Privacy-Preserving Retrieval Scheme Over Encrypted Medical Records with Relevance Ranking

Wanting Lei[1], Xiehua Li[1](✉), Yingzhu Wang[1], and Xiaoyu Mei[1,2]

[1] College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, Hunan, China
beverly@hnu.edu.cn

[2] New Lynn School, 1 Hutchinson Avenue, New Lynn, Auckland 0600, New Zealand

Abstract. Electronic medical records (EMRs) contain a large amount of highly private and sensitive information of patients and medical institutions. For privacy concerns, EMRs are usually encrypted before outsourcing them to the cloud storage platform. However, it is difficult to retrieve the encrypted EMRs accurately and efficiently. The existing encrypted data retrieval schemes can hardly achieve the goals of fuzzy multi-keyword search, relevance ranking and high retrieval accuracy. Thus, this paper proposes a Privacy-preserving Retrieval scheme over Encrypted Medical Records (PREMR) that can satisfy those goals. We utilize the Possibility-Levenshtein based Spelling Corrector (PLSC) to support fuzzy multiple input keywords. A homomorphic-based encryption algorithm is proposed for relevance score encryption and calculation so that the encrypted medical records can be ranked without leaking private information. We theoretically prove that our scheme can achieve data confidential and privacy preserving. With the experiments' evaluation, we analyze the costs and efficiency of our scheme. Finally, the comparison of PREMR with other related schemes shows that our scheme is more efficient and secure.

Keywords: Encrypted medical records · Fuzzy multi-keyword search · Homomorphic encryption · Relevance ranking · Searchable encryption

1 Introduction

Electronic medical records (EMR) are widely used in healthcare systems as the healthcare industry is moving toward digitization. Many hospitals and medical institutes use EMRs for online diagnosis, health screening, and new drug development. Since a large amount of EMRs and images are generated everyday, most hospitals and institutes use the public cloud storage platform to store these data. However, medical records contain highly sensitive personal information that should not be outsourced without protection. A simple way to protect EMRs information is to encrypt them before outsourcing, but this reduces the accuracy and efficiency of EMR retrieval. Aim to solve this problem, the first searchable encryption scheme was proposed by Song *et al.* [1], in which some

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 81–92, 2024.
https://doi.org/10.1007/978-981-99-8211-0_9

basic search approaches over encrypted data were discussed. Boneh *et al.* [2] proposed the first public key encryption scheme. After that searchable encryption becomes an important technology for encrypted data retrieval with privacy preserving [3–5].

Another issue that affects the retrieval accuracy and efficiency is the correctness of the input keywords. Errors in the input keywords would cause inaccurate search results and even retrieval failure. In this paper, we utilize our former proposed Probability-Levenshtein based Spelling Correction (PLSC) algorithm [6, 7] in recommended keywords ranking and medical keywords correction. So that PLSC can support fuzzy multiple keywords input and provide a more accurate search query. Then, we propose a correlation encryption and calculation algorithm based on homomorphic encryption, so that the cloud server can securely complete the calculation of the sum of keywords the relevance scores in the EMR. In addition, proxy is introduced in our scheme to support multiple EMR owners and multi-keyword relevance score ranking. Finally, we compared our PREMR scheme with the newly published searchable encryption scheme for performance evaluation. Our contributions can be summarized as follows.

- In order to test the accuracy of PLSC, we build a library that contains 2000 medical records with more than 3000 medical words. Based on this library, we compare our work with Norvig's spelling corrector and edit distance.
- We design a relevance score encryption and ranking algorithm based on homomorphic encryption to support secure keyword-based query and retrieval. The algorithm adopts Paillier-based encryption to sum up encrypted multi-keyword relevance scores.
- We built up an encrypted EMR retrieval system that can support data outsourcing and dynamic updates for multiple EMR owners. We also implement the performance comparison among PREMR and several similar searchable encryption schemes.

The rest of the paper is organized as follows. Section 2 is the related work on searchable encryption. Section 3 introduces the template of EMR and PLSC correction evaluation. Section 4 presents the constructions and definitions of our scheme. The detailed description of our PREMR scheme is represented in Sect. 5. Theoretical security analysis is given in Sect. 6. We give the scheme implementation results and comparison in Sect. 7. Section 8 is the conclusion of the whole paper.

