Chinese-keyword Fuzzy Search and Extraction over Encrypted Patent Documents

Wei Ding[1], Yongji Liu[1] and Jianfeng Zhang[2]
[1]*China Defense Science and Technology Information Center, 100036, Beijing, China*
[2]*National University of Defense Technology, 410073, Changsha, China*

Keywords: Chinese Keywords, Fuzzy Search, Extraction, Encrypted Documents.

Abstract: Cloud storage for information sharing is likely indispensable to the future national defence library in China e.g., for searching national defence patent documents, while security risks need to be maximally avoided using data encryption. Patent keywords are the high-level summary of the patent document, and it is significant in practice to efficiently extract and search the key words in the patent documents. Due to the particularity of Chinese keywords, most existing algorithms in English language environment become ineffective in Chinese scenarios. For extracting the keywords from patent documents, the manual keyword extraction is inappropriate when the amount of files is large. An improved method based on the term frequency–inverse document frequency (TF-IDF) is proposed to auto-extract the keywords in the patent literature. The extracted keyword sets also help to accelerate the keyword search by linking finite keywords with a large amount of documents. Fuzzy keyword search is introduced to further increase the search efficiency in the cloud computing scenarios compared to exact keyword search methods. Based on the Chinese Pinyin similarity, a Pinyin-Gram-based algorithm is proposed for fuzzy search in encrypted Chinese environment, and a keyword trapdoor search index structure based on the n-ary tree is designed. Both the search efficiency and accuracy of the proposed scheme are verified through computer experiments.

1 INTRODUCTION

Along with the development strategy of civil-military integration in China, the sharing and remote acquisition of Chinese defence information becomes increasingly desirable. Cloud storage (Weiss, 2007) is likely inevitable to the future national defence library in China, but the sensitive data have to be encrypted before outsourcing. Data encryption makes effective data utilization more challenging, since there could be a large amount of outsourced data files when data owners in cloud computing can share their outsourced data with many users (Li, 2010). When one user might only want certain specific data files, keyword-based search technique is suitable for selectively retrieving files of interest. Exact keyword search greatly enhances system usability by returning the matching files when users' searching inputs exactly match the predefined keywords or the closest possible matching files, when exact match fails. Various keyword search protocols for encrypted data were proposed (Liu, 2009; Liu, 2012; Boneh, 2004; Chang, 2005), and (Song, 2000) designed a double-encryption structure based on a searchable encryption model used in (Goh, 2003) to guarantee the data privacy. The aforementioned exact-keyword search method enjoys higher security but its fatal drawback of little tolerance about keyword typos and format inconsistencies makes the search experience very frustrating (Chor, 1995). For retrieving the data in a privacy preserving manner but more efficiently, the fuzzy keyword search is introduced, e.g., in (Ji, 2009), where fuzzy sets based on wild card technique and gram based technique are constructed and a symbol-based trie-traverse search scheme is adopted. However, these methods are adapted to English language, and thus turned to be invalid for Chinese keywords because of their diverse semantic analysis philosophy. In order to utilizing fuzzy search engine for Chinese keywords, a pre-process procedure on the initial Chinese characters is necessary (Cao, 2009). Keyword extraction is also helpful to accelerate the keyword search progress since it links finite keywords with a far more number of files. The drawbacks of the traditional method of

Ding, W., Liu, Y. and Zhang, J..
Chinese-keyword Fuzzy Search and Extraction over Encrypted Patent Documents.
In *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015) - Volume 1: KDIR*, pages 168-176
ISBN: 978-989-758-158-8

