necessary to encrypt the data, store the data in the cloud in the form of ciphertext, and prevent the disclosure of privacy information [1]. Data is stored in the cloud in the form of ciphertext, which changes the original characteristics and association between data, making the traditional data retrieval method based on plaintext no longer effective. Searchable encryption technology is widely used in the cloud computing environment to protect the privacy of data and retrieve ciphertext [2]. However, there are some problems in the existing searchable encryption methods, such as inconvenient data query update and deletion operation, low time efficiency, and not support sorting the query results according to the accuracy. Therefore, it is of great theoretical significance and practical value to design a secure, efficient, and easy to operate and support sorting ciphertext retrieval method.

Manuscript received May 06, 2019; revised May 28, 2020. This work was supported in part by the National Natural Science Foundation of China (61841205, 11761057), the Yunnan Provincial Local Universities Joint Project (2017FH001-056), the Yunnan Provincial Department of Education Scientific Research Fund Project (2019J0602), and the Qujing Normal University Excellent Course Project (JPKC2016005).

K. Y. Juan was with the Department of Computer, Hunan Mass Media College, Changsha 410100, China (e-mail: 154334326@qq.com)

L. Y. was with the Department of Information Engineering, Qujing Normal University, Qujing Yunnan 655011, China (Correspondence e-mail: qjsfxyly@163.com).

L. P. was with the Department of Information Engineering, Qujing Normal University, Qujing Yunnan 655011, China (e-mail: qjncliping@163.com)

II. RELATED WORK

Searchable encryption technology can be tracked back to the literature [3] Song et al in 2000, when they proposed the first symmetric searchable encryption scheme (SWP), however, this scheme does not support the file retrieval index, it based on symmetric encryption technology, divides the file into “words” and then encrypts them, scans the ciphertext file and matches the ciphertext words to judge whether a retrieval keyword exists in the ciphertext file, the algorithm has large storage overhead and low time efficiency. Later, many scholars paid attention to improving the security, time overhead, accuracy and operability of the searchable encryption algorithm. In 2003, Goh [4] proposed the index-based searchable encryption scheme (Z-DIX), which is much better than the scheme of SWP. However, when the number of files in the retrieval system is large, it still takes a long time, the retrieval time is less efficient. In 2006, Curtmola et al. [5] proposed the SSE-1 and SSE-2 schemes on the basis of SWP. When performing file retrieval, the server only needs O (1) time complexity to complete the retrieval operation, but when a file needs to be added or deleted, the server needs to rebuild the index to adapt to the update state, which leads to more cost of maintenance. In terms of time overhead and operability research, they paid great attention to the study of dynamic increase, update and delete operation, and put forward a algorithm which is suitable dynamic updating under ciphertext state [6],[7],[8]. Researching on how to improve the accuracy and sequencing of ciphertext search results, they proposed an improved algorithm based on the tree structure retrieval [9], but the large storage overhead is a disadvantage of this algorithm. They constructed a sorted-file retrieval structure based on binary trees [10], which reduces the storage space overhead, meanwhile, its retrieval time performance is also reduced. To solve the problems of large maintenance and low time performance existed [11],[12], they proposed a distributed-text-retrieval model CBFTRM based on a

counting bloom filter [11], which achieves higher precision for searching documents, efficient file retrieval and low cost of maintenance and communication. However, CBFTRM model cannot sort the query results because of the absence of semantic function for counting bloom filter. They constructed retrieval index on basis of binary balance tree and proposed a multi- keyword retrieval scheme to rank the cipher text [12], which uses vector space model and the TF-IDF model to generate the index vector and the query vector, and the vector inner product as ranking of search results. However, it should be remarked that relevance scores of different keywords in the scheme cannot effectively explain the correlation between files and keywords, so the ranking results are not accurate enough. They firstly proposed the fuzzy multi-keyword retrieval scheme based on privacy protection [13], which used locality sensitive hashing (LSH) and bloom filter to build an index so that it greatly improves the efficiency and accuracy of file retrieval. However, this scheme does not have the sorting function for retrieval results. On basis of the work [14], they built vector index based on counting bloom filter [15], which used Min Hash algorithm to reduce the dimension of the keywords, and the time of index construction and query trapdoor generation was further reduced comparing to the work shown in [13]. At the same time, it used the number of vector index of counting bloom filter to denote the weight of keywords and sorted query results. However, the number of vector index of counting bloom filter cannot accurately represent the relationship between keywords and files because of the false positive probability event existed in counting bloom filter, i.e., different keywords in different files may appeared in the same vector position after hash mapping, which thus leads to the inaccuracy of query results. In terms of safety research, they proposed two unconditionally secure searchable symmetric encryption (SSE) new structural schemes [15], but one of the schemes has some restrictions on the security definition, while the other scheme achieves complete security, but it need to increase the size of key. Besides, they proposed an effective multi-user searchable encryption scheme based on password policy attributes (CP-ABE) [16], which proved to be semantic security under indistinguishability against chosen keyword attack (IND-CKA) in random oracle model.

