


An Efficient Searchable Encryption Against Keyword Guessing Attacks for Sharable Electronic Medical Records in Cloud-based System

Yilun Wu¹  · Xicheng Lu² · Jinshu Su² · Peixin Chen¹

Received: 28 June 2016 / Accepted: 13 September 2016 / Published online: 8 October 2016
© Springer Science+Business Media New York 2016

Abstract Preserving the privacy of electronic medical records (EMRs) is extremely important especially when medical systems adopt cloud services to store patients' electronic medical records. Considering both the privacy and the utilization of EMRs, some medical systems apply searchable encryption to encrypt EMRs and enable authorized users to search over these encrypted records. Since individuals would like to share their EMRs with multiple persons, how to design an efficient searchable encryption for sharable EMRs is still a very challenge work. In this paper, we propose a cost-efficient secure channel free searchable encryption (SCF-PEKS) scheme for sharable EMRs. Comparing with existing SCF-PEKS solutions, our scheme reduces the storage overhead and achieves better computation performance. Moreover, our scheme can guard against keyword guessing attack, which is neglected by most of the existing schemes. Finally, we implement both our scheme and a latest medical-based scheme to evaluate the performance. The evaluation results show that our scheme performs much better performance than the latest one for sharable EMRs.

Keywords Electronic medical record · Searchable encryption · Cloud storage · Secure channels · Privacy · Security

Introduction

In recent years, cloud computing has gained increasing attention because it provides a more convenient and cost-efficient solution for users to manage the data [1]. Due to the tremendous benefits of cloud computing, the electronic medical records (EMRs) providers are willing to deploy their EMRs storage and application services into the cloud instead of maintaining a specialized data center [2]. Since EMRs involve lots of information about patients' privacy, it is important to prevent the contents of EMRs from being revealed to both unauthorized users and the cloud server.

The basic way to protect EMRs from being disclosed to the unauthorized users is user authentication. In current medical information systems, smart card based authentication schemes [3–7] are widely used to verify the correctness of remote users. He et al. proposed an efficient authentication scheme [3] which can be deployed in the mobile cloud environment. Chaturvedi et al. [5] found the security vulnerabilities of previous works, and proposed an improved three-factor remote user authentication scheme. Khan et al. [7] proposed a more cost-efficient scheme which can defend both active attacks and passive attacks. Although the user authentication is a reliable technique to protect EMRs, it is hard for the authentication schemes to keep the cloud server from accessing EMRs.

To prevent the sensitive information of EMRs from being revealed to the cloud server, patients/practitioners are willing to encrypt EMRs [8]. Encrypting EMRs before

This article is part of the Topical Collection on *Systems-Level Quality Improvement*

✉ Yilun Wu
yl.wu@nudt.edu.cn

¹ College of Computer, National University of Defense Technology, Changsha, Hunan, People's Republic of China

² National Key Laboratory for Parallel and Distributed Processing, National University of Defense Technology, Changsha, Hunan, People's Republic of China

outsourcing would be a reliable solution for protecting EMRs, but it makes data utilization, such as keyword search, a very challenge task [9]. To solve this problem, Boneh et al. [10] have designed the first public key encryption based searchable encryption (PEKS) to allow users to search over the encrypted data. Hereafter, many follow-up schemes [11–13] have been proposed to enrich the functionalities of the searchable encryption. With PEKS, users can quickly sort out the information of interest from a large amount of data without leaking sensitive information to the cloud server. Recently, some cloud-based EMRs systems [14, 15] have applied PEKS to build a secure storage environment.

All aforementioned PEKS schemes require secure channels to transmit some sensitive information. Otherwise, a potential eavesdropper can easily get the sensitive information and break the system. To solve the secure channel problem, Beak et al. [16] have proposed a novel PEKS scheme, referred to as SCF-PEKS, which guarantees the secure keyword search without secure channels. Following this work, Rhee et al. have designed two SCF-PEKS schemes [17, 18] based on an enhanced security model. To improve efficiency, Gu et al. [19] have proposed a productive solution that requires no pairing operation in the encryption procedure. In addition, Fang et al. [20] proposed a new scheme, whose security does not rely on random oracles. And afterwards, Fang et al. proposed the improved scheme [21] which is secure against keyword guessing attack (IND-KGA). In 2015, Guo et al. [22] proposed an very efficient SCF-PEKS scheme that is practical to deployed in the cloud-based EMRs system.

However, existing SCF-PEKS solutions only consider the scenario with the one-receiver setting. In other words, these solutions assume that there is only one receiver in the system. In reality, one patient/practitioner would like to share her electronic medical record with a wide range of users [14]. For example, a patient may share her electronic medical record with her family, her friends or her practitioner. And, with the consent of the patient, a practitioner may share the electronic medical record with other practitioners to discuss the rehabilitation program. In the existing SCF-PEKS schemes, the EMR owner, referred to as the sender in this paper, has to encrypt each keyword for each receiver. If many receivers are authorized to search over the sender's EMR involving many keywords, it would incur noticeable overhead in terms of the computation overhead and the storage overhead. For example, if there are fifty authorized receivers can search over the sender's EMR which contains fifty keywords, the sender has to generate 2,500 ciphertexts corresponding for these keywords. Even worse, the sender has to outsource these ciphertexts to the cloud server, which will increase the communication overhead. On the

view point of the EMRs provider, more storage space should be rented from the cloud with the increasing consumption of the storage. Hence, a more cost-efficient solution in the multi-receiver setting is required.

