Soft Computing (2022) 26:8949–8960
https://doi.org/10.1007/s00500-022-07292-5

FOUNDATIONS

Specified keywords search scheme for EHR sharing

Shufen Niu[1] · Fei Yu[1] · Mi Song[1] · Song Han[1] · Caifen Wang[2]

Accepted: 1 June 2022 / Published online: 25 July 2022

Abstract
Searchable encryption allows data users to search for encrypted files by keywords without restriction. However, electronic health record (EHR) contains sensitive information, and data users should search for and share EHR with restriction. If data users are not restricted when EHR is searched and shared, there is a high risk that EHR will be misused and reveal large amounts of private patient information. This paper proposes a specified keywords search scheme for EHR sharing based on searchable encryption and proxy re-encryption to address this problem. In the scheme, the data user searches with the keywords specified by the doctor and obtains EHR from the medical cloud. Proxy re-encryption is used to implement the sharing of EHR and privacy preservation securely. The security proof demonstrates that our scheme is secure against chosen keyword attack. Furthermore, the experimental results show that the scheme achieves computational efficiency

Keywords Electronic health record · Searchable encryption · Specified keywords · Conjunctive keywords

1 Introduction

With the development of information technology, more and more hospitals use EHR (Heart et al. 2017) to record treatment information. The use of EHR reduces the burden on doctors and brings convenience to clinical work. Moreover, compared with the traditional medical record, EHR reduces the cost of storage and can be stored for a long time. In the process of patient diagnosis, the transmission of EHR between doctors or hospitals is inevitable. Since the EHR contains patients' private information, the EHR must be encrypted, and it is necessary to focus on data security during EHR sharing(Riad et al. 2019; Chi et al. 2019; Gautam et al. 2019). Therefore, how to retrieve and share encrypted EHR while ensuring data security and privacy preservation is an essential research direction.

Searchable encryption and proxy re-encryption can effectively ensure the security of encrypted data search and sharing(Wu et al. 2016; Yu et al. 2021; Chen et al. 2021). Searchable symmetric encryption (SSE) scheme was proposed by Song et al. (2000). The scheme realizes the search for encrypted data, but the key management is complicated. In 2004, Boneh et al. (2004) introduced the public key cryptosystem into the searchable encryption and proposed the first public key encryption with keyword search scheme (PEKS). Still, the scheme can only be used under the secure channel. Baek et al. (2008) proposed a public key searchable encryption scheme under public channel. Currently, public key searchable encryption is widely used in a variety of scenarios(Jiang et al. 2018; Hassan et al. 2019; Xu et al. 2020). Blaze et al. (1998) put forward the proxy re-encryption (PRE). Shao et al. (2010) proposed a proxy re-encryption scheme with a keyword search by combining searchable encryption and proxy re-encryption, which achieves search and sharing of ciphertext.

Many schemes for searching and sharing EHR based on searchable encryption and proxy re-encryption have been

Mi Song, Song Han and Caifen Wang have contributed equally to this work.

✉ Fei Yu
yf_1997163@163.com

Shufen Niu
sfniu76@nwnu.edu.cn

Mi Song
1744391811@qq.com

Song Han
565904313@qq.com

Caifen Wang
wangcf@nwnu.edu.cn

[1] College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, Gansu, China

[2] College of Data and Internet, Shenzhen University of Technology, Shenzhen 518118, Guangzhou, China

proposed in recent years. Wang et al. (2019) proposed a scheme for sharing EHR among multi-institution based on cloud. The scheme supports conjunctive keyword search and guarantees access legitimacy through an identity-based access control mechanism. To reduce the overhead, Ying et al. (2018) provided a cloud-based lightweight EHR sharing scheme. Ciphertext policy attribute-based encryption (Bethencourt et al. 2007) is used to achieve fine-grained access control of EHR in the scheme. With the development of blockchain (Nakamoto 2008), some scholars apply it to the EHR searching and sharing scheme. Wang et al. (2019) proposed a cloud-assisted EHR data sharing scheme via blockchain. The scheme stores keyword index and EHR ciphertext in blockchain and cloud, respectively. Searchable encryption and proxy re-encryption are used to realize data sharing security and privacy preservation. Niu et al. (2020) introduced a blockchain-based EHR sharing scheme that implements multi-keyword search and uses attribute-based encryption to ensure the confidentiality and fine-grained access control of EHR. For the integrity and access control of EHR, a security protocol for a cloud-assisted EHR system via blockchain is proposed by Kim et al. (2020). In (Qin et al. 2021), a secure sharing scheme for EHR based on consortium blockchain is presented to achieve controllable sharing and precise search of EHR. Dagher et al. (2018) introduced a blockchain-based EHR sharing framework. Under this framework, patients, data providers, and third parties can access EHR safely and efficiently. Chen et al. (2019) proposed a blockchain-based searchable encryption scheme for EHR, and the scheme ensures the traceability and integrity of the index using blockchain. In the above schemes, when searching EHR, either the data owner generates a trapdoor for the data user, or the data user constructs a trapdoor by himself. However, the former will result in low search autonomy, while the latter can search unrestricted, which is not conducive to privacy preservation.

Considering that EHR contains patient privacy and sensitive information, EHR should be restricted from searching and sharing to achieve data security and privacy protection. We propose a specified keywords search scheme for EHR sharing. In our scheme, the doctor specifies the keywords for the data user, and the data user can only search with the keywords specified. Therefore, the data user can only obtain EHR containing keywords specified by the doctor, preventing malicious data users from abusing EHR data. Multi-keyword search and keyword access control, privacy preservation, and EHR data security are implemented in our scheme. The contributions of our scheme are as follows.

- The keyword index is generated using searchable encryption and stored in the public cloud. The data user can make conjunctive keyword search in the cloud, which supports multi-keyword search and reduces the communication overhead of search.
- Searchable encryption and proxy re-encryption are used to realize the search of specified keywords and the secure sharing of ciphertext, respectively, protecting patient privacy and ensuring EHR security.
- Before the data user obtains data from the medical cloud, the medical cloud verifies the identity of the data user to determine its legitimacy and ensure the security of EHR data.

The paper is organized as follows: Sect. 2 reviews preliminaries. Section 3 presents the system model. Section 4 provides the algorithm framework and security model. Section 5 describes the scheme construction and security proof in detail. The experiments and efficiency analyses are performed in Sect. 6. Finally, Sect. 7 concludes our work.

2 Preliminaries

In this section, we review the preliminaries required for this paper.

2.1 Bilinear pairing

Let G_1 and G_2 be cyclic groups of prime order p, a bilinear pairing $e : G_1 \times G_1 \rightarrow G_2$ must satisfy the following properties.

1. Bilinear: For any $x, y \in G_1$ and $a, b \in Z_p^*$, we have $e(x^a, y^b) = e(x, y)^{ab}$.
2. Non-degenerate: $x, y \in G_1$ exists such that $e(x, y) \neq 1$.
3. Computable: For any $x, y \in G_1$, an efficient algorithm computes $e(x, y)$.

2.2 Complexity assumptions

Definition 1 *DDH Assumption* (Bao et al. 2003) For any $a, b \in Z_p^*$, given a tuple $(g, g^a, g^b, T \in G_1)$ as input. We define an algorithm $\mathcal{A}$ has advantage ε as $|Pr[\mathcal{A}(g, g^a, g^b, g^{ab}) = 1] - Pr[\mathcal{A}(g, g^a, g^b, T) = 1]|$. We say that the decisional Diffie–Hellman assumption holds if advantage ε is negligible for all probabilistic polynomial time $\mathcal{A}$.