To solve the above-mentioned problems, in this paper we proposed, based on bloom filter, a searchable ciphertext retrieval method (SCRM) which is suitable for cloud counting environment. First of all, we build a secure index based on counting bloom filter to achieve highly effective retrieval on ciphertext data and dynamic updating function of index. Then we construct frequency matrix of keywords by using the frequency of central keywords appeared in files. Finally, we use TF-IDF model to compute relevance score between retrieval keywords and target files so that we can achieve accurate rank on query results.

III. PRELIMINARIES

A. Counting Bloom Filter

The bloom filter (BF) originated in 1970. It was the first proposed by Burton bloom in the literature [17]. It is a kind of data structure with high spatiotemporal efficiency. It consists of a long binary vector and a set of independent hash functions. It is mainly used to determine whether an element is in a set. If an element is mapped by a set of independent hash functions, the output is Results if the values of the vector position sequence in the bloom filter are all 1, the element can be judged to belong to the set; otherwise, if the values of any corresponding positions are not 1, the element can be judged not to belong to the set. The basic principle of BF is that we firstly define a k bit vector M and initialize each position to 0. Then it maps n elements in the set W in turn to an address sequence $(h_1, h_2, ..., h_r)$ by using r mutually independent hash functions whose ranges are $[0,\ k-1)$ integers. Finally, the value that corresponds to address sequence in vector M is set to 1. If one position is mapped many times by hash functions, then taking 1 at the position only for the first hash mapping, and the subsequent does not change its value. BF only supports insert and search operations rather than delete operations. It is a simple data structure which is only suitable for the judgment of elements in static set, and is powerless for dynamic set.

Counting Boom Filter(CBF) [18] is an improved application data structure based on Boom Filter, the basic principle of which is to expand each bit in the BF vector to be a counter. When an element is inserted, if different elements are mapped to the same position multiple times by r hash functions, the value of the corresponding position counter is incremented by 1, and when element is deleted, the value of the corresponding position counter is decreased by 1. CBF adds delete operation on the base of BF, so it can support dynamic update of the elements in the set, which is suitable for element judgment in dynamic collection, such as file retrieval, web page de-duplication and so on. The mapping structure of BF and CBF is as follows, shown in Fig.1.

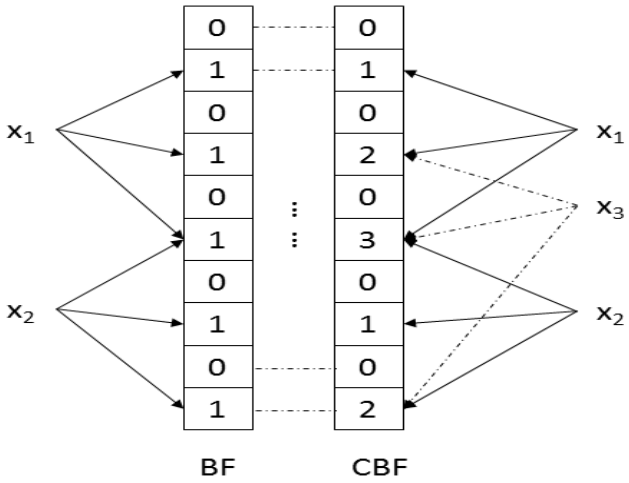

Fig.1. The mapping structure of BF and CBF.

B. TF-IDF model

Term Frequency-Inverse Document Frequency (TF-IDF) model [19] is a weighted statistical method that commonly used for information retrieval, which is consisted of term frequency (TF) and inverse document frequency (IDF), and is used to evaluate the importance of a certain keyword to a certain file in document collection. Wherein, TF indicates the number of a keyword occurred in document. If a keyword appears frequently in a document but appears rarely in other documents, it shows that the keyword has a good ability to distinguish documents and is defined as in formula (1).

$$TF_{ij} = n_{ij} \Big/ \sum_j n_{ij} \quad (1)$$

Wherein, n_{ij} and $\sum_j n_{ij}$ denote respectively the number of the keyword w_j occurred in the $i\text{-}th$ file, and the total number of the keywords that contained in the $i\text{-}th$ file.

