 CMC, vol.60, no.1, pp.351-363, 2019

Fuzzy Search for Multiple Chinese Keywords in Cloud Environment

Zhongjin Fang[1, 2], Jinwei Wang[1, *], Baowei Wang[1], Jianjun Zhang[3] and Yunqing Shi[4]

Abstract: With the continuous development of cloud computing and big data technology, the use of cloud storage is more and more extensive, and a large amount of data is outsourced for public cloud servers, and the security problems that follow are gradually emerging. It can not only protect the data privacy of users, but also realize efficient retrieval and use of data, which is an urgent problem for cloud storage. Based on the existing fuzzy search and encrypted data fuzzy search schemes, this paper uses the characteristics of fuzzy sounds and polysemy that are unique to Chinese, and realizes the synonym construction of keywords through Chinese Pinyin and Chinese-English translation, and establishes the fuzzy word and synonym set of keywords. This paper proposes a Chinese multi-keyword fuzzy search scheme in a cloud environment, which realizes the fuzzy search of multiple Chinese keywords and protects the private key by using a pseudo-random function. Finally, the safety analysis and system experiments verify that the scheme has high security, good practicability, and high search success rate.

Keywords: Cloud storage, fuzzy search, Chinese keyword, synonym, encrypted search.

1 Introduction

With the development and popularization of information technology, the number of data files stored by local users and enterprises in the local area is growing, and the pressure on local storage is increasing [Patnaik (2016)]. The local hardware failure or severe damage will greatly affect the users and enterprises for the use of data, or even the loss of important data will always be. Therefore, cloud storage services are popular with more and more users with the advantages of convenience and cost saving [Li and Zhang (2013)]. However, the use of cloud storage services has some constraints. For example, some data related to the trade secrets of enterprises must be protected from being used illegally. As a result, the data is typically encrypted locally and then outsourced to a cloud storage server, which brings great trouble with the use of data [Fu, Shu and Sun

[1]School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing, 210044, China.

[2] Binjiang College, Nanjing University of Information Science & Technology, Wuxi, 214105, China.

[3] College of Engineering and Design, Hunan Normal University, Changsha, 410081, China.

[4] Department of Electrical and Computer Engineering, New Jersey Institute of Technology, NJ 07102, USA.

[*] Corresponding Author: Jinwei Wang. Email: wjwei@nuist.edu.cn.

(2014)]. Due to the limitation of network bandwidth and local storage capacity, it is impossible for users to download all the data and then decrypt it. In brief, the research design of cloud data services that support privacy protection and encrypted search is a research topic of great significance and practical value [Lucas, Seny and Fabian (2005)].

Researchers at home and abroad have made many research results in keyword search based on public key encryption. For example, Song et al. proposed a ciphertext keyword search based on symmetric encryption in 2000 [Song, Wagner and Perrig (2000)]; Boneh et al. proposed a ciphertext keyword search based on public key encryption in 2007 [Boneh and Waters (2007)]. In recent years, many achievements have been made. For example, Li et al. put forward the use of edit distance to quantify the similarity between keywords and form a new technology based on keyword fuzzy search [Li, Wu and Yuan (2016)]; Hore et al. proposed a range search method based on range grouping in 2012 [Hore, Mehrotra, Canim et al. (2012)].

From the earlier analysis, it can be seen that the research on encrypted data search has achieved productive results, but there are still many problems to be solved in semantic search and fault tolerance: the earlier keyword-based search method only focuses on the precise or fuzzy matching of keywords, which is not completely applicable to the Chinese environment [Cao, Wang, Li et al. (2014)]. In addition to the glyphs, the Chinese characters contain two parts: pinyin and meaning. Pinyin consists of initials and finals. Therefore, under normal circumstances, a Chinese keyword has many synonyms, synonyms, and similar words. At present, there are few studies on Chinese keyword search with similar semantics and speech at home and abroad [Wang, Cao and Li (2010)]. Hence, this paper proposes a fuzzy search strategy based on keyword-based encrypted cloud data, and explores the execution efficiency and method of fuzzy search of Chinese fuzzy sounds and synonymous keywords in the cloud storage environment.