2 Related Work

Most researches on searchable encryption are aiming at improving accuracy and security of data retrieval. For improving accuracy, Sun *et al.* [9] proposed a multi-keyword search scheme using a vector space model and a cosine measure with TF (word frequency) × IDF (inverse text frequency index) to provide order-preserving document retrieval. Kabir *et al.* [10] improved Sun's scheme by writing the plaintext TF values in the index tree orderly. However, the plaintext TF values may leak information about keywords and documents. To improve security of encrypted document retrieval, Liu *et al.* [11] proposed a verifiable searchable encryption scheme that can verify the correctness of retrieval results over dynamic data collection. Du *et al.* [12] proposed a searchable symmetric encryption scheme that combines access control and boolean queries. Liu *et al.* [13] adopted attribute hierarchy with the comparison-based encryption to achieve dynamic access control over encrypted personal health records. Those searchable encryption schemes are usually considered as a way to guarantee data privacy and search efficiency. Also, there are many researches on searchable encryption schemes with multiple keyword support [14, 15] and are applied in many other areas [16]. However, these schemes have some limitations in retrieval efficiency, accuracy or privacy. In cloud computing applications, especially in medical cooperation projects, the searchable encryption should be able to support precise and efficient retrieval on outsourced medical records for further diagnosis.

Another research topic on searchable encryption is fuzzy search for multiple keywords. Li *et al.* [13] proposed a scheme that used kNN and Euclidean distance to select *k* nearest database records, but the search accuracy is not desirable. Traditional spelling correction algorithms, such as the Levenstein distance, do not achieve high correction accuracy if the spelling error is more than two letters. Zhong [8] proposed a fuzzy search scheme that used k-gram to construct a fuzzy keyword set and Jaccard coefficient to calculate the similarity of keywords. Gnanasekaran [18] converted keyword into a vector, and used LSH (Local Sensitive Hash) to support fuzzy keyword search. Aritomo [19] used simhash to realize the keyword fuzzy search, and the VP-tree to improve search accuracy. K. Wang [20] used LSH to build index, and used Bloom filter to realize fuzzy search over multiple keywords. However, those schemes did not consider the misalignment of letters in the keywords, which may lead to less accurate search results.

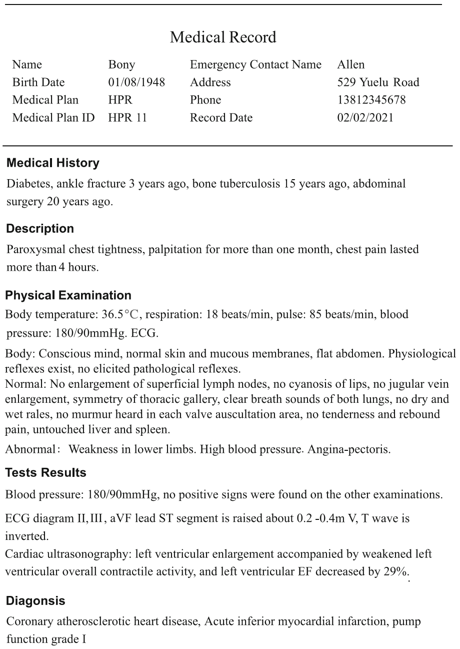
3 Spelling Correction on Electronic Medical Records

3.1 Electronic Medical Records Templates

In order to support fuzzy search, we adopt our previous PLSC (Probability-Levenshtein based Spelling Correction) algorithm [6] to correct the ambiguous input search words. We build a library with 2000 EMRs that contain 3000 common medical terms. The medical terms are selected from [17]. The format example of EMR is shown in Fig. 1. This is a typical EMR, which contains private information such as the patient's name, address and phone number, and also sensitive information such as the patient's condition, diagnosis and prescription.

3.2 Spelling Correction Evaluation of EMRs

We evaluate the PLSC algorithm using our EMRs library. The experiment tests the correction accuracy of PLSC, Norvig's spelling corrector, and edit distance. Table 1 gives the correction probability of three spelling correctors, where spelling errors in each keyword are random. The test result shows that PLSC is able to give more accurate candidate correction especially when there are more than two random errors in the input keywords.



