International Journal of Security and Its Applications
Vol. 10, No. 12 (2016), pp.335-346
http://dx.doi.org/10.14257/ijsia.2016.10.12.27

A Multi-attribute Keyword Retrieval Mechanism for Encrypted Cloud Data

Yunfa Li, Mingyi Li, Yangyang Shen

Key Laboratory of Complex Systems Modeling and Simulation, School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China
yunfali@hdu.edu.cn

Abstract

With the development of cloud computing technology, more and more users have outsourced their data to cloud platform. To protect the safety of these data, some encryption methods are usually used. But it is very difficult question that how to effective retrieval and to use the encrypted data, we propose a multi-attribute keyword retrieval mechanism for encrypted cloud data. In this mechanism, we first define the local feature attributes of the keywords and the global feature attributes of the document. Then, we construct the mapping relationship between keywords and document attributes according to the basic principle of inverted index algorithm and establish the security index. Based on the above steps, we improve the weight of the key words and calculate the similarity score of the document by combining the multi-attribute score function. Experiments show that this mechanism not only can effectively improve the accuracy of the data retrieval, but also can greatly reduce the bandwidth consumption of system.

Keywords*: cloud computing, data service, multi-attribute, retrieval*

1. Introduction

Now, more and more sensitive information are stored in the cloud. The sensitive information includes some personal medical records, financial information, trade secrets and so on. People will lose the direct control-right to data after they sent the data to the cloud server, which may lead to the disclosure of personal privacy data. Therefore, the security problem of cloud computing has restricted seriously its development. In 2010, Google fired two staffs who invaded Google Voice, Gtalk and other accounts in order to obtain private data [1]. It shows that the cloud computing server confronts the risk of the leakage of sensitive data when it provides services to user. In June 2010, Apple Inc also appears the information leakage event [2]. These series of security incidents leads to the result that people become more worried about the security of cloud computing.

The protection method of data safety in current cloud computing environment is mainly by using encryption method. In general, the data owner will encrypt the data before these data are uploaded to cloud storage platform. In this situation, the encrypted data will loss the characteristics of the original data. It is very difficult problem that how to effectively retrieval and use encrypted data. The traditional way is generally based on the weights of keywords to calculate the similarity score of documents and is through matching keyword to return the most relevant result to user. But, in fact, this method has many limitations. In order to resolve these problems, we propose a multi-attribute keyword retrieval mechanism for encrypted cloud data.

We first define the local feature attributes of the keywords and the global feature attributes of the document. Then, we construct the mapping relationship between keywords and document attributes according to the basic principle of inverted index algorithm and establish the security index. Based on the above steps, we improve the

ISSN: 1738-9976 IJSIA

weight of the key words and calculate the similarity score of the document by combining the multi-attribute score function.

The mechanism introduced the keywords fuzzy set method is based on wildcard to enhance the experience. Its purpose is to return the *Top-k* [3] of the most relevant results users in the event of a spelling error. Compared with the traditional methods, the method considers not only the weight of the keyword, but also the global attributes of the document, which makes it more reasonable. So, it can improve the efficiency of data retrieval based on the security services.

This paper is organized as follows: we discuss the related works in section 2. In section 3, the system model of retrieval method in cloud computing is described. In section 4, a retrieval method about multi-attribute fuzzy keyword is proposed. A series of experiments are done and the results are analyzed in section 5. Finally, the conclusions are drawn in section 6.

2. Related Works

In order to solve the problem of data security in cloud environment, many researchers have made research contributions and obtained some achievements. The contributions and the achievements are described as follows.

