Searchable Encryption For Text Document With Ranked Results

Iuon-Chang Lin[#1], Ya-Hui Liu[*2]

[#*]*Department of Management Information Systems, National Chung Hsing University*

Taichung,Taiwan R.O.C

[1]iclin@dragon.nchu.edu.tw

[2]thankuever@gmail.com

***Abstract*—With the Internet develops fast and most the cloud storages are free, more and more data outsource to the cloud. They may include some sensitive and privacy data, such as financial data, health data and so on. There is a problem that the cloud can't fully trusted. If the cloud leaks some important data, it may cause some risk. Document owner must encrypted document before uploading it to a remote, untrusted server. Encryption keeps the document privacy and security. However, the capability of searching will lose at the same time. When the document owner wants to retrieve some data over encrypted data, it will be a big problem. This paper proposes a method that not only can keep data secure, but the feature of search isn't lost. The method of the paper is embedding the keyword into the text document. When the document owner or user want to search the document, he/she can transmits keywords to the cloud. The cloud can restore the keywords from each document to search instead of decrypting the whole document database.**

***Keywords*— Searchable Encryption, Cloud Storage Security**

1. INTRODUCTION

Cloud storage offers a free space to people can save their documents, and it's convenient to access documents anytime and anywhere through the Internet. It saves many cost such as hardware cost and human cost. However,44% people in EU [1] who aware of cloud storage still keep away from the useful technology in 2014. Security and privacy problems are the main reason for not using cloud storage. That is, if the cloud can overcome the problem ,the intention of use will rise. In order to solve the problem, documents will be encrypted before uploading to the cloud. Document encryption keeps documents security. Any adversary can't know the content of encrypted documents without the keys .Unfortunately, it also removes all search capabilities from the document owner. If document owner wants to get the documents there have two bad choices. One downloads the all encrypted documents, decrypt, and search in decrypt form. The method will bring more cost because the user must download the whole database for just only getting a document. The other method the document owner gives the key to the cloud, let it decrypt document for document owner to search. The movement makes encryption meaningless. To response the demand, document owner need a new technique can search over encrypted data in an efficient and security way. For this propose of above demand, this paper proposes a method which use keyword search over encrypted text document more efficiently.

2. RELATED WORK

The goal of our proposed paper is authorized a group of user-defined people can search on the encrypted text documents which are been uploaded to the untrusted cloud by one document owner. In order to achieve the objective, the technique called searchable encryption have been extensively studied in the literature. Briefly speaking, searchable encryption means a cloud stored encrypted documents which outsourced by document owner, he/she has the capability to search on the encrypted documents and retrieve them. The first searchable encryption is proposed by Song at al. [2] in 2000.It was called SWP which base on secret key cryptography. Even though SWP achieves the goal to enable full-text search over encrypted documents, its efficiency, search functionalities and query expressiveness have much need to improve. After SWP, some

searchable encryption schemes have been proposed to improve. It can classified into based on secret key cryptography [2-3] and based on public key cryptography [4-5] .Searchable encryption based on secret key cryptography means the secret-key user can create the searchable content which will outsourced to the cloud server, and he/she also have the search capability to retrieve the encrypted document. As mentioned above, SWP [2] belongs to the category. SWP searches the encrypted data with sequence scan using a special two-layered encryption. The Goh [3] improve the SWP [2] limitation. The scheme adds a Bloom filter as an index for each document. Searchable encryption based on public key cryptography means the private key owner can search the document which encrypted by corresponding public key. The first public-key searchable encryption scheme is proposed by Boneh et al. [4] .Yang et al. [5] proposed a multi-user searchable encryption scheme. The scheme builds a U-HKey list in the cloud side to manage user enrolment and user revocation.

Keyword-based searchable encryption technology develops so far. The goal of search not only for search the document. It develops more enhanced functions. At the fuzzy keyword search, Li et al. [6] presents "Dictionary-based Fuzzy Set Construction" , and Liu et al. [7] proposed "Wildcard-based Fuzzy Set Construction". According to the set of fuzzy keywords, Li et al. [6] is smaller Liu et al. [7].

