Achieving Fine-grained Multi-keyword Ranked Search over Encrypted Cloud Data

Guowen Xu[1] and Hongwei Li[1,2]

[1] School of Computer Science and Engineering,
University of Electronic Science and Technology of China, China
[2] Science and Technology on Communication Security Laboratory, Chengdu, China

Abstract. With the advancement of Cloud computing, people now store their data on remote Cloud servers for larger computation and storage resources. However, users' data may contain sensitive information of users and should not be disclosed to the Cloud servers. If users encrypt their data and store the encrypted data in the servers, the search capability supported by the servers will be significantly reduced because the server has no access to the data content. In this paper, we propose a Fine-grained Multi-keyword Ranked Search (FMRS) scheme over encrypted Cloud data. Specifically, we leverage novel techniques to realize multi-keyword ranked search, which supports both mixed "AND", "OR" and "NO" operations of keywords and ranking according to the preference factor and relevance score. Through security analysis, we can prove that the data confidentiality, privacy protection of index and trapdoor, and the unlinkability of trapdoor can be achieved in our FMRS. Besides, Extensive experiments show that the FMRS possesses better performance than existing schemes in terms of functionality and efficiency.

Keywords: Cloud computing, Multi-keyword search, Privacy-preserving

1 Introduction

Cloud computing is an emerging data storage and computing service, which is available to the public users over the Internet. It significantly saves the local data storage and computing cost of data owners [1–5]. However, users' data may contain sensitive information of users and should not be disclosed to the cloud server. Simply encrypting the data may prevent the server from accessing it, but it also significantly reduces the search capability of the server. In addition, there are other security concerns, like some sensitive information leakage of users.

Recently, the searchable encryption [6–9] has been developed as a fundamental approach to enable searching over encrypted cloud data. Sun et al. [10] propose a multi-keyword text search scheme, which builds the search index based on term frequency and the vector space model with cosine similarity measure to achieve higher search result accuracy. To improve the search efficiency, Strizhov et al. [11] propose a tree-based Substring Position Searchable Symmetric Encryption (SSP-SSE) to handle substring search queries over encrypted data, which also involves identifying the position

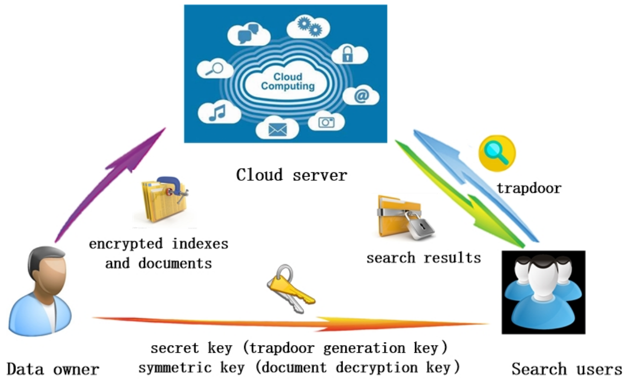


Fig. 1: System model

of the substring within the document. Li et al. [12] utilize the relevance score and k-nearest neighbor techniques to design an efficient multi-keyword search scheme, which supports the mixed"AND", "OR" and "NO" operations of keywords. However, these proposed schemes cannot achieve the ranking according to the preference factor and relevance score simultaneously.

In this paper, we propose a Fine-grained Multi-keyword Ranked Search(FMRS) scheme over encrypted cloud data. Specifically, our original contributions can be summarized as follows:

- Both mixed "AND", "OR" and "NO" operations of keywords and fine-grained ranking according to the preference factor and relevance score were supported in our proposed scheme.
- Data confidentiality, privacy protection of index and trapdoor, and the unlinkability of trapdoor can be achieved in our FMRS. Besides, Extensive experiments show that the FMRS possesses better performance than existing schemes in terms of functionality and efficiency.

The remainder of this paper is organized as follows. In Section 3, we describe the preliminaries of the proposed schemes. In Section 2, we outline the system model, threat model and security requirements. We present the developed scheme in Section 4. Then we carry out the security analysis and performance evaluation in Section 5 and Section 6, respectively. Finally, Section 7 concludes the paper.