In 2000, Song *et al*. [4] put forward searchable symmetric encryption (SSE). First, the plaintext file is divided into "word" and then encrypts respectively. Based on these steps, it scans the entire ciphertext document and makes comparison with the ciphertext word. Thus, it can confirm the existence of keywords and the number of times that it appears in the file. The advantage of the method is fast, but its disadvantage is easy to suffer from statistical attack because the tolerance mechanisms for input format errors are lack. So, it is not suitable for large scale data in cloud computing environment. Afterwards, many people proposed a traditional single keyword search scheme [5-8] based on the above theory. In these schemes, the file is first encrypted. Then, the encrypted file is indexed by using a single keyword. Thus, the privacy information of the file can't be obtained by the cloud server. The search results will be decrypted through the trap door function and the corresponding key. Subsequently, some improved connection search schemes are built and most of them adopt the bilinear mapping, which will leads to high computational complexity and large amount of communication consumption [9][10]. Moreover, these schemes only support precise query results. Li *et al*. [11] proposed a fuzzy keyword search scheme, which defines and measures the similar degree of the keywords by editing distance. In the search scheme, the two construction methods are used in order to construct two kinds of fuzzy keyword set which are based on wildcard and G (gram). In 2007, Swaminathan *et al*. [12] proposed a privacy ranking retrieval algorithm. In the algorithm, the ordering and encryption algorithm is used to encrypt the frequency of keywords in the file. In 2011, Bosch *et al*. [13] also put forward a keyword retrieval method based on wildcard. In the method, the keywords contained in the file are inserted into the bloom filter in order to avoid attack correlation. Moreover, the pseudo random number can be produced in the method based on document identifiers which is used to hide the binary vectors in the bloom filter. To achieve the retrieval, all pre-generated keywords as wildcard can convert into keywords and can convert these keywords retrieval containing wildcard into exact match retrieval. As for the optimization of results, it makes a sort mainly according to the relevant degree of the document and returns the most relevant results.

In 2012, Wang *et al*. [14] further research fuzzy word retrieval scheme and give the formal security proof. But, the above algorithms still have two deficiencies. First, these algorithms require each keyword as the leaf nodes of the index tree which generate very large space overhead in storage. Second, these algorithms don't support to index dynamic updating. After that, Wang *et al*. [15] first proposed a sort of encryption search scheme in

cloud computing. In the scheme, the data owner calculates the related degree for each keyword and establishes an inverted index structure after using one-to-many security sequence mapping encryption. In the keyword search phase, the server matches the encrypted file by receiving the keyword trap door function and returns the most relevant documents according to the relevant score sort. In 2012, this algorithm (OPSE) was further improved the literature [16]. Thus, the security and efficiency of the algorithm was improved.

Subsequently, Cao *et al*. [17] proposed a sort retrieval algorithm based on the encryption, which can support multiple keyword search (MRSE). In the algorithm, the safety ranking of keyword is extended from one-dimensional to multidimensional keywords. Its basic method is based on "the coordinate matching [18]" principle and the degree of similarity between the keywords is measured by quoting "the inner product similarity". Thus, the security ranking of encrypted data can realized under the environment of cloud computing. In 2014, the authors further expanded the MRSE algorithm to support more search semantics in the literature [19]. Although they have made some achievements, there are still some problems. These problems can be described as follows: (1) the algorithm doesn't support dynamic index updating and the new keywords need to be reconstructed when there is a new keyword needed to be added because the algorithm used the static keyword dictionary. (2) The frequency of MRSE is not considered, and the practicability of the system is reduced. Xu *et al*. [20] constructed a keyword set by introducing partition matrix and achieved the dynamic update. Meanwhile, they pull in an random factor in the set of keyword in order to reduce the effect of the redundant keyword in normal distribution between the correctness of the query results. Since the multiplication of matrix will take up too much space, it will have great effect on the query efficiency of high dimensional data.

Anuradha *et al*. [21] proposed a system that supports multi-owner keyword ranked search over the encrypted cloud data by using the concept of ranked searchable symmetric encryption model. Amol *et al*. [22] explored the problem of multi-keyword ranked search over encrypted cloud data, and established a set of strict privacy requirements for such a secure cloud data utilization system. Then, they proposed two MRSE schemes [23] based on the similarity measure of "coordinate matching" while meeting different privacy requirements.

In order to resolve the above problems, we will build a multi-attribute keyword retrieval mechanism for encrypted cloud data. The main idea of the mechanism is: the data owner first builds a multi-attribute eigenvectors security index through the global feature of document and the local characteristic attributes of keyword. Then, the cloud server decides their weight through the sorting way selected by user. On this basis, the cloud server improves the weight calculation algorithm. After that, the similarity score of the document is computed through combining multi-attribute scoring function in the cloud server.

3. System Model of Retrieval Method

The architecture of cloud service generally consists of three main entities, namely the data owner, the cloud server and the user. The data owner uploads the data, the cloud server stores the data, and the user retrieves the data. Meanwhile, the whole service process is faced with a variety of threats. For example, the data is stolen by the fake owner, the data is used by the fake user, and the data is stored by the fake cloud server, and so on. In view of the various security threats that the data resource faced, people usually encrypt the data by using a certain encryption method to ensure the security of data resource service. However, with the increasing demand of cloud services, it is a very difficult problem to retrieve the encrypted data and provide effective data services to users.