Medical Record

Name	Bony	Emergency Contact Name	Allen
Birth Date	01/08/1948	Address	529 Yuelu Road
Medical Plan	HPR	Phone	13812345678
Medical Plan ID	HPR 11	Record Date	02/02/2021

Medical History

Diabetes, ankle fracture 3 years ago, bone tuberculosis 15 years ago, abdominal surgery 20 years ago.

Description

Paroxysmal chest tightness, palpitation for more than one month, chest pain lasted more than 4 hours.

Physical Examination

Body temperature: 36.5 °C, respiration: 18 beats/min, pulse: 85 beats/min, blood pressure: 180/90mmHg. ECG.

Body: Conscious mind, normal skin and mucous membranes, flat abdomen. Physiological reflexes exist, no elicited pathological reflexes.
Normal: No enlargement of superficial lymph nodes, no cyanosis of lips, no jugular vein enlargement, symmetry of thoracic gallery, clear breath sounds of both lungs, no dry and wet rales, no murmur heard in each valve auscultation area, no tenderness and rebound pain, untouched liver and spleen.

Abnormal: Weakness in lower limbs. High blood pressure. Angina-pectoris.

Tests Results

Blood pressure: 180/90mmHg, no positive signs were found on the other examinations.

ECG diagram II, III, aVF lead ST segment is raised about 0.2 -0.4m V, T wave is inverted.
Cardiac ultrasonography: left ventricular enlargement accompanied by weakened left ventricular overall contractile activity, and left ventricular EF decreased by 29%.

Diagonsis

Coronary atherosclerotic heart disease, Acute inferior myocardial infarction, pump function grade I

Fig. 1. Medical Record Template

Table 1. Accuracy comparison with random errors

Errors	PLSC (%)	Norvig's corrector (%)	Edit distance (%)
1 - 2	94.7	89.5	85.1
2	93.2	81.1	73.4
1 - 3	71.6	64.8	60.1

4 System Construction and Preliminaries

This section first introduces our system structure, and then describes threat models, system goals, notations, and cryptographic preliminaries.

4.1 System Model

There are four principals in the PREMR system. EMR owners is responsible for medical data encryption and index building. They upload encrypted EMRs to the cloud service provider (CSP), and send indexes to the Proxy. Proxy merges the indexes from all EMR owners and encrypts the merged index. Then Proxy uploads the secured index to the CSP. The encrypted EMRs and index are uploaded by EMR owners and Proxy, respectively. Meanwhile, EMR owners distribute decryption keys to authorized users via secure channel.

In our scheme, the EMR storage server is considered as an "honest-but-curious" entity. Specifically, the storage server will honestly implement the protocol, but also curiously analyze the index, stored data and queries to capture more information associated with plaintext EMRs. EMR owners are suppose to be honest because they have the original plaintext records. Proxy is a trustworthy entity who builds up secured index for outsourced data, and generates trapdoors for users' searching queries. Users are untrusted, they may collude with others to get more information about the encrypted EMRs. Secret keys are uncompromised.

4.2 Notations

- R: plaintext EMR set, R = $\{R_1, R_2, \ldots, R_n\}$;
- R$'$: encrypted EMR set, R$'$ = $\{R'_1, R'_2, \ldots, R'_n\}$;
- ID: EMR identifier in plaintext $ID = \{id_1, id_2, \ldots, id_n\}$;
- ID': encrypted EMR identifier, $ID' = \{id'_1, id'_2, \ldots, id'_n\}$;
- SW: keywords set in plaintext, SW = $\{W_1, W_2, \ldots, W_m\}$;
- SW$'$: keywords set in ciphertext, SW$'$ = $\{\ W'_1, W'_2, \ldots, W'_n\}$;
- $S_{i,j}$: plaintext relevance score of keyword W_i in R_j; S_j: sum of relevance score in R_j in plaintext;
- $S'_{i,j}$: encrypted relevance score of keyword W_i in document R_j; S'_j: sum of relevance score in R_j in ciphertext;
- $\mathbb{Z}^*_{y^2}$ is the set of integers range between 1 and y^2.

Our PREMR scheme includes three major processes: EMR index building, encrypted EMR searching and queue-based ciphertext retrieval.