In this paper, we focus on the implementation of Chinese fuzzy keyword search strategy which is suitable for cloud environment and can protect privacy. We propose a method to construct the synonymy and homonym sets of Chinese keywords by inter-translation between Chinese and English and the fuzzy pinyin strategy. We provide an effective fuzzy keyword search scheme based on Chinese keywords for cloud data retrieval which can protect the privacy of keywords. Fuzzy keyword search greatly improves the availability of the system by returning the matched files. When the keyword entered by the user matches the predefined keyword exactly, the matching files will be returned. When the exact matching fails, keyword-based synonyms and homonyms are used to return the closest possible matching files. Specifically, the similarity of different language expressing is used to achieve the similarity of Chinese keywords, and it develops a new ciphertext retrieval technology. Based on the homophone and synonym keyword sets, we propose an efficient fuzzy keyword search scheme. The strict security analysis shows that the proposed scheme is secure and privacy-preserving, which is the goal of the Chinese fuzzy keyword search scheme.

The rest of paper is organized as follows: Section 2 introduces the system model, threat model, our design goal and briefly describes some necessary background for the techniques used in this paper. Section 3 shows a straightforward construction of fuzzy keyword search scheme. Section 4 provides the detailed description of our proposed

schemes, including the efficient constructions of fuzzy keyword set and fuzzy keyword search scheme. Section 5 presents the security analysis. Section 6 presents the experiments analysis. Finally, Section 7 concludes the paper.

2 Problem descriptions

2.1 System and threat model

The architecture of the data storage service in the cloud environment is shown in Fig. 1, which presents that architecture contains three main entities: a data owner, user, and cloud service provider. Where, the data owner can be an individual or enterprise user who stores the data file set $C = (F_1, F_2, \cdots, F_n)$ on the cloud server. Different keyword sets related to file set *C* are predefined and expressed as $W = (w_1, w_2, \cdots, w_p)$. To ensure that sensitive data is not used by unauthorized persons, data set C needs to be encrypted before outsourcing to the cloud server. Since there are a large number of similar sounds and synonyms in Chinese, in order to improve the utilization efficiency and the retrieval success rate of cloud data, the architecture needs to provide fuzzy search function of fuzzy sounds and synonyms for encrypted data. The data owner needs to generate the private key *sk* for the search request and distribute it to other authorized users, such as team members or enterprise employees. When the private key allocation is completed, for any input keyword w, in order to safely search out the relevant file set, the authorized user uses the private key *sk* and one-way generation function to convert the keyword which need be searched into a search request (hereinafter referred to as a trap door) and submit it to the cloud server. The cloud server caries out the search without decrypting the data and sends the searched set of target files (denoted as FID_w) associated with the keyword w or the ambiguous sound or synonym of w to the data searcher.

This paper considers that the cloud server involved in the cloud data service architecture is honest, but curious, it can correctly execute the specified protocol specification, but it will infer and analyze the relevant information through the input of users. Therefore, we still follow the security definitions involved in traditional symmetric encryption when designing the synonymous keyword search scheme. Except for search results and search models, anything else related to stored files and indexes should be not revealed.

2.2 Goal of the design

In order to realize a safe and efficient synonym keyword search for the above model of cloud data, this paper needs to achieve the following goals: 1) fuzzy keyword search function: explore efficient and correct fuzzy keyword search strategy for outsourcing cloud data of different mechanism design; 2) guarantee security: prevent the cloud server from learning knowledge related to data files or keywords in the search process; 3) guaranteed efficiency: achieve the above goals with the smallest possible occupation of storage, communication, and computing resources.

2.3 Preliminary knowledge

C: The set of outsourced files, represented as the set of n data files, i.e., $C = (F_1, F_2, \cdots, F_n)$.

W: A collection of different keywords extracted from the file set *C*, expressed as a set of p words, i.e., $W = (w_1, w_2, \cdots, w_p)$.

I: An index established for a privacy-protected fuzzy keyword search.

T_w: Trapdoor, which is a search request, generated by a one-way function after the user inputs the search keyword w.

FID_{w_i}: The collection consists of a file set C containing the keyword w_i or its near or synonymous file *ID*.

$f(key,\cdot), g(key,\cdot)$: Pseudo-random function (PRF), defined as: $\{0,1\}^* \times key \rightarrow \{0,1\}^1$.

$Enc(key,\cdot), Dec(key,\cdot)$: Symmetric key encryption/decryption function based on semantic security.

Edit distance: Edit distance is a description of the similarity of strings. For the two words w_1 and w_2, edit distance $ed(w_1, w_2)$ represents the minimum number of operations required for both to implement the transformation, which can be operations to add, modify, and delete characters. For a given word w and integer d, $S_{w,d}$ is used to represent the similar word w', satisfying $ed(w, w') \leq d$.