Based on the above analyzing about cloud service, we construct a system model of fuzzy keyword retrieval method in cloud computing. The model is shown as Figure 1.

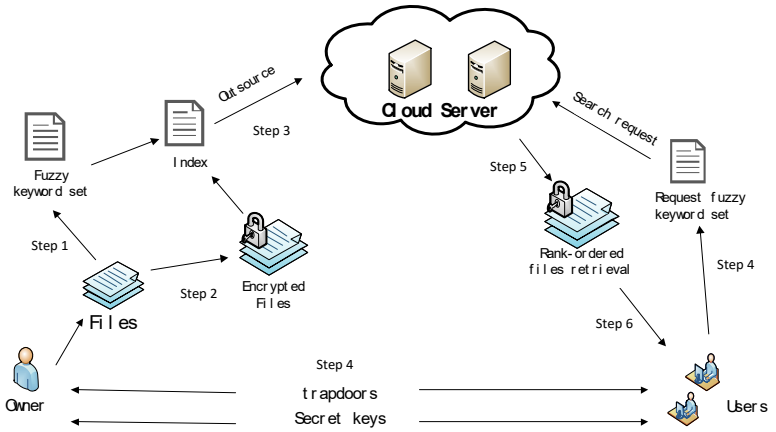

Figure 1. The System Model of Fuzzy Keyword Retrieval Method in Cloud Computing

In this model, the data owner needs to encrypt the file set $F = (f_1, f_2, ...f_m)$ before the file set is uploaded to the cloud server (the data owner can choose an encryption mode in term of the security of requirement). Its purpose is to protect the security of data transmission. The process can be described as follows:

Step 1: The data owner extracts the collection of key words from the $F = (f_1, f_2, ...f_m)$ document and the local features of the key words. And then it establishes the fuzzy keyword set.

Step 2: The data owner extracts the global features of each document from the F document, and then constructs the multi-attribute feature vector of the document.

Step 3: According to the feature vector of document, the similarity score is computed and the inverted index of the keyword-document structure is built. In the following step, the fuzzy key word set and the document set is encrypted by the data owner. Then, the information is uploaded to the cloud server with the index construction.

Step 4: The user requests to obtain the authorization of the data owner in order to get the trapdoor function and the key. Then, the user sends the keyword search request to the cloud server. Thus, the corresponding fuzzy keyword set is generated and encrypted. At last, the encrypted fuzzy keyword is sent to the cloud service for retrieval.

Step 5: Receiving a request for the retrieval, the cloud server begins to execute the request and returns to the Top-K encryption document, which is asked by the user.

Step 6: The user receives the retrieval results from the cloud server and decrypts the encrypted document by using the trapdoor function and the key.

4 Retrieval about Multi-attribute Keyword

4.1 The Extraction of Keyword Multi-attribute Feature Vector

The extraction of keyword is the preprocessing stage of the document set. The key words set will be extracted from the documents through algorithm generally to produce the segmentation result set. We filter the segmentation result set and delete the invalid word and so on. Thus, the key words which reflect the document information are extracted from them. In view of the authority of the document, it is a crucial step for the preprocessing of document that we extract multi-attribute feature value before the above operation. The multi-attribute feature value of the document contains global and local feature attributes. The feature vector of feature values will affect the final score of the document, the cloud server make sorting according to the document relevance score when it returns the retrieval ranking results.

Definition 1 *GAD* (the global-attribute of the document): *GAD* is essential attribute of the document, such as the number of document quoted and downloaded, the date of publication and so on.

Definition 2 *LAK* (the local-attribute of the keyword): *LAK* is essential attribute of the keyword, such as the *TF-IDF* of keyword, keyword length, keyword frequency, location of the document and so on.

Definition 3 (multi-attribute feature vector): The multi-attribute feature vector is made up of the local attributes of the key words, the global attributes from the document and so on, which is shown as Figure 2.