4.3 Cryptographic Preliminaries

In PREMR scheme, we adopt both symmetric key algorithm and homomorphic encryption to guarantee the security of EMR and the value of relevance scores. The symmetric key algorithm (SKA) is used to encrypt keywords, EMR identifiers, and EMRs. The homomorphic encryption (HE) is used to encryption the relevance score of each keyword in every EMR. The algorithms that are involved in the PREMR system are defined as followed.

- SKA = (T, K, ENC1, DEC1) is a symmetric key encryption algorithm, where T is the input data, K is the symmetric key, ENC1 is the encryption algorithm; DEC1 is the decryption algorithm.
- HE = (RS, PK, SK, ENC2, DEC2) is a Paillier-based homomorphic encryption, where RS is the relevance score of a keyword, PK is the public key to encrypt RS, SK is the secret key. ENC2 and DEC2 are the encryption and decryption algorithms. PK and SK are generated with the followed method:

1. Suppose $p, q \in Z_n$ are two large prime numbers, and $\gcd(pq,(p-1)(q-1)) = 1$, $\Phi(n) = (p-1)(q-1)$. Let $n = pq$, $\lambda = \mathrm{lcm}(p-1, q-1)$.

2. The multiplicative subgroup $Z_n \times Z_n^* \rightarrow Z_{n^2}^*$. $\left|Z_{n^2}^*\right| = \Phi\left(n^2\right) = n\Phi(n)$. g is some element of $Z_{n^2}^*$, $r \in (0, n)$ is a random integer, and $\gcd(r, n) = 1$, $r^{(p-1)} \equiv 1(\bmod\ p)$. $r^\lambda = 1 \bmod n$, $r^{n\lambda} = 1 \bmod n^2$.
3. Let $L(x) = \frac{x-1}{n}$, the modular multiplicative inverse $\mu = (\mathrm{L}(g^\lambda \bmod n^2))$-1mod n.
4. The public key is PK = (n, g) and the secret key SK = λ.

5 Encrypted EMR Searching with Privacy Preserving

This section introduces the index building process and EMR searching. Then it describes the relevance score calculation and ranking algorithms.

5.1 EMR Index Building

Before encrypting EMRs, the owners first extract keywords, and build inverted plaintext indexes. Subsequently, EMR owners encrypt and upload the medical records to the cloud server, and at the same time send the plaintext index to the Proxy. Proxy collects indexes from all EMR owners, merges and builds up the secure inverted index.

Plaintext Index Building and EMR Encryption. EMR owners first extract keywords from EMRs, calculate the TF-IDF value for each keyword as its relevance score, and then build the plaintext index **I**. $\mathbf{I} = \{\mathrm{I}_1,\mathrm{I}_2,\mathrm{I}_3\ldots\mathrm{I}_m\}$, $\mathrm{I}_i = (W_i, \bigcup_j < id_j, S_{i,j} >)$, I_i is the inverted index of keyword W_i, id_j is the identifier of the EMR that contains W_i, $S_{i,j}$ denotes the TF-IDF score of keyword W_i, in the EMR with the identifier id_j.

Furthermore, EMR owner implements SKA (*, K_1, ENC1) to encrypt EMRs. Equation (1) describes the encryption process.

$$\begin{aligned} id_j' &\leftarrow \mathrm{KA}(id_j, K_1,\ \mathrm{ENC1}) \\ R_j' &\leftarrow (R_j, K_1,\ \mathrm{ENC1}) \\ C = &\left\{\left(id_1', R_1'\right), \left(id_2', R_2'\right), \ldots, \left(id_n', R_n'\right)\right\} \end{aligned} \tag{1}$$

EMR owners then send **I** to the Proxy, and upload C to the CSP.

Indexes Merging and Encryption. In our system, we support multiple EMR owners to outsource their medical records. Proxy is introduced to handle multiple indexes merging and secure index building, so that even though EMRs are encrypted with different keys the retrieval can still be accurate and efficient. The secure index $\mathbf{I}'$ is generated with the followed steps.