Definition 2 *q-ABDHE Assumption* (Fang et al. 2009) For any $x, z \in Z_p^*$, given a tuple $(g, g^x, g^{x^2}, \cdots, g^{x^q}, g^z, g^{zx^{q+2}} \in G_1, e(g, g)^{zx^{q+1}}, T \in G_2)$ as input. We define an algorithm $\mathcal{A}$ has advantage ε as $|Pr[\mathcal{A}(g, g^x, g^{x^2}, \cdots, g^{x^q}, g^z, g^{zx^{q+2}}, e(g, g)^{zx^{q+1}}) = 1] - Pr[\mathcal{A}(g, g^x, g^{x^2}, \cdots, g^{x^q}, g^z, g^{zx^{q+2}}, T) = 1]|$. We say that the augmented bilinear

decisional Diffie–Hellman exponent assumption holds if advantage ε is negligible for all probabilistic polynomial time $\mathcal{A}$.

3 System model

This section presents the system model of the specified keywords search scheme for EHR sharing.

There are five entities in the system: patient, doctor, data user, public cloud, and medical cloud, as shown in Fig. 1. The five entities are described as follows.

Patient The patient needs to register and obtain a unique identity (ID_i) from the registration system. Before interacting with the doctor, the patient shows ID_i to the doctor as an authorization to manage EHR. When the patient needs an EHR, they generate a trapdoor to search for the EHR ciphertext storage address ($address$) on the public cloud and obtain the EHR ciphertext from the medical cloud through the $address$ and ID_i.

Doctor The doctor generates an EHR for the patient and extracts keywords. The EHR is encrypted and stored in the medical cloud. Meanwhile, keywords are encrypted by doctors with searchable encryption and stored in the public cloud. When a data user requests an EHR, the doctor specifies keywords for the data user. In addition, the doctor acts as a trusted third party to generate proxy re-encryption keys for the data user.

Data user The data user generates a trapdoor to search within the keywords specified by the doctor. Then, the data user gets $address$ from the public cloud. Then, the data user sends the $address$ and search token to the medical cloud to obtain the EHR ciphertext.

Public cloud The public cloud is used to store keyword index. The public cloud is highly open and can be accessed by any public cloud user, but only patients and authorized data users can search in the public cloud. Before data users search, the public cloud acts as a proxy to re-encrypt the keyword index.

Medical cloud The medical cloud is used to store EHR ciphertext and return $address$ to the doctor. The medical cloud is exclusive and can be accessed by authenticated users. When the EHR is shared, the medical cloud acts as a proxy to re-encrypt the EHR ciphertext.

4 Scheme framework and security model

In this section, we provide the scheme framework and security model.

4.1 Scheme framework

In the scheme, we consider two different cases of EHR search and sharing (Case I and Case II). Case I describes searching and obtaining EHR by the patient, including eight polynomial algorithms. Case II describes searching and sharing EHR by the data user, including nine polynomial algorithms. The definition of each polynomial algorithm is as follows.

- $Setup(\lambda) \rightarrow PP$: The algorithm takes a security parameter λ as input, and outputs the public parameters PP.
- $KeyGen(PP) \rightarrow (sk, pk)$: The algorithm inputs PP and outputs the public/private key pairs (sk, pk) for the patient, doctor and data user.
- $Enc(PP, m, sk_d, pk_p) \rightarrow C_m$: Given PP, an EHR m, the doctor's private key sk_d, and the patient's public key pk_p, the algorithm encrypts the EHR and outputs the EHR ciphertext C_m.
- $IndGen(PP, W, sk_d, pk_p) \rightarrow I$: Given PP, the keyword set $W = \{w_1, w_2, \cdots, w_n\}$, sk_d, pk_p, the algorithm outputs the keyword index I.

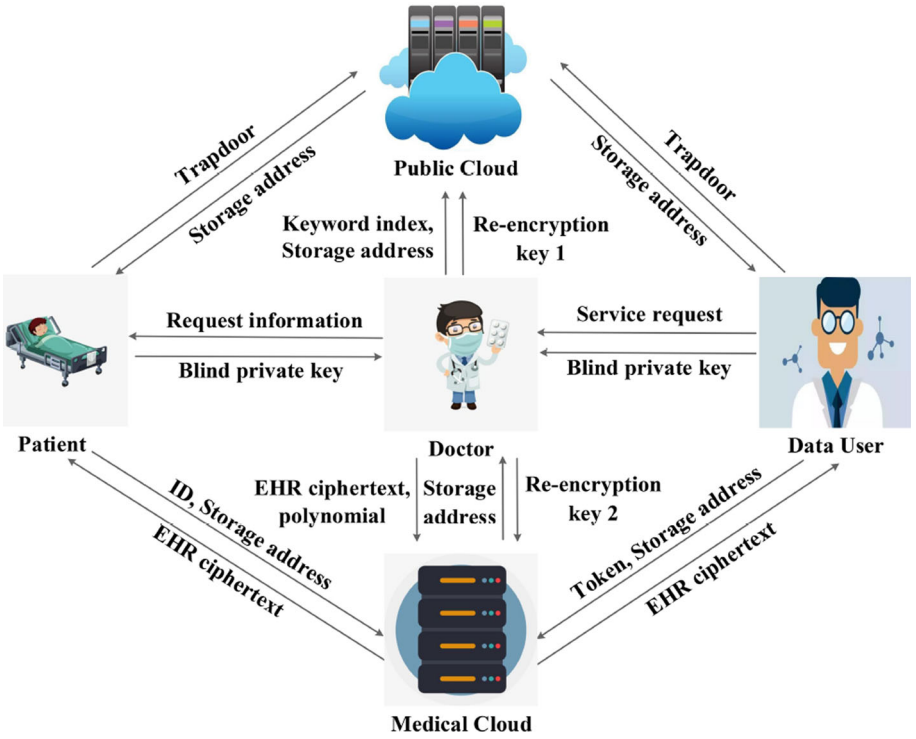
Case I: the patient searches and obtains EHR

- $TrapdoorGen(PP, W', sk_p, pk_d) \rightarrow Tw$: Given PP, the search keyword set $W' = \{w'_1, w'_2, \cdots, w'_l\}$, the patient's private key sk_p, and the doctor's public key pk_d, this algorithm outputs the trapdoor Tw.
- $DataSearch(I, Tw) \rightarrow address \ or \ \perp$: The algorithm takes I and Tw as input, and outputs $address$ or $\perp$.
- $DataAccess(address, ID_i) \rightarrow C_m \ or \ \perp$: The algorithm takes $address$ and ID_i as input, and outputs C_m or $\perp$.
- $Dec(C_m, sk_p) \rightarrow m$: The algorithm takes C_m and sk_p as input, and outputs m.

Case II: the data user searches and obtains EHR

- $IndRe/enc(I, rk_1) \rightarrow I'$: The algorithm inputs I and the re-encryption key rk_1, and outputs the re-encrypted keyword index I'.
- $TrapdoorGen(PP, W'', sk_p, pk_d) \rightarrow Tw'$: The algorithm inputs PP, the specified keyword set $W'' = \{w''_1, w''_2, \cdots, w''_l\}$, the data user's private key sk_d and pk_p, and outputs the trapdoor Tw'.
- $DataSearch(I', Tw') \rightarrow address \ or \ \perp$: The algorithm inputs I' and Tw', and outputs $address$ or $\perp$.
- $DataAccess(address, C_m, rk_2, token) \rightarrow C'_m \ or \ \perp$: The algorithm inputs $address$, C_m, the re-encryption key rk_2, and the token $token$, and outputs the re-encrypted EHR ciphertext C'_m or $\perp$.

Fig. 1 System model



- $Dec(C'_m, sk_u) \rightarrow m$: The algorithm takes C'_m and sk_u as input, and outputs m.