IDF indicates the general importance of a keyword in a collection of documents and is usually used as the weight factor of a keyword, which is defined as in formula (2).

$$IDF_{ij} = log\left[\sum_i F_i / (\sum_{w_j \in F_i} F) + 1\right] \quad (2)$$

Wherein, $\sum_i F_i$ represents the total number of files contained in the files set and $\sum_{w_j \in F_i} F_i$ represents the number of files containing a keyword in the files set. Here, the denominator plus 1 is to prevent special cases where the denominator is zero.

Finally, formula (3) is used as TF-IDF model, and the value of TF-IDF is used as the relevance score of query keyword and target file, ranking according to the value from top to bottom and returned to the top-p targets that users are most interested in.

$$TF\text{-}IDF = TF_{ij} * IDF_{ij} \quad (3)$$

IV. System Security Model

The security model of searchable encryption system over cloud environment is shown in Fig.2, which consists of data owner, authorized user, and untrusted cloud server.

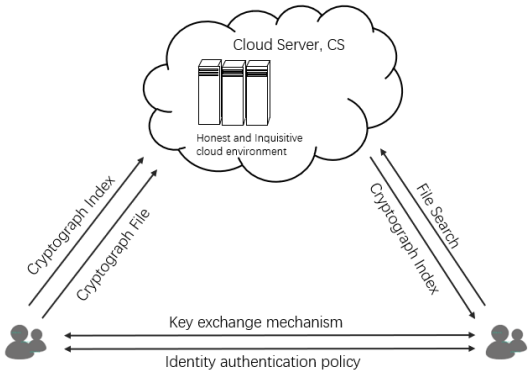

Fig.2. System security model of searchable encryption over cloud

(1) Data owner (abbreviated as OU), the publishers and sharers of files over cloud system, who mainly complete the operation of encrypting and updating on file set, building secure search indexes, authorizing users' authentication, key distribution, and sending ciphertext file, security index, file update information to cloud server.

(2) Authentication user (abbreviated as AU), requesters of files over cloud system, who possesses authority to access the shared files that granted by OU, sends generated search trap to server by inputting file retrieval keywords to cloud server. Finally, the communication key distributed by OU is used to decrypt the ciphertext search results to obtain the interested top-p plaintext files.

(3) Cloud server (abbreviated as CS), the storage center of ciphertext file set and security index, which is also responsible for receiving file update information that sent by OU and file retrieval request of AU, completing file retrieval index update and feeding back the retrieval results.

Because cloud computing systems work on the basis of complying with user data hosting and communication protocols, this paper believes that the server in cloud computing system is a "honest and curious" semi trusted entity, that is, the cloud server will not deliberately attack the system maliciously, disclose user privacy, or conspire with other unauthorized users to attack the system, but it will be out of "curiosity" Mining user data and pushing application services, disclosing user privacy. Therefore, this paper considers the known ciphertext attack model as the security target. In the communication environment of the cloud computing system, the cloud server or other unauthorized users cannot obtain any plaintext information. The file set uploaded by the data owner, the security retrieval index, the query request and query results of the authorized users all exist in the form of ciphertext. The attacker can only choose ciphertext only attack.

V. SCRM Method

In the cloud computing environment, data retrieval has the characteristics of large amount of data, large number of users, frequent retrieval and update operations. The existing searchable public key encryption methods that need linear search time to match ciphertext information one by one and need a lot of logarithm operations are obviously not suitable. Although the existing sorted ciphertext search algorithms have small amount of calculation and fast speed, they also have low query accuracy. Therefore, based on CBF, this paper proposes a sort ciphertext retrieval method SCRM, which can achieve precise ciphertext sorting [20]. The specific implementation process of SCRM is described in detail below.

A. Notations definition and description

Some of main notations involved in SCRM method are defined and shown in TABLE I.

TABLE I
MAIN SYMBOL DEFINITION AND DESCRIPTION

Symbol	Definition and Description
F	Represent a set of plaintext files, $F = (F_1, F_2, ...F_m)$.
C	Represent a set of ciphertext files, $C = (C_1, C_2, ...C_m)$.
FID	Represent a set of identifiers for corresponding files, $FID = (FID_1, FID_2, ..., FID_m)$, it's just an identifier for files distinguishing without any semantics.
MID	Represent a set of storage identifier for corresponding files in cloud server, $MID = (MID_1, MID_2, ..., MID_m)$.
QID	Represent a set of files identifiers associated with retrieval keywords. $QID = (QID_1, QID_2, ..., QID_{ml})$, and $QID \subseteq FID$.
W	Represent a set of keywords in all files,

	$W=(w_1,w_2,...,w_n)$, the same of keyword is allowed to appear repeatedly in files.
ψ	It is a matrix of keywords frequency in files.
I	It is a file retrieval index by hashing keyword into CBF.
M	It is a vector with k bits storage space.
C_ψ	It is the ciphertext form of ψ.
C_I	It is a security index, ie the ciphertext form of I.
Q_w	Retrieval keyword.
Q	Retrieval keyword vector.
T_Q	Retrieval trapdoor.