In this paper, we propose a novel SCF-PEKS scheme, aiming to reduce both the computation overhead and the storage overhead for sharable EMRs in the multi-receiver setting. In our scheme, the sender only needs to generate one ciphertext for each keyword, no matter how many receivers our scheme has. In addition, our scheme guarantees the IND-KGA secure as the scheme in [22]. Our contributions can be summarized as follow: First, our SCF-PEKS scheme is a low overhead solution which is practical in the cloud. Comparing with existing works, the sender undertakes less computation tasks and costs less storage space in our scheme. Second, our scheme is IND-KGA secure that can guard against the keyword guessing attack. Moreover, our scheme requires no secure channel, meaning that no secret information will be transmitted on channels. Finally, we present a comprehensive comparison between our scheme and some other SCF-PEKS schemes. Our comparison consists of both the theoretical analysis and the performance evaluation. Both of them prove that our scheme is a better solution in the cloud.

The remainder of the paper is organized as follows. In “**Statement**”, we present the system model, the security model, the design objectives and some algorithm definitions. The cryptographic primitives and assumptions are introduced in “**Preliminaries**”. “**Details of our proposed scheme**” gives the details of our proposed scheme, followed by the security analysis and theoretical comparison in “**Analysis**”. The performance is evaluated in “**Performance evaluation**”. Finally, we conclude the paper in “**Conclusion**”.

Statement

System model

As illustrated in Fig. 1, there are three entities involved in our system.

Sender The sender is an entity who has one EMR which will be shared with some authorized users. To protect the privacy of the EMR, the sender should encrypt it before outsourcing. In addition, to enable authorized users to efficiently search over the encrypted EMR, the sender should generate a secure index involving some keywords for the EMR. After that, the sender outsources the encrypted EMR together with the corresponding index to the cloud server. To delegate the search ability to authorized users, the sender

computes a re-encryption key for each authorized user, and sends these keys to the cloud server. The number of the keys depends on the number of authorized users.

Receivers In our scheme, receivers are the authorized users who can perform the keyword search on the sender’s encrypted EMR. Each receiver requests the search by generating a trapdoor associated with a certain query keyword, and obtains the sender’s encrypted EMR from the cloud server if the query keyword is involved in the index.

The cloud server The cloud server is an entity which stores the sender’s EMR, and performs the search operation after receiving the trapdoor from one receiver.

Security model

In this paper, we assume that there is no secure channel in the EMRs system, meaning that each entity is forbidden to transmit any secret information, such as secret keys and trapdoors, via transmission channels. Otherwise, a potential eavesdropper will get the secret information, and try to break the system. Similar to existing work, we consider a semi-trust cloud server, which honestly follows our proposed scheme, but curiously learns the underlying meanings of the sender’s EMR. In other words, the cloud server will try to learn the content of the sender’s EMR by decrypting it. In addition, the cloud server is also interested in retrieving the keywords from the index and the trapdoors.

Design goal

Our goals consists of the following aspects:

- *Security.* First, the confidentiality of the sender’s EMR should be guaranteed in our scheme. The cloud server

cannot retrieve any EMR from the encrypted data. Second, the cloud server cannot learn any keyword from neither the index nor the trapdoors. Third, the trapdoors should not be linkable, which means the trapdoors should be totally different even if they contain the same keyword. In this paper, our scheme should also guard against the keyword guessing attack.

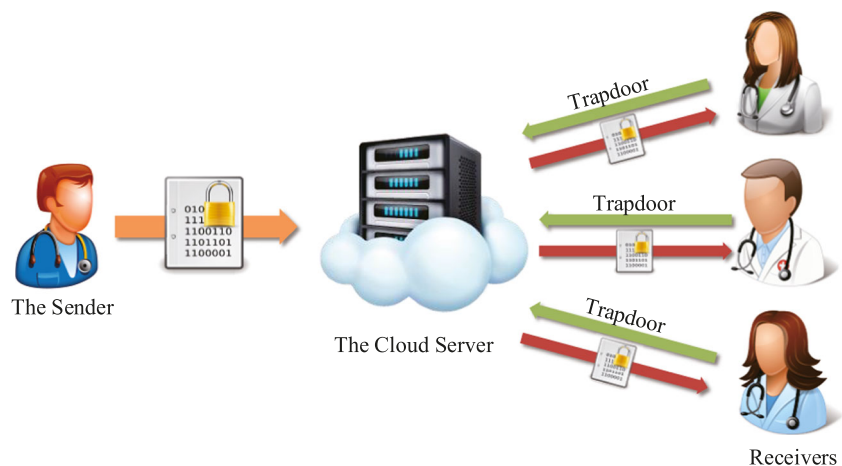
- *Low Storage Overhead.* As the main purpose in this paper, our scheme should require less storage overhead than other SCF-PEKS solutions. Since the data will be outsourced from the sender to the cloud server, reducing the storage overhead is equivalent to reducing the communication overhead.
- *Low Computation Overhead.* The sender should cost acceptable computational resources on generating the index for the EMR and computing the re-encryption key for each receiver. In addition, our scheme should achieve better search efficiency when receivers request the keyword query.

Algorithm definition

We define some algorithms used in our scheme as follows. The detail of each algorithm will be introduced in “[Details of our proposed scheme](#)”.

- $GlobalSetup(\lambda)$: The algorithm takes the security parameters λ as input, and outputs the global parameter \mathcal{GP} .
- $KeyGen(\mathcal{GP})$: Given the global parameter \mathcal{GP} , the algorithm outputs a public/secret key pair (pk, sk) .
- $Enc(\mathcal{GP}, M, sk_S)$: Given the global parameter \mathcal{GP} , an electronic medical record M and a sender’s secret key sk_S , the algorithm encrypts the record and outputs the corresponding ciphertext.
- $IndexGen(\mathcal{GP}, sk_S, \mathcal{W})$: The algorithm inputs the global parameter \mathcal{GP} , a sender’s secret key sk_S and a keyword set \mathcal{W} , outputs the secure index.