$SP_{w,d}$: Keyword set corresponding to fuzzy pinyin of keyword w. For a given Chinese keyword w and an integer d, the set of fuzzy tones corresponding to the Pinyin PY_w is $SP_{w,d} = \{w_1', w_2' \ldots\ldots\}$, and the set of all similar Pinyin keyword that satisfies the requirement that the edit distance of the Pinyin of the keyword w is less than d is expressed as $ed(PY_w, PY_{w'}) \leq d$, i.e., $w_i' \in SP_{w,d}, PY_{w_i'} \in S_{PY_w,d}$.

SY_w: A collection of keywords synonymous with the keyword. When a different keyword set $SY_w = (w_1', w_2', \ldots\ldots)$ describing the same thing in the same language is converted into another language, w_i' generally corresponds to the same keyword w_e, then the set $SY_w = (w_1', w_2', \ldots\ldots)$ is a synonym set of the keyword w, where $syn(w) = syn(w_i')$, $syn()$ is a synonymous conversion function. In this paper, synonymous conversion is implemented in Chinese and English.

Fuzzy keyword search: Given a set of n encrypted data files $C = (F_1, F_2, \cdots, F_n)$, a predefined set of different Chinese keywords $W = (w_1, w_2, \cdots, w_p)$, the combination of the input multiple search keywords w and d, i.e., $\{w, d\}$. After performing the synonym keyword search, a file *ID* set $\{FID_{\{wi,di\}}\}$ will be returned, where $w \in \{w, d\}$, $w_i \in SY_w$ or $w_i \in SP_{w,d}$.

3 System framework

3.1 Representation of system framework

According to the above target analysis, the overall framework design of the system is shown in Fig. 1.

3.2 Overview of design

The goal of a fuzzy search is to return as many results as possible (including synonymous and fuzzy tones) based on the set of keywords inputted by different users. However, such

fuzzy search based on keyword synonym and fuzzy word is very challenging for matching cloud data. Any two Chinese words can easily obtain the fuzzy words or synonyms in the plaintext state, but it is difficult to find similar rules after one-way encryption function encryption (such as pseudo-random function or another encryption algorithm). The traditional encryption search strategy searches through equal comparisons between user-submitted search traps and searchable encrypted indexes, but is not available in fuzzy searches here [Chen, Shen, Hu et al. (2016); Cheang, Wang, Cai et al. (2018)].

In order to solve this problem, this paper proposes a step-by-step scheme to reduce the difficulty of fuzzy matching with cloud-encrypted data. In the first step, the data owner constructs a fuzzy keyword set on the client side, and the set mainly includes three parts: keywords, Pinyin and fuzzy tones of keywords, English words corresponding to keywords, and corresponding index information (Chinese keyword and document ID table, Chinese and English keyword comparison table and Pinyin and fuzzy tones comparison table). In the second step, based on the fuzzy keyword set, a safe and efficient fuzzy search method is designed, which will be elaborated in the following chapters.

For data outsourced to the cloud, in addition to security issues, the user is most concerned with the efficiency of the operation. Therefore, this paper uses symmetric encryption as a searchable encryption framework.

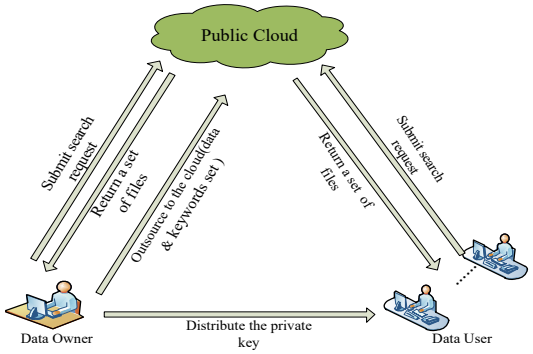

Figure 1: Multi-keyword Chinese fuzzy search strategy system framework

4 Implementation of project

In the design of Chinese fuzzy keyword search scheme framework, this paper first considers the establishment of fuzzy keyword set, then analyzes how to generate search request, and finally how to implement secure and efficient encrypted data search.

4.1 A fuzzy keyword set establishment

Establishing a set of keywords is a prerequisite for efficient fuzzy search. $SP_{w,d}$ and SY_w are generated by keyword w and similarity constraint d, where $w' \in SP_{w,d}, ed(PY_w, PY_{w'}) \leq d; w' \in SY_w, syn(w) = syn(w')$. The specific implementation is as follows:

(1) Fuzzy pinyin keyword set establishment

The pinyin of Chinese characters is mainly composed of initials and finals, and the combination of initials and finals conforms to specific laws. The following is a list of specific collections:

Initial set: {b, p, m, f, d, t, n, l, g, k, h, j, q, x, zh, ch, sh, r, z, c, s, y, w}