3. PROPOSED ARCHITECTURE

We illustrate the system model in this section. As shown in Fig. 1, three entities are defined in our system including the cloud server, the user(s) and the document owner.

3.1. Cloud Server

Cloud server acts as document storage. Search and retrieval document for users. The architecture assumes cloud server is semi-honest which would follow the prescribed procedure of the protocol to return the right document to the user.

3.2. User(s)

One or more people are defined by document owner can use keyword to search and retrieve the document.

3.3 Document Owner

The people own one or more documents which store in cloud and share it to other users.

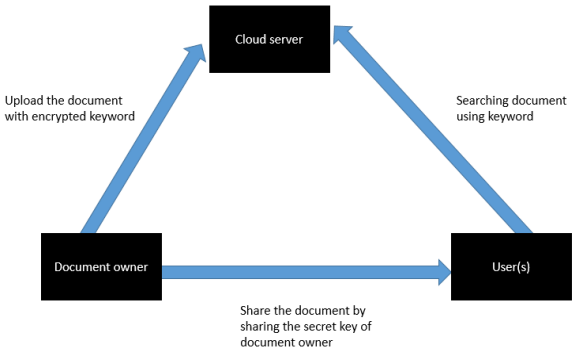

Fig. 1 Communication model for keyword search over encrypted document

4. OUR METHOD

The paper proposes a multi-user multi-keyword searchable encryption scheme means a group of owner-defined users can access the documents that the document owner upload to the cloud using multi-keyword search. In this scheme, document owner uses the concept of histogram shifting [8] to embed keywords into the encrypted text document. Because the cloud server only compares the keywords key-in to the keyword hide in the document instead of decrypting all cloud storage in the phase of query, the scheme improves process efficiency. The following of Figure 2 describes the concept of our proposed method:

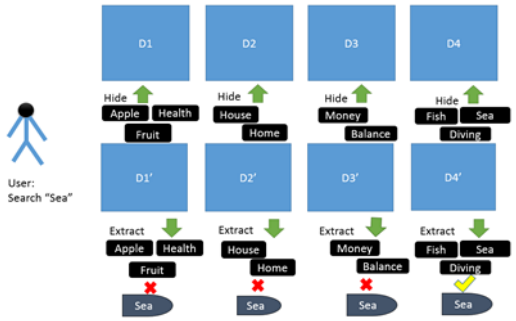

Fig. 2 The concept of proposed method

Our scheme consists of four phases (Setup keyword and document, Embed searchable keyword and auxiliary information, Share key, Query).

4.1. Setup Keyword and Document

Before embedding keywords into document, the keywords and documents need to process the following steps.

4.1.1. Document Filter

Text document certainly includes some unimportant words such as preposition, conjunction, and interjection. These words must frequently appears in the article, but it isn't meaningless for keyword search. As the result, this scheme filters these insignificant words before producing a keyword list for embedding into documents.
Input: Document D
Output: A list of keywords called LIST.

4.1.2. Ranking Importance

We use the technique called TF-IDF(Term Frequency-inverse Document Frequency) to evaluate how important every keyword in the LIST. Finally, we obtain the ranked result of keywords.
Input: LIST
Output: Ranked result of keywords called $RANK_{keyword}$

4.1.3. Document Encryption

In order to protect the security of D they will be encrypted using owner's secret key $Sk_{document}$.
Input: Document D
Output: Encrypted D called D'

4.1.4. Document Encryption

We use the concept of histogram shifting [7] .First, this scheme builds the histogram of D'. The D' is consist of binary bits (0 or 1).We divide them (one bit) into groups of n bits. Second, each group is converted into 0 to 2^n. The x-axis of histogram ranges from 0 to 2^n.The value of 2^ncan control the height of histogram. If 2^n is small, the y-axis (frequency) of the histogram will high. Finally, we observe the size of the peak point of histogram to evaluate how many keywords can embedding into the D'.
Input: Encrypted Document D'
Output: The size of peak point of histogram called P_{size}.

4.1.5. Keyword encryption

In order to protect the security of keywords, they will be encrypted using owner's secret key $Sk_{keyword}$.We choose the P_{size} of $RANK_{keyword}$.
Input: The P_{size} of $RANK_{keyword}$.
Output: The Psize of encrypted $RANK_{keyword}$ for D' called $D_{keywords}$.