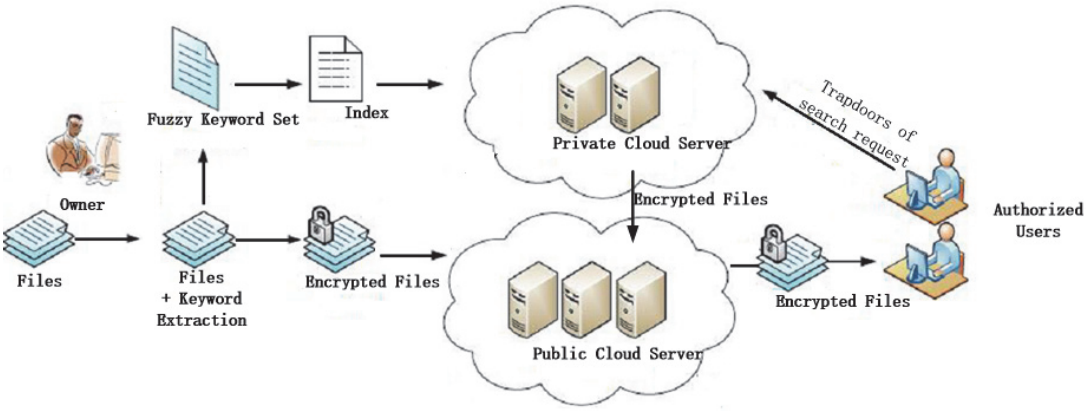

Figure 1: Adopted architecture of the fuzzy keyword search.

manual keyword extraction are obvious, e.g., including the inconformity of keyword combination and low efficacy, especially in the scenarios of cloud computing. Many research can be found for the automatic keyword extraction, e.g., (Witten, 2009) adopts statistical methods based on English dictionary to build a KEA system for automatic keyword extraction, and (Yang, 2002) uses PAT-tree structure to auto-collect keywords, while an improved scheme based on the co-occurrence frequency of Chinese phases is reported in (Du, 2011). However, few studies are done particularly for automatic keyword extraction over Chinese patent documents.

In this paper, for increasing the accuracy of keyword extraction, we overall consider the influence of the word frequency in the special regions, the penalty function of parallel structure and the weighted lexical morphemes upon the subjects of Chinese patent literature. After removing the common words, Chinese keywords are automatically extracted based on an improved method of the term frequency-inverse document frequency (TF-IDF) algorithm. To efficiently search Chinese keywords, a Pinyin-Gram-based algorithm is proposed to build the fuzzy keyword set, since Chinese Pinyin offers a unique method to study the Chinese word similarity, which is substantially different from English. Encrypted Files and keyword sets are transferred to the private cloud server. From the side of authorized users, a keyword trapdoor search index structure based on the n-ary tree is designed, and the searched encrypted files are outsourced by the public server, which usually has much more memories than the private server. The efficiency of the proposed scheme is verified through computer experiments, which is significantly higher than the traditional methods.

2 SYSTEM DESIGN AND TASK

2.1 System Description

In this paper, the adopted system architecture is consist of four components, i.e., the owner, the private cloud server, the public cloud server and the authorized users as indicated in Fig. 1. The difference compared to general system architecture, e.g., in (Li, 2010), lies in that a private cloud server is introduced. The advantages of such arrangement is to doubly enhance the security of sensitive files, since information leakage may happen through the index analysis if all the data are stored together in the public server.

The flow of the fuzzy keyword search is depicted as the follows. The keywords are extracted automatically from the patent files, and then the fuzzy keyword sets and search index are constructed. Patent files are encrypted and transferred to the private server by the owner. These encrypted files are uploaded to the public cloud server with necessary remarks or extra encryption. The authorized users deliver the search request and the responding trapdoor functions are processed at the private server. The file indexes and the found encrypted files are outsourced to the user. Besides these features, the encryption of certain patent literature, e.g., the national defense patents, are desired for cloud computing. As implied by (Li, 2010), the cloud server cannot be fully trusted. On one hand, it does not delete the encrypted files and the index, and only response to the query requests from authorized authors with unchanged search results. On the other hand, it may analyze the data stored in the server for certain purposes and sell the analyzed results as additional information to opponents. Inspirited by (Shi, 2007), the following two models are considered to protect the data security.

1) Cipher-text Model. With this model, the cloud server only knows the encrypted files without the secret key. Hence, the data information would be safe and the query request is conceal.

2) Trapdoor Model. In this model, the cloud server is potential to perform additional operations on stored data, e.g., attaining static data and extra analysis information according to the utilization frequency of certain keyword strings or outsourcing files (Zerr, 2008). Using the trapdoors of keyword fuzzy set, exact keywords are hidden and thus it is unlike that the cloud server learns the specific information about keywords.

It is noteworthy that Chinese characters have a different formation philosophy compared with English language, and Chinese-Pinyin plays an essences role in constructing fuzzy set (Cao, 2009) which is adopted in this paper.

2.2 Design Target

Our purpose is to design an efficient and safety Chinese-keyword fuzzy search system based on cloud storage. More specifically, the design tasks include:

1) Automatic Chinese keyword extraction;

2) Constructing the keyword fuzzy set, which should be of small size and fast building period.