4.2 Security model

The security is based on the DDH assumption and q-ABDHE assumption. Indistinguishability of chosen keyword attack (IND-CKA) game is designed to prove the security. There are two types of attackers in the IND-CKA game.

IND-CKA game Suppose $\mathcal{A}$ is a polynomial time adversary, $\mathcal{B}$ is the challenger, $\mathcal{A}$ and $\mathcal{B}$ play the following two games:

Game 1: $\mathcal{A}$ is assumed to be an outside attacker.

Setup: The algorithms $Setup(\lambda)$ and $KeyGen(PP)$ are executed. Then, $\mathcal{B}$ sends (PP, pk_p, sk_p, pk_d) to $\mathcal{A}$.

Query phase 1: $\mathcal{A}$ makes keyword index query.

Keyword index query: $\mathcal{A}$ can adaptively ask $\mathcal{B}$ for the keyword index for w_i, $\mathcal{B}$ generates the keyword index $I = (R_0, R_1, R_2, I_{w_i})$ as a response.

Challenge: $\mathcal{A}$ outputs a keyword pair (w_0, w_1). (Neither w_0 nor w_1 has been queried in phase 1). $\mathcal{B}$ randomly chooses $\delta \in \{0, 1\}$, let $w^* = w_\delta$, and responds the keyword index $I^* = (R_0^*, R_1^*, R_2^*, I_{w_i^*})$ to $\mathcal{A}$.

Query phase 2: $\mathcal{A}$ continues to adaptively query as in query phase 1, $w_i \neq w_0, w_1$.

Guess: $\mathcal{A}$ outputs $\delta' \in \{0, 1\}$, if $\delta' = \delta$, the adversary $\mathcal{A}$ wins the game.

Game 2: $\mathcal{A}$ is assumed to be an inside attacker.

Setup: The algorithms $Setup(\lambda)$ and $KeyGen(PP)$ are executed. Then, $\mathcal{B}$ sends (PP, pk_p, pk_d) to $\mathcal{A}$.

Query phase 1: $\mathcal{A}$ makes the following queries.

Private key query: $\mathcal{A}$ makes private key query, $\mathcal{B}$ sends sk_d and maintains $List = (sk_d, c)$.

Trapdoor query: $\mathcal{A}$ can adaptively ask $\mathcal{B}$ for the trapdoor for w_i, $\mathcal{B}$ constructs $Tw = (Tw_1, Tw_2)$ as a response.

Challenge: $\mathcal{A}$ outputs a keyword pair (w_0, w_1). (Neither w_0 nor w_1 has been queried in phase 1). $\mathcal{B}$ randomly chooses $\delta \in \{0, 1\}$, let $w^* = w_\delta$, and responds the keyword index $Tw^* = (Tw_1^*, Tw_2^*)$ to $\mathcal{A}$.

Query phase 2: $\mathcal{A}$ continues to adaptively query as in query phase 1, $w_i \neq w_0, w_1$.

Guess: $\mathcal{A}$ outputs $\delta' \in \{0, 1\}$, if $\delta' = \delta$, the adversary $\mathcal{A}$ wins the game.

5 Scheme construction and security proof

This section presents our scheme and the security proof in detail.

5.1 Scheme construction

The scheme comprises three phases: system setup, data generation and storage, data search and access.

- $Setup(\lambda)$: Let λ be the security parameter and (g, p, G_1, G_2, e) be the bilinear pairing. Select a one-way collision-resistant hash function $H : \{0, 1\}^* \rightarrow Z_p^*$. Set $Y =$

$e(g, g)$.

The public parameters are $PP = (g, p, e, G_1, G_2, H, Y)$.

- $KeyGen(PP)$: The patient randomly chooses $sk_p \in Z_p^*$ as the private key, and computes the public key $pk_p = g^{sk_p}$. What is more, the patient needs to register in hospital's system and obtain a unique identity ID_i. The doctor randomly chooses $sk_d \in Z_p^*$ as the private key, and computes the public key $pk_d = g^{sk_d}$. Similarly, the data user randomly chooses $sk_u \in Z_p^*$ as the private key, and computes the public key $pk_u = g^{sk_u}$.

Phase 2: Data generation and storage

- $Enc(PP, m, sk_d, pk_p)$: The patient shows ID_i to the doctor as authorization for the doctor to manage his/her EHR. The doctor generates an EHR m and extracts keywords $W = \{w_1, w_2, \ldots, w_n\}$ after diagnosis. Then, the doctor encrypts the EHR as follows.

– Randomly chooses $\alpha \in Z_p^*$ and calculates $C_1 = mY^{\alpha sk_d}$, $C_2 = pk_p^{\alpha sk_d}$, $h_m = H(m)$ and $C_3 = g^{\alpha h_m}$.
– Computes $h_{ID_i} = H(ID_i)$, $1 \le i \le n$ and constructs a n-degree polynomial $h(x) = (x - h_{ID_1})(x - h_{ID_2}) \cdots (x - h_{ID_n})$.

The EHR ciphertext can be denoted as $C_m = (C_1, C_2, C_3)$. Then, the doctor uploads C_m and $h(x)$ to the medical cloud and obtains $address$.

- $IndGen(PP, W, sk_d, pk_p)$: Moreover, the doctor generates a keyword index of the keyword set $W = \{w_1, w_2, \ldots, w_n\}$ by the following operations.

– Randomly chooses $r \in Z_p^*$ and computes $R_0 = Y^r$, $R_1 = pk_p^r$ and $R_2 = Y^{sk_d r}$.
– Computes $I_i = g^{-rh_{w_i}}$, where $h_{w_i} = H(w_i)$, $1 \le i \le n$.

Let the keyword index $I = (R_0, R_1, R_2, I_i)$. Then, the doctor sends I and $address$ to the public cloud.

Phase 3: Data search and access

Case I: the patient searches for and obtains EHR

When a patient is transferred to another hospital or requests a claim from an insurance company, the patient needs access to the electronic medical record. Case I describes the patient searching and obtaining EHR. The patient generates the trapdoor and sends it to the public cloud. After receiving the trapdoor, the public cloud runs the search algorithm and returns the EHR ciphertext storage address to the patient. The patient sends the address and identity to the medical cloud. If the medical cloud verifies the identity, the patient will obtain the EHR ciphertext. Finally, the patient decrypts the EHR ciphertext to get EHR. The system model of case I is shown in Fig. 2. In addition, we give a flow diagram, as shown in Fig. 3 to make our scheme clearer.

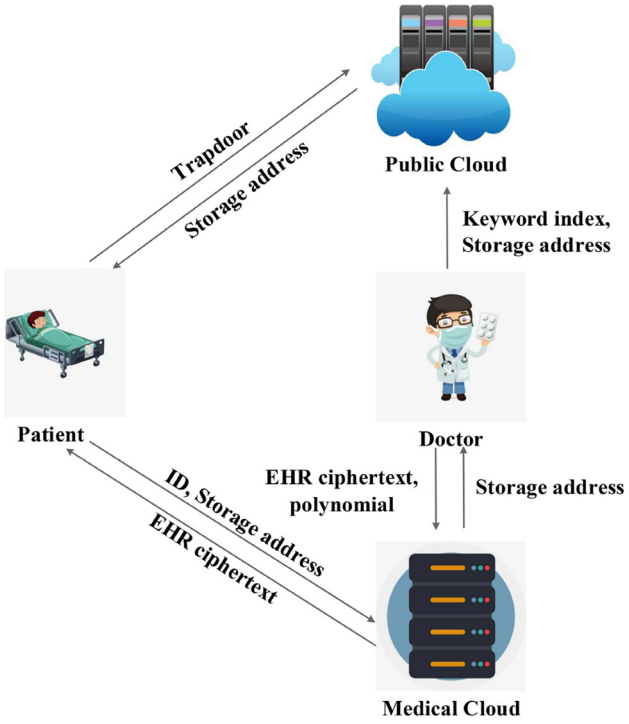


Fig. 2 System model of Case I

- $TrapdoorGen(PP, W', sk_p, pk_d)$: The patient generates the trapdoor of the keyword set $W' = \{w_1', w_2', \ldots, w_l'\}$ that he/she wants to search, where $l < n$.