B. Detailed implementation process

The detailed implementation process of SCRM method includes following steps.

Step 1 Initial().

Initialization operation, OU extracts keywords that can correctly reflect the semantics of each file from F, obtains keywords set W, and carries out vectorization processing on keywords.

Step 2 Preproc().

Preprocessing operation, OU counts the frequency of keyword in each file and names as ψ_{ij}, and constructs the matrix of ψ by the value of ψ_{ij}, the structure of which is shown in formula (4).

$$\psi = \begin{array}{c} \\ FID_1 \\ FID_2 \\ \cdot \\ \cdot \\ \\ FID_m \end{array} \begin{array}{c} \begin{array}{cccc} w_1 & w_2 & \cdots & w_n \end{array} \\ \begin{bmatrix} \psi_{11} & \psi_{12} & \cdots & \psi_{12} \\ \psi_{21} & \psi_{22} & \cdots & \psi_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \psi_{m1} & \psi_{m2} & \cdots & \psi_{mn} \end{bmatrix} \end{array} \tag{4}$$

On the basis of ψ, calculating the number of keywords in file F_i, and the number of files that contain keyword w_j, respectively name it as $N_{w_j,F}$, $N_{w_j,F}$. Among of them, $N_{w_j,F}$ is shown in formula (5), $N_{w_j,F}$ is shown in formula (6).

$$N_{F_i,w_j} = \sum_{j=1}^{j=n} \psi_{ij} \tag{5}$$

$$N_{w_j,F} = \left\{ N_{w_j,F_{i-1}} + 1, i = i-1 \mid \psi_{ij} \neq 0, i = m,\ j = 1,2,...,n \right\} \tag{6}$$

Step 3 BuildIndex_I(), CreatLink().

Construction of the file retrieval index and the keyword file linklist. Firstly, OU maps each keyword in W into vectors M in CBF by using r mutually independent hash functions with ranges of integer $[0,\ k-1)$ and obtains the file retrieval index. Then, extends the Counter of the position $M[k]$ in index I to linklist header shown in Figure 3, and insert the element behind the header. Linklist header and linklist elements are respectively expressed into Link.head={ $M[k]$, Next}, Link.element={ FID, Next}. Among of them, FID is file identifier that associated with keywords of the position $M[k]$, and Next is a pointer to the next element, the Next pointer of the last element points to Null. Lastly, obtained the file retrieval index and keyword file correlated linklist, and its structure is shown in Fig.3.

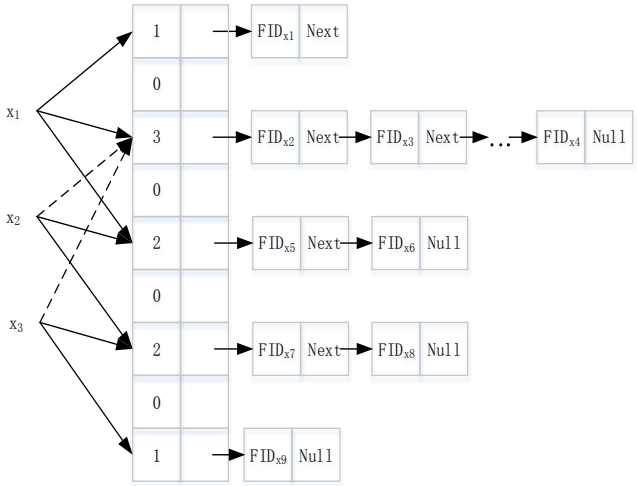

Fig.3. The file retrieval index and keyword file correlated linklist structure

Step 4 KenGen(k,n).

Generation security key, OU inputs security parameters k and n, uses probabilistic key generation algorithm to generate security key $sk=\left(M_1,M_2,M_3,S,P_{plu}\right)$, and sends sk to authorized users by secret channel, such as kerberos protocol. In security sk, M_1 and M_1 are k order random invertible matrix, M_3 is n order random invertible matrix, $S=\left(0,1\right)^k$ is k bits two elements vector, P_{plu} is a secret big prime number.