3) Keyword fuzzy search, which supports a rapid encrypted file searching using an efficient trapdoor index structure;

4) Data security, which prevent the leakage of sensitive files and keyword information;

5) System usability, which costs low level communication bandwidth and computation complexity to complete all the above tasks.

2.3 Notation Definition

Herein we define the notations used in the following sections. Let

1) $DW = \left\{DW_1, DW_2, \ldots, DW_K\right\}$ denotes the collect of the specialty domains, with K as the number of all concerned specialties.

2) $WD = \left\{wd_1, wd_2, \ldots, wd_{\aleph}\right\}$ denotes the collection of all the Chinese words or phrases in the dictionary of science and technology, where $\aleph$ is the total number of Chinese words;

3) $WS = \left\{ws_1, ws_2, \ldots, ws_{\Im}\right\}$ denotes the collection of all the initial keywords contained in outsourcing files, where $\Im$ is the total number of initial keywords;

4) $W = \left\{w_1, w_2, \ldots, w_M\right\}$ denotes the collections of all the pre-processed keywords w_l where *M* is their total number；

5) $F = \left\{F_1, F_2, \ldots, F_N\right\}$ denotes the collection of the outsourcing files with *N* as the file number;

6) $C = \left\{F_1^{'}, F_2^{'}, \ldots, F_N^{'}\right\}$ denotes the collection of the encrypted files of the outsourcing files with *N* as the file number;

7) $T_{w_l} = H(w_l, Sk)$ denotes the trapdoor of length τ according to the keyword w_l , where $H(\cdot)$ represents the keyed-Hass message authentication code, e.g., HMAC-MD5, and *Sk* is the shared key between the owner and the authorized users;

7) FID_l denotes the sole label of the encrypted file;

8) $S_{w_l,d} = \left\{w_l^{'} \mid ed(w_l, w_l^{'}) \leq d\right\}$ denotes the collection of the keywords whose edit-distance with the keyword w_l is no more than *d*, where $ed(w_l, w_l^{'})$ is the edit distance between the keywords w_l and $w_l^{'}$;

9) A_{w_l} =" $I_1 I_2 \cdots I_K$" denotes the adapted Pinyin syllable string according to the keyword $w_l \in \mathrm{W}$, where *K* is the number of Chinese characters in this keyword w_l and I_j represents the *j*-th syllable according the *j*-th Chinese character in the keyword;

10) $S_{A_{w_l},d} = \left\{A_{w_l^{'},d'} \mid d' \leq d\right\}$ denotes the collection of the adapted Pinyin syllable strings according to the keyword collection $S_{w_l,d}$, with $A_{w_l,d}$ = $\left\{A_{w_l^{'}} \mid ed(A_{w_l^{'}}, A_{w_l}) = d\right\}$.

$\overline{A}_{w_l}$ =" $\overline{I}_1 \overline{I}_2 \cdots \overline{I}_K$" denotes the fuzzy counterpart of A_{w_l} =" $I_1 I_2 \cdots I_K$" , where $\overline{I}_j$ may lack no more than a final compound, an initial consonant or a

Pinyin tone compared with I_j in this paper, and $\bar{S}_{A_{w_l},d} = \left\{\bar{A}_{w_l,d'} \mid d' \le d\right\}$ collects all the fuzzy strings whose edit-distance with the keyword A_{w_l} is no more than *d*, with $\bar{A}_{w_l,d} = \left\{\bar{A}_{w_l} \mid ed(\bar{A}_{w_l}, A_{w_l}) = d\right\}$.

It is noteworthy that each syllable I_j contains three parts, i.e., the Pinyin spelling, the Pinyin tone and an underline symbol “_”, where the Pinyin tone ranges from 1 to 4, and the Pinyin spelling contains at most two parts, which are the final compound and the initial consonant picked from a pre-fixed Pinyin spelling combination table while a small number of Pinyin syllables only contain the final compound in the spelling. For example, I_j =“zhong1_” according to the Chinese character “中” with “zhong” being its Pinyin spelling and “1” being its Pinyin tone as well as an underline symbol “_” for separating adjacent syllables, where “zh” is the initial consonant and “ong” is the final compound.