– Randomly chooses $\varphi \in Z_p^*$ and sets $Tw_1 = \varphi$.
– Computes $Tw_2 = (pk_d g^{-\varphi})^{\frac{1}{sk_p - \sum h'_{w_i}}}$, where $h'_{w_i} = H(w_i')$.

Then, the doctor sends $Tw = (Tw_1, Tw_2)$ to the public cloud.

- $DataSearch(I, Tw)$: After receiving the trapdoor from the patient, the public cloud verifies the equation $e(R_1 \prod I_i, Tw_2) R_0^{Tw_1} = R_2$. If the equation holds, the public cloud returns $address$ to the patient. Otherwise, the public cloud returns $\perp$.

Correctness:

$$e(R_1 \prod I_i, Tw_2) R_0^{Tw_1}$$

$$= e(pk_p^r \prod g^{-rh_{w_i}}, (pk_d g^{-\varphi})^{\frac{1}{sk_p - \sum h'_{w_i}}}) Y^{r\varphi}$$

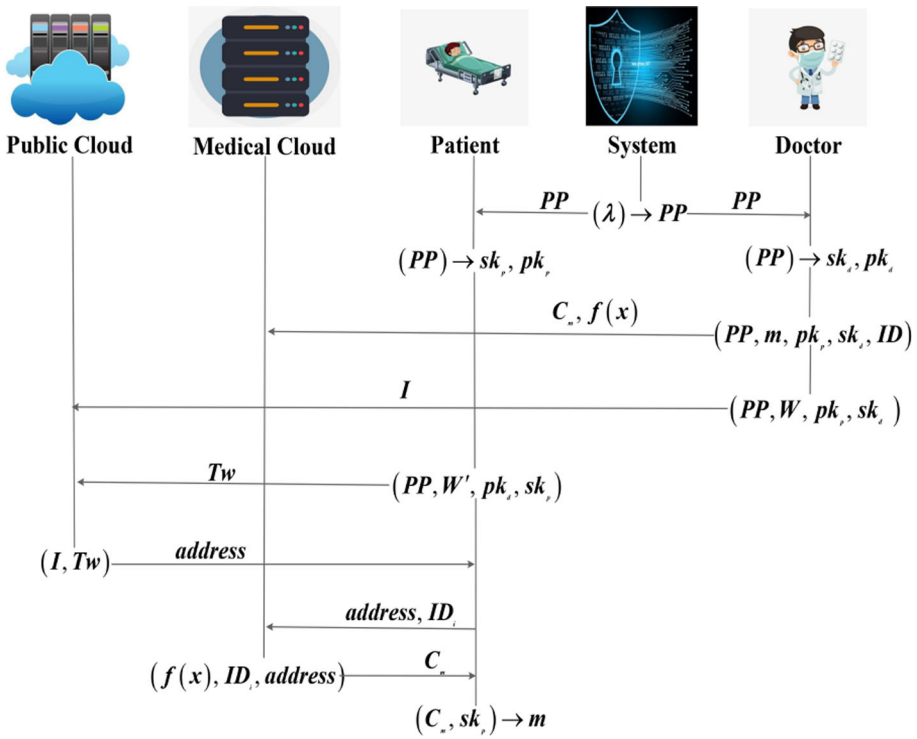


Fig. 3 Flow diagram of algorithms (Case I)

$$
\begin{aligned}
&= e(g^{(sk_p - \sum h_{w_i})r}, (g^{sk_d-\varphi})^{\frac{1}{sk_p - \sum h'_{w_i}}}) Y^{r\varphi} \\
&= e(g^r, g^{(sk_d-\varphi)}) e(g^r, g^{\varphi}) \\
&= e(g^r, g^{sk_d}) \\
&= R_2
\end{aligned}
$$

- $DataAccess(address, ID_i)$: The patient sends ID_i and $address$ to the medical cloud. The medical cloud computes $h_{ID_i} = H(ID_i)$ and verifies the equation $h(h_{ID_i}) = 0$. If the equation holds, the medical cloud returns C_m stored at $address$ to the patient. Otherwise, the medical cloud returns $\perp$.

Correctness: $h(h_{ID_i}) = (h_{ID_i} - h_{ID_1})(h_{ID_i} - h_{ID_2}) \cdots (h_{ID_i} - h_{ID_n}) = 0$

- $Dec(C_m, sk_p)$: The patient calculates $C_1/e(C_2, g^{1/sk_p})$ to get m and checks whether the equation $e(g^{h_m sk_p}, C_2) = e(C_3, pk_d)$ holds or not. If the equation holds, the decryption is correct.

Correctness:

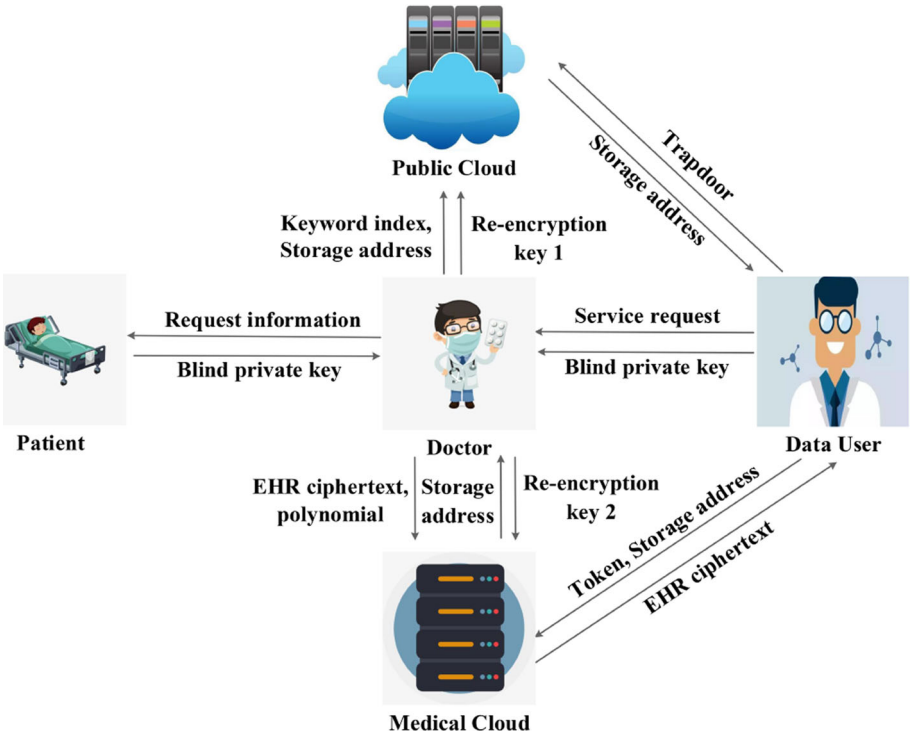
$$
\begin{aligned}
&C_1/e(C_2, g^{1/sk_p}) \\
&= mY^{\alpha sk_d}/e(pk_p^{\alpha sk_d}, g^{1/sk_p}) \\
&= mY^{\alpha sk_d}/Y^{\alpha sk_d} \\
&= m
\end{aligned}
$$

$$
\begin{aligned}
&e(g^{h_m sk_p, C_2}) \\
&= e(g^{h_m sk_p}, pk_p^{\alpha sk_d}) \\
&= e(g^{\alpha h_m}, g^{sk_p sk_d/sk_p}) \\
&= e(C_3, pk_d)
\end{aligned}
$$