Fig. 1 System model



- $\text{ReKeyGen}(\mathcal{GP}, sk_S, pk_R)$: Given the global parameter \mathcal{GP} , a sender’s secret key sk_S and a receiver’s public key pk_R , the algorithm outputs a re-encryption key.
- $\text{Trapdoor}(\mathcal{GP}, sk_R, w')$: Given the global parameter \mathcal{GP} , a receiver’s secret key sk_R and a query keyword w' , the algorithm outputs the trapdoor.
- $\text{Search}(\mathcal{GP}, \mathcal{I}, \mathcal{T}, rk)$: Given the global parameter \mathcal{GP} , the index \mathcal{I} , the trapdoor \mathcal{T} and a re-encryption key rk , the algorithm outputs 1 if $w = w'$, otherwise outputs 0. Noting that keyword w is involved in \mathcal{I} , and query keyword w' is involved in \mathcal{T} .
- $\text{Dec}(\mathcal{GP}, \mathcal{C}, sk_R, rk)$: Given the global parameter \mathcal{GP} , a ciphertext \mathcal{C} , a receiver’s secret key sk_R , and a re-encryption key rk , the algorithm outputs the record if each input parameter is correct.

Preliminaries

Bilinear map

Let \mathbb{G}_1 and \mathbb{G}_2 be two cyclic groups of a large prime p . Let g be a generator of \mathbb{G}_1 . A bilinear map can be defined as $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ if following three conditions hold. 1) *Bilinear*: for any $a, b \in \mathbb{G}_1$, $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$. 2) *Non-degeneracy*: $\hat{e}(g, g) \neq 1$. 3) *Computability*: Given $u, v \in \mathbb{G}_1$, \hat{e} can be efficiently computed.

Assumptions

Let \mathbb{G} be a cyclic group of a large prime p with a generator of g . The following assumptions hold in our scheme.

Divisible Decision Diffie-Hellman (DDDH) assumption [23]: Given (g, g^a, g^b, r) where a, b, r are randomly chosen in \mathbb{Z}_p , we define the advantage function of an adversary \mathcal{A} as:

$$\text{Adv}_{\mathbb{G}_1, \mathcal{A}}^{\text{DDDH}}(\lambda) = |\Pr[\mathcal{A}(g, g^a, g^b, g^{a/b}) = 1] - \Pr[\mathcal{A}(g, g^a, g^b, g^r) = 1]|$$

where, λ is the security parameter. We say that DDDH assumption holds if $\text{Adv}_{\mathbb{G}_1, \mathcal{A}}^{\text{DDDH}}$ is negligible for \mathcal{A} .

Divisible Computation Diffie-Hellman (DCDH) Assumption [23]: Given (g, g^a, g^b) where a, b are randomly chosen in \mathbb{Z}_p , the advantage for an adversary \mathcal{A} to compute $g^{a/b}$ is negligible.

Inverse Computational Diffie-Hellman (InvCDH) assumption [23]: Given (g, g^a) where a are randomly chosen in \mathbb{Z}_p , the advantage for an adversary \mathcal{A} to compute $g^{1/a}$ is negligible.

SCF-PEKS secure against Keyword Guessing Attack (IND-KGA)

In this subsection, we review the definition of SCF-PEKS against keyword guessing attack (IND-KGA) [21]. We first review the IND-KGA game. Let \mathcal{A} be an outside adversary who makes the keyword guessing attack, and \mathcal{B} be a challenger. The security game is defined as follows.

- **Setup**: Both the algorithm $\text{GlobalSetup}(\lambda)$ and the algorithm $\text{KeyGen}(\mathcal{GP})$ are executed by \mathcal{B} . Then the generated \mathcal{GP} and pk_R are given to \mathcal{A} .
- **Query 1**: \mathcal{A} asks \mathcal{B} for the trapdoor for any query keyword from the keyword space. \mathcal{B} responds the trapdoor $\mathcal{T} = \text{Trapdoor}(\mathcal{GP}, sk_R, w)$ to \mathcal{A} .
- **Challenge**: Once Query 1 is over, \mathcal{A} outputs two query keywords (w_0, w_1) , and sends these two keywords to \mathcal{B} . Noting that neither w_0 nor w_1 is queried in Query 1. Upon receiving the keywords, \mathcal{B} chooses a random value $\gamma \in \{0, 1\}$, creates a challenge $\text{Trapdoor}(\mathcal{GP}, sk_R, w_\gamma)$, and sends it to \mathcal{A} .
- **Query 2**: \mathcal{A} continues to request a number of trapdoors as in Query 1. It is worth noting that \mathcal{A} cannot query w_0, w_1 .
- **Guess**: \mathcal{A} outputs the guess γ' , and wins the game if $\gamma' = \gamma$.

The advantage for \mathcal{A} to win IND-KGA game is

$$\text{Adv}_{\mathcal{A}}^{\text{IND-KGA}}(\lambda) = |\Pr[\gamma' = \gamma] - 1/2|. \tag{1}$$

The scheme is said to be IND-KGA secure if the advantage $\text{Adv}_{\mathcal{A}}^{\text{IND-KGA}}(\lambda)$ is negligible.