3 KEYWORD EXTRACTION

3.1 Common Word Removal

The common words are defined in this paper as those words which have a high word frequency in most patent documents. These words are known to strongly affect the keyword extraction based on the TF-IDF algorithm (Yang, 2002). We collect the common words according to the following criterion

$$f_{cw}(wd_i, DW) = \begin{cases} 1, & \min\left(\dfrac{f_{TF}(wd_i, DW_p)}{N_p}, \dfrac{f_{TF}(wd_i, DW_k)}{N_k}\right) > \delta \\ 0, & \min\left(\dfrac{f_{TF}(wd_i, DW_p)}{N_p}, \dfrac{f_{TF}(wd_i, DW_k)}{N_k}\right) \le \delta \end{cases} \quad (1)$$

which indicates that the word wd_i shall be seen as a common word if $f_{cw}(wd_i, DW) = 1$ when its occurrence frequency in any two specialty domains (e.g., $DW_p \in DW$ and $DW_k \in DW$) is more than the threshold number δ, with N_p and N_k being the patent document number in these two corresponding domains.

$$f_{TF}(wd_i, DW_p) = \sum_k f_{TF-IDF}(wd_i, F_k, DW_p)$$

summarizing the TF-IDF value of the word wd_i in all the files (i.e., F_k) which belongs to the domain DW_p, while $f_{TF-IDF}(wd_i, F_k, DW_p)$ returns the TF value of wd_i in the file F_k.

3.2 Keyword Weighted Function

Traditional methods based on TF-IDF usually take the influence of the word frequency in the special regions of the literature (e.g., the title) into account in order to accelerate the extraction speed (Witten, 1999). Different from other literature, patent literature also characteristics of particular features, which can help a fast and accurate keyword auto-extraction. The following two features are considered in this paper for Chinese patent literature:

1) Combined phrases or words probably contains the keyword. In Chinese literature, there are many combined phrases which usually share the same lexical morphemes, and it is likely that keywords lie in these phrases. For instance, two Chinese phrases $wd_1 =$ “战术电台” and $wd_2 =$ “背负电台” share the same lexical morpheme $wd_3 =$ “电台”，which is likely one of the keywords in that document.
2) Words in the parallel structure are not the keywords. The words or phrases in the parallel structure in Chinese literature, which are normally remarked by Chinese conjunction words (e.g., “和”, “与” and “或”) and punctuation mark (i.e., ”、”), usually have a high word frequency, but do not be considered as the keywords.

The weighted function used for extracting the keywords in a document is defined as follows

$$f_{\text{wei}}(wd_i) = \lambda_1 W_{\text{reg}}(wd_i) + \lambda_2 W_{\text{mor}}(wd_i) - \lambda_3 f_{\text{pen}}(wd_i) \quad (2)$$

where $W_{\text{reg}}(wd_i)$ reflects the weight of the word occurrence frequency given by

$$W_{\text{reg}}(wd_i) = f_{TF}(wd_i) + f_{\text{title}}(wd_i) + f_{\text{other}}(wd_i) \quad (3)$$

with $f_{TF}(wd_i)$ denoting the word frequency of wd_i in this document while $f_{\text{title}}(wd_i)$ and

$f_{\text{other}}(wd_i)$ denoting the weights if the word lies in the title and other special regions respectively;

$W_{\text{mor}}(wd_i)$ on behalf of the weight of the lexical morphemes described in details by (Du, 2011), which responds to the first aforementioned feature of the patent literature;

$f_{\text{pen}}(wd_i)$ representatives of the penalty function when the word lies in a parallel structure as described in the second aforementioned feature of the patent literature. More specifically,

$$f_{\text{pen}}(wd_i) = \alpha_1 f_{\text{para}}(wd_i) + \alpha_2 \sum_{i \neq j} f_{\text{mor}}(wd_i, wd_j) \quad (4)$$

given that $f_{\text{para}}(wd_i)$ counts the total number of the word wd_i appearing in the parallel structure and $f_{\text{mor}}(wd_i, wd_j)$ counts the number of the same lexical morphemes shared by wd_i with other words (i.e., wd_j 's) in the parallel structure, while α_1 and α_2 are the responding weight parameters. For instance, two Chinese phrases $wd_1 =$ “战术电台” and $wd_2 =$ “背负电台” show in a parallel structure, and then $f_{\text{mor}}(wd_1, wd_2) = 1$ by knowing the they share one lexical morpheme “电台”.

Plus, λ_1 , λ_2 and λ_3 represents the weight of the word appearance frequency, the weight of the lexical morphemes and the penalty of the parallel structure, respectively.