Step 5 Enc(F, ψ, I, sk).

Encrypting operation, firstly, OU encrypts F, ψ, I with sk, and also calculates the storage identifier $MID_i = MD5(FID_i)$ for each file in CS. secondly, uploads C_ψ to CS. Encrypting ψ and I are shown in formula (7) and (8).

$$C_\psi = \psi^T \cdot M_3^T \tag{7}$$

$$C_I = Enc(I,sk) = (M_1^T I', M_2^T I'') \tag{8}$$

I' and I'' are two sub vectors decomposed from I, the detailed decomposition process is completed in reference to the operation rules in [12]. Lastly, considering the fastly speed and highly security level of AES symmetric encryption, it is suitable for encrypting large scale files set, so, this paper uses AES encryption algorithm and key P_{plu} to encrypt F.

Step 6 Trap(Q, sk).

Generation retrieval trapdoor, AU inputs retrieval keywords $Q_w = (w_1, w_2, ..., w_j)$. Firstly, uses the same set of hash functions in Step3 to map Q_w into the vector M in

CBF and obtain Q. Secondly, adopts the similarity method to obtain T_Q in step 5, is shown in formula (9). Lastly, sends T_Q to CS.

$$T_Q = Enc_{sk}(Q) = (M_1^{-1}Q', M_2^{-1}Q'') \tag{9}$$

Step 7 Ranking_Search(C_I, C_ψ, T_Q).

Sortable ciphertext retrieval, the following 5 steps are carried out.

(1)When CS received T_Q, completes inner product operation between security C_I and T_Q, the detailed calculation process is as follows in the literature [11],[12], shown in the formula (10).

$$\begin{aligned} C_I \cdot T_Q &= (M_1^T I', M_2^T I'')^T \cdot (M_1^{-1}Q', M_2^{-1}Q'') \\ &= (M_1^T I')^T \cdot M_1^{-1}Q' + (M_2^T I'')^T \cdot M_2^{-1}Q'' \\ &= I'^T \cdot Q' + I''^T \cdot Q'' \\ &= I^T \cdot Q \end{aligned} \tag{10}$$

According to the calculation results of the above file retrieval, then finds the file identifier set QID that associated with retrieval keyword in keyword file correlated linklist, and sends to AU.

(2)AU downloads C_ψ from CS, and decrypts C_ψ to obtain ψ with M_3^{-1}, shown in formula (11).

$$C_\psi^T \cdot M_3^{-1} = \psi \tag{11}$$

(3)On the basis of ψ, AU calculates the TF-IDF values of retrieval keywords depend on formulas (1),(2),(3),(4),(5),(6), that is to obtain the relevance scores of retrieval keywords for distinguishing target files with QID, and ranking target files and returning identifier of the top-p interested target files.

(4)AU uses the same $MD5$ function to calculate the value $MD5[top\text{-}p(QID)]$ of the top-p target files, and sends it to CS in turn.

(5)When CS received the value of $MD5[top\text{-}p(QID)]$, according to the value *MID* of the stored files in CS, the *top-p* ciphertext target files in encrypted file set C are sequentially responded to AU. Then, AU uses the same decryption algorithm AES and key P_{plu} to decrypt ciphertext target files, restores the result of ranking plaintext target files, and Completes the whole communication process.

VI. Performance Analysis

A. Security analysis

On the security protection of files retrieval index, retrieval trapdoor and keywords frequency matrix, this paper draws on the practices in [13] and [14], randomly generates invertible matrices M_1, M_2,and M_3, and binary vector S to encrypt file index vector, keyword retrieval vector and keyword frequency matrix. Since the space of key matrix is infinite, each time a randomly generated key matrix has only one invertible matrix, implying that the correctness probability that unauthorized users want to forge key matrix to generate retrieval trapdoor and decrypt keywords frequency matrix is 0, which effectively ensure the security of file retrieval index, trapdoor and keywords frequency matrix. At the same time, in order to ensure that privacy information will not be leaked due to the leakage of related retrieval trapdoor, this paper also uses binary vector to split index vector so that two completely different traps are generated for the same retrieval vector, which ensures the irrelevance of trapdoor. Finally, on the security protection of file privacy and key management, this paper uses AES symmetric encryption algorithm and Kerberos protocol to complete files set encryption and key management, which effectively guarantees the security of files set and key. Since the security of AES algorithm and Kerberos protocol has been proved in detail in many references, we will not mention them in this paper. To sum up, as long as the key *sk* is not artificially leaked, SCRM is security against known ciphertext attack model.