Details of our proposed scheme

In this section, we present the details of our proposed scheme. Each entity in our scheme invokes at least one of the algorithm mentioned in “Algorithm definition”. Roughly, our scheme can be divided into four main stages: *Initialization, Data Processing, Search and Record Retrieval*.

Initialization

The cloud server runs $\text{GlobalSetup}(\lambda)$ to generate the global parameter \mathcal{GP} , where λ is the security parameter. Specifically, the cloud server takes λ to generate two cyclic groups \mathbb{G}_1 and \mathbb{G}_2 with the same prime order p , having g as a generator of \mathbb{G}_1 . Then the cloud server initializes a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, and chooses a hash function $H : \{0, 1\}^* \rightarrow \mathbb{Z}_p$. In addition the cloud server selects a random value sk_C from \mathbb{Z}_p as the secret key, and computes the corresponding public key $pk_C =$

g^{sk_C} . Thus, the global parameter can be denoted as $\mathcal{GP} = \{\mathbb{G}_1, \mathbb{G}_2, g, p, \hat{e}, H, pk_C\}$.

After that, the sender runs $\text{KeyGen}(\mathcal{GP})$ to generate the public/secret pair. More precisely, the sender randomly chooses a secret value sk_S from \mathbb{Z}_p as the secret key. Then the sender computes $pk_S = g^{1/sk_S}$. Similarly, each receiver $R_i \in \mathcal{R}$, where \mathcal{R} is the receiver set, can generate his secret key $sk_{R_i} \in \mathbb{Z}_p$ and public key $pk_{R_i} = g^{1/sk_{R_i}}$, respectively.

Data processing

In this stage, the sender prepares the necessary data which will be outsourced to the cloud server, including a secure index, the encrypted EMR, and a re-encryption key set. The sender first extracts a keyword set \mathcal{W} from the electronic medical record M . Then the sender runs $\text{IndexGen}(\mathcal{GP}, sk_S, \mathcal{W})$ to generate a secure index. Specifically, for each keyword $w \in \mathcal{W}$, the sender computes:

$$\tau_w = pk_C^{sk_S \cdot H(w)}. \tag{2}$$

Thus, the index can be denoted as $\mathcal{I} = \{\tau_w\}_{w \in \mathcal{W}}$.

In order to encrypt the electronic medical record $M \in \mathbb{G}_2$, the sender runs $\text{Enc}(\mathcal{GP}, M, sk_S)$. More precisely, the sender chooses a random value k from \mathbb{Z}_p , and computes:

$$C_1 = M \oplus \hat{e}(g^{sk_S}, g^k), C_2 = g^k. \tag{3}$$

The encrypted record is denoted as $\mathcal{C} = \{C_1, C_2\}$.

Besides that, the sender should also generate a re-encryption key $rk_{S \rightarrow R_i}$ for each receiver $R_i \in \mathcal{R}$ by invoking the algorithm:

$$\text{ReKeyGen}(\mathcal{GP}, sk_S, pk_{R_i}) : rk_{S \rightarrow R_i} = pk_{R_i}^{sk_S}. \tag{4}$$

The re-encryption key set can be denoted as $\mathcal{RK} = \{rk_{S \rightarrow R_i}\}_{R_i \in \mathcal{R}}$.

Finally, the sender outsources the secure index \mathcal{I} , the encrypted record \mathcal{C} and the re-encryption key set \mathcal{RK} to the cloud server.

Search

To search over the encrypted record \mathcal{C} , one receiver R_i needs to compute the trapdoor for a query keyword w' by invoking $\text{Trapdoor}(\mathcal{GP}, sk_{R_i}, w')$. More specifically, the receiver chooses a random value $r \in \mathbb{Z}_p$, and computes:

$$T_1 = pk_C^r, T_2 = pk_C^{H(w') \cdot r \cdot sk_{R_i}}. \tag{5}$$

Then, the receiver R_i sends the trapdoor $\mathcal{T}_{w'} = \{T_1, T_2\}$ to the cloud server.

Upon receiving the trapdoor, the cloud server performs $\text{Search}(\mathcal{GP}, \mathcal{I}, \mathcal{T}_{w'}, rk_{S \rightarrow R_i})$ to check whether the encrypted record \mathcal{C} involves the keyword w' . Precisely, for each τ_w in \mathcal{I} , the cloud server check if

$$\hat{e}(\tau_w, T_1) = \hat{e}(T_2, rk_{S \rightarrow R_i})^{sk_C}. \tag{6}$$

Search outputs 1 only if Eq. 6 holds, which implies $w = w'$. In that case, the cloud server sends $\{\mathcal{C}, rk_{S \rightarrow R_i}\}$ back to the receiver R_i . Otherwise, sends \perp .

Record retrieval

Once the receiver R_i gets $\{\mathcal{C}, rk_{S \rightarrow R_i}\}$ from the cloud server, he decrypts the ciphertext \mathcal{C} to retrieve the record M by invoking $\text{Dec}(\mathcal{GP}, \mathcal{C}, sk_{R_i}, rk_{S \rightarrow R_i})$. The record can be retrieved as follows:

$$M = C_1 \oplus \hat{e}(rk_{S \rightarrow R_i}, C_2)^{sk_{R_i}}. \tag{7}$$

Analysis

Correctness

We first show that the correctly generated index can be correctly searched if the receiver R_i generates the correct trapdoor. It is equivalent to proving the correctness of Eq. 6. Assume that there is a keyword $w \in \mathcal{W}$ that satisfies $w = w'$, we have:

$$\begin{aligned} \hat{e}(\tau_w, T_1) &= \hat{e}(pk_C^{sk_S \cdot H(w)}, pk_C^r) \\ &= \hat{e}(g^{H(w) \cdot r \cdot sk_C}, g^{sk_S \cdot sk_C}) \\ &= \hat{e}(g^{H(w) \cdot r \cdot sk_C}, g^{\frac{sk_S}{sk_{R_i}} \cdot sk_{R_i} \cdot sk_C}) \\ &= \hat{e}(g^{H(w) \cdot r \cdot sk_{R_i} \cdot sk_C}, g^{sk_S / sk_{R_i} \cdot sk_C}) \\ &= \hat{e}(pk_C^{H(w') \cdot r \cdot sk_{R_i}}, pk_{R_i}^{sk_S \cdot sk_C}) \\ &= \hat{e}(T_2, rk_{S \rightarrow R_i}^{sk_C}). \end{aligned} \tag{8}$$

Then, we show that the receiver R_i can correctly retrieve the record M . It is equivalent to proving the correctness of Eq. 7. We have

$$\begin{aligned} &C_1 \oplus \hat{e}(rk_{S \rightarrow R_i}, C_2)^{sk_{R_i}} \\ &= M \oplus \hat{e}(g^{sk_S}, g^k) \oplus \hat{e}(g^{sk_S / sk_{R_i}}, g^k)^{sk_{R_i}} \\ &= M \oplus \hat{e}(g^{sk_S}, g^k) \oplus \hat{e}(g^{sk_S}, g^k) = M \end{aligned} \tag{9}$$

Therefore, the receiver R_i can successfully perform the keyword search on the encrypted record, and decrypt it.

Security analysis

In this subsection, we analyze the security of our proposed scheme. The security of EMRs and the security of the corresponding keywords are discussed in Theorem 1 and Theorem 2, respectively. We first prove that the cloud server cannot retrieve plaintext of any EMR if both DCDH assumption and InvCDH assumption hold. Then, by constructing two equivalent games, we prove that the keywords in our scheme are secure against keyword guessing attacks in standard model.

Theorem 1 *The electronic medical record M is secure if both DCDH assumption and InvCDH assumption hold in \mathbb{G}_1 .*

Proof As show in Eq. 3, the probability for an adversary \mathcal{A} to decrypt the record is equivalent to computing g^{sk_S} . According to our scheme, the cloud server could obtain $g^{1/sk_S}, g^{sk_S/sk_{R_i}}, g^{1/sk_{R_i}}$. \square

Case 1 As soon as DCDH assumption holds in \mathbb{G}_1 , it is hard for the cloud server to compute g^{sk_S} from g^{1/sk_S} with non-negligible probability.

Case 2 Denote sk_S/sk_{R_i} as a , $g^{sk_{R_i}}$ as b , thus sk_S can be denoted as a/b . According to InvCDH assumption, it is hard for an adversary to compute $g^{a/b}$ from (g, g^a, g^b) with non-negligible probability. Equivalently, the cloud server cannot retrieve g^{sk_S} from $(g, g^{sk_S/sk_{R_i}}, g^{1/sk_{R_i}})$ with non-negligible probability.

In conclusion, the cloud server cannot retrieve the record M from the ciphertext \mathcal{C} with non-negligible probability as soon as both DCDH assumption and InvCDH assumption hold in \mathbb{G}_1 .

Theorem 2 *Our scheme is IND-KGA secure in the standard model, if DDDH assumption holds in \mathbb{G}_1 .*

Proof Since sk_S is not owned by the cloud server, the cloud server cannot retrieve the keywords from the index. Therefore, this theorem is equivalent to proving that the keywords are secure in our scheme.

Suppose there exists a polynomial-time adversary \mathcal{A} in IND-KGA game. We build a simulator \mathcal{B} that can play a DDDH game. Denote pk_C as g_1 . \mathcal{B} inputs a DDDH instance $(A = g_1^a, B = g_1^b, V)$, and tries to distinguish $V = g_1^{a/b}$ from a random element in \mathbb{G}_1 . We construct the following games to prove the security.

Game 1. Let $V = g_1^{a/b}$. Game 1 is essentially the same as IND-KGA game except for the following changes:

- **Setup:** \mathcal{B} chooses a random value $l \in \mathbb{Z}_p$, and sets the receiver R_i 's public key as $pk_{R_i} = B^l = g_1^{b \cdot l}$. Naturally, the receiver R_i 's secret key is $sk_{R_i} = \frac{1}{b \cdot l \cdot sk_C}$. \mathcal{B} send pk_{R_i} to \mathcal{A} .
- **Challenge:** Upon receiving keywords (w_0, w_1) , \mathcal{B} picks a random bit $\gamma \in \{0, 1\}$. Then \mathcal{B} sets $T_1 = A^l$ and $T_2 = V^{H(w_\gamma)}$, respectively. Finally, \mathcal{B} sends the trapdoor $\mathcal{T}_{w_\gamma} = \{T_1, T_2\}$ to \mathcal{A} . \square

Game 1 is equivalent to IND-KGA game only if the generated trapdoor is valid. Let $r' = a \cdot l$, we can have:

$$\begin{aligned} T_1 &= A^l = g_1^{a \cdot l} = g_1^{r'} = pk_C^{r'}, \\ T_2 &= V^{H(w_\gamma)} = g_1^{\frac{a}{b} H(w_\gamma)} = g_1^{\frac{a \cdot l}{b \cdot l} H(w_\gamma)} \\ &= g_1^{H(w_\gamma) \cdot r' \cdot sk_{R_i}} = pk_C^{H(w_\gamma) \cdot r' \cdot sk_{R_i}}. \end{aligned} \tag{10}$$