Step 1. Proxy receives multiple indexes from different EMR owners and merges them into a new index based on keywords.
Step 2. Proxy implements SKA(($*$, K_2, ENC1) to encrypt keywords and EMR identifiers.

$$\begin{aligned} W_i' &\leftarrow \mathrm{SKA}(W_i, K_2,\ \mathrm{ENC1}) \\ id_j'' &\leftarrow \mathrm{SKA}(W_i, K_2,\ \mathrm{ENC1}) \end{aligned} \tag{2}$$

Comparing Eq. (1) and Eq. (2) we can see that different encryption keys(K1, K2) are used to encrypt the same EMR identifier(id_j), so that the linkability of the index and stored EMR is broken.

Step 3. Proxy runs HE(RS, PK, ENC2) to encrypt the relevance score $S'_{i,j}$. The encryption process is defined in Eq. (3).

$$S'_{i,j} = g^{S_{i,j}} \times r^n mod n^2 \tag{3}$$

At last, Proxy establishes the secure index $\mathbf{I}'$ and upload it to the CSP. The format of the secure index is defined in Eq. (4).

$$\mathrm{I'} = \left\{\mathrm{I}_1^{'}, \mathrm{I}_2^{'}, \ldots, \mathrm{I}_i^{'}\right\}, \mathrm{I}_i^{'} = \left\{W_i^{'}, \bigcup\nolimits_j \left\langle id_j^{''}, S_{i,j}^{'}\right\rangle\right\} \tag{4}$$

5.2 Encrypted EMR Retrieval

When user tries to search a set of keywords, the PLSC algorithm will first correct the misspelled ones. Then user sends the plaintext keywords set $\mathbf{SW} = \{W_1, W_2, \ldots, W_t\}$ to the Proxy. Proxy generate the query trapdoor $\mathbf{SW}' = \{W'_1, W'_2, \ldots, W'_t\}$, where $W'_i =$ SKA(W_i, K_2, ENC1).

Algorithm 1 Ciphertext searching by CSP

Input: $SW' = \{W_1', W_2', ..., W_t'\}$;

Output: EMR identifiers, S_j'

```
function EMR SEARCHING
$\mathbf{I}_r' = \mathbf{I}'$;
    for ($i = 1$; $i \le t$; $i$++) do
        Search $\mathbf{I}'$;
        if $W_i' \in \mathbf{I}_i'. W_i'$ then $\mathbf{I}_r' = \mathbf{I}_r' \cap \mathbf{I}_i'$;
        end if
    end for
    while $\mathbf{I}_r' \neq \emptyset$ do
        for each $\mathbf{I}_r'.id_j$ do
            $\mathbf{I}_r'. S_j' = \prod_{i,j} S_{i,j}'$;
        end for
    end while
    return ($\mathbf{I}_r'$)
end function
```

CSP Searching Algorithm. SP searches $\mathbf{SW}'$ in $\mathbf{I}'$. The searching algorithm is described in Alg.1. Search result is the conjunction of EMRs that contain all queried keywords in $\mathbf{SW}'$. Subsequently, CSP sums the encrypted relevance scores of multiple keywords in each EMR. The relevance score calculation is defined in Eq. (5).

$$S'_j = \prod\nolimits_{i,j} S'_{i,j} = g^{\sum S_{i,j}} \times \prod\nolimits_i r_i^n mod n^2 \tag{5}$$

CSP returns the search result to the Proxy for relevance score decryption and ranking.

Relevance Ranking Algorithm. After receiving the search results from CSP, Proxy needs to decrypt and rank the summation of relevance scores for each returned EMR. Proxy implements SKA($*$, K_2, DEC1) to get the plaintext keywords W_i and EMR identifiers idi. Then, Proxy runs HE(S'_j, SK, DEC2) to decrypt the sum of relevance score. The decryption process is defined in Eq. (6).

$$\begin{aligned} S_j &= \frac{L\left(S_j'^{\lambda} mod n^2\right)}{L\left(g^{\lambda} mod n^2\right)} mod n \\ &= \frac{L\left(g^{\lambda \sum S_{i,j}} \times \prod_i r_i^{\lambda n} mod n^2\right)}{L\left(g^{\lambda} mod n^2\right)} mod\ n \\ &= \sum S_{i,j} \end{aligned} \tag{6}$$

where $\prod\limits_i r_i^{\lambda n} \equiv 1$. Proxy ranks the top-$k$ EMRs based on their $\sum S_{i,j}$ and returns the EMR identifiers back to users. Upon receiving the EMR identifiers, users send downloading requests to the CSP directly.