2 SYSTEM MODEL, THREAT MODEL and SECURITY REQUIREMENTS

2.1 System Model

As shown in Fig. 1, we consider a system consists of three entities, i.e., *Data owner*, *Cloud server* and *Search user*. The tasks of dtaa owner is to encrypt raw data utilizing symmetric encryption such as AEE, and send them to the cloud server. Then, for each original data, corresponding a encrypted index will be created and be also sent to the cloud server for query expediently. The cloud server is an intermediate entity which stores the encrypted documents and corresponding indexes that are received from the data owner, and provides data access and search services to given search users, and the main task of search user is to query on cloud assisted with the cloud server.

2.2 Threat Model and Security Requirements

In our threat models, the cloud server is generally considered "honest-but-curious", which is the same as most related works on secure cloud data search [10,13–16]. Specifically, on one hand, the cloud server is honest and honestly follows the designated protocol specification and provides appropriate services. However, the cloud server is also curious, and could be "curious" to infer and analyze data (including index) stored in the cloud server so as to learn additional information. Based on this situation, we consider two threat models depending on the information available to the cloud server.

– **Known Ciphertext Model:** In this model, the cloud server only knows the encrypted document collection $\mathcal{C}$ and the corresponding index collection $\mathcal{I}$, both of which are outsourced from the data owner.
– **Known Background Model:** In this stronger model, the cloud server is supposed to possess more knowledge than what can be accessed in the known ciphertext model, such as the correlation relationship of trapdoors and the related statistics of other information, i.e., cloud server can obtain a large amount of statistical information through a known database which bears the similar nature to the targeting dataset.

Based on the above threat model, we define the security requirements as follows:

– *Confidentiality of documents*: The documents of the data owner are stored in the cloud server. Due to the privacy of data, the contents of documents should not be identifiable except by the data owner and the authorized search users.
– *Privacy protection of index and trapdoor*: Index and trapdoor are closely related to file information, the contents of index and trapdoor privacy should be ensured and cannot be identified by the cloud server.
– *Unlinkability of trapdoor*: Unlinkability of trapdoor, i.e the cloud server can not get any keyword information according to the trapdoors. For the same keywords, trapdoors should be generated randomly, rather than deterministic.

3 PRELIMINARIES

3.1 Secure kNN Computation

We leverage the work of Wong et al. [17]. Wong et al. propose a secure k-nearest neighbor (kNN) scheme which can encrypt two vectors and calculate their Euclidean distance secretly. Firstly, the secret key (S, M_1, M_2) needs to be generated by data owner. The role of binary vector S is to split plaintext vector into two random vectors, which can change the original value of plaintext vector. Then the M_1 and M_2 are used to encrypt the split vectors. Detailed introduction for kNN can refer to the literature [17].

3.2 Relevance Score

We adopt a widely used expression in [18] to evaluate the relevance score as

$$Score(\widetilde{\mathcal{W}}, F_j) = \sum_{w \in \widetilde{\mathcal{W}}} \frac{1}{|F_j|} \cdot (1 + \ln f_{j,w}) \cdot \ln(1 + \frac{N}{f_w}) \tag{1}$$

where $f_{j,w}$ represents the frequency of keyword w in document F_j; f_w represents the number of documents that contain the keyword w; N represents the number of files in the collection; and $|F_j|$ represents the length of F_j, which obtained by counting the number of indexed keywords.

3.3 Reference Factor

The preference factors are defined by search user for custom search, which can set the weight of keywords according to one's preferences. In this paper, we exploit super-increasing sequence $(a_1 > 0, a_2, \cdots, a_l)$ (i.e., $\sum_{i=1}^{j-1} a_i \cdot D < a_j (j = 2, 3, \cdots, l)$) to custom search for everyone, where a_i is the preference factor of keyword w_i.

4 PROPOSED SCHEME

In this section, we discuss our fine-grained multi-keyword ranked search scheme(FMRS) in detail.

Initialization A $(m+1)-$ dimensional binary vector$S$, two $(m+1)\times(m+1)$ invertible matrices M_1 and M_2 and symmetric key are generated by data owner. where sk is to encrypt documents outsourced to the cloud server.