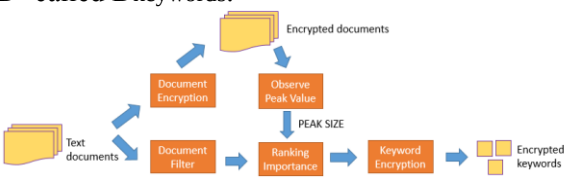

Fig. 3 The process of setup keyword and document

4.2. Embed Searchable Keyword and Hide Auxiliary Information

The following step uses the peak point of histogram to embed keywords to make every document can be searched from the cloud by the users.

4.2.1. Shift Histogram

Shift the pixels of histogram for embedding a keyword. P means the highest frequency of histogram, Z means the lowest frequency of histogram.
1. If P>Z → To shift the range of the histogram, [Z+1, P-1], to the left-hand side by 1 unit.
2. If P<Z → To shift the range of the histogram, [P+1, Z-1], to the right-hand side by 1 unit.

Input: Encrypted document D'
Output: The position for embedding keywords, peak point P, zero point Z

4.2.2. Embed Keyword

Embedding the Encrypted keywords for D' into the peak of histogram of D'
The $D_{keywords}$ turn into binary sequence and embed into D'. By the way, adding a symbol between $D_{keywords}$,
1. If P>Z → To be embedded bit is "1", the pixel value is changed to P-1.
If the bit is "0", the pixel value remains.
2. If P<Z → To be embedded bit is "1", the pixel value is changed to P+1.
If the bit is "0", the pixel value remains.

Input: $D_{keywords}$,D'
Output: The D' includes $D_{keywords}$

4.2.3. Hide auxiliary information

The peak point P and zero point Z are hidden into the D' for the auxiliary information at the query time.Hide the peak point P and zero point Z into a document using pseudo-random number generator (PRNG). The seed of PRNG is $Sk_{document}$ and $Sk_{keyword}$, PRNG($Sk_{keyword}$ & $Sk_{document}$) generates a sequence of numbers. PRNG($Sk_{keyword}$ & $Sk_{document}$)->{Number}.The random number sequence serves as the hiding location of peak point and zero point. Put the P and Z into the D with Psize of encrypted keywords.
Input: The D' include $D_{keywords}$.
Output: The D' include $D_{keywords}$ and auxiliary information.

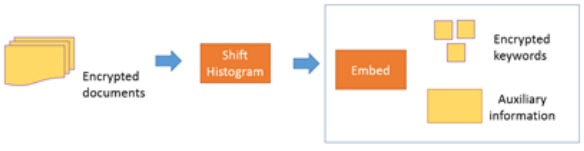

Fig. 4 The process of Embed searchable keyword and hide auxiliary information

4.3. Share Key

The document owner not only uses the document by himself/herself but shares with other peoples in real life. For the propose to share documents,we need to share the $Sk_{keyword}$ and $Sk_{document}$ with the people can uses the auxiliary information to help them search document. We encrypted the $Sk_{keyword}$ and $Sk_{document}$ by the public key pk of user.

$pk_{user}(Sk_{keyword} \& Sk_{document})$ be sent to the user be allowed to access the document. pk_{user} means the user's public key.

The user get the $pk_{user}(Sk_{keyword} \& Sk_{document})$.Then, he/she uses their private key to receiving the $Sk_{keyword}$ and $Sk_{document}$ of document owner.

Input: $pk_{user}(Sk_{keyword} \& Sk_{document})$

Output: $Sk_{keyword} \& Sk_{document}$

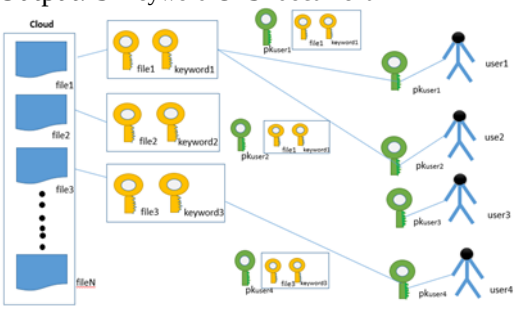

Fig. 5 The process of share key

4.4. Query

In order to access the document with keywords, the user sends N keyword to the cloud.