Case II: the data user searches for and shares EHR

When a research institute or pharmaceutical company investigates the symptoms of a certain disease (e.g., COVID-19) to formulate treatment plans better, the research institute or pharmaceutical company needs to search and share the EHR for the specified keywords as a data user. Case II describes the data user searching and sharing EHR. When the data user requests to share EHR, the doctor specifies a set of search keywords for the data user. For example, suppose the data user requests to share EHR about COVID-19. In that case, the doctor specifies the search keywords such as "nasal congestion", "headache", "cough", "fever", and other COVID-19 related keywords. Still, the data user cannot search for "stomachache", "heart disease", and other non-COVID-19 related keywords. The doctor, the patient and the data user interact to generate proxy re-encryption keys $rk_1 = sk_u/sk_p$ and $rk_2 = g^{rk_1}$. The doctor uploads rk_1 to the public cloud. Then, the public cloud re-encrypts the keyword index. The data user generates the trapdoor and sends it to the public cloud. After receiving the trapdoor, the public cloud runs the search algorithm and returns the EHR ciphertext storage address to the data user. The doctor sends rk_2 to the medical cloud, and the medical cloud re-encrypts the

Fig. 4 System model of Case II



EHR ciphertext. At the same time, the data user computes the token and sends it and the address to the medical cloud. If the medical cloud verifies the token, the data user will obtain the re-encrypted EHR ciphertext. Finally, the data user decrypts the ciphertext to get EHR. The system model of case II is shown in Fig. 4. In addition, we give a flow diagram, as shown in Fig. 5 to make our scheme clearer.

- $IndRe/enc(I, rk_1)$: The doctor uploads rk_1 to the public cloud, then the public cloud computes $R_1' = R_1^{rk_1}$. Let the new keyword index $I' = (R_0, R_1', R_2, I_i)$.
- $TrapdoorGen(PP, W'', sk_p, pk_d)$: For the specified keyword set $W'' = (w_1'', w_2'', \cdots, w_l'')$, the data user constructs the trapdoor.

– Randomly chooses $\varphi' \in Z_p^*$ and sets $Tw_1' = \varphi'$.

– Computes $Tw_2' = (pk_d g^{-\varphi'})^{\frac{1}{sk_u - \sum h_{w_i}''}}$, where $h_{w_i}'' = H(w_i'')$.

The data user sends $Tw' = (Tw_1', Tw_2')$ to the public cloud.

- $DataSearch(I', Tw')$: Upon receiving the trapdoor, the public cloud checks whether the equation $e(R_1' \prod I_i, Tw_2') R_0^{Tw_1'} = R_2$ holds or not. If the equation holds, the public cloud returns $address$ to the patient. Otherwise, the public cloud returns $\perp$.

Correctness:

$$
\begin{aligned}
& e(R_1' \prod I_i, Tw_2') R_0^{Tw_1'} \\
&= e(pk_u^r \prod g^{-rh_{w_i}}, (pk_d g^{-\varphi'})^{\frac{1}{sk_u - \sum h_{w_i}''}}) Y^{r\varphi'} \\
&= e(g^{(sk_u - \sum h_{w_i})r}, (g^{sk_d - \varphi'})^{\frac{1}{sk_u - \sum h_{w_i}''}}) Y^{r\varphi'} \\
&= e(g^r, g^{(sk_d - \varphi')}) e(g^r, g^{\varphi'}) \\
&= e(g^r, g^{sk_d}) \\
&= R_2
\end{aligned}
$$

- $DataAccess(address, C_m, rk_2, token)$: The doctor sends rk_2 to the medical cloud. At the same time, the data user computes $token = pk_p^{1/sk_u}$ and sends $token$ and $address$ to the medical cloud. Upon receiving $token$ and $address$, the medical cloud verifies the equation $e(rk_2, token) = Y$. If the equation holds, the medical cloud computes $C_2' = e(C_2, rk_2)$ to re-encrypt the EHR ciphertext stored at $address$. Then, the medical cloud returns the re-encrypted EHR ciphertext $C_m' = (C_1, C_2', C_3)$ to the data user. Otherwise, the medical cloud returns $\perp$.

Correctness:

$$
\begin{aligned}
& e(rk_2, token) \\
&= e(g^{rk_1}, pk_p^{1/sk_u}) \\
&= e(g^{sk_u/sk_p}, g^{sk_p/sk_u})
\end{aligned}
$$

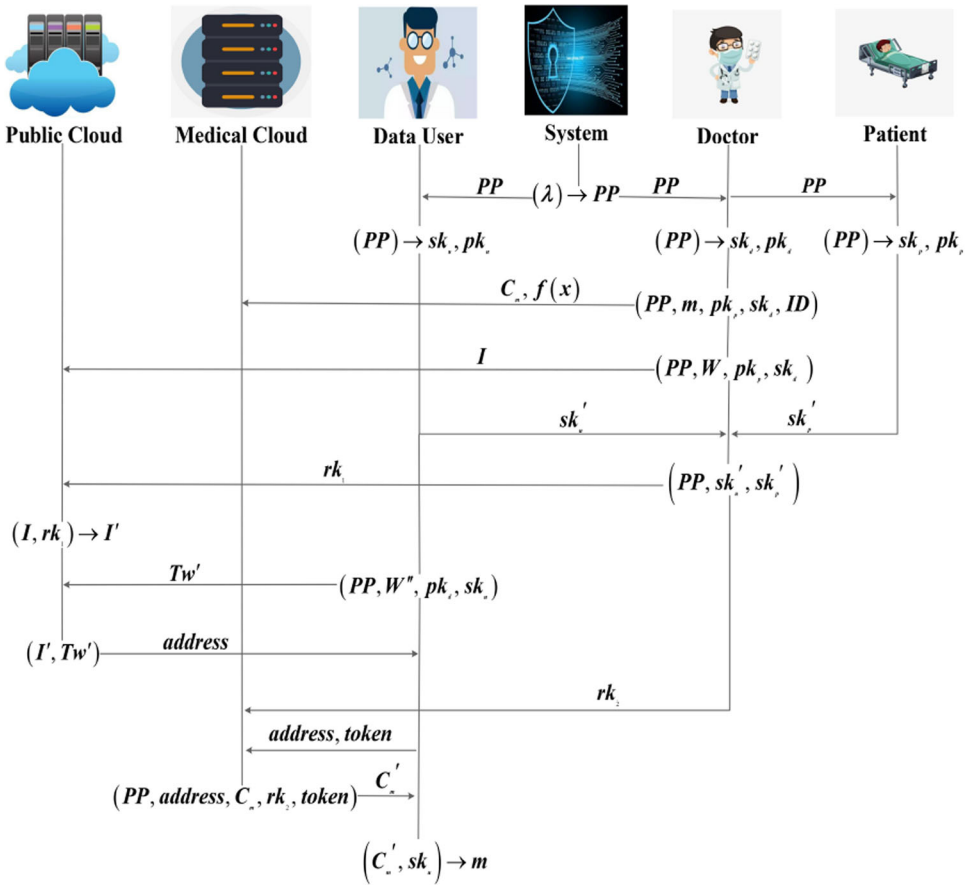


Fig. 5 Flow diagram of algorithms (Case II)

$= e(g, g)$

$= Y$

- $Dec(C'_m, sk_u)$: The data user computes $C_1/C_2'^{1/sk_u}$ to get m and verifies the equation $C_2'^{h_m}/e(C_3^{sk_u}, pk_d) = 1$. If the equation holds, the decryption is correct.