Index building document collection $(F_1, F_2, \cdots F_N)$ will be encrypted formed as C_j $(j = 1, 2, \cdots N)$ by symmetric key sk, and sent to the cloud server firstly. Then, for each document C_i,a m-dimensional binary vector P_i are generated asP_j $(j = 1, 2, \cdots N)$, where each element in P_j was determined by the $TF \times IDF$ weighting technique [17]. The data owner extends the P to a $(m + 1) -$ dimension vector P', where $P'[m+1] = 1$. The data owner splits P' into two $(m + 1)$ $-$dimension vectors (P_a, P_b) using the key S. i.e, if $S[j] = 0$, $P_a[i] = P_b[i] = P'[i]$, otherwise $P'[i] = P_a[i] + P_b[i]$, (the value of $P'[i]$ will be randomly split into $P_a[i]$ and P_b). Therefore, the index of encrypted document C_j can be denoted as $I_j = (P_a M_1, P_b M_2)$. Finally, the data owner sends $C_j \parallel FID_j \parallel I_j$ $(j = 1, 2, \cdots, N)$to the cloud server.

Trapdoor generation Keyword set $\widetilde{\mathcal{W}}$ will be created by search user firstly, where we utilize a m-dimensional binary vector Q to indicate the preference factors of w_j. Then, Q will be extended to $(m + 1)$ dimension vector Q', where $Q'[m + 1]$ was set as $-s$ (the value of $-s$ will be explained in following schemes), then Q' is scaled by a random number $r \neq 0$ to generate $Q'' = r \cdot Q'$. After applying the same splitting and encryption processes as above, the trapdoor $T_{\widetilde{\mathcal{W}}}$ is generated as $\left(M_1^{-1} q_a, M_2^{-1} q_b\right)$. Finally, the search user sends $T_{\widetilde{\mathcal{W}}}$ to the cloud server.

Query With the index I_j $(j = 1, 2, \cdots, N)$ and trapdoor $T_{\widetilde{\mathcal{W}}}$ the final query result is as follows

$$\begin{aligned} R_j = I_j \cdot T_{\widetilde{\mathcal{W}}} &= (P_a M_1, P_b M_2) \cdot \left(M_1^{-1} q_a, M_2^{-1} q_b\right) \\ &= P_a \cdot q_a + P_b \cdot q_b = P' \cdot Q'' \\ &= r \cdot (P \cdot Q - s) \end{aligned} \tag{2}$$

4.1 Model analysis

Compared with the traditional model, we replace the values of $P[i]$ and $Q[i]$ by the relevance scores and the preference factors of keywords, respectively. For ease of calculation, the score is rounded up, i.e, $score(w_i, F_j) = \lceil 10 * score(w_i, F_j)\rceil$, and we assume that the score is not greater than D, i.e. $score(w_i, F_j) < D$. And we also assume that the keyword sets of the the "OR" "AND" and "NO" opreations are $(w_1', w_2', \cdots, w_{l_1}')$, $(w_1'', w_2'', \cdots, w_{l_2}'')$ and $(w_1''', w_2''', \cdots, w_{l_3}''')$, respectively, the "OR","AND" and "NO" operations denoted by $\vee, \wedge$ and $\neg$, respectively. Here we assume that the "NO" keywords have maximum weight, "AND" second, "OR" minimum. Thus the corresponding rule can be represented as $(w_1' \vee w_2' \vee \cdots \vee w_{l_1}') \wedge (w_1'' \wedge w_2'' \wedge \cdots \wedge w_{l_2}'')$ $\wedge(\neg w_1''' \wedge \neg w_2''' \wedge \cdots \wedge \neg w_{l_3}''')$ by the ascending order of keyword weight, For "OR","AND" and "NO" opreations, the search user chooses a super-increasing sequence $(a_1, a_2, \cdots, a_{l_1})$, $(a_{l_1+1}, a_{l_1+2}, \cdots, a_{l_1+l_2})$ and $(a_{l_1+l_2+1}, a_{l_1+l_2+2} \cdots, a_{l_1+l_2+l_3})$ $(\sum_{i=1}^{j-1} a_i \cdot D < a_j (j = 2, 3 \cdots N))$ to achieve searching with keyword weight, respectively, and we assume that $l_1 + l_2 + l_3 = N$. So according to the search keyword set $(w_1', w_2', \cdots, w_{l_1}', w_1'', w_2'', \cdots, w_{l_2}'', w_1''', w_2''', \cdots, w_{l_3}''')$, the corresponding values in Q are set as $(a_1, \cdots, a_{l_1}, a_{l_1+1}, \cdots, a_{l_1+l_2}, a_{l_1+l_2+1}, \cdots, a_{l_1+l_2+l_3})$. Other values in Q are set as 0.