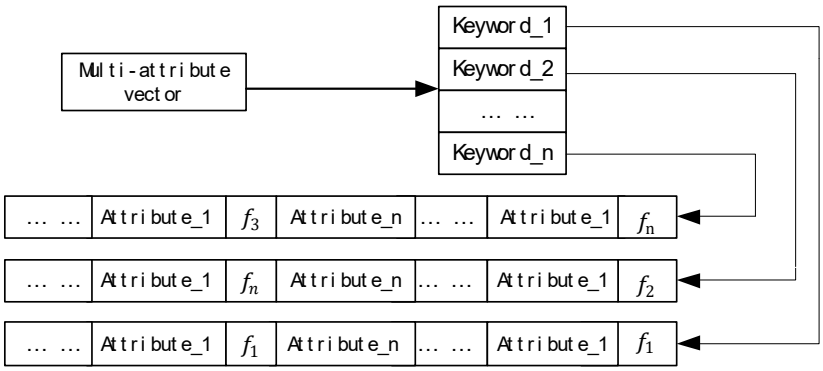

Figure 2. Multi-attribute Vector of Documents

Keyword_1, *keyword_2*, ⋯, *keyword_n* are the keywords extracted from the document. And *attributes_1*, *attribute_2*, ⋯, *attribute_n* are the attribute values which contains local and global attribute. According to the above definition, it can be seen that the traditional ranking algorithm calculates the score of the document according to the weight of the keyword, which is similar to the *TF-IDF* algorithm. However, the traditional ranking algorithm has only a single local property, which couldn't represent the entire document information. Thus, it may generate the insufficient retrieval accuracy during the processing of calculation.

4.2 The Construction method of Keyword Fuzzy Set

How to construct some keyword fuzzy sets according to a keyword? Although there are various kinds of methods for judging the similarity of string, the method of

edit distance is used in this paper. Edit distance [24] $ed(w_1, w_2)$ refers to the minimum number of operations about converting a string w_1 into another string w_2, which is mainly composed of three basic operations. (1) Insert: it inserts an arbitrary character into the string anywhere. (2) Replace: it replaces the other character with any character in the string. (3) Delete: it deletes the random character in the string. In this paper, $S_{w,d}$ shows that the w keyword and a given d threshold which meet all of the fuzzy keyword set w.

In order to complete the retrieval and the relevance sort of encrypted data in cloud, the fuzzy set construction method is used in this paper based on wildcard. In the construction method, we use wildcard to replace edit operation for the keyword string. As for the w_i keyword whose distance is d, its fuzzy set is $S_{w_i,d} = \{S'_{w_i,0}, S'_{w_i,1}, ...S'_{w_i,d}\}$, which means that there are d wildcards in the fuzzy keyword set and each wildcard represents an editing operation.

For example, If we structure the fuzzy set of $FUZZY$ keyword based on wildcard and edit distance is 1, the fuzzy keyword sets is that $S_{FUZZY,1}$ = { $FUZZY$, $F*UZZY$, $*FUZZY$, $*UZZY$, $F*ZZY$, $FU*ZZY$, $FU*ZZY$, $FUZ*ZY$, $FU*ZY$, $FUZZ*Y$, $FUZZ*$, $FUZZY*$}. If the length of the w_i keyword is L, the size of the $S_{w,1}$ is $2*L+1+1$. Moreover, the edit distance of $S_{w,2}$ is 2, its size is $C_{L+1}^1 + C_L^1 * C_L^1 + 2C_{L+2}^2$ and the edit distance of $S_{w,3}$ is 3, its size is $C_L^1 + C_L^3 + 2C_L^2 + 2C_L^2 * C_L^1$, Therefore, if we construct the fuzzy set whose distance is d, the size of the fuzzy keyword set is only $O(L^d)$.

In the traditional file retrieval field, people usually use the classic inverted index structure to achieve the rapid retrieval of documents. By establishing the mapping structure of the keyword and the document, the key words are used as the index list to point to the address of the inverted document. The index list is a collection of all keywords in the document, which is composed of a fuzzy keyword, each containing a fuzzy keyword that has been constructed and the corresponding document storage address.

(1) The calculation of the weight of keyword

TF-IDF algorithm is generally used in the calculation of the weight of keyword, but there are great limitations. *TF-IDF* mainly includes *TF* (*Term Frequency*) and *IDF* (*Inverse document frequency*). Weight calculation formula is shown as follows:

$$Weight_w = tf_{p,m} \times \log \frac{N}{pf + 1} \quad (1)$$

N represents the total number of the document sets, and tf represents the frequency of the w_p, p_f represents that how many w_p of documents had appeared. According to the above formula, we can know that the frequency of a word is very high and the document containing this word is less, which denotes that this word has good category discrimination and can represent the main content of this document. On the contrary, the lower the frequency, the more the number of the document, the main content is not representative about the document and its weight value will be lower. It can be seen that the formula only takes the frequency of keywords into account which neglects the actual location of the key words in the document as well as the characteristics of the document. So that it couldn't represents the authority of the document. The traditional calculation is based on the total amount of the given file to calculate the weights of the keyword. Once we add the new file to it, according to the traditional calculation rules, all of the index established need to be established again, which will obviously reduce the efficiency of time.