Thus, Eq. 10 is equivalent to Eq. 6. For \mathcal{A} , Game 1 is equivalent to IND-KGA game. Therefore, the advantage for \mathcal{A} to win Game 1 is:

$$Adv_{\mathcal{A}}^{Game1}(\lambda) = Adv_{\mathcal{A}}^{IND-KGA}(\lambda) \tag{11}$$

Game 2. Game 2 is essentially the same as Game 1 except that the value $V = g_1^{a/b}$ is replaced by a random value $V \in \mathbb{G}_1$. Since V is uniform in \mathbb{G}_1 , we have:

$$Pr[(\gamma' = \gamma)] = \frac{1}{2}. \tag{12}$$

Thus, the advantage for \mathcal{A} to win Game 2 is:

$$Adv_{\mathcal{A}}^{Game2}(\lambda) = |Pr[(\gamma' = \gamma)] - \frac{1}{2}| < \epsilon', \tag{13}$$

where ϵ' is a negligible value.

Since the probability for \mathcal{A} to distinguish Game 1 and Game 2 is equal to the probability to distinguish $g^{a/b}$ and random value, we can have:

$$\begin{aligned} Adv_{\mathcal{A}}^{IND-KGA}(\lambda) &= Adv_{\mathcal{A}}^{Game1}(\lambda) \\ &\leq Adv_{\mathcal{A}}^{Game2}(\lambda) + Adv_{\mathbb{G}_1, \mathcal{A}}^{DDDH}(\lambda) \\ &= \epsilon' + \epsilon = \epsilon, \end{aligned} \tag{14}$$

where ϵ is negligible if DDDH assumption holds in \mathbb{G}_1 . Hence, the advantage for \mathcal{A} to win IND-KGA game is negligible.

Theoretical comparison

Suppose there are n receivers who are authorized to search on the sender's EMR which involves m keywords. Let P denote an pairing operation, E denote an exponentiation operation. Let $|\mathbb{Z}_p|, |\mathbb{G}_1|, |\mathbb{G}_2|$ denote the length of the element in $\mathbb{Z}_p, \mathbb{G}_1$ and \mathbb{G}_2 , respectively. We compare our scheme with Rhee et al.'s scheme [18], Fang et al.'s scheme [21] and Guo et al.'s scheme [22], and show the comparison results in Table 1. Noting that both the scheme [21] and the scheme [22] are optimized in the multi-receiver setting before the comparison. To reduce as much computation overhead as possible, some parameters will be calculated only once in both [21] and [22]. For example, in the scheme [21], the ciphertext C_1 and C_5 will only be calculated once even though the sender invokes PEKS several times.

Table 1 Comparison among Rhee et al. [18], Fang et al. [21], Guo et al. [22] and our scheme

Schemes	Rhee et al. [18]	Fang et al. [21]	Guo et al. [22]	Our scheme
IndexGen+ReKeyGen	$(n+m)E+mP$	$(4+m+3mn)E+(1+m+mn)P$	$(n+2m+3mn)E+mP$	$(n+m)E$
Search	$2E+P$	$5E+3P$	$4E+3P$	$E+2P$
(Index+ReKey) size	$n \mathbb{G}_1 +m\lambda$ -bit	$ \mathbb{Z}_p +(2+mn) \mathbb{G}_1 +(m+mn) \mathbb{G}_2 $	$(n+mn) \mathbb{G}_1 +m\lambda$ -bit	$(m+n) \mathbb{G}_1 $
IND-KGA	Yes	Yes	Yes	Yes

Computation Overhead In our scheme, the computational cost for generating the index is mE . Besides, our scheme needs additional n exponentiation operations to generate the re-encryption key set. Comparing with the other three schemes, it is obvious that the sender in our scheme costs less computational resources in the multi-receiver setting. Since an pairing operation P consumes more computation resource than an exponentiation operation E in general, the scheme [18] performs a little better on Search than ours. But our scheme needs less mP operations on Index-Gen than the scheme [18]. Overall, the performance in our scheme is more efficient than the scheme [18]. Hence, our scheme offers the best computation efficiency when multiple receivers are authorized to search over the sender’s EMR in computation comparison.

Storage Overhead On the view point of the sender, the main additional storage overhead is the index in both

[21] and [22]. Unlike these two schemes, the sender needs extra storage overhead to store the re-encryption set in our scheme. As illustrated in Table 1, the total additional storage overhead in our scheme is $(m+n)|\mathbb{G}_1|$, which is much lower than the schemes in [21] and [22].

Performance evaluation

To show the performance more intuitively, we implement both our scheme and Guo et al.’s scheme [22] in C language using Pairing-Based Cryptographic (PBC) library [24], and compare these two schemes on a computer running Ubuntu Linux with 3.4 GHz Intel Core i3 processor and 4 Gigabyte memory. We adopt the type A elliptic curve with 160-bit group order and 1024-bit field order to build the cryptographic environment. The experiment focuses on evaluating the computation overhead and the storage overhead. Each experimental result is the average value from 10 runs.