We demonstrate that our model(FMRS) can achieve logical search operation as following(for the convenience of deduction, we still use w_i to replace w_i', w_i'' and w_i''' in the following paragraphs):

Step 1: A search user needs to be sure of the keywords to be searched for. In FMRS, firstly, the values of $score\,(w_i, F_j)$ which belongs to"NO" keyword set will be set as 1, then the search results is:

$$R_j = r \cdot (P \cdot Q - s) = r \cdot (\sum_{i=1}^{N} score(w_i, F_j) a_i - s) \tag{3}$$

(here s is equal to the minimum value of the "NO" keywords weight, i.e,$s = a_{l_1+l_2+1}$)

Step 2: Check whether R_j is less than 1 by computing following equation (4)

$$\begin{aligned} R_j &= r \cdot (P \cdot Q - s) \\ &= r \cdot (\sum_{i=1}^{N} score(w_i, F_j) \cdot a_i - s) \\ &= r(\sum_{i=1}^{l_1+l_2} score(w_i, F_j) a_i + \sum_{i=l_1+l_2+1}^{N} score(w_i, F_j) a_i - s) \end{aligned} \tag{4}$$

We know that $\sum_{i=1}^{j-1} a_i \cdot D < a_j (j = 2, 3 \cdots N)$, if all the keywords in the "NO" keyword set are not in the keyword sets of $F_j (j = 2, 3 \cdots N)$, we can infer $\sum_{i=l_1+l_2+1}^{N} score(w_i, F_j) \cdot a_i = 0$, therefore

$$\begin{aligned} R_j &= r(\sum_{i=1}^{l_1+l_2} score\left(w_i, F_j\right) \cdot a_i - s) \\ &= r(\sum_{i=1}^{l_1+l_2} score\left(w_i, F_j\right) \cdot a_i - a_{l_1+l_2+1}) \\ &< \sum_{i=1}^{l_1+l_2} D \cdot a_i - a_{l_1+l_2+1} \\ &< 0 \end{aligned} \tag{5}$$

In the same way, If there is a keyword belongs to "NO" keyword set and which is in the keyword sets of $F_j (j = 2, 3 \cdots N)$, there must be $R_j > 0$. So, if $R_j > 0$, we choose a new R_j and return to **Step 2**. Otherwise, we go to the next step.

Step 3: Use R_j to mod $(-r \cdot a_{l_1+l2+1}, r \cdot a_{l_1+l_2}, \cdots, r \cdot a_{l_1+1})$ in turn, then check whether the quotient is over or equal to 1 each time. Besides, the remainder can't be zero. For the first time, $R_j = r(\sum_{i=1}^{l_1+l_2} score(w_i, F_j) \cdot a_i - s) = r \cdot \sum_{i=1}^{l_1+l_2} score(w_i, F_j) \cdot a_i - rs$. Then $R_j \quad mod - r \cdot a_{l_1+l_2+1} = r \cdot \sum_{i=1}^{l_1+l_2} score(w_i, F_j) \cdot a_i$. Obviously, quotient is 1, then we go to the next step. (the purpose of this step is to eliminate the effect of s).

For the second time , the value of R_j is equal to the remainder operated last time, which is $r \cdot \sum_{i=1}^{l_1+l_2} score(w_i, F_j) \cdot a_i$, then we mod $r \cdot a_{l_1+l_2}$, here we know

$$\begin{aligned} R_j &= r \cdot \sum_{i=1}^{l_1+l_2} score(w_i, F_j) \cdot a_i \\ &= r \cdot \sum_{i=1}^{l_1+l_2-1} score(w_i, F_j) \cdot a_i + r \cdot score(w_{l_1+l_2}, F_j) \cdot a_{l_1+l_2} \end{aligned} \tag{6}$$

If the keyword $w_{l_1+l_2}$ (i.e., $w''_{l_1+l_2}$) is not in the keyword sets of $F_j(j = 2, 3 \cdots N)$, then

$$\begin{aligned} R_j &= r \cdot \sum_{i=1}^{l_1+l_2-1} score(w_i, F_j) \cdot a_i \\ &< r \cdot \sum_{i=1}^{l_1+l_2-1} D \cdot a_i \\ &< r \cdot a_{l_1+l_2} \end{aligned} \tag{7}$$

The quotient of $R_j mod \quad r \cdot a_{l_1+l_2} = 0$, we should also choose a new R_j and return to **Step 2**. On the contrary, if the keyword $w_{l_1+l_2}$ (i.e., $w''_{l_1+l_2}$) is in the keyword sets of $F_j, R_j mod \quad r \cdot a_{l_1+l_2} \geqslant 1$, we go to the next step.