6 Security Analysis

This section analyzes the data confidentiality and private-preserving of our scheme. We have proved that our scheme can guarantee the security of ciphertext retrieval by using the queue-based search strategy, and can protect the EMR privacy through different encryption algorithms.

Data Confidential. The original EMRs are encrypted before outsourcing to the CSP and the decryption keys are distributed to users via secure channel. Based on the assumption we made in the system model in Sect. 4, EMRs can not be compromised without correct secret keys. Thus, EMR data confidential can be guaranteed.

Indexes are constructed separately by the EMR owners, then merged and encrypted by the Proxy. EMR identifiers in the index and in the outsourced EMRs are encrypted with different keys so that CSP cannot get the relationship of the encrypted EMRs and the encrypted index. Keywords relevance scores of each EMR are encrypted and calculated with the homomorphic encryption, CSP cannot get any information from the keywords and their relevance scores. Therefore, as long as the encryption keys are not compromised, the confidentiality of data, index, keywords and relevance scores can be guaranteed.

Possibility of Privacy Leakage. Queries are encrypted by proxy and then forwarded to CSP. So that, CSP cannot get user information and user privacy is protected. Meanwhile, the file downloading requests and query trapdoors are sent by users and proxy separately. It is impossible for the CSP to guess the exact correspondence between the queried keyword and the downloaded EMRs.

7 Performance Test

The performance test experiments are implemented by C++ programming language on Windows 7 machines, each of which is with an Intel(R) Core(TM) i5 6500 3.2 GHz processor and a 2GB RAM. The performance is evaluated with our own EMR dataset. Our dataset uses more than 3000 medical keywords to generate 2000 EMRs containing various diseases. We compare our scheme with the most relevant researches on searchable encryption: FMS [13], TBMSM [**Error! Reference source not found.**] and Zhong's scheme [8]. In the experiments, the number of keywords in 2000 EMRs varies from 1000 to 3000, and the number of EMRs varies from 100 to 2000.

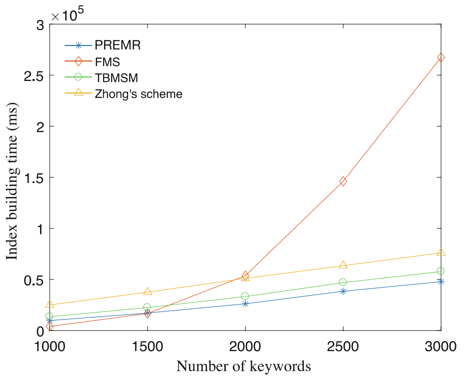
7.1 Index Building Efficiency

We compare the index building time and storage cost among four schemes. Figure 2(a) shows the time overhead required to build an index with the increasing number of keywords. The index building time of FMS grows exponentially since it needs to create an index vector for each document. When the number of keywords exceeds 1500 the index building time of FMS is more than that of other three schemes. While the time cost on building index with other three schemes are stable and increase linearly. The index structure of our PREMR scheme is the inverted index based on keywords. Therefore, the index generation time increases linearly with the increase of keywords. Figure 2. Index Building Timeshows the index generation time with the increase number of EMRs. Our PREMR takes less time to build the index than other three schemes. Compared with other searchable encryption methods, our PREMR is the most efficient one on index building stage.

Figure 3(a) shows the index storage size when the number of index keywords is 1000, 1500, 2000, 2500, and 3000 respectively. When the number of keywords in the index is greater than 1000 or the number of EMRs in the data set is greater than 300, the index storage overhead of our PREMR is less than that of other three schemes. Figure 3(b) shows the required index storage space with the number of EMRs ranges from 100 to 2000. I It indicates that PREMR scheme has better index generation efficiency and less index storage overhead than the other three schemes.