In this paper, we introduce a new method to calculate the dynamic data, which is used to generate the index of dynamic documents. We call this algorithm for the *TF-ICF* [25] (*Term Frequency-Inverse Corpus Frequency*) algorithm. The main idea of this algorithm is that the document set (e.g., a document set composed of the journals in computer) is stable, and the frequency of fixed key words will be stable. So we can replace the inverse document frequency of a document which possesses the same type and different quantity document with the document set with a certain number.

Definition 4: In the *TF-ICF*, the calculation method of feature weight on the keyword in the document is shown as follow:

$$Weight_w = \ln(1+tf) \times \ln(\frac{N+1}{pf+1}) \quad (2)$$

N represents the total number of the document sets, *tf* represents the frequency of the w_p occurring in the f_m document, and p_f represents that how many documents w_p appear. Experiment shows that *ICF* will be stable when the number of document to a certain extent. For example, the threshold value of the number is 1000; the document will update below the threshold, such as delete, modify, and add and so on. The frequency of *ICF* needs to be calculated again. If else, it would be opposite, there is no need to recalculate the *ICF* value. You can directly use the prior *ICF* replace the new *ICF*. By improving the *TF-ICF* algorithm, the dynamic update of the data is realized in the cloud computing.

(2) The score function of document

The cloud server makes a sort and returns the most relevant *top-K* documents to users, which are the highest score. In this scheme, the multi-attribute score function of the document will use the multi-attribute to calculate the score of the document, which is composed of the local attribute and the global attribute. Each weight ratio for the attributes is not identical each other.

Definition 5: The multi-attributes of the w_i keyword respectively is $a_1, a_2, \ldots, a_n$, the corresponding weights respectively is $\beta_1, \beta_2, .., \beta_n$, and $\sum_1^n \beta_n = 1$, the multi-attribute score function of the document ranking algorithm is:

$$Score_{FID_{w_i}} = \sum_{i=1}^{n} a_i \times \beta_i \quad (3)$$

The frequency of each keyword is calculated through the existing *TF-ICF* algorithm and the location of each keyword is calculated by the corresponding Table 1. Different multi-attribute weights have the different proportion to the total score function. After the actual test, we believe that the more ideal weight percentage is: the keyword is 0.5, the position weight is 0.2, the frequency of the cited document is 0.15, and the weight of the download times is 0.15. The multi-attribute score function for the document is:

$$0.5Weight_w + 0.2loc\_wei + 0.15ref\_num + 0.15dow\_num \quad (4)$$

So far, the final similarity score of the document is obtained, and the fuzzy keyword set constructed by the data owner is:

$$\left\{ \left\{ T_{w_i} \right\} w_i \in S_{w_i,d}, f(sk, FID_{w_i} \| w_i), Score_{FID_{w_i}} \right\} w_i \in W$$

4.4 The Retrieval Process of Fuzzy Keyword with Multiple Attributes

The retrieval process based on the cloud system model of retrieval method in Section 3 can be described as follows:

(1) The preprocessing of the document: the data owner preconditioned the document in the initial stage of the document. It includes the extraction of keywords and document multi-attribute vector. Firstly, we extract keyword according to the segmentation algorithm and construct the fuzzy set $W = (w_1, w_2, ..., w_p)$ by using the fuzzy sets algorithm based on wildcard. Secondly, we extract multi-attribute feature vector for the document. At last, the trapdoor $T_{w_i} = f(sk, w_i)$ is generated through the symmetric encryption function, where *sk* is the key.

(2) Create an index: the trapdoor sets $\{T_{w_i}\} w_i \in S_{w_i,d}$ is generated after each fuzzy keyword is built. Then the index set and the fuzzy keyword sets $\{\{T_{w_i}\} w_i \in S_{w_i,d}, f(sk, FID_{w_i} \| w_i), Score_{FID_{w_i}}\}\ w_i \in W$ are encrypted through a one-way function *F*( ). After that, they are uploaded to the cloud server.