In a similar way, using the R_j to mod $(-r \cdot a_{l_1+l2-1}, r \cdot a_{l_1+l_2-2}, \cdots, r \cdot a_{l_1+1})$ in turn, if all the keywords in the "AND" keyword set are in the keyword sets of $F_j(j = 2, 3 \cdots N)$, the quotient is $\geqslant 1$ in each time. Besides, the final remainder can't be zero, i.e., the remainder of R_j mods $a_{l_1+1} \neq 0$. (The remainder is not 0 guarantee that at least one "OR" keyword in the document)can satisfy the above matching rule with "OR", "AND"and "NO".

Step 4: For all of the documents that conform to the above query scheme. Return K documents with the highest scores by equation (2).

5 SECURITY ANALYSIS

In this section, we analyze the security of the proposed model. In particular, we focus on how to achieve confidentiality of documents, privacy protection of index and trapdoors, and unlinkability of trapdoors of our proposed model. Other security features are not the key issues of our scheme.

5.1 Confidentiality of Documents

In FMRS, raw documents are stored in cloud. Taking privacy into account, we exploit symmetric encryption (e.g., AES) to encrypt every document before outsourcing them. Because AES was proved to be secure in [19], and it is unable to spy any information or content of documents if attacker have not the secret key sk. Therefore, the confidentiality of encrypted documents can be protected well.

5.2 Privacy Protection of Index and Trapdoor

For convenience of searching, indexes and the trapdoors are created accord to keywords utilizing in our query process. All the index $I_j = (p_a M_1, p_b M_2)$ and the trapdoor $T_{\widetilde{\mathcal{W}}} = (M_1^{-1} q_a, M_2^{-1} q_b)$ are ciphertexts of vectors (P, Q). The secret key is $K = (S, M_1, M_2)$ that generated by data owner in our model, where S functions as a splitting indicator that divides P and Q into (p_a, p_b) and (q_a, q_b) respectively, then, we use two invertible matrices M_1 and M_2 to encrypt (p_a, p_b) and (q_a, q_b). Because the high security of KNN has been proved under the known ciphertext model [17]. Therefore, the privacy of index and trapdoor are protected well in FMRS.

5.3 Unlinkability of Trapdoor

Some documents stored in the cloud server may be frequently retrieved. Unlinkability refers to the case where the cloud server can not obtain keyword information from the trapdoors. Once unlinkability of trapdoor is broken, the cloud server can deduce relationship of trapdoors, and threaten the privacy of keywords. Therefore, for the same keywords, trapdoor should be generated randomly, rather than deterministic. We prove FMRS can achieve the unlinkability of trapdoors in a strong threat model, i.e., known background model [15].

In our model, the trapdoor is made up of two parts. The values of $a_i (i = 1, 2, \cdots, N)$ are the super-increasing sequence randomly selected by the search user (assume there are α possible sequences). And the $(m + 1)$ dimension is $-s$ defined by the search user, where the value of s is equal to the minimum value of the "NO" keyword weights. i.e,$s = a_{l_1+l_2+1}$. Assuming that the number of different $a_{l_1+l_2+1}$ is represented as β. Further, $Q'' = r \cdot Q'$, Q' is used to multiply a positive random number r, assuming that all the possible values of r is 2^{η_r} (if the search user chooses η_r-bit r). Finally, Q'' is split into (q_a, q_b) by the splitting indicator S. Specifically, if $S[i] = 0 (i = 1, 2, \cdots, m + 1)$, the value of $Q''[i]$ will be randomly split into $q_a[i]$ and $q_b[i]$, assuming the number of '0' in S is μ, and each dimension with q_a and q_b is η_q bits. Note that η_s, η_r, μ and η_q are independent of each other. Then in our model, we calculate the probabilities of two trapdoors which are the same as follows:

$$P_2 = \frac{1}{\beta \cdot 2^{\eta_r} \cdot (2^{\eta_q})^{\mu}} = \frac{1}{\beta \cdot 2^{\eta_r + \mu\eta_q}} \tag{8}$$