Single final set: {a, o, e, i, u, ü}

Complex final set: {ai, ei, ui, ao, ou, iu, ie, üe, er}

Front nose vowel: {an, en, in, un, ün}

Post nasal vowel: {ang, eng, ing, ong}

Chinese Pinyin does not have any combination like English words. For example, when the initial is b, the set of finals can only contain a fixed number: {a, o, i, u, ai, ei, ao, ie, an, en, in, ang, eng, ing, ian, iao}. Moreover, the most common mistakes in Chinese Pinyin spelling is the fuzzy sound, such as the flat tongue and the squeaky tongue, l and n in the initial, the front and back nasal, ian and iang, and uan and uang in the final. Therefore, the easiest way to create a fuzzy sound keyword set is to enumerate the possible pinyin combinations and find the set of keywords that are the same as these combinations. Examples are as follows:

Assuming that the user gives $d = 2$ and the pinyin of the input keyword w is lin, the corresponding fuzzy key keyword combination $SP_{w,d} = \{w_1', w_2' \ldots\ldots\}$ is generated according to the pinyin composition rule, and the pinyin of w_i' should be included in the set {lin, nin, ling, ning, *in}.

(2) Synonym set establishment

There are many words in Chinese that have the same or similar meanings, but these are not reflected in the synonym dictionary, e.g., "计算机", "电脑" and "微机". These three Chinese words have the same meaning. When the user executes the cloud data search, the file IDs related to these three words should be returned to the user, but there is no good way to achieve a similar synonym comparison. By comparing the similarity between Chinese and English in describing the same thing, a method for realizing synonym conversion using language differences is proposed. For example, the English words of the above three words are "computer", so if the English translation of the keyword w_i is consistent, then the words are synonymous. The implementation process is as follows:

Assuming the keyword w input by the user, the function $syn(w)$ is executed, w is translated into the English word we, and then the Chinese keyword translated into w_e is searched in the Chinese-English comparison table and the keyword set SY_w is returned.

After generating the corresponding fuzzy sound and synonym set, the $\boldsymbol{SP_{w,d}}$ and $\boldsymbol{SY_w}$ are encrypted by the encryption function $\boldsymbol{Enc(key,\cdot)}$, and sent to cloud together with the encrypted file for saving.

4.2 Generate search request

After the user inputs the keywords $\{w, d\}$, the scheme carries out a fuzzy search and returns a corresponding set $\{FID_{w_i}\}$ of file IDs, where $w \in \{w, d\}$, $w_i \in SY_w$ or $w_i \in SP_{w,d}$. The generation process of the search request is similar to the generation of the keyword index,

that is, according to the input w and d, the fuzzy pinyin and synonym generation function is called to obtain the fuzzy pinyin keyword set $SP_{w,d}$ and the synonym set SY_w. A search trapdoor is generated by w, $SP_{w,d}$ and SY_w, which all is encrypted, and then submitted to the cloud server. Finally, the search request generation work is completed.

4.3 Fuzzy search scheme

In the cloud service system, in order to avoid the cloud to obtain sensitive information, part of the work needs to be performed on the client side, e.g., the establishment of search indexes and the generation of trapdoors. Executing the search in a large amount of data is a very resource-consuming work, which should be done by the cloud server. The execution flow of the keyword-based encrypted cloud data fuzzy search scheme is as follows:

Scheme preprocessing stage:

(1) The data owner randomly selects two numbers a and b as the private key sk and distributes the private key to the data user.

(2) Build index. Index $I_1 = \left\{f(a, w_i'), Enc\left(sk_{w_i'}, FID_{w_i}\right)\right\}$, $w_i' \in SP_{w,d}$, $I_2 = \left\{f(a, w_i'), Enc\left(sk_{w_i'}, FID_{w_i}\right)\right\}$, $w_i' \in SY_w$, where $1 \le i \ge p$, $sk_{w_i'} = g(b, w_i')$.

(3) Outsource the index tables I_1 and I_2 and the encrypted data files to the cloud server storage.

Search stage:

(1) The user inputs sk and $\{w, d\}$. The client system generates $SP_{w,d}$ and SY_w, and generates a trapdoor $T_{1w'} = \left(f(a, w'), g(b, w')\right)$, $w' \in SP_{w,d}$; $T_{2w'} = \left(f(a, w'), g(b, w')\right)$, $w' \in SY_w$. The trapdoor sets $T_{1w'}$ and $T_{2w'}$ are sent to the cloud.