B. Renewable capability analysis

Assume that OU has a new file F_{m+1} to be released, extracting keywords $(w_{n+1}, w_{n+2}, w_{n+3})$ from F_{m+1} after Step 1, and then generating new index I_{m+1} after Step 3, finally, using security key generated of Step 4 and method of Step 5 to encrypt the index I_{m+1} to generate security ciphertext index $(M_1^T I'_{m+1}, M_2^T I''_{m+1})$, performing the same-dimensional matrix addition on $(M_1^T I', M_2^T I'')$ and the original security index matrix can obtain the updated security index $[M_1^T(I'_{m+1} + I'), M_2^T(I'' + I''_{m+1})]$. Similarly, if OU needs to delete a file, an updated security index can be obtained by performing same-dimensional matrix subtraction operation on the basis of original security ciphertext index. After updating the security index, we use method of Step 2 to build new keywords frequency matrix, and update the keyword file associated linklist of Step 4, which is actually the insertion and deletion operation of linear linked list and will not be repeated in this paper. From which we can see that SCRM method has the ability of updating security index.

C. Ranking capability analysis

Considering CBF index without semantic function, cannot realization ciphertext retrieval ranking, this paper introduces keyword file correlated list and keyword frequency matrix ψ on the basis of CBF index. Firstly, AU determines whether search keyword belongs to keyword set or not through the CBF index, if it is the case, then find the target file *FID* which is associated with retrieval keyword in keyword file correlated list, and then read the frequency of retrieval keywords through *FID* in keyword frequency matrix ψ, using TF-IDF model to statistically calculate the relevancy scores of retrieval keywords to discriminatory power of target file. Finally, the document retrieval results can be ranked according to relevance scores, and two alignments process of searching linklist and word frequency matrix can also eliminate the false positive probability of CBF index.

D. Efficiency analysis

In order to evaluate the performance of time cost and accuracy on SCRM in file retrieval, this paper uses Enron Email Dataset (EED) as experimental data set [21]. Select files randomly from the dataset, completes word segmentation using Java open source parser (MMSEG4J), build a keyword dictionary, run on an Intel ® Core ™ i5-3230 2.60GHz dual core CPU, 2.0GB RAM memory, Windows 7 64-bit operation system platform for experimental testing.

(1) Retrieval time overhead

File retrieval time overhead is mainly affected by the size of file set, the efficiency of file retrieval index, the generation trapdoor time and the ranking mechanism of retrieval results, which can comprehensively measure the overall performance of a searchable ciphertext retrieval scheme, and it is also an important performance index to measure the service quality of system. In order to objectively evaluate SCRM in this paper and those in [13], [14] on file retrieval time, in this paper, we use respectively SCRM and the methods presented in [13],[14] to build the file retrieval index and trapping trapdoor, and returns the top-10 results as target file. The testing is carried out in different file sets scale and different retrieval keywords, and the test results are shown in Fig.4(a) and Fig.4(b) below.

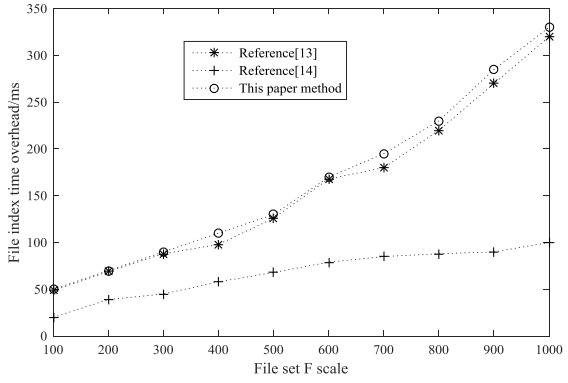

Fig.4(a). Retrieval keywords are equal to 10, the number of files are different.