Therefore, the larger β, η_r, μ and η_q can achieve the stronger security, we choose 1024-bit r, then the probability $P_1 < 1/2^{1024}$. Thus, the probabilities of two trapdoors which are the same is negligible. In summary, we present the comparison results of security level in Table **??**, where (FMRS)represents our model. Clearly, all the schemes can achieve confidentiality of documents and privacy protection of index and trapdoor, but the OPE schemes [20] cannot achieve the unlinkability of trapdoor very well because of the similarity relevance mentioned in [13]. Comparison with our model, the scheme [12] can not return precise results because the relevance scores is not utilized.

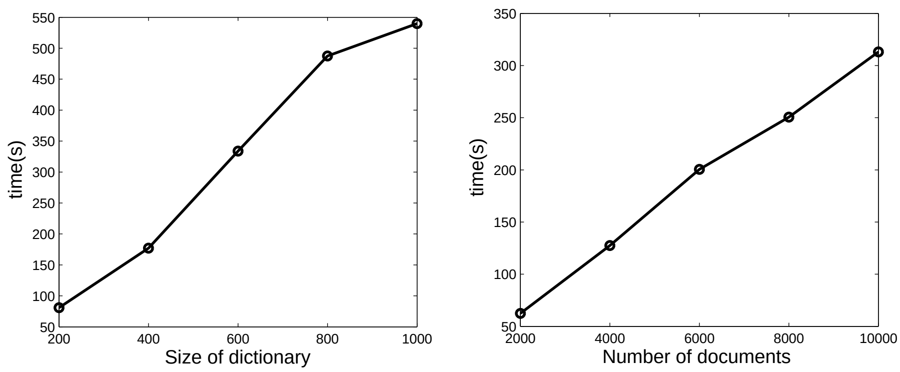
6 PERFORMANCE EVALUATION

In this section, we analyze the performance of the model by using the method of simulation and comparison with the existing models [10, 13, 15]. We randomly select a certain number of data through a real database 1990-2003 [21], and conduct real-world experiments on an Intel Core i5 2.6 GHz system.

6.1 Computation overhead

In order to provide a comprehensive analysis of computation overhead, we discuss it from following phases.

Index building. Note that the *Index building* phase of FMRS, which contains the relevance score computing. Considering the cost of calculating the relevance score, it is negligible in comparison with the cost of index building, we do not distinguish them. Moreover, because the index instruction mainly involves with the two multiplications of a $(m+1) \times (m+1)$ invertible matrix and a $(m+1)-$ dimension splitting vector. as shown in Fig. 2, we can see the time of building index is significantly associated with the number of documents and dictionaries.

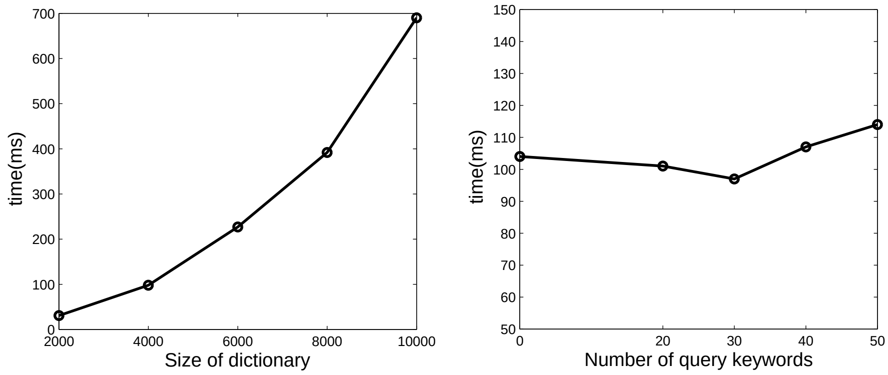


(a) (b)

Fig. 2: Time for building index. (a) Different size of dictionary N=6000. (b) Different number of documents $|\mathcal{W}| = 400$.

Trapdoor generation. In *Trapdoor generation* phase, our model randomly generates a super increasing sequence and a weight sequence, respectively, which is same as [12]. As shown in Fig. 3, the time of generating trapdoors is also significantly associated with the number of dictionaries, instead of the number of query keywords. It is partly because even if we do not want to search some keywords, we will still need to set values for the corresponding elements. With the increase in the number of the keywords in the dictionary, the time cost rises.