(2) The cloud server will compare the trapdoor $T_{1w'} = f(a, w')$, $w' \in SP_{w,d}$; $T_{2w'} = f(a, w')$, $w' \in SY_w$ with the indexes I_1 and I_2 to achieve file *ID* set $\{FID_{w_i}\}$ respectively, where $w \in \{w, d\}$, $w_i \in SY_w$ or $w_i \in SP_{w,d}$. The cloud sever sends the final result to the client.

(3) The client uses the corresponding $g(b, w')$ to decrypt the file *ID*, and calls $Dec(key, \cdot)$ to decrypt the required file.

5 Security analysis

In the encrypted search scheme designed in this paper, when the user inputs the same search request, the cloud will always return the same search result. Although the cloud server does not see what the underlying plaintext is, it can still establish access patterns and search patterns in interaction with the user. So the scheme ensures that content other than access and search requests are not compromised. This section will prove that the fuzzy search scheme designed in this paper is in line with the non-adaptive semantic security requirements [Raghavendra, Girish and Geeta (2018)]. The non-adaptive attack model only considers adversaries (e.g., cloud servers), who cannot select trapdoor-based search requests and previous search results, because only users with authorized private keys can

generate search traps [Wang, Chen, Li et al. (2017); Shen, Wang, Li et al. (2018)]. Below we introduce some of the concepts to analyze the security of fuzzy search schemes.

History: The interaction between the user and the cloud server, consisting of a set of files C and a set of keywords searched by the user, expressed as $H_q = (C, w_1, w_2, \cdots, w_q)$.

View: According to the key K, the history H_q is given, and the cloud server can only see the encrypted history. The view $V_k(H_q)$ includes: an index I of the file set C, a trapdoor of the query keyword $T_{1w'}$ $w' \in SP_{w,d}$ and $T_{2w'}$ $w' \in SY_w$, and a set C of the encrypted file, denoted as $\{e_1, \cdots, e_n\}$.

Track: Given a history H_q and an encrypted file set C, $Tr(H_q)$ captures the precise information learned by the cloud server, including the size of the encrypted file $(|F_1|, \cdots, |F_n|)$, search result $\{FID_{w_i}\}$, where $w_i = w$, $w_i \in SY_w$ or $w_i \in SP_{w,d}$. Pattern for each search $\prod_q[i,j]$. $\prod_q[i,j]$ is a symmetric matrix and stores the intersection of two sets $\prod_{q_1}$ and $\prod_{q_2}$. $\prod_{q_1}$ and $\prod_{q_2}$ are the intersection of the fuzzy note and the intersection of the synonym record, respectively: $\prod_{q_1}=\{T_{w'}\}_{w' \in SP_{w_i,d}} \cap \{T_{w'}\}_{w' \in SP_{w_j,d}}$, $\prod_{q_2}=\{T_{w'}\}_{w' \in SY_{w_i}} \cap \{T_{w'}\}_{w' \in SY_{w_j}}$.

In general, the security strength of this solution is reflected in the fact that cloud servers cannot distinguish their views for two historical records with the same trajectory. In other words, the cloud server cannot extract more information content based on the information leaked in the query (i.e., the trajectory), so the solution is safe. The security conclusions of the fuzzy search scheme in this paper are explained in the following theorem. Since the fuzzy search scheme has been described in the above, the following conclusions are equally applicable to the instantiated fuzzy search.

Theorem: This fuzzy keyword search scheme satisfies non-adaptive semantic security.

Proof: To prove semantic security, we construct an emulator *S* with a trajectory $Tr(H_q)$, which can simulate a view V_q^*, which can fully simulate the view $V_k(H_q)$ of the cloud server for any $q \in N$, any H_q, and a randomly chosen K. The security parameters l of the pseudo-random function $f(key,.)$, $T = max\{|SP_{w_i,d}|, |SY_{w_i}|\}_{w_i \in W}$, $|W|$ and the size of FID_{w_i} are known to *S*.

For $q = 0$, *S* generates $V_q^* = \{e_1^*, e_2^*, \cdots, e_n^*, I^*\}$, and e_i^* is randomly selected from $\{0,1\}^{|F_i|}$ $(1 \le i \le n)$. Let $I^* = (T^*, C^*)$, where $T^*[i]$ and $C^*[i]$ represents the ith row records in T^* and C^* respectively, as follows:

T^*: For $1 \le i \le \tau|W|$, *S* selects a random $t_i^* \in \{0,1\}^l$ and makes $T^*[i] = t_i^*$.

*C**: For $1 \le i \le \tau|W|$, *S* selects a random $c_i^* \in \{0,1\}^{|FID_{w_i}|}$ and makes $C^*[i] = c_i^*$.