Table 2 illustrates the computational performance of both Guo et al.’s scheme and ours. In Table 2(a), we fix the number of receivers as fifty, and conclude that the time for generating the index is linear to the number of keywords. According to Table 2(b), we can get that the number of receivers would influence the time for generating the index. Both Table 2(a) and Table 2(b) demonstrate that our scheme

Table 2 Index Generation Time Comparison

	Guo et al. [22]		Our scheme	
	Operations	Time (ms)	Operations	Time (ms)
(a) Index generation time for the different number of keywords with the same number of receivers, $n=50$				
The number of keywords				
10	$1570E+10P$	4986.9	$60E$	190.4
20	$3090E+20P$	9852.6	$70E$	221.9
30	$4610E+30P$	14719.1	$80E$	255.1
40	$6130E+40P$	19568.2	$90E$	283.3
50	$7650E+50P$	24458.0	$100E$	319.4
(b) Index generation time for the different number of receivers with the fixed number of keywords, $m=50$				
The number of reviewers				
10	$1610E+50P$	5077.6	$60E$	190.1
20	$3120E+50P$	9897.6	$70E$	220.9
30	$4630E+50P$	14622.2	$80E$	256.8
40	$6140E+50P$	19504.2	$90E$	283.0
50	$7650E+50P$	24378.3	$100E$	311.1

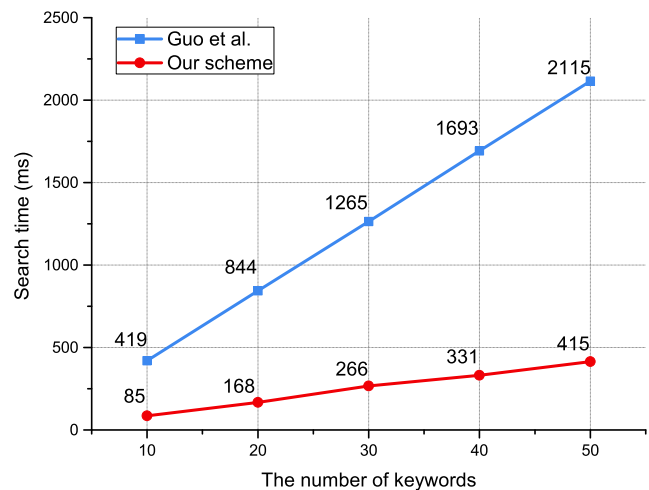
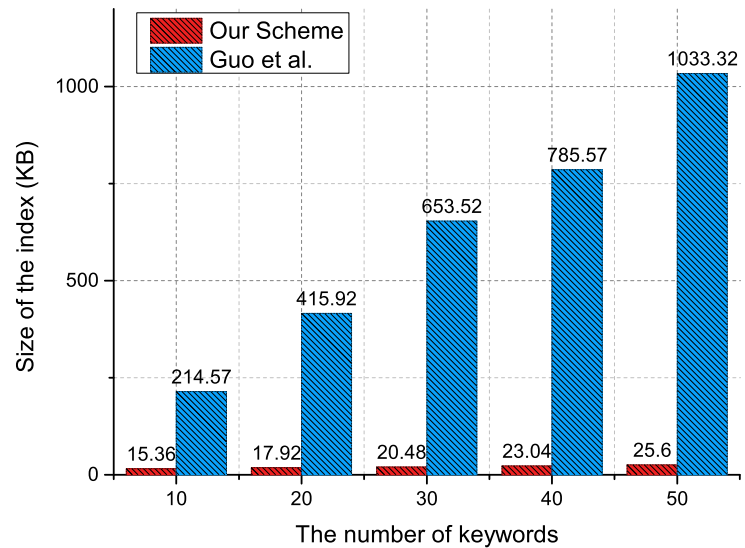
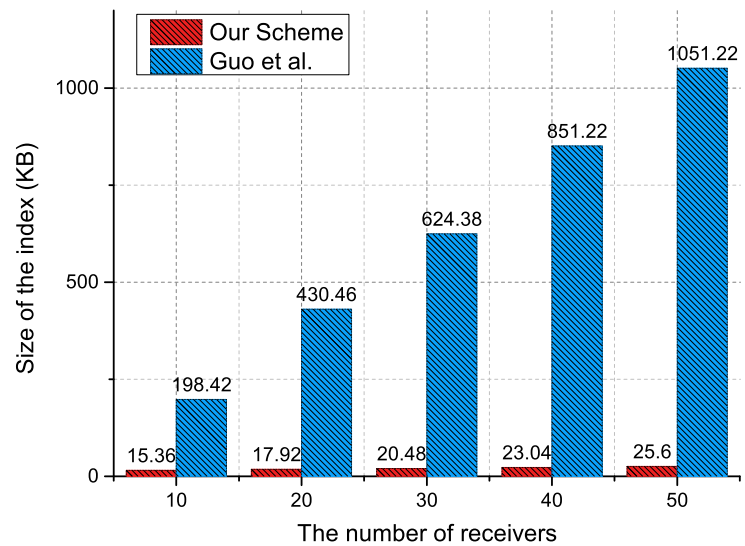


Fig. 2 The search time for the different number of keywords

Fig. 3 Storage overhead

(a) The index storage overhead with 50 receivers



(b) The index storage overhead with 50 keywords

consumes much less computational resources on generating the index than the scheme in [22].

Figure 2 shows that the time for search is linearly increasing with the number of keywords in the index. It is worth noting that we consider the worst case, in which each search operation should go through the whole index. Likewise, Fig. 2 proves that our scheme presents the better computational performance on the search.

Figure 3 illustrates the storage overhead of the index in both [22] and this paper. In our scheme, the storage overhead of the index consists of both the index itself and the re-encryption key set. In our implementation, both $|\mathbb{G}_1|$ and $|\mathbb{G}_2|$ are 2048-bit length. And the security parameter λ

appeared in [22] is set as 2048. As showed in Fig. 3, both the number of keywords and the number of receivers will affect the storage overhead of the index. Apparently, the storage overhead of the index is significantly reduced in our scheme.