Query. The computation overhead in *Query* phase, as shown in Fig. 4, is significantly associated with the size of dictionary and the number of documents, instead of the number of query keywords. Note that, in *Trapdoor generation* and *Query* phases, the computation overheads are irrelative to the number of query keywords. Thus our schemes are more efficient compared with some multiple-keyword search schemes [22, 23], as their cost is linear with the number of query keywords. Besides, comparing with [12], our model returns more precise results because of using the relevance scores.



(a) (b)

Fig. 3: Time for generating trapdoor. (a) Different size of dictionary, $|\widetilde{\mathcal{W}}|$=20. (b)Different number of query keywords, $|\mathcal{W}| = 4000$.

7 Conclusion

In this paper, we propose a Fine-grained Multi-keyword Ranked Search (FMRS) shceme over the encrypted cloud data. Specifically, we develop the multi-keyword ranked search to support both mixed “AND”, “OR” and “NO” operations of keywords and ranking according to the preference factor and relevance score. Security analysis indicates that FMRS scheme can preserve confidentiality of documents, privacy protection of index and trapdoor and unlinkability of trapdoor. Real-world experiments demonstrate that FMRS can achieve better performance in terms of functionality and efficiency compared to the existing proposals.

8 Acknowledgement

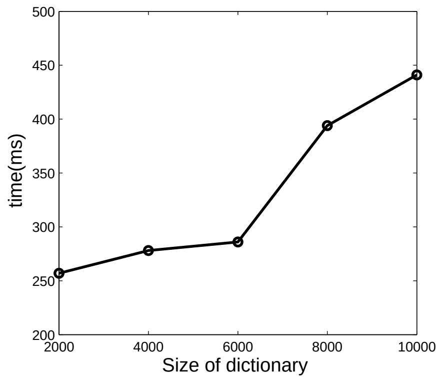
This work is supported by the National Key R&D Program of China under Grants 2017YFB0802300 and 2017YFB0802000, the National Natural Science Foundation of China under Grants 61772121, 61728102, and 61472065, the Fundamental Research Funds for Chinese Central Universities under Grant ZYGX2015J056.

References

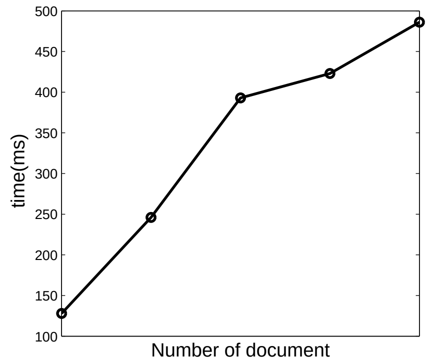
1. Wang, C., Cao, N., Ren, K., Lou, W.: Enabling secure and efficient ranked keyword search over outsourced cloud data. IEEE Transactions on Parallel and Distributed Systems **23**(8) (2012) 1467–1479
2. Yu, M., Yang, K., Wei, L., Sun, J.: Practical private information retrieval supporting keyword search in the cloud. In: 2014 Sixth International Conference on Wireless Communications and Signal Processing (WCSP),. (Oct 2014) 1–6