Due to the semantic security of symmetric encryption, it is impossible for an adversary to distinguish between e_i and e_i^* or $Enc(sk_{w_i'}, FID_{w_i})$ and c_i^*. Due to the pseudo-randomness of the trapdoor generation function, there is also no case where $f(x, w_i')$ and a random string t_i^* can be distinguished. Therefore, $V_k(H_0)$ and V_0^* are indistinguishable.

For $q \geq 1$, $V_q^* = \{e_1^*, e_2^*, \ldots, e_n^*, I^*, \{T_{1w'}\}_{w' \in SP_{w,d}}, \{T_{2w'}\}_{w' \in SY_w}\}$ is constructed by *S*, where e_i^* is randomly generated by $\{0,1\}^{|F_i|}$ $(1 \leq i \leq n)$. Let $I^* = (T^*, C^*)$, the return result $\{FID_{w_i}\}$ of the jth search request, where $w_i \in SY_{w_j}$ or $w_i \in SY_{w_{j,d}}$ is the set of record corresponding file IDs that match the index. We can rewrite it to $U_{k=1}^{\alpha j} FID_{w_{j,k}}$, each $w_{j,k}$ satisfies $w_{j,k} \in SY_{w_j}$ or $w_{j,k} \in SY_{w_{j,d}}$, and α_j represents the number of jth search matches. For the search input w^1, the first trapdoor set $\{T_{w'}^*\} = \{\{T_{1w'}\}_{w' \in SP_{w,d}}, \{T_{2w'}\}_{w' \in SY_w}\}$ is constructed. The emulator *S* performs the following operations:

(1) Select α_1 random strings $t_{1,1}^*, \ldots, t_{\alpha,1}^* \in \{0,1\}^l$ and set them to $T^*[i_{1,1}], \ldots, T^*[i_{1,\alpha 1}]$.

(2) Select α_1 random strings $\rho_{1,1}^*, \ldots, \rho_{\alpha,1}^* \in \{0,1\}^l$ and set them to $C^*[i_{1,k}] = \mathrm{Enc}(\rho_{1,k}^*, FID_{w_{i,k}})$, $1 \leq k \leq \alpha_1$

(3) Set other $\tau - \alpha_1$ trapdoors to random value pairs $(t_{1,k}^*, \rho_{1,k}^*) \in \{0,1\}^l \times \{0,1\}^l$, where $\alpha 1 \leq k \leq \tau$.

For trapdoor simulation $w^j (2 \leq j \leq q)$ $i < j$, if $|\prod_q [i,j]| = 0$, trapdoor simulation will repeat the same operation as trapdoor w^1. Otherwise, the file number of the file *ID* list $\{FID_{w_{j,k}}\}_{1 \leq k \leq \alpha_j}$ is represented by β_1, and these file IDs have been successfully matched in the list $\{FID_{w_{i,k}}\}_{1 \leq k \leq \alpha_i, i<j}$. Next, the execution process of *S:*

(1) Select β_j trapdoors from the existing trapdoor set $\{T_{w'}^*\}_{w' \in \{SP_{w^i,d}, SY_{w^i}\}, i<j}$, corresponding to β_j file ID lists $\{FID_{w_{j,k}}\}_{1 \leq k \leq \alpha_j} \cap (\bigcup_{i=1}^{j-1} \{FID_{w_{j,k}}\}_{1 \leq k \leq \alpha_i})$, and then assign them to trapdoor simulation w^j.

(2) If $\alpha_j > \beta_j$, S will create $\alpha_j - \beta_j$ records in the index I^* using the same process as the simulation trapdoor.

(3) *S* further checks $\prod_q [i,j]$, where $i < j$, then finds γ_j trapdoors from the already existing trapdoor set $\{T_{w'}^*\}_{w' \in \{SP_{w^i,d}, SY_{w^i}\}, i<j}$ (no matches in index I^*) and assigns them to the current trapdoor simulation w^j.

(4) Value pairs are randomly selected from $\{0,1\}^l \times \{0,1\}^l$ to assign values to the remaining $\tau - \alpha_j - \gamma_j$ trapdoors.

The correctness of the constructed view is easily demonstrated by searching for trapdoors built into the index. There is no case in this scenario where an attacker can distinguish between $V_k(H_q)$ and V_q^*. Moreover, the simulated ciphertext uses a symmetric encryption scheme, and its semantic security determines that the ciphertext is indistinguishable. Indexes and trapdoors are also indistinguishable based on the nature of pseudo-random functions. Therefore, the proof theorem is correct.