(3) The retrieval phase: the authorized user gets the private key *sk* and retrieves the keyword *Q*. The user will input *(Q, k)*, where *k* is the edit distance. First of all, the trapdoor set $T_Q = f(sk, Q)$ of *Q* is computed according to the fuzzy set scheme based on wildcard is and is sent to the cloud server.

(4) The back stage: the cloud server begins to search for matching in the encrypted index table after it receives retrieval request service sent by the user. If the query does not exist, the keyword does not exist. Otherwise, FID_Q is document identifier, and the relevant Score are obtained. After all trapdoor queries are completed, the fuzzy keywords that are likely to be consistent with all the queries are sorted according to the correlation score and the final sort of top-k encryption file's $ID: \{f(sk, FID_{w_i} \| w_i\}$ is returned. Finally, the encrypted file sequence received by the user is decrypted with the authorization private key *sk* to obtain the goal file retrieved.

5 Experiments and Result Analysis

Our experiment environment is: the operating system is Window 7 with 32 bit, CPU is the Intel Pentium Dual-Core Processor with 2.6GHz and 4G memory, language development environment is original Lucene-3.6 search engine toolkit, the *Java* toolkit which includes *PDFBox*, and Java language development is selected which version is the *JDK* is 1.7. In this paper, we randomly select 1030 documents which belong to the *IEEE* database and the Springer database as the test file. The size of the document set is 780MB approximately. We use *PDFBox* to process the document, and we extract the keyword and filter out the forbidden word. Moreover, the multi-attribute (such as two-attribute, three-attribute, four-attribute and so on) characteristic of the document is specified which contains cited frequency and download times.

In order to make the experimental result more persuasive, we use the relevant information to retrieve two evaluation indexes, namely the recall rate and the precision rate. Considering the actual situation in the retrieval, each page will show 20 results and most users may only read the first few pages. In order to enhance the authenticity and the accuracy of the experimental results, we only choose the first 3 pages of the retrieval results.

5.1 The Result of Retrieval Experiments

(1) The precision rate of retrieval query

First, we choose single-attribute, two-attribute, three-attribute and four-attribute to test the accuracy of retrieval query, respectively. In order to show the experimental results, we use top-10, top-20, top-30, top-40, top-50, top-60 as choosing condition to choose the returning results of retrieval query, respectively. The results are shown as Figure 3.

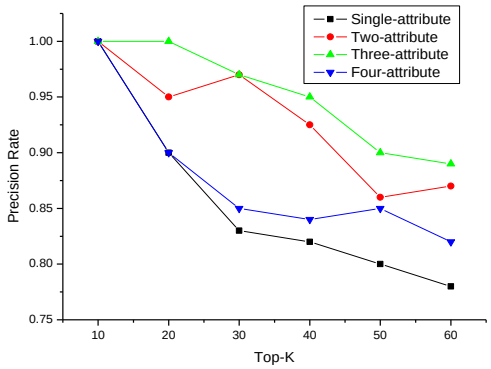

Figure 3. The Precision Rate of The Retrieval System with Different Attribute

(2) The recall rate of retrieval

We first selected keyword w_i whose related document total is 93 to test the recall rate. Then we use w_i to retrieval in the single-attribute, two-attribute, three-attribute and four-attribute retrieval system, respectively. The experimental results are shown as Figure 4.

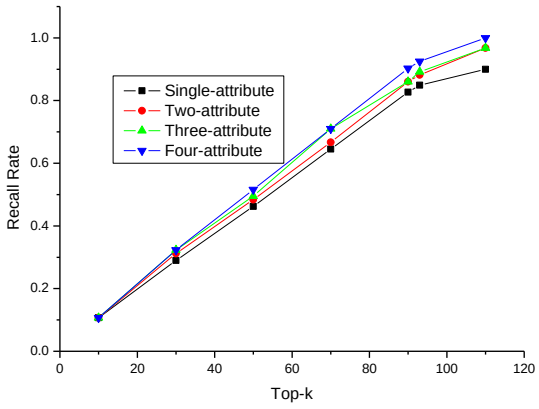

Figure 4. The Recall Rates of The Retrieval System with Different Attribute

(3) The consumption of retrieval time

We choose the same keyword w_i to query in the single-attribute, two-attribute, three-attribute and four-attribute retrieval system, respectively. In order to test the keyword retrieval time, we use top-10, top-20, top-30, top-40, top-50, top-60 as choosing condition to choose the retrieval time, respectively. The results are shown as Figure 5.