To this end, we define the keyword as

$$ws_l = \{wd_i \mid f_{\text{wei}}(wd_i) > \Gamma, f_{cw}(wd_i, DW) = 0\} \, (5)$$

which means that the extracted keyword ws_l is the word whose weighted function value is higher than the threshold Γ as well as outside the common word collection.

4 KEYWORD FUZZY SEARCH

The extracted keywords form a pre-defined keyword set by the owner, which offers to the keyword search engine as the reference. However, in order to facilitate the search experience of the user, the fuzzy search scheme over the encrypted data is introduced.

4.1 Pinyin-Gram-based Algorithm

In the informatics and computer science, the edit distance between two strings are counted by the number of the letter replacement to convert one string into the other. The Gram of a string is defined as the core sub-string utilized for an efficient fuzzy search, and a Gram-based keyword fuzzy set was introduced to build fuzzy search schemes over encrypted data. One of its basic hypotheses is that all original operations by users only make changes on a single word within the keyword, and do not disorganize the positions of all the letters within such keyword. Such condition is usually true in reality and thus the Gram-based algorithm to construct keyword fuzzy sets is widely adopted, e.g., in (Li, 2010). However, Chinese keywords are no like English keywords whose adjacent words are separated by a space symbol, while Chinese characters are connected with each other. Hence a pre-process operation is first needed to isolate all the characters, where Pinyin syllable strings are introduced (Cao, 2009). In additions, the traditional method used in English scenarios treats one replacement of a letter as the edit distance of 1, and such idea shall results in a hug fuzzy set not only because Chinese Pinyin contains both spelling and tone, but also since the same Pinyin syllable string can represent various characters. It is also vain to consider Pinyin string as a generalized English letter strings since deleting, inserting or replacing a letter of Pinyin spelling may introduce an illegal Pinyin string. In (Wang, 2007; Bellare, 1997), Pinyin strings are used for matching approximate Chinese strings in plaintext scenarios. For a Chinese Pinyin syllable, the difference between other syllables can be classified into 3 aspects: the initial consonant, the final compound and the tone. Based on this fact, we define the edit distance used in this paper as follows:

Definition 1: Edit Distance for Pinyin Syllable

1) When the tone varies, the edit distance increases by 0.5;

2) When either the initial or the final varies but the changed notes belong to the prefixed collection of Pinyin similar pairs, the edit distance increases by 0.5;

3) When either the initial or the final varies but the changed notes do not belong to the prefixed

collection of Pinyin similar pairs, the edit distance increases by 1;

4) When both the initial and the final vary, the edit distance increases by 4.

In this definition, note that the prefixed collection of Pinyin similar pairs have finite options (Cao, 2009), e.g., ’z’ and ‘zh’ form a pair of similar initial consonants while ‘en’ and ‘eng’ form a pair of similar final compounds.

Definition 2: Edit Distance between Pinyin Syllable and Its Fuzzy String

1) When an initial or a final is deleted, the edit distance increases by 1;

2) When both the initial and the final vary, the edit distance increases by 4;

3) When the tone varies, the edit distance increases by 0.5.

Based on the above definitions, we introduce the Pinyin-Gram-based algorithm for constructing Chinese keyword fuzzy set as Algorithm 1:

Algorithm 1: Constructing Chinese-keyword Fuzzy Set

1) Build the Pinyin syllable table and the collection of Pinyin similar pairs;

2) Let $\overset{\leftrightarrow}{A}_{w_l} = \left\{ A_{w_l^{'},0} \mid S_{A_{w_l^{'}},0} \text{ and } \mathrm{ASCII}_{w_l^{'}} = \mathrm{ASCII}_{w_l} \right\}$ distinguish two different keywords w_l and $w_l^{'}$ when $A_{w_l^{'}} = A_{w_l}$, e.g., w_l =“意义” and $w_l^{'}$ =“异议”, where ASCII_{w_l} denotes the ASCII code of w_l.

3) Traverse $S_{A_{w_l},d}$ and $\overline{S}_{A_{w_l},d}$ recursively from $S_{A_{w_l},d-0.5}$ and $\overline{S}_{A_{w_l},d-0.5}$ by varying and deleting only one parameter Ψ, respectively, with the initial condition of $S_{A_{w_l},0} = \overline{S}_{A_{w_l},0} = \left\{ \overset{\leftrightarrow}{A}_{w_l} \right\}$ and with Ψ representing an initial consonant, a final compound or the tone symbol;

4) Let $\mathrm{Fuzzyword}_{A_{w_l},d} = S_{A_{w_l},d} \cup \overline{S}_{A_{w_l},d}$, where $\mathrm{Fuzzyword}_{A_{w_l},d}$ collects all the fuzzy set of the keyword w_l with an edit distance threshold of *d*.