Correctness:

$$
\begin{aligned}
&C_1/C_2'^{1/sk_u} \\
&= mY^{\alpha sk_d}/e(g^{\alpha sk_d}, g^{sk_u})^{1/sk_u} \\
&= mY^{\alpha sk_d}/Y^{\alpha sk_d} \\
&= m \\
&C_2'^{h_m}/e(C_3^{sk_u}, pk_d) \\
&= e(g^{\alpha sk_d}, g^{sk_u})^{h_m}/e(g^{\alpha h_m sk_u}, g^{sk_d}) \\
&= e(g, g)^{\alpha sk_d sk_u h_m}/e(g, g)^{\alpha sk_d sk_u h_m} \\
&= 1
\end{aligned}
$$

5.2 Security proof

In this subsection, we analyze security. The analysis of games is as follows.

Theorem 1 *If the DDH assumption and q-ABDHE assumption hold, the scheme is secure against a chosen keyword attacks.*

Lemma 1 *The proposed scheme is secure against a chosen keyword attack in Game 1, assuming the DDH problem is intractable.*

Proof Suppose there exists a polynomial time adversary $\mathcal{A}$ in Game 1, which can attack our scheme. We build a challenger $\mathcal{B}$ that can solve the DDH problem.

$\mathcal{B}$ inputs a DDH instance $(g, A = g^a, B = g^b, T)$, and has to distinguish $T = g^{ab}$. □

Setup: Let λ be the security parameter and (g, p, G_1, G_2, e) be the bilinear pairing parameters. Select a one-way collision-resistant hash function $H : \{0,1\}^* \rightarrow Z_p^*$. The public parameters are $PP = (g, p, e, G_1, G_2, H, Y)$. $\mathcal{B}$ randomly chooses $sk_d, x \in Z_p^*$ and sets the doctor's public key as $pk_d = g^{sk_d}$. Let $pk_p = A^x = g^{ax}$ and $sk_p = x$ as the patient's public key and private key. Then, $\mathcal{B}$ sends (PP, pk_p, sk_p, pk_d) to $\mathcal{A}$.

Query phase 1: $\mathcal{A}$ makes keyword index query.

Keyword index query: $\mathcal{A}$ makes keyword index query on the keyword on w_i. $\mathcal{B}$ randomly chooses $r \in Z_p^*$ and computes

$R_0 = e(g,g)^r$, $R_1 = pk_p^r$, $R_2 = e(g,g)^{sk_d r}$ and $I_{w_i} = g^{-rh_{w_i}}$, $h_{w_i} = H(w_i)$. $\mathcal{B}$ sends $I = (R_0, R_1, R_2, I_{w_i})$ to $\mathcal{A}$.
Challenge: $\mathcal{A}$ outputs a keyword pair (w_0, w_1). $\mathcal{B}$ randomly chooses $\delta \in \{0,1\}$, let $w^* = w_\delta$ and computes

$$R_0^* = e(g,B)^{1/x},$$
$$R_1^* = T,$$
$$R_2^* = e(g,B)^{sk_d/x},$$
$$I_{w^*} = B^{-h_{w^*}/x}, h_{w^*} = H(w^*).$$

Then, $\mathcal{B}$ responds $I^* = (R_0^*, R_1^*, R_2^*, I_{w_i^*})$ to $\mathcal{A}$.
Query phase 2: $\mathcal{A}$ continues to adaptively query as in query phase 1, $w_i \neq w_0, w_1$.
Guess: $\mathcal{A}$ outputs $\delta' \in \{0,1\}$, if $\delta' = \delta$, then output 1 meaning $T = g^{ab}$; else output 0 meaning $T \neq g^{ab}$.

Lemma 2 *The proposed scheme is secure against a chosen keyword attack in Game 2, assuming q-ABDHE problem is intractable.*

Proof Suppose there exists a polynomial time adversary $\mathcal{A}$ in Game 2, which can attack our scheme. We build a challenger $\mathcal{B}$ that can solve the q-ABDHE problem. □

Let q_k is the number of trapdoor queries, $q > q_k + 1$. $\mathcal{B}$ inputs a q-ABDHE instance $(g, g^x, g^{x^2}, \cdots, g^{x^q}, g^z, g^{zx^{q+2}}, T)$ and has to distinguish $T = e(g,g)^{zx^{q+1}}$.
Setup: Let λ be the security parameter and (g, p, G_1, G_2, e) be the bilinear pairing parameters. Select a one-way collision-resistant hash function $H : \{0,1\}^* \to Z_p^*$. The public parameters are $PP = (g, p, e, G_1, G_2, H, Y)$. Choose a random degree q polynomial $f(x)$. $\mathcal{B}$ sets $pk_p = g^x$ and $pk_d = g^{f(x)}$ as the patient's public key and the doctor's public key, respectively. Then, $\mathcal{B}$ sends (PP, pk_p, pk_d) to $\mathcal{A}$.
Query phase 1: $\mathcal{A}$ makes the following queries.
Private key query: $\mathcal{B}$ maintains $List = (sk_d, c)$. $\mathcal{A}$ makes private key query. $\mathcal{B}$ checks if sk_d is in the $List$. If sk_d exists, $\mathcal{B}$ returns sk_d to $\mathcal{A}$, or randomly chooses $c = \{0,1\}$. If $c = 1$, $\mathcal{B}$ returns $sk_d = f(x)$ to $\mathcal{A}$. Otherwise, $\mathcal{B}$ outputs a random number and aborts.
Trapdoor query: When $\mathcal{A}$ makes trapdoor query on the keyword w_i, $\mathcal{B}$ queries $List$. If $c = 0$, $\mathcal{B}$ sets $Tw_1 = f(h_{w_i})$ and computes $Tw_2 = g^{(f(x)-f(h_{w_i}))/(x-h_{w_i})}$, where $h_{w_i} = H(w_i)$. Then $\mathcal{B}$ sends $Tw = (Tw_1, Tw_2)$ to $\mathcal{A}$. When $q > q_k + 1$, $Tw_1 = f(h_{w_i})$ is a random value, since $f(x)$ is a random degree q polynomial. Otherwise, $\mathcal{B}$ returns an error message and aborts.
Challenge: $\mathcal{A}$ outputs a keyword pair (w_0, w_1). $\mathcal{B}$ randomly chooses $\delta \in \{0,1\}$, let $w^* = w_\delta$. Then, $\mathcal{B}$ runs the above algorithms to get $List = (sk_d^*, c^*)$. If $c^* = 1$, $\mathcal{B}$ outputs an error message and aborts. Otherwise, $\mathcal{B}$ sets $Tw_1^* = f(h_{w^*})$ and computes $Tw_2^* = g^{(f(x)-f(h_{w^*}))/(x-h_{w^*})}$, where $h_{w^*} = H(w^*)$. Defines the degree $q+1$ polynomial $F^*(x) = (x^{q+2} - h_{w^*}^{(q+2)})/(x - h_{w^*}) = \sum_{i=0}^{q+1}(F_i^* x^i)$. Computes

$$R_0^* = T^{F_{q+1}^*} e\left(g^z, \prod_{i=0}^{q}(g^{x^i})^{F_i^*}\right),$$
$$R_1^* = g^{z\sum_{i=0}^{q+1}(F_i^* x^{i+1})},$$
$$R_2^* = e(R_1 I_{w^*}, Tw_2^*) R_0^{Tw_1^*},$$
$$I_{w^*} = g^{-zF^*(x)h_{w^*}}.$$