Conclusion

In this paper, we proposed a low overhead SCF-PEKS scheme which is suitable to be deployed in a medical cloud environment. Our scheme is a practical solution in the multi-receiver setting. By using our scheme, the sender can

efficiently delegate the search ability to receivers with both low computation overhead and low storage overhead. Each authorized receiver can easily search over the encrypted EMRs, and retrieve the matched records. Our correctness analysis and security analysis demonstrated that the proposed scheme is soundness and secure against the IND-KGA attack. The comprehensive comparisons, including theoretical comparison and performance evaluation, showed that our scheme can achieve better efficiency in terms of the computation overhead and the storage overhead compared with existing ones. Since our scheme is a more cost-efficient solution, it will be more competitive to be deployed in the cloud. For the future work, we will investigate on enriching the functionalities of the search based on our scheme, such as the fuzzy keyword search and the ranked keyword search.

References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., et al., A view of cloud computing. *Communications of the ACM* 53(4):50–58, 2010.
2. Li, M., and et al, Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings, *Security and Privacy in Communication Networks*, pp. 89–106. Berlin Heidelberg: Springer, 2010.
3. He, D., Chen, J., Hu, J., He, D., Chen, J., and Hu, J., An id-based client authentication with key agreement protocol for mobile clientserver environment on ecc with provable security. *Information Fusion* 13(3):223–230, 2012.
4. Mishra, D., A study on id-based authentication schemes for tele-care medical information system. *Computer Science* 24(6):621–625, 2013.
5. Chaturvedi, A., Mishra, D., and Mukhopadhyay, S., Improved biometric-based three-factor remote user authentication scheme with key agreement using smart card. *Information systems security*, pp. 63–77. Berlin Heidelberg: Springer, 2013.
6. Mishra, D., Das, A. K., and Mukhopadhyay, S., A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards. *Expert Systems with Applications* 41(18):8129–8143, 2014.
7. Khan, M. K., Chaturvedi, A., Mishra, D., and Kumari, S., On the security enhancement of integrated electronic patient records information systems. *Computer Science and Information Systems* 12(2):857–872, 2015.
8. Xhafa, F., Li, J., Zhao, G., Li, J., Chen, X., and Wong, D. S., Designing cloud-based electronic health record system with attribute-based encryption. *Multimedia Tools and Applications* 74(10):3441–3458, 2014.
9. Sun, W., Lou, W., Hou, Y. T., and Li, H., Privacy-preserving keyword search over encrypted data in cloud computing. In: *Secure Cloud Computing*, pp. 189–212. New York: Springer, 2014.
10. Boneh, D., Crescenzo, G. D., Ostrovsky, R., and Persiano, G., Public key encryption with keyword search. In: *EUROCRYPT 2004*, pp. 506–522, 2004.
11. Park, D. J., Kim, K., and Lee, P. J., Public key encryption with conjunctive field keyword search. In: *Information security applications*, pp. 73–86. Berlin Heidelberg: Springer, 2004.
12. Hwang, Y. H., and Lee, P. J., Public key encryption with conjunctive keyword search and its extension to a multi-user system. In: *Pairing-Based Cryptography Pairing*, pp. 2–22. Berlin Heidelberg: Springer, 2007.
13. Dong, C., Russello, G., and Dulay, N., Shared and searchable encrypted data for untrusted servers. *J. Comput. Secur.* 19(3):367–397, 2011.
14. Benaloh, J., Chase, M., Horvitz, E., and Lauter, K., Patient controlled encryption: Ensuring privacy of electronic medical records. In *ACM cloud computing security workshop*. *ACM*, 103–114, 2009.
15. Liu, Z., Weng, J., Li, J., Yang, J., Fu, C., and Jia, C., Cloud-based electronic health record system supporting fuzzy keyword search. *Soft. Comput.* 20(8):1–13, 2015.
16. Baek, J., Safavi-Naini, R., and Susilo, W., Public key encryption with keyword search revisited. In: *ICCSA 2008, LNCS 5072*, pp. 1249–1259, 2008.
17. Rhee, H. S., Park, J. H., Susilo, W., and Lee, D. H., Improved searchable public key encryption with designated tester. In: *ASIACCS 2009, ACM*, pp. 376–379, 2009.
18. Rhee, H. S., Park, J. H., Susilo, W., and Dong, H. L., Trapdoor security in a searchable public-key encryption scheme with a designated tester. *J. Syst. Softw.* 83(5):763–771, 2010.
19. Gu, C., Zhu, Y., and Pan, H., Efficient public key encryption with keyword search schemes from pairings. *Information security and cryptology*, pp. 372–383. Berlin Heidelberg: Springer, 2007.
20. Fang, L. M., Susilo, W., Ge, C. P., and Wang, J. D., A secure channel free public key encryption with keyword search scheme without random oracle. In: *CANS 2009, LNCS 5888*, pp. 248–258, 2009.
21. Fang, L. M., Susilo, W., Ge, C. P., and Wang, J. D., Public key encryption with keyword search secure against keyword guessing attacks without random oracle. *Inf. Sci.* 238(7):221–241, 2013.
22. Guo, L., and Yau, W. C., Efficient secure-channel free public key encryption with keyword search for EMRs in Cloud Storage. *J. Med. Syst.* 39(2):1–11, 2015.
23. Bao, F., Deng, R. H., and Zhu, H., Variations of diffie-hellman problem. In *Information and Communications Security*. *Springer Berlin Heidelberg*, 301–312, 2003.
24. PPBC Library, <https://crypto.stanford.edu/pbcl/>.