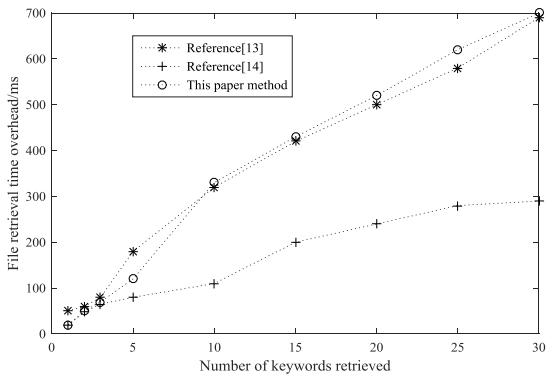

Fig.4(b). Files are equal to 1000, retrieval keywords are different.
Fig.4. File retrieval time overhead

Fig.4(a) shows that, input 10 keywords to carry out file retrieval in different files set scale, the time overhead of the method in [14] is lowest, SCRM is close to the method in [13]. Fig.4(b) shows that, input 1000 files to carry out file retrieval in different keywords, when retrieval keywords number less than 5, the time cost of SCRM is close to the method in [14] and is slightly better than the method in [13]. With the increase of the number of keywords, the method in this paper is close to the method in [13] and is slightly worse than [14]. The main reason is that the mechanisms of indexing and generating trapdoor are difference among on the three methods, literature [14] uses minhash algorithm to reduce the dimension of keywords and the number of hash functions needed to generate file retrieval index are less, which reduces the time overhead of constructing the index of the file retrieval. However, comparing the results of shown in Figure 4(a) and Figure 4(b), it is found that the effect of this dimensionality reduction is not obvious in the time overhead on a small amount of keyword retrieval. Compared with the method in this paper, the method in [13] has less time cost for sorting search results, but it takes more time to calculate the similarity when constructing the index. However, the ordering of the method in this paper only reads and computes the associated files and the exact word frequency of constant item. The time complexity of the algorithm is O(1) and there are not many associated file with keywords in retrieval results when the number of keywords is small. In other words, the time cost on word frequency reading and calculation is not large. So, on the whole, the retrieval time overhead of the method in this paper is feasible.

(2) Retrieval accuracy

The query accuracy is another important performance index to measure the service quality of ciphertext retrieval system. In order to objectively evaluate the performance of SCRM and the methods in [13],[14] on the accuracy of file retrieval, we randomly select 1000 files from the dataset to build security index and input 10 retrieval keywords, and test the relevance scores of keywords in the returned top-10 results to measure the accuracy of file retrieval results. The test results are shown in Fig.5, which indicates that the higher of relevance score, the more accurate of retrieval result. The results of shown in figure 5 show that, the retrieval accuracy of SCRM is much higher than [14], and the method in [13] is lowest. The main reason is that, index vector counting is only used as the keyword weight to establish the ranking mechanism [14], the sorting accuracy is affected by false positive probability, leading to a lower sorting accuracy. The method of this paper establishes a precise word frequency matrix, and calculates the relevance score of keywords according to frequency-frequency matrix, which is more accurate and reliable relative to [14]. Finally, the accuracy is lowest [13], where there is no effective ranking mechanism.

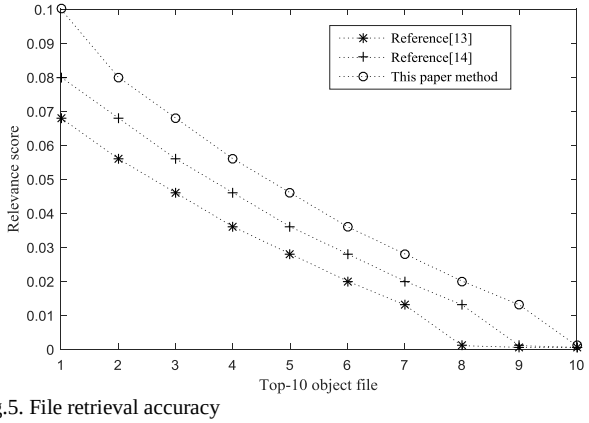

Fig.5. File retrieval accuracy

VII. Conclusions

This paper proposed a method SCRM which is suitable for ciphertext retrieval and sorting in cloud computing environment. By constructing CBF index vector, the ciphertext retrieval and index dynamic update are realized. At the same time, considering that the index vector constructed by CBF does not have semantic function and cannot achieve the precise sorting function of the target file, the paper introduces the word frequency matrix and TF-IDF model to calculate the correlation score of the ability to distinguish the key words from the file, which effectively realizes the sorting function of the target file. Finally, the paper theoretically analyzes the security, renewability and orderability of the method, and tests and analyzes the time cost and accuracy of the document retrieval, which proves the feasibility and effectiveness of the method. However, the disadvantage of this method is that in the process of sorting, because of the introduction of word frequency matrix and file association linked list, the storage space and index update cost are increased. Therefore, the next research direction is to further reduce the storage space and index update cost.