$\mathcal{B}$ sends $I^* = (R_0^*, R_1^*, R_2^*, I_{w^*})$ to $\mathcal{A}$. Let $r^* = zF^*(x)$, if $T = e(g,g)^{zx^{q+1}}$, then

$$\begin{aligned} R_0^* &= T^{F_{q+1}^*} e\left(g^z, \prod_{i=0}^{q}(g^{x^i})^{F_i^*}\right) \\ &= e(g,g)^{zF^*(x)} = e(g,g)^{r^*}, \end{aligned}$$
$$R_1^* = g^{z\sum_{i=0}^{q+1}(F_i^* x^{i+1})} = g^{xz\sum_{i=0}^{q+1}(F_i^* x^i)} = g^{xr^*} = pk_p^{r^*},$$
$$R_2^* = e(R_1 I_{w^*}, Tw_2^*) R_0^{Tw_1^*} = e(g,g)^{sk_d r},$$
$$I_{w^*} = g^{-zF^*(x)h_{w^*}} = g^{-r^* h_{w^*}}.$$

Query phase 2: $\mathcal{A}$ continues to adaptively query as in query phase 1, $w_i \neq w_0, w_1$.
Guess: $\mathcal{A}$ outputs $\delta' \in \{0,1\}$, if $\delta' = \delta$, then output 1 meaning $T = e(g,g)^{zx^{q+1}}$; else output 0 meaning $T \neq e(g,g)^{zx^{q+1}}$.

6 Scheme construction and security proof

In this section, we analyze the performance and efficiency of the proposed scheme, and compare the scheme with Wu et al. (2016), Wang et al. (2019), Xue (2022) and Liu et al. (2021).

6.1 Theory analysis

(1) Functionality comparison

The functions of the four schemes are compared, as shown in Table 1. It is evident that all the schemes support secure search. Whereas our scheme and the scheme of Wang et al. (2019) support multi-keyword search, only ours supports specified keywords search.

(2) Comparison of communication overhead

We compare the communication overhead of the four schemes in terms of EHR storage, data search, and data access, as shown in Table 2. We define the element length of G_1, G_2, and Z_p^* as $|G_1|$, $|G_2|$, and $|Q|$. Since the data communicated in each scheme belongs to the elements in G_1, G_2, and Z_p^*, the communication overhead of the scheme can be effectively represented by counting the above parameters. Since the support of conjunctive keyword search, our

Table 1 Comparison of Functionality

	Secure search	Multi-keyword search	Specified keywords search
Wu et al. (2016)	√	×	×
Wang et al. (2019)	√	√	×
Xue (2022)	√	×	×
Ours	√	√	√

Table 2 Comparison of communication overhead

	EHR storage	Data search	Data access
Wu et al. (2016)	$(n+2)\|G_1\|+\|G_2\|$	$2l\|G_1\|$	$2\|G_1\|+\|G_2\|$
Wang et al. (2019)	$(2n+7)\|G_1\|+\|G_2\|$	$(3+l)\|G_1\|$	$4\|G_1\|+2\|G_2\|$
Xue (2022)	$5\|G_1\|+(n+1)\|G_2\|$	$l\|G_1\|$	$7\|G_1\|+2\|G_2\|$
Ours	$(n+4)\|G_1\|+2\|G_2\|$	$\|G_1\|+\|Q\|$	$2\|G_1\|+\|G_2\|$ $\|G_1\|+2\|G_2\|$

Table 3 System configuration and mathematical parameters

CPU	Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz
OS	Linux and Ubuntu10.10
Program language	The C Programming Language
Program library	Pairing-Based Cryptography (PBC)
Pairing	Type A
Elliptic curve	$y^2 = x^3 + x$
Base field	512 bit
Group order	$2^{159} + 2^{107} + 1$

scheme has a low data search communication overhead. Our communication overhead is close to that of Wu et al. (2016) in the EHR storage and data access phase.

6.2 Numerical analysis

We emulate the proposed protocol, Wu et al. (2016), Wang et al. (2019), Xue (2022), and Liu et al. (2021). Liu et al. (2021) is currently a popular attribute-based searchable encryption (ABSE) scheme. Since the encryption and decryption of ABSE are related to the number of attributes, we only compare the search algorithm of Liu et al. (2021). The experiments are implemented using C language and Pairing-Based Cryptography (xxx yyy) on a PC with Linux operating system. The system configuration and mathematical parameters are shown in Table 3. For Pairing, Type A is a common type, and 512 bit is a safe length for base field. The experimental results are shown in Figs. 6, 7, and 8.

Figure 6 shows that the encryption running time of the proposed scheme and Wu et al. (2016) is much less than that of Wang et al. (2019) and Xue (2022). Since the encryption of Wang et al. (2019) and Xue (2022) is related to the number of keywords, the time cost is more as the number of keywords increases. In addition, our encryption time is very

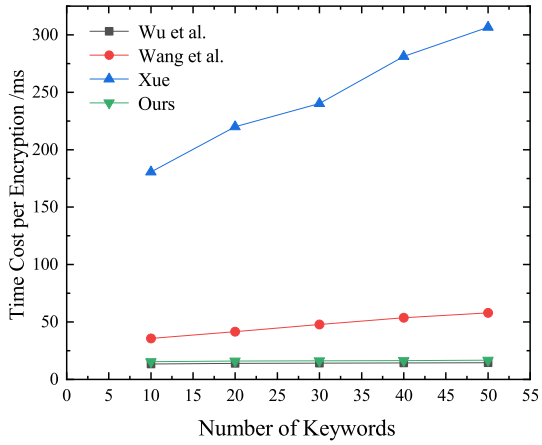


Fig. 6 Time cost of encryption

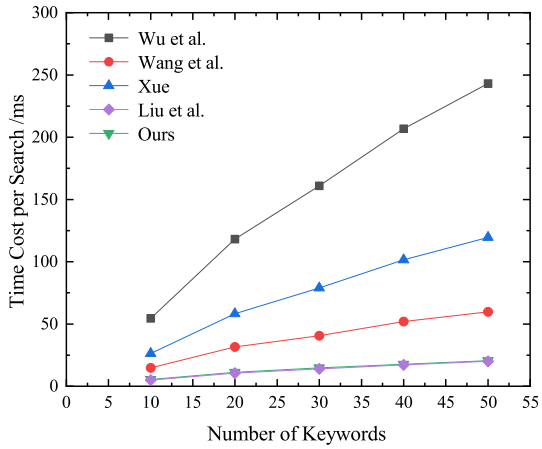


Fig. 7 Time cost of search

close to that of Wu et al. (2016), but a bit longer than Wu et al. (2016).

From Fig. 7, our keyword search algorithm is the most efficient, while Wu et al. (2016) is the lowest. Compared with the more current ABSE (Liu et al. 2021), the search efficiency of

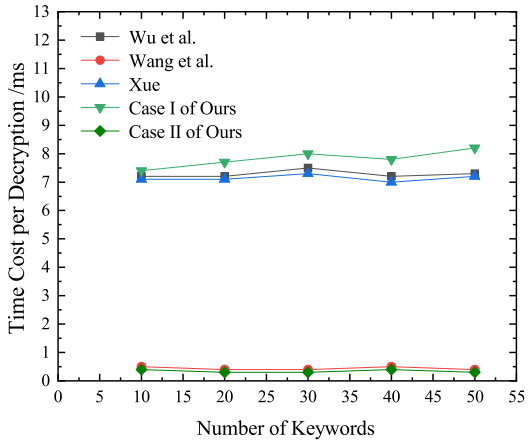


Fig. 8 Time cost of decryption

our scheme is slightly better. With the increase in keywords, our advantage will become more and more apparent, which means that our scheme is suitable for big data search.