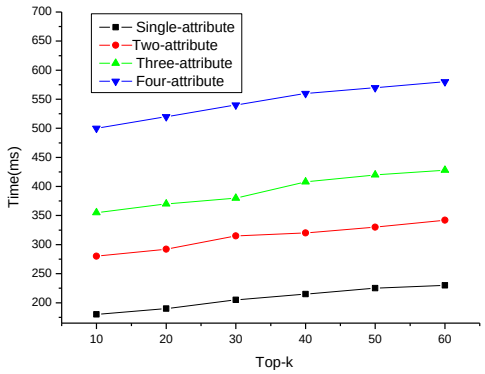

Figure 5. The Retrieval Time of The Retrieval System with Different Attribute

5.2 The Analysis of Retrieval Results

From the Figure 3, when the number of documents returned is less,we can see that the accuracy rate of single-attribute and multi-attribute retrieval results are not very different. But, the accuracy rate of multi-attribute retrieval was significantly higher than the single-attribute when results of return are increased. it can be see that accuracy of multi-attributes has increased about 9% than accuracy of single-attribute retrieval. So, we can see when on the same conditions,the more attribute is,the higher accuracy is. It can be seen that the multi-attribute ciphertext retrieval system is better than the traditional single attribute on the precision.

From the Figure 4, we can see when on the same conditions ,the recall rate of the single-attribute retrieval is significantly less than the multi-attribute retrieval. Especially,the multi-attribute retrieval completed the full recall rate, that is to say, it have recalled all of the related document.

From the Figure 5, it can be seen that the retrieval time is similarly related to the number of returned documents. If the number of documents increases,the retrieval time also will increase. But in the same condition, the consumption time of multi-attribute is greatly shorter than the single-attribute, which shows that the multi-attribute is effective in the real environment.

6 Conclusions

Based on the above experiment results, it is proved that the mechanism proposed in this paper not only can effectively improve the accuracy of the data retrieval, but also can greatly reduce the bandwidth consumption of system.

Although this mechanism has some advantages, it also has some disadvantages. The main performance is that it needs to be detected constantly to find the most reasonable keyword. In large-scale cloud computing, this may need a large amount of time. Therefore, in future work, we will further consider the technology and method to support multi-keyword search. Thus, it can be more effective and accurate for the retrieval of encrypted data.

Acknowledgments

This paper is a revised and expanded version of a paper entitled “Multi-attribute Fuzzy Keyword Retrieval Method for Secure Data Service in Cloud Computing” presented at the 9th International Conference on Security Technology, Jeju Island, Korea (2016).