Based on Algorithm 1, an example would be that according to the keyword w_l =“法制”, the fuzzy set with the edit distance threshold of 1 shall be $\mathrm{Fuzzyword}_{A_{w_l},1} = \overset{\leftrightarrow}{A}_{w_l} \cup \left(A_{w_l,0.5} \cup \overline{A}_{w_l,0.5} \right) \cup \left(A_{w_l,1} \cup \overline{A}_{w_l,1} \right)$= {fa2_zhi4_$|_{\mathrm{ASCII}_{w_l}}$ } ∪ {fa_zhi4_, fa2_zhi_, fa2_zi4_} ∪ {a2_zhi4_, f2_zhi4_, fa_zhi_, fa2_i4_, fa2_zh4_, fa_zi4_, fa2_zi_}, by noticing that ’z’ and ‘zh’ belong to the collection of Pinyin similar pairs while w_l =“法制” varies from $w_l^{'}$ =“法治” although they share the same pronunciations.

4.2 Multi-node Search Tree Structure

The balanced binary search tree is widely adopted to build keyword index and several improvements were proposed, e.g., in (Li, 2010) by using symbol-based tire-traverse search scheme and in (Li, 2012) by using binary sort search tree. It is known that HMAC is featured to be unidirectional and collision resistant, and its output string has the same length. We divide $T_{w_l} = H(w_l, Sk)$ into π/λ segments (Li, 2010), with π being the length of this trapdoor and λ being the length of each segment. For example, for $A = \left\{ a_i \right\}$ as a full permutation sequence of length λ in the binary sense, where $a_i = a_{i-1} + 1$ with $a_0 = 0$, hence each divided segment can be represented by a parameter a_i. According to (Li, 2010), we introduce the adopted keyword trapdoor search index structure as Algorithm 2:

Algorithm 2: Balanced Multi-node Search Tree

1）The data owner outsources the encrypted files and the trapdoors $\left(FID_{w_l} \parallel T_{w_l} \right)$ of all keywords in the fuzzy set to the private cloud server;

2）Let Gw be the index root node at the private cloud server and each node can have 2^{λ} sub nodes with Node_i as the *i*-th sub node as indicated in Fig. 2, performing the following operations：

3）For a segment of T_{w_l} with $T_{w_l,j}$ as the value of the j-th segment at a node denoted as $\mathrm{Node}_{T_{w_l,j}}$, with the initial father node as Gw.

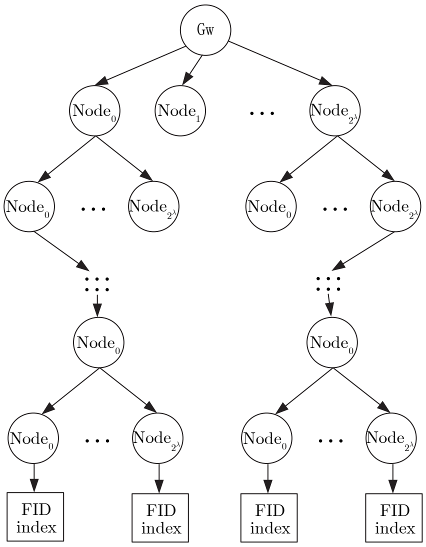

Figure 2: Adopted balanced multi-node search tree.

4）Process $T_{w_l,j}$ from j=1 to j= 2^λ , and judge whether the last segment of T_{w_l} has been treated or not; If yes, the index FID_{w_l} is added to the index table of the sub node (i.e., $\mathrm{Node}_{T_{w_l,j}}$) and terminate the search; If not, when the sub node $\mathrm{Node}_{T_{w_l,j}}$ is null, then fill this node with 2^λ empty sub nodes and consider $\mathrm{Node}_{T_{w_l,j}}$ as the new father, otherwise directly consider $\mathrm{Node}_{T_{w_l,j}}$ as the new father, iterating the step (1). Using Algorithm 2, the update operations like inserting or deleting can be performed too by similarly updating the file identification, i.e., FID_{w_l} in the index table, after finding the leaf node.