In Fig. 8, the decryption time of the Case II is much less than the scheme of Wu et al. (2016) and the scheme of Xue (2022), and the decryption running time of Case I is most. Since the decryption algorithm of the Case II and Wang et al. (2019) contain similar operations, the decryption time cost of Case II is very close to that of Wang et al. (2019).

7 Conclusion

This paper proposes an EHR search and sharing scheme based on searchable encryption with specified keyword search and proxy re-encryption, achieving multi-keyword search and protecting the privacy and EHR security. The scheme is secure against chosen keyword attack. In addition, the experimental results show that the search of the scheme is more effective than other existing schemes. What is not perfect is that the proposed scheme's encryption and decryption running time is slightly more than that of the comparison scheme.Therefore, how to improve the encryption and decryption efficiency while ensuring security is the focus of the subsequent research. For example, in a multiuser scenario, we can consider applying an attribute-based encryption algorithm or broadcast encryption algorithm to the scheme to improve the efficiency of encryption and decryption.

Funding This work was supported by The National Natural Science Foundation of China (No. 61772022).

Data availability Data available on request from the authors.

Declarations

Conflict of interest The authors have not disclosed any competing interests.

References

Baek J, Safavi-Naini R, Susilo W (2008) Public key encryption with keyword search revisited. Computational science and its applications - ICCSA 2008. Perugia, Italy, pp 1249–1259

Bao F, Deng R, Zhu H (2003) Variations of Diffie-Hellman problem. In: information and communications security. ICICS 2003, Huhehaote, China, pp 301-312

Bethencourt J, Sahai A, Waters B (2007) Ciphertext-policy attribute-based encryption. 2007 IEEE symposium on security and privacy (SP '07). CA, USA, Berkeley, pp 321–334

Blaze M, Bleumer G, Strauss M (1998) Divertible protocols and atomic proxy cryptography. Advances in cryptology - EUROCRYPT 98. Espoo, Finland, pp 127–144

Boneh D, Crescenzo G, Ostrovsky R, Persiano G (2004) Public key encryption with keyword search. Advances in cryptology - EUROCRYPT 2004. Interlaken, Switzerland, pp 506–522

Chen L, Lee W, Chang C, Choo K, Zhang N (2019) Blockchain based searchable encryption for electronic health record sharing. Future Gener Comput Syst 95:420–429. https://doi.org/10.1016/j.future.2019.01.018

Chen Z, Wu A, Li Y, Geng S (2021) Blockchain-enabled public key encryption with multi-keyword search in cloud computing. Secur Commun Netw 2021(2):1–11. https://doi.org/10.1155/2021/6619689

Chi T, Qin B, Zheng D (2019) An efficient searchable public-key authenticated encryption for cloud-assisted medical internet of things. Wirel Commun Mob Comput 2020:1–11. https://doi.org/10.1155/2020/8816172

Dagher G, Mohler J, Milojkovic M, Marella P (2018) Ancile: privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. Sustain Cities Soc 2018:S2210670717310685. https://doi.org/10.1016/j.scs.2018.02.014

Fang L, Susilo W, Ge C, Wang J (2009) A secure channel free public key encryption with keyword search scheme without random oracle. In: cryptology and network security. CANS 2009, Kanazawa, Japan, pp 248–258. https://doi.org/10.1007/978-3-642-10433-6_16

Gautam P, Ansari M, Sharma SK (2019) Enhanced security for electronic health care information using obfuscation and RSA algorithm in cloud computing. Int J Inf Secur Priv 13(1):59–69. https://doi.org/10.4018/IJISP.2019010105

Hassan A, Hamza R, Yan H, Li P (2019) An efficient outsourced privacy preserving machine learning scheme with public verifiability. IEEE Access 7:146322–146330. https://doi.org/10.1109/ACCESS.2019.2946202

Heart T, Ben-Assuli O, Shabtai I (2017) A review of PHR, EMR, and EHR integration: a more personalized healthcare and public health policy. Health Policy Technol 6(1):20–25. https://doi.org/10.1016/j.hlpt.2016.08.002

Jiang L, Li T, Li X, Atiquzzaman M, Ahmad H, Wang X (2018) Anonymous communication via anonymous identity-based encryption and its application in IoT. Wirel Commun Mob Comput 2018:1–11. https://doi.org/10.1155/2018/6809796

Kim M, Yu S, Lee J, Park Y, Park Y (2020) Design of secure protocol for cloud-assisted electronic health record system using blockchain. Sensors. https://doi.org/10.3390/s20102913

Liu X, Yang X, Luo Y, Zhang Q (2021) Verifiable multi-keyword search encryption scheme with anonymous key generation for medical internet of things. IEEE Internet Things J 2021:1. https://doi.org/10.1109/JIOT.2021.3056116

Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system. Decentralized Business Review, 21260

Niu S, Chen L, Wang J, Yu F (2020) Electronic health hecord sharing scheme with searchable attribute-based encryption on Blockchain. IEEE Access 8:7195–7204. https://doi.org/10.1109/ACCESS.2019.2959044

Qin Q, Jin B, Liu Y (2021) A secure storage and sharing scheme of stroke electronic medical records based on consortium blockchain. BioMed Res Int 5:1–14. https://doi.org/10.1155/2021/6676171

Riad K, Hamza R, Yan H (2019) Sensitive and energetic IoT access control for managing cloud electronic health records. IEEE Access 7:86384–86393. https://doi.org/10.1109/ACCESS.2019.2926354

Shao J, Cao Z, Liang X, Lin H (2010) Proxy re-encryption with keyword search. Inf Sci 180(13):2576–2587. https://doi.org/10.1016/j.ins.2010.03.026

Song D, Wagner D, Perrig A (2000) Practical techniques for searches on encrypted data. In: proceeding 2000 IEEE symposium on security and privacy. S&P 2000, Berkeley, CA, USA, pp 44-45

The pairing-based cryptography library (2013) http://crypto.stanford.edu/pbc/address

Wang X, Zhang A, Xie X, Ye X (2019) Secure-aware and privacy-preserving electronic health record searching in cloud environment. Int J Commun Syst 32(8):e3925.1-e3925.11. https://doi.org/10.1002/dac.3925

Wang Y, Zhang A, Zhang P, Wang H (2019) Cloud-assisted EHR sharing with security and privacy preservation via consortium blockchain. IEEE Access 7:136704–136719. https://doi.org/10.1109/ACCESS.2019.2943153

Wu Y, Lu X, Su J, Chen P (2016) An efficient searchable encryption against keyword guessing attacks for sharable electronic medical records in cloud-based system. J Med Syst 40(12):258. https://doi.org/10.1007/s10916-016-0609-z

Xu X, He C, Xu Z, Qi L, Wan S, Bhuiyan M (2020) Joint optimization of offloading utility and privacy for edge computing enabled IoT. IEEE Internet of Things J 7(4):2622–2629. https://doi.org/10.1109/JIOT.2019.2944007

Xue L (2022) DSAS: a secure data sharing and authorized searchable framework for e-healthcare system. IEEE Access 10:30779–30791. https://doi.org/10.1109/ACCESS.2022.3153120

Ying Z, Wei L, Li Q, Liu X, Cui J (2018) A lightweight policy preserving EHR sharing scheme in the cloud. IEEE Access 6:53698–53708. https://doi.org/10.1109/ACCESS.2018.2871170

Yu X, Xu C, Dou B, Wang Y (2021) Multi-user search on the encrypted multimedia database: lattice-based searchable encryption scheme with time-controlled proxy re-encryption. Multimed Tools Appl 80(8):1–19. https://doi.org/10.1007/s11042-020-09753-1