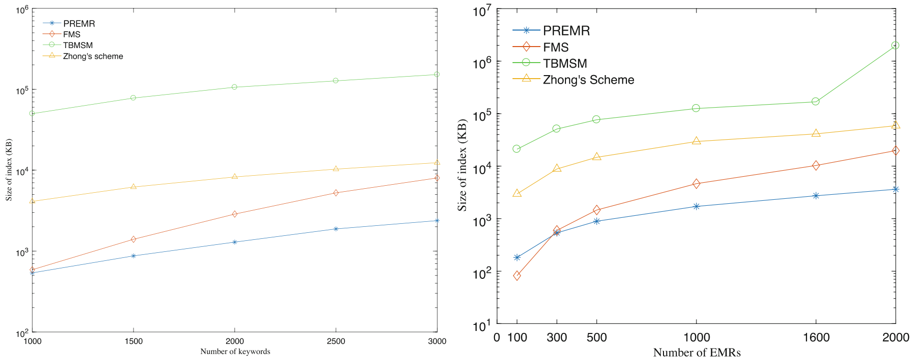
7.2 Trapdoor Generation Time

Figure 4 compares the trapdoor generation efficiency of these four schemes when there are 1000 queries, and the keywords in each query ranging from 10 to 50. Figure 4 shows that the trapdoor generation time of FMS is not affected by the number of queried



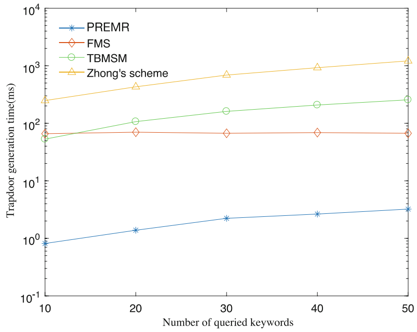
(a) Number of keywords (b) Number of EMRs

Fig. 2. Index Building Time



(a) Number of keywords (b) Number of EMRs

Fig. 3. Index Storage Space



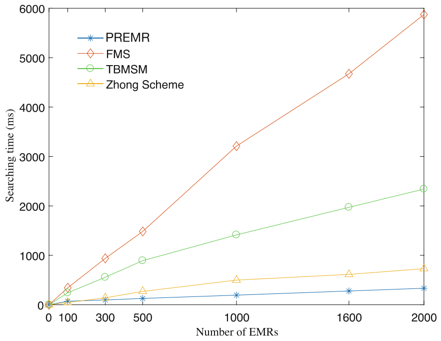


Fig. 4. Trapdoor Generation Time **Fig. 5.** Search Efficiency Comparison

keywords. This is because that the trapdoor in FMS is a fixed-length one-dimensional vector corresponding to the keywords, even though the number of keywords increases, the trapdoor generation time remains basically unchanged. The trapdoor generation time of PREMR, TBMSM and Zhong's scheme grows linearly with the increase of queried keywords. From comparison result, it shows that the PREMR scheme has a better performance on trapdoor generation efficiency, especially in supporting multiple keywords and simultaneous queries.

7.3 Search Efficiency

Figure 5 shows the search efficiency of compared schemes. All schemes are evaluated with the number of EMRs ranging from 100 to 2000, and the number of keywords in each query is 5. In FMS, a matrix calculation is carried out between the retrieval vector and index vector of each EMR, which increases the search time significantly with the increase of stored EMRs. In TBMSM scheme, a search sequence should be obtained firstly by matching each search keyword with that in the index. So that, the search time in TBMSM increases linearly with the number of keywords in the index. The search efficiency in Zhong's scheme is mainly affected by the mapping operation of the index and query vectors with LSH (Local Sensitive Hash) function. Although our PREMR scheme is also affected by the number of keywords, the search time grows slowly. Form Fig. 5 we can see that our PREMR scheme has less search time than the other schemes. The search time of PREMR is less than 1s even though there are 2000 encrypted EMRs in the database.

8 Conclusion

This paper proposed a privacy-preserving retrieval scheme over encrypted medical records. The proposed scheme can achieve multi-keyword fuzzy search and relevance ranking. In this paper, we use PLSC to support the fuzzy input keywords and improve spelling correction. In addition, homomorphic encryption algorithm is introduced to support keywords relevance scores calculation and ranking securely. Then, the theoretical proofs show that our PREMR scheme can guarantee the security of query vectors and stored EMRs. Finally, we experimentally analyzed and compared the PREMR with three other similar schemes, and the experimental results proved that the PREMR has better performance in index building, query trapdoor generation and search efficiency.