5 PERFORMANCE VALIDATION

We randomly choose 800 documents from the patent reports stored in China Defence Science and Technology Information Centre from their titles. The test environment is 3.3GHz CPU, 4G DRR3-1333MHz RAM and 32bit Windows 7 operating system.

To remove the common words from these documents, we first let the threshold $\delta = 0.4$ and recall $f_{TF-IDF}()$ from (Yang, 2002) in (1). To extract keywords, we use $\Gamma = 0.7$ in (3), given that $\lambda_1 = \lambda_2 = \lambda_3 = 1/3$ in (2) where $W_{\mathrm{reg}}(wd_i)$ and $W_{\mathrm{mor}}(wd_i)$ are recalled from (Du, 2011) while $\alpha_1 = 5$ and $\alpha_2 = 2$ is used for $f_{\mathrm{pen}}(wd_i)$. We note that the optimal combination of these parameters is beyond the range of this paper. To depict the performance the proposed method of keyword extraction, we choose the F-measure defined as

$$F = \frac{2 \times P \times R}{P + R} \tag{6}$$

where the precision value P gives the percentage of the correctly extracted keywords versus all the extracted keywords, while the recall value R is the percentage of the correctly extracted keywords versus all the true keywords. Fig. 3 depicts the comparison of several keyword extraction methods, which indicates the proposed method in this paper is much more effective than the traditional TF-IDF method. Compared with (Du, 2011), similar performance is witnessed, however, when the number of the keywords is large as the case in cloud computing, our method shows performance superiority.

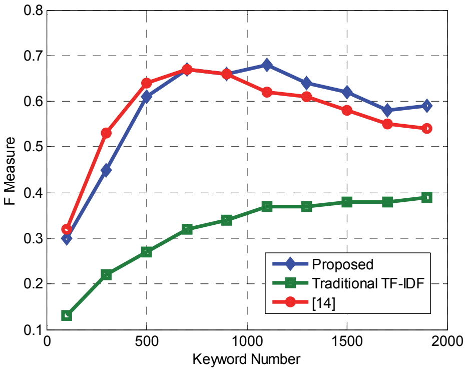

Figure 3: Automatic Keyword Extraction Performances.

From the extracted keywords, we choose 1800 keywords for testing our fuzzy search scheme. Fig. 4 shows the comparison of the proposed Pinyin-Gram-based algorithm and the wild-card-based method used in (Li, 2010) for constructing Chinese keyword fuzzy set, which indicates that the proposed scheme is more efficient. In additions, it also shows that the construction time increases linearly along the keyword number while a dramatic increase is witnessed when the edit distance threshold d of the fuzzy set is raised.

Fig. 5 shows the comparison of the proposed multi-node search tree and symbol-based trie-traverse search scheme used in (Li, 2010) for constructing Chinese keyword search tree, which indicates that the proposed scheme is faster. In additions, it also shows that the construction time increases linearly along the number of the keyword trapdoors.

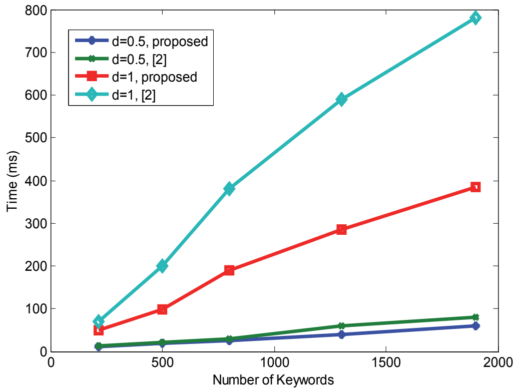

Figure 4: Consumed Time for constructing fuzzy set.

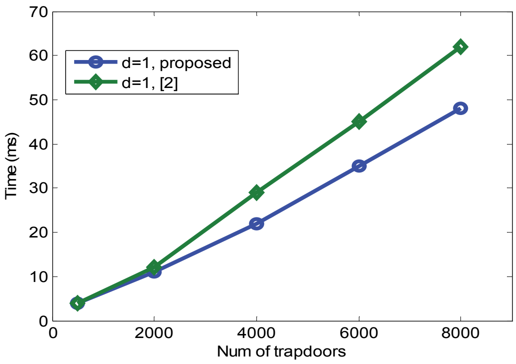

Figure 5: Consumed Time for constructing search tree.