6 Experimental analysis

6.1 Time and space consumption

The fuzzy search scheme of Chinese keywords proposed in this paper introduces synonym and fuzzy phonetic words of Chinese keywords based on the existing search schemes. Therefore, the proposed scheme needs extra time and space to process keyword synonyms and fuzzy sounds in the scheme preprocessing stage and search stage. However, it is still at the same level of time complexity and space consumption as the original solution. The time complexity of the pre-processing stage index construction is only related to the number of keywords, i.e., $O(\tau|W|)$, the size of the index is also only related to the number of keywords, i.e., $O(\tau|W|)$. During the search, due to the support of multi-threading by the cloud server system, the retrieval of synonyms and fuzzy words can be realized at the same time, so it does not increase too much time consumption, and the time complexity of the search is $O(\tau|W|)$.

6.2 Experimental comparison

This paper uses the free cloud platform provided by Amazon as the experimental platform and uses the journal and magazine literature as the search object to conduct experiments and verification objects to verify the effectiveness and practicability of the project. When building an index, as the number of keywords increases, the CPU and memory usage gradually increases, and the time consumed increases accordingly.

The time consumption of building an index is shown in Fig. 2.

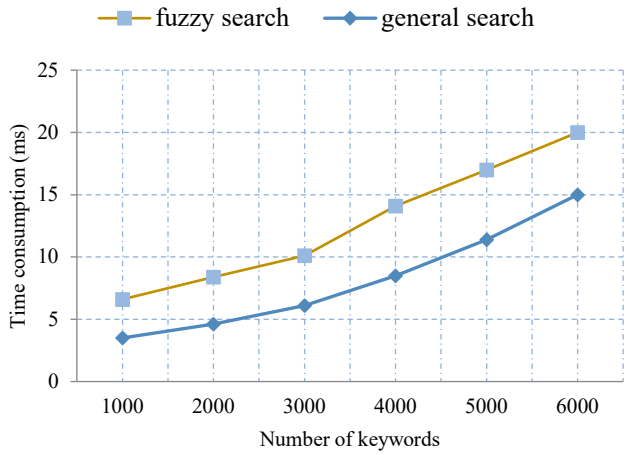

Figure 2: The time consumption of index building

The time consumption of keyword query is shown in Fig. 3.

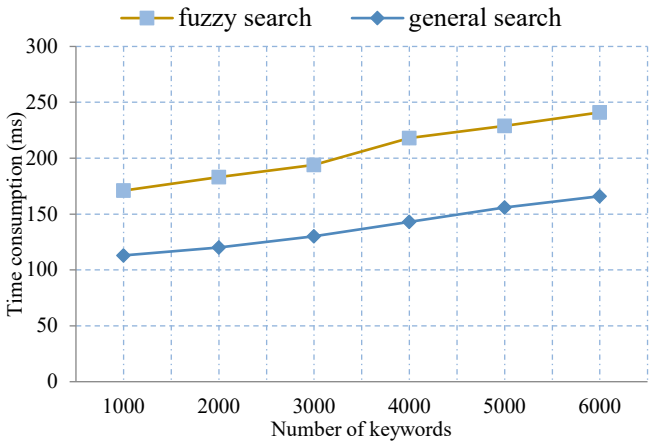

Figure 3: The time consumption of keyword query

The success rate of keyword query is shown in Fig. 4.

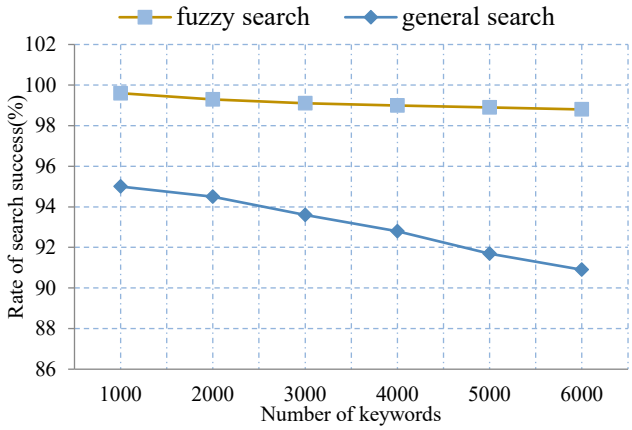

Figure 4: The comparison results of the success rate of keyword query

It can be seen from the above experimental results that the proposed scheme realizes the storage of synonyms and fuzzy words of keywords by appropriately increasing storage. Although the scheme preprocessing and search time are increased, the success rate of the search is greatly improved.