References

[1] D. G. Feng, M. Zhang, Y. Zhang, Z. Xu, "Study on Cloud Computing Security", Journal of Software , vol. 22, no. 1,(2011) , pp. 71-83.
[2] Z. Xiao and Y. Xiao, "Security and Privacy in Cloud Computing", IEEE Communications Surveys & Tutorials, vol.15, no.2, (2013), pp. 843-859.
[3] D. Aggeliki, T. Dimitri, S. Timos, "Top-k-size Keyword Search on Tree Structured Data", Information Systems, vol.47, (2015), pp. 178-193.
[4] D.X. Song, D. Wagner, A. Perrig, "Practical Techniques for Searches on Encrypted Data", IEEE Symposium on Security & Privacy, (2000), pp. 44-55.
[5] Y. C. Chang, M. Mitzenmacher, "Privacy Preserving Keyword Searches on Remote Encrypted Data", Proceedings of the 3th International Conference on Applied Cryptography and Network Security, New York, USA, (2005) June 7-10
[6] D. Boneh, E. Kushilevitz, R. Ostrovsky, E. William, "Public Key Encryption that Allows PIR Queries", Proceedings of the 27th Annual International Cryptology Conference, Santa Barbara, USA, (2007)August 19-23.
[7] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, l. J. Malone, G. Neven, P. Pailier, "Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions", Journal of Cryptology, vol. 21, no.3, (2008) , pp. 350-391.
[8] R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky, "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions", Journal of Computer Security, vol.19, no.5, (2011), pp.895-934.
[9] P. Golle, J. Staddon, B. Waters, "Secure Conjunctive Keyword Search over Encrypted Data", Proceedings of the 2th International Conference on Applied Cryptography and Network Security, Yellow Mountain , China, (2004) June 8-11
[10] D. Boneh, B. Waters, "Conjunctive, Subset, and Range Queries on Encrypted Data", Proceedings of the 4th Theory of cryptography Conference, the Netherlands, (2007) February 21-24.
[11] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, & W. Lou, "Fuzzy Keyword Search over Encrypted Data in Cloud Computing" , Proceedings of the 29th conference on Information communications, NJ, USA, (2010) March 14-19.
[12] A. Swaminathan, Y. Mao, G. M. Su, "Confidentiality-preserving Rank-ordered Search", Proceedings of the 2007 ACM workshop on Storage security and survivability, Alexandria, USA, (2007) October 7-12.
[13] B. Christoph, B. Richard, H. Pieter, W. Jonker, "Conjunctive Wildcard Search over Encrypted Data", Proceedings of the 8th VLDB Workshop on Secure Data Management, Seattle, WA, USA, (2011) September 2.
[14] C. Wang, K. Ren, S. Yu, "Achieving Usable and Privacy-assured Similarity Search over Outsourced Cloud Data", Proceedings of the 31th conference on Information communications, NJ, USA, (2012) March 25-30.
[15] C. Wang, N. Cao, J. Li, K. Ren, W.J. Lou, "Secure Ranked Keyword Search over Encrypted Cloud Data", Proceedings of the 30th International Conference on Distributed Computing Systems, Genoa, Italy, (2010) June 21-25.
[16] C. Wang, N. Cao, K. Ren, "Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data", IEEE Transactions on parallel and distributed systems, vol.23, no.8, (2012), pp. 1467-1479.
[17] N. Cao, C. Wang, M. Li, K. Re, and W.J. Lou, "Privacy-preserving Multi-keyword Ranked Search over Encrypted Cloud Data", Proceedings of the 32th IEEE International conference on Computer Communications, USA ,(2011) April 10-15.
[18] I.H. Witten, A. Moffat, T.C. Bell, "Managing Gigabytes: Compressing and Indexing Documents and Images", Morgan Kaufmann Publishing, San Francesco, (1999).
[19] N. Cao, C. Wang, M. Li, K. Ren, W. Lou, "Privacy-preserving Multi-keyword Ranked Search over Encrypted Cloud Data", IEEE Transactions on Parallel and Distributed Systems, vol.25, no.1, (2014), pp. 222-233.
[20] Z. Xu, W. Kang, R. Li, and K. Yow, C.Z. Xu, "Efficient Multi-keyword Ranked Query on Encrypted Data in the Cloud", Proceedings of 18th International Conference on Parallel and Distributed Systems, Singapore, (2012) December 17-19.
[21] M. Anuradha, G. A. Patil, "Efficient Keyword Search over Encrypted Cloud Data", International Conference on Information Security & Privacy, Procedia Computer Science, Vol. 78, (2016), pp.139-145.
[22] A. D. Amol, J. Sanjay, "Confidentiality-conserving Multi-keyword Ranked Search above Encrypted Cloud Data", Procedia Computer Science , vol 79, (2016) , pp. 845-851.
[23] L. Yunfa, L. Mingyi, X. Nannan, "Multi-attribute Fuzzy Keyword Retrieval Method for Secure Data Service in Cloud Computing", The 9th International Conference on Security Technology, Advanced Science and Technology Letters, Jeju Island, Korea , (2016) November 24-26.
[24] E. S. Ristad, P. N. Yianilos, "Learning String-edit Distance", Pattern Analysis and Machine Intelligence, vol.20, no.5, (1998), pp.522-532.

[25] J. W. Reed, Y. Jiao, T. E. Potok, B.A. Klump, and M.T. Elmore, A.R. Hurson, "TF-ICF: A New Term Weighting Scheme for Clustering Dynamic Data Streams", Proceedings of the 5th International Conference on Machine Learning and Applications, Orlando, Florida ,(2006) December 14-16.

Authors

Yunfa Li, born in 1969, is a Ph.D. and associate professor in school of Computer Science and Technology at Hangzhou Dianzi University. His research interests include cloud computing, cluster computing, grid computing, big data and system security. Contact him at yunfali@hdu.edu.cn.

Mingyi Li, born in 1991, is a postgraduate in school of Computer Science and Technology at Hangzhou Dianzi University. His research interests include cloud computing, big data and system security. Contact him at 952921628@qq.com

Yangyang Shen, born in 1991, is a postgraduate in school of Computer Science and Technology at Hangzhou Dianzi University. His research interests include cloud computing, big data and system security. Contact him at 2644294165@qq.com