Fig. 6 shows the comparison of the proposed scheme, the methods used respectively in (Li, 2010), (Li, 2012) and a basic method for searching Chinese keywords. The basic method means to use a standard wild-card-based method for constructing the fuzzy set as used in (Li, 2010) combined with the standard balanced binary search tree. It indicates that the proposed scheme is faster than all other methods.

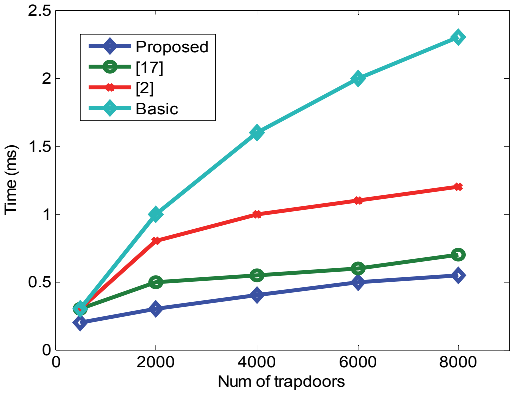

Figure 6: Consumed Time for searching keywords.

6 CONCLUSIONS

The Chinese Pinyin string construction is purposely tailored for Chinese keyword scenarios. The Pinyin-Gram-based algorithm and a balanced multi-node search tree are proposed for Chinese keyword fuzzy search over encrypted data. In the system design, a private cloud is introduced isolated from public cloud servers for higher security purpose. The complexity of our scheme is shown to increase linearly along the keyword number and the proposed search method shows a more efficient performance than existing methods.

ACKNOWLEDGEMENTS

We thank Dr. Tao Xu for his invaluable advice and suggestions, who is in part supported by China Postdoctoral Science Foundation (2015M570230) and Tianjin Enterprise-Postdoctoral Fund for Selected Innovation Program, at University-Enterprise joint postdoctoral station between Tsinghua University and Tianjin Zhonghuan Electronic & Information (Group) Co., Ltd.

REFERENCES

Weiss, A., 2007. Computing in the Clouds. *netWorker*.

Li, J., et al, 2010. Fuzzy keyword search over encrypted data in cloud computing, *Proceedings of IEEE*.

Liu, Q., et al, 2009. An Efficient Privacy Preserving Keyword Search Scheme in Cloud Computing, *IEEE Int. Sym. on Trusted Computing and Communications*.

Liu, Q., et al, 2012. Secure and Privacy Preserving Keyword Search for Cloud Storage, *Journal of Network and Computer Applications*.

Boneh, D., et al, 2004. Public Key Encryption with Keyword Search, *Int. Conf. on Theory and Applications of Crypto-graphic Technique*.

Chang, Y., et al, 2005. Privacy Preserving Keyword Searches on Remote Encrypted Data, *Applied Cryptography and Network Security*.

Song, D., et al, 2000. Practical techniques for searches on encrypted data, *IEEE Sym. on Security and Privacy*.

Goh, E.-J., 2003. Secure indexes, *Cryptology ePrint Archive Report*.

Chor, B., et al, 1995. Private information retrieval, *Annual Sym. on Foundations of Computer Science*.

Ji, S., et al, 2009. Efficient interactive fuzzy keyword search, *VLDB Journal*.

Cao, J., et al, 2009. A Pinyin indexed method for approximate matching in Chinese, *CMMSC'2009*.

Witten, I., et al, 1999. KEA: Practical Automatic Keyphrase Extraction, *ACM Confrence on Digital Libraries*.

Yang, W., et al, 2002. Chinese Keyword Extraction based on Max-duplicated Strings of the Document, *ACM SIGIR Conf. on Research and Development in Information Retrieval*.

Du, Y., et al, 2011. Automatic extraction of keyword based on word co-occurrence frequency, *Journal of Beijing Institute of Machinery*.

Shi, E., et al, 2007. Multidimensional range query for encrypted data, *IEEE Symposium on Security and Privacy*.

Zerr, S., et al, 2008. r-Confidential indexing for distributed documents, *IEEE Symposium on Security and Privacy*.

Wang, J., et al, 2007. An Approximate String Matching Algorithm for Chinese Information Retrieval Systems, *Journal of Chinese Information Processing*.

Bellare, M., et al, 1997. HMAC: Keyed-hashing for message authentication, *Internet Request for Comment RFC*.

Ma, X., 2005. Analysis of Chinese homonym, *Contents of Major Papers*.

Li, Q., et al, 2012. Efficient Multi-keyword research over Secure Cloud Storage, *Computer Science*.