3. Xu, G., Li, H., Liu, S., Wen, M., Lu, R.: Efficient and privacy-preserving truth discovery in mobile crowd sensing systems. IEEE Transactions on Vehicular Technology **68**(4) (2019) 3854–3865
4. Xu, G., Li, H., Lu, R.: Practical and privacy-aware truth discovery in mobile crowd sensing systems. In: Proceedings of ACM CCS. (2018) 2312–2314
5. Xu, G., Li, H., Liu, D., Ren, H., Dai, Y., Liang, X.: Towards efficient privacy-preserving truth discovery in crowd sensing systems. In: Proceedings of GLOBECOM. (2016) 1–6
6. Xu, G., Ren, Y., Li, H., Liu, D., Dai, Y., Yang, K.: Cryptmdb: A practical encrypted mongodb over big data. In: 2017 IEEE International Conference on Communications (ICC). (May 2017) 1–6
7. Li, R., Xu, Z., Kang, W., Yow, K.C., Xu, C.Z.: Efficient multi-keyword ranked query over encrypted data in cloud computing. Future Generation Computer Systems **30** (2014) 179–190
8. Li, H., Liu, D., Dai, Y., Luan, T.H., Shen, X.: Enabling efficient multi-keyword ranked search over encrypted cloud data through blind storage. IEEE Transactions on Emerging Topics in Computing (2014, DOI10.1109/TETC.2014.2371239)
9. Xu, G., Li, H., Dai, Y., Yang, K., Lin, X.: Enabling efficient and geometric range query with access control over encrypted spatial data. IEEE Transactions on Information Forensics and Security **14**(4) (2019) 870–885
10. Sun, W., Wang, B., Cao, N., Li, M., Lou, W., Hou, Y.T., Li, H.: Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking. IEEE Transactions on Parallel and Distributed Systems **DOI: 10.1109/TPDS.2013.282** (2013)
11. Strizhov, M., Ray, I.: Substring position search over encrypted cloud data using tree-based index. In: 2015 IEEE International Conference on Cloud Engineering (IC2E). (March 2015) 165–174
12. Li, H., Yang, Y., Luan, T.H., Liang, X., Zhou, L., Shen, X.S.: Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data. IEEE Transactions on Dependable and Secure Computing (2015, DOI: 10.1109/TDSC.2015.2406704)
13. Yu, J., Lu, P., Zhu, Y., Xue, G., Li, M.: Towards secure multi-keyword top-k retrieval over encrypted cloud data. IEEE Transactions on Dependable and Secure Computing **10**(4) (2013) 239–250
14. Xu, G., Li, H., Dai, Y., Bai, J., Lin, X.: Efrs:enabling efficient and fine-grained range search on encrypted spatial data. In: 2018 IEEE International Conference on Communications (ICC). (May 2018) 1–6
15. Cao, N., Wang, C., Li, M., Ren, K., Lou, W.: Privacy-preserving multi-keyword ranked search over encrypted cloud data. IEEE Transactions on Parallel and Distributed Systems **25**(1) (2014) 222–233
16. Xu, G., Li, H., Tan, C., Liu, D., Dai, Y., Yang, K.: Achieving efficient and privacy-preserving truth discovery in crowd sensing systems. Computers & Security **69** (2017) 114–126
17. Wong, W.K., Cheung, D.W.l., Kao, B., Mamoulis, N.: Secure knn computation on encrypted databases. In: Proceedings of SIGMOD International Conference on Management of data, ACM (2009) 139–152
18. Zobel, J., Moffat, A.: Exploring the similarity space. In: ACM SIGIR Forum. Volume 32., ACM (1998) 18–34
19. Ferguson, N., Schroeppel, R., Whiting, D.: A simple algebraic representation of rijndael. In: Selected Areas in Cryptography, Springer (2001) 103–111
20. Wang, C., Cao, N., Li, J., Ren, K., Lou, W.: Secure ranked keyword search over encrypted cloud data. In: Proceedings of ICDCS, IEEE (2010) 253–262
21. `http://kdd.ics.uci.edu/databases/nsfabs/nsfawards.html`

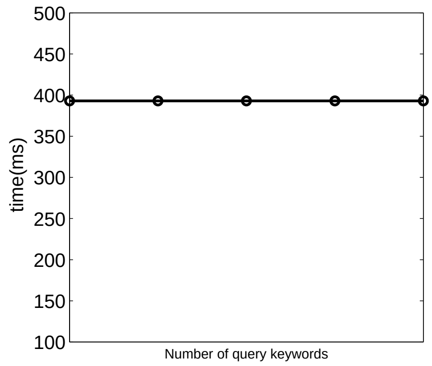
22. Golle, P., Staddon, J., Waters, B.: Secure conjunctive keyword search over encrypted data. In: Applied Cryptography and Network Security, Springer (2004) 31–45
23. Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. In: Theory of cryptography. Springer (2007) 535–554



(a)



(b)



(c)

Fig. 4: Time for query. (a) For the different size of dictionary with the same number of documents and number of search keywords, $N = 6000, |\widetilde{\mathcal{W}}| = 20$. (b) For the different number of documents with the same size of dictionary and number of search keywords, $|\mathcal{W}| = 8000, |\widetilde{\mathcal{W}}| = 20$. (c) For the different number of search keyword with the same size of dictionary and number of documents, $N = 6000, |\mathcal{W}| = 8000$.