References

[1] Isaac Odun-Ayo, Olasupo Ajayi, and Sanjay Misra, "Cloud Computing Security: Issues and Developments," Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2018, 4-6 July, 2018, London, U.K., pp175-181

[2] S Pradeep, and Yogesh kumar Sharma, "Effectual Secured Approach for Internet of Things with Fog Computing and Mobile Cloud Architecture Using Ifogsim," Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2019, 3-5 July, 2019, London, U.K., pp101-104

[3] Song D X, Wagner D, Perring A. Practical Techniques for Searches on Encrypted Data. Proceedings of the IEEE Symposium on Security and Privacy. Washington: IEEE Computer Society Press, 2000:44-55.

[4] E.-J. Goh,"Secure Indexes," Cryptology ePrint Archive, 2003. http:// eprint. iacr. org/2003/216.

[5] Curtmola R, Garay J, Kamara S, et al. Searchable symmetric encryption: improved definitions and efficient constructions. ACM Conference on Computer and Communications Security. New York: ACM, 2006:79-88.

[6] Liesdonk P V, Sedghi S, Doumen J, et al. Computationally Efficient Searchable Symmetric Encryption. Secure Data Management,2010, 6358: 87−100.

[7] Kamara S, Papamanthou C, Roeder T. Dynamic searchable symmetric encryption. Proc. of the 19th ACM Conf. on Computer and Communications Security (CCS 2012). New York: ACM, 2012. 965−976.

[8] Kamara S, Papamanthou C. Parallel and Dynamic Searchable Symmetric Encryption. International Conference on Financial Cryptography and Data Security. Berlin: Springer, 2013:258-274.

[9] Wang C, Cao N, Li J, et al. Secure Ranked Keyword Search over Encrypted Cloud Data. IEEE International Conference on Distributed Computing Systems. IEEE, 2010:253-262.

[10] Li Q, Yue F S, Wang G J. Efficient multi-keyword search over secure cloud storage, Computer Science, 2012,39(12):158-161. (in Chinese)

[11] Feng J J, Wang X L, Tiang Q. Text retrieval model based on Counting Bloom Filter, Computer Engineering, 2014, 40(2):58-61. (in Chinese)

[12] Xia Z H, Wang X H, Sun X M, et al. A secure and Dynamic Multi-keyword Ranked Search Scheme over Encrypted Cloud Scheme over Encrypted Cloud Data. IEEE Transactions on Parallel and Distributed Systems, 2016, 7(2):340-352.

[13] Wang B, Yu S, Lou W, et al. Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud. Proceedings IEEE INFOCOM. Toronto: Institute of Electrical and Electronics Engineers, 2014:2112-2120.

[14] He H, Xia W, Zhang J. Fuzzy multi-keyword retrieval scheme over encrypted data in cloud computing, Computer Science, 2017, 44(5):146-152. (in Chinese)

[15] Yoshizawa T, Watanabe Y,Shikata J . "Unconditionally secure searchable encryption." Information Sciences & Systems. IEEE, 2017.

[16] Shangping W, Xiaoxue Z, Yaling Z , et al. "Efficiently Multi-User Searchable Encryption Scheme with Attribute Revocation and Grant for Cloud Storage." PLOS ONE 11.11(2016):157-167.

[17] Bloom B H. Space/time trade-offs in hash coding with allowable errors. Communications of the ACM, 1970, 13(7):422-426.

[18] Rottenstreich O, Kanizo Y, Keslassy I. The Variable-Increment Counting Bloom Filter. IEEE/ACM Transactions on Networking, 2012, 22(4):1092-1105.

[19] Zhang W, Xiao S, Lin Y, et al. Secure Ranked Multi-keyword Search for Multiple Data Owners in Cloud Computing. IEEE International Conference on Parallel and Distributed Systems. Atlanta: IEEE, 2017:276-286.

[20] Feng G L, Tang L. Multi-attribute ranked keyword search over encrypted cloud data, Computer Science ,2013, 40(11):131-136. (in Chinese)

[21] COHEN W W. Enron email dataset. 2015. http://www.cs. cmu.edu/～ enron/.

Kuang YueJuan, received the Master degree in Computer Science from Hunan University in 2010. His research interests include computer network, information security, and cloud computing. He is a Lecturer at the Department of Computer, Hunan Mass Media College.

Li Yong, received the Master degree in Computer Science from Wuyi University in 2010. His research interests include computer network, information security, and cloud computing. He is an Assistant Professor at the Department of Information Engineering, Qujing Normal University.

Li Ping, received the Master degree in Computer Science from Kunming University of Science and Technology in 2006. His research interests include computer network, information security, cloud computing, and database application technology. She is an Assistant Professor at the Department of Information Engineering, Qujing Normal University.