4.4.1. Generate Trapdoor

User sends (PRNG($Sk_{keyword} \& Sk_{document}$) , $Sk_{keyword}\&Sk_{document}$(keyword1)…$Sk_{keyword}\&Sk_{document}$(KeywordN)) as trapdoor to the cloud server.

Input:Trapdoor

4.4.2. Search and Count Keyword

The cloud server use PRNG($Sk_{keyword} \& Sk_{document}$) to get the peak point P and zero point Z. Using the P and Z to get $D_{keywords}$. Equal($D_{keyword}$.,$Sk_{keyword}\&Sk_{document}$(keyword 1)…,$Sk_{keyword}\&Sk_{document}$(KeywordN)) is true or false.

A number Count to record how many the number of keyword is equal.

The following step extract the $D_{keywords}$

1. If P>Z → P can extract 0 and P-1 can extract 1.
2. If P<Z → P can extract 0 and P+1 can extract 1.

If it is true,COUNT+1, continue to compare keywords until Count=N.

If it is false, continue to compare keywords until COUNT=N.

Then, the value of COUNT=0.It will output ⊥.

Input: Trapdoor

Output: D' with COUNT rank

4.4.3. Recovery

Extract all keywords and auxiliary information recover the encrypted document .

Using the value of PRNG($Sk_{keyword} \& Sk_{document}$) to extract P and Z.

Shift the pixels of histogram for recovering the encrypted document.

1. If P>Z → To shift the range of the histogram , [Z+1, P-1], to the right-hand side by 1 unit.
2. If P<Z → To shift the range of the histogram , [P+1, Z-1], to the left-hand side by 1 unit.

Return the D' which contain the keywords that user query in the form of ranked result by the value of COUNT to user.

Input: D' with keywords and auxiliary information

Output: D'

4.4.4. Decryption

The user choose a result D' to use the $Sk_{document}$ to get the original document D.

Input: D', $Sk_{document}$

Output: D

Fig. 6 The process of query.

5. Conclusions

The paper proposes a multi-user multi-keyword searchable encryption scheme which a group of owner-defined user can searches over encrypted data. Because the search result is be ranked, it more matches the demand of users. Cloud server searches using the equal of encrypted keyword and trapdoor. It can't know what the keyword is, so user can keep the keyword privacy. Anyone without secret key can't access document, so the document is secure. The architecture satisfied the demand of keyword privacy and document security.

REFERENCES

[1] Heidi SEYBERT, Petronela REINECKE,"Half of Europeans used the internet on the go and a fifth saved documents on internet storage space in

2014”http://ec.europa.eu/eurostat/statisticsexplained/index.php/Internet_and_cloud_services_-_statistics_on_the_use_by_individuals

[2] D. Song, D. Wagner, and A. Perrig, Practical Techniques for Searches on Encrypted Data. Proc. IEEE Symp. on Security and Privacy, S&P’00, pp. 44-55, 2000

[3] E-J. Goh. Secure indexes. Technical Report 2003/216, IACR ePrint Cryptography Archive, 2003. See http://eprint.iacr.org/2003/216.

[4] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, “Privacy-preserving multikeyword ranked search over encrypted cloud data,” Proceedings of IEEE INFOCOM 2011, pp. 829-837, 2011.

[5] D. Boneh, G. di Crescenzo, R. Ostrovsky, and G. Persiano, Public key encryption with keyword search. Proc. Eurocrypt’04, pp. 506-522, 2004.

[6] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, “Fuzzy Keyword Search Over Encrypted Data In Cloud Computing,” in Proc of IEEE INFOCOM’10 MiniConference, San Diego, CA, USA, 2010

[7] C. Liu, “Fuzzy keyword search on encrypted cloud storage data with small index”, Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference, (2011), pp. 269-273.*Specification*, IEEE Std. 802.11, 1997.

[8] Z. Ni, Y. Q. Shi, N. Ansari and W. Su: Reversible data hiding. IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 3, pp. 354-362, March 2006.