7 Conclusion

According to the actual needs of users' Chinese search in a cloud storage environment, this paper presents a fuzzy search strategy based on multi-keywords for encrypted cloud data. By constructing fuzzy sounds and synonym sets in the scheme, the fuzzy sounds and synonymous problems between the input text and the words that the user is looking for are well solved in the Chinese environment, and the pseudo-random function is used

to effectively avoid the problem of information disclosed in the query process. Therefore, the scheme has high security, good practicability, and high search success rate.

Acknowledgement: This work is supported in part by the Natural Science Foundation of China under Grants (Nos. 61772281, U1636219, 61702235, 61502241, 61272421, 61232016, 61402235 and 61572258), in part by the National Key R&D Program of China(Grant Nos. 2016YFB0801303 and 2016QY01W0105), in part by the Major Program of the National Social Science Fund of China under Grant (Nos. 17ZDA092), in part by the Electronic Information and Control of Fujian University Engineering Research Center Fund under Grant (No. EIC1704), in part by the plan for Scientific Talent of Henan Province Grant (No. 2018JR0018), in part by the Natural Science Foundation of Jiangsu Province, China under Grant (No. BK20141006), and in part by the Natural Science Foundation of the Universities in Jiangsu Province under Grant (No. 14KJB520024), the PAPD fund and the CICAEET fund. This work is implemented at the 2011 Collaborative Innovation Center for Development and Utilization of Finance and Economics Big Data Property, Universities of Hunan Province, Open project, grant number 20181901CRP04.

References

Boneh, D.; Waters, B. (2007): Conjunctive, subset, and range queries on encrypted data. *Proceedings of TCC'07*, vol. 4392, pp. 535-554.

Cao, N.; Wang, C.; Li, M.; Ren, K.; Lou, W. J. (2014): Privacy-preserving multi-keyword ranked search over encrypted cloud data. *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 222-233.

Cheang, C. F.; Wang, Y.; Cai, Z.; Xu, G. (2018): Multi-VMs intrusion detection for cloud security using dempster-shafer theory. *Computers, Materials & Continua*, vol. 57, no. 2, pp. 297-306.

Chen, C.; Shen, P.; Hu, J.; Guo, S.; Tari, Z. et al. (2016): An efficient privacy-preserving ranked keyword search method. *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 4, pp. 951-963.

Fu, Z.; Shu, J.; Sun, X. (2014): Smart cloud search services: verifiable keyword-based semantic search over encrypted cloud data. *IEEE Transactions on Consumer Electronics*, vol. 60, no. 4, pp. 762-770.

Hore, B.; Mehrotra, S.; Canim, M.; Kantarcioglu, M. (2012): Secure multidimensional range queries over outsourced data. *The VLDB Journal*, vol. 21, no. 3, pp. 333-358.

Li, C.; Zhang, X. (2013): Research on benchmark-based HPC in cloud. *Computer Science*, vol. 40, no. 6, pp. 23-30.

Li, F.; Wu, C.; Yuan, X. (2016): Multi-keyword ranked fuzzy search over encrypted data in cloud supporting dynamic update. *Journal of Computational and Theoretical Nanoscience*, vol. 13, no. 12, pp. 9705-9709.

Lucas, B.; Seny, K.; Fabian, M. (2005): Achieving efficient conjunctive keyword searches over encrypted data. *Proceedings of the International Conference on*

Information and Communications Security, vol. 3783, pp. 414-426.

Patnaik, L. M. (2016): DRSMS: domain and range specific multi-keyword search over encrypted cloud data. *International Journal of Computer Science and Information Security*, vol. 14, no. 5, pp. 69-78.

Raghavendra, S.; Girish, S.; Geeta, C. M. (2018): Split keyword fuzzy and synonym search over encrypted cloud data. *Multimedia Tools and Applications*, vol. 77, no. 8, pp. 10135-10156.

Shen, J.; Wang, C.; Li, T.; Chen, X.; Huang, X. et al. (2018): Secure data uploading scheme for a smart home system. *Information Sciences*, vol. 453, no. 7, pp. 186-197.

Song, D.; Wagner, D.; Perrig, A. (2000): Practical techniques for searching on encrypted data. *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, pp. 44-55.

Wang, C.; Cao, N.; Li, J. (2010): Secure ranked keyword search over encrypted cloud data. *Proceedings of the IEEE International Conference on Distributed Computing Systems*, pp. 253-262.

Wang, J.; Chen, X.; Li, J.; Zhao, J.; Shen, J. (2017): Towards achieving flexible and verifiable search for outsourced database in cloud computing. *Future Generation Computer Systems*, vol. 67, no. 2, pp. 266-275.