Acknowledgement. This work was supported by the National Natural Science Foundation of China under Project 92067104 and 61872134, and also supported by the Hunan Provincial Natural Science Foundation under Project 2021JJ30140.

References

1. Song, D.X., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: Proceedings of S&P, Berkeley, CA, USA, pp. 44–55 (2000)

2. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Proceedings of EUROCRYPT, Interlaken, Switzerland, pp. 506–522 (2004)
3. Li, H., Liu, D., Dai, Y., Luan, T.H., Shen, X.S.: Enabling efficient multi-keyword ranked search over encrypted mobile cloud data through blind storage. IEEE Trans. Emerg. Top. Comput. **3**(1), 127–138 (2015)
4. Li, R., Liu, A.X., Wang, A.L., Bruhadeshwar, B.: Fast and scalable range query processing with strong privacy protection for cloud computing. IEEE/ACM Trans. Networking **24**(4), 2305–2318 (2016)
5. Lei, X., Tu, G.-H., Liu, A.X., Xie, T.: Fast and secure kNN query processing in cloud computing. In: Proceedings of CNS, pp. 1–9 (2020)
6. Li, X., Li, F., Jiang, J., Mei, X.: Paillier-based fuzzy multi-keyword searchable encryption scheme with order-preserving. Comput. Mater. Continua **65**(2), 1707–1721 (2020)
7. Li, X., Long, G., Li, S.: Encrypted medical records search with supporting of fuzzy multi-keyword and relevance ranking. In: Proceedings of ICAIS, Dublin, Ireland, pp. 85–101 (2021)
8. Zhong, H., Li, Z., Cui, J., Sun, Y., Liu, L.: Efficient dynamic multi-keyword fuzzy search over encrypted cloud data. J. Netw. Comput. Appl. **149**, 102469 (2020)
9. Sun, W., et al.: Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking. IEEE Trans. Parallel Distrib. Syst. **25**(11), 3025–3035 (2014)
10. Kabir, T., Adnan, M.A.: A dynamic searchable encryption scheme for secure cloud server operation reserving multi-keyword ranked search. In: Proceedings of SysS, Dhaka, Bangladesh, pp. 1–9 (2017)
11. Liu, Q., Tian, Y., Wu, J., Peng, T., Wang, G.: Enabling verifiable and dynamic ranked search over outsourced data. IEEE Trans. Serv. Comput. **15**(1), 69–82 (2022)
12. Du, L., Li, K., Liu, Q., Wu, Z., Zhang, S.: Dynamic multi-client searchable symmetric encryption with support for boolean queries. Inf. Sci. **506**, 234–257 (2020)
13. 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 Trans. Dependable Secure Comput. **13**(3), 312–325 (2016)
14. Pakniat, N., Shiraly, D., Eslami, Z.: Certificateless authenticated encryption with keyword search: enhanced security model and a concrete construction for industrial IoT. J. Inf. Secur. Appl. **53**, 102525 (2020)
15. Wang, C., Yuan, X., Cui, Y., Ren, K.: Toward secure outsourced middlebox services: practices, challenges, and beyond. IEEE Network **32**, 166–171 (2018)
16. Hao, J., Huang, C., Ni, J., Rong, H., Xian, M., Shen, X.S.: Fine-grained data access control with attribute-hiding policy for cloud-based IOT. Comput. Netw. **153**, 1–10 (2019)
17. Stedman, T.L.: The American Heritage Stedman's Medical Dictionary, Edition 2. Houghton Mifflin Company (2002)
18. Xia, Z., Wang, X., Sun, X., Wang, Q.: A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. IEEE Trans. Parallel Distrib. Syst. **27**(2), 340–352 (2016)
19. Aritomo, D., Watanabe, C., Matsubara, M., Morishima, A.: A privacy-preserving similarity search scheme over encrypted word embeddings. In: Proceedings of IIWAS, New York, NY, USA, pp. 403–412 (2019)
20. Li, M., Wang, G., Liu, S., Yu, J.: Multi-keyword fuzzy search over encrypted cloud storage data. Procedia Comput. Sci. **187**, 365–370 (2021)