ISSN 1392–124X (print), ISSN 2335–884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2014, T. 43, Nr. 3

A New Public Key Encryption with Conjunctive Field Keyword Search Scheme

Min-Shiang Hwang

Asia University, Department of Computer Science and Information Engineering
No. 500, Lioufeng Raod, Wufeng Shiang, Taichung, Taiwan, R.O.C.
and China Medical University, Department of Medical Research, China Medical University Hospital
No.91, Hsueh-Shih Road, Taichung 40402, Taiwan, R.O.C.
e-mail: mshwang@asia.edu.tw

Shih-Ting Hsu

National Chung Hsing University, Department of Management Information Systems
250 Kuo Kuang Road, 402 Taichung, Taiwan, R.O.C.

Cheng-Chi Lee*

Fu Jen Catholic University, Department of Library and Information Science
No. 510, Zhongzheng Rd., Xinzhuang Dist., New Taipei City 24205, Taiwan, R.O.C.
e-mail: cclee@mail.fju.edu.tw

crossref http://dx.doi.org/10.5755/j01.itc.43.3.6429

Abstract. The scheme that searching over the encrypted data, which is also named conjunctive keyword searchable scheme, enables one to search the encrypted data by using conjunctive keywords. The concept was first presented by Golle *et al*., and then Park *et al*. extended their scheme into a public key system. According to the existing conjunctive keyword searchable schemes and the assumption that Golle *et al.* proposed, there are two types: the fixed keyword field scheme and the variable keyword field scheme. However, there are still rooms for both kinds of the schemes to improve both the performance and the security. In this paper, we propose an efficient secure channel free public key encryption with conjunctive field keyword search scheme that can stand against the off-line keyword-guessing attacks, which is more suitable for the weak devices used by users.

Keywords: public key encryption; conjunctive keyword; SCF-PEKS; SCF-PECKS.

1. Introduction

1.1. Motivation

As the Internet has been adopted worldwide, a number of daily activities have been shifted into the personal PC, mobile devices, and so on; moreover, the remote tools and resource can be accessed easily through the Internet. For example, suppose user Alice adopts the remote server as the database to store the personal documents. Whenever Alice wants to store the documents at the remote server, she transfers these documents via the Internet. Since the Internet is public, there exists a lot of risks for the documents in the transmission process [16, 17]. In order to protect the content of each document, Alice usually encrypts the documents before sending to the remote server [7, 14, 25, 27, 29]. However, as the documents are encrypted, the content is changed into the sequential of characters which cannot be distinguished. The content of the encrypted documents cannot be learned by the attackers and Alice. To solve this problem, Song *et al.* [24] gave the concept of searching over the encrypted data with a certain word in 2000. Boneh *et al.* [2] further proposed Public key Encryption with Keyword Search

* Corresponding author

(PEKS) scheme which enables one to search the encrypted data by using a keyword in 2004.

In PEKS scheme [11], there are three entities: the data sender, the receiver, and the server. The data sender owns the documents and wishes to share the documents with the receiver. The server provides the storage space to store the encrypted documents and is regarded as the third part who responds the query from the receiver; we can take the server as an untrusted server since the data owner cannot manage their documents directly. Suppose user Bob is a data sender and he wishes to send a document M to a receiver Alice, he sets keywords for M and uses Alice's public key to encrypt the document and keywords. Then, Bob sends the encrypted data with the following form:

$$E_{A_{pub}}(M)||PEKS(A_{pub}, w_1)|| \dots ||PEKS(A_{pub}, w_m)$$

where A_{pub} is Alice's public key and $w_1, w_2, \dots, w_m$ are the keywords that Bob sets. When Alice wishes to search the encrypted documents with keyword W, she generates a $trapdoor$ containing W and sends to the server. The server finds the corresponding encrypted documents by comparing the trapdoor and the keyword ciphertexts and sends them to Alice. Please refer to Fig. 1. However, PEKS scheme needs a secure channel between the server and the receiver. However, building the secure channel is costly and not suitable for overall situations such as the applications in the public-key setting. Furthermore, if Alice wishes to retrieve the encrypted documents related with "urgent", "Monday" and "Business", PEKS scheme cannot accomplish this query since the user can only search the encrypted documents with one keyword. Thus, we address the two important issues of PEKS scheme as follows:

1. Constructing a secure channel is costly and not suitable for the overall applications.
2. Searching the encrypted data only using one keyword is not enough in the practical applications.

To solve the above two issues, Baek *et al.* [1] and Golle *et al.* [9] proposed the solutions, respectively. For the first issue, Baek *et al.* [1] presented a new security model that removes the secure channel assumption in PEKS scheme, which is named Secure Channel Free Public key Encryption with Keyword Search (SCF-PEKS) scheme. The basic idea of SCF-PEKS scheme is to make the server to keep its own public/private key pairs. Whenever the data sender produces the keyword ciphertexts, he inputs the server's public key in the algorithm; only the corresponding private key can execute the Test algorithm in SCF-PEKS scheme. For another issue, Golle *et al.* [9] first presented the notion of *secret key* encryption with conjunctive keyword field scheme which enables users to search the encrypted documents with more than one keyword. They assumed that each document has m keyword fields, and they identified a vector of m keywords and denoted the ith document by $D_i = (w_{i,1}, w_{i,2}, \dots, w_{i,m})$. Taking an email system as an example, there are four keyword fields defined for each email: "From", "To", "Date" and "Subject". Also, Golle *et al.* proposed two assumptions as follows:

1. The same keyword will not appear in two keyword fields. For example, "From:Bob" cannot be confused with "To:Bob" since they do not belong to the same keyword fields.
2. Every keyword field is defined for every document. If the number of keywords is less than the number of keyword fields, we should assign the keyword "NULL" to the keyword fields which do not have the content.

Whenever the user searches the encrypted documents, he has to identify the corresponding keyword fields as well as the keywords that he wishes to search. However, Golle *et al.*'s scheme is constructed in the *secret-key* setting which is also not suitable for a public key cryptosystem.

1.2. Related Works

We can categorize the existing conjunctive keyword searchable schemes into two types: the fixed keyword field and the variable keyword field [8]. The fixed keyword field schemes [5, 9, 12, 21, 23] are based on the assumption in [9] which identifies m keyword fields for each document. When the receiver generates the trapdoor, he has to identify the keyword fields that he wants to search. It is not difficult to apply since the query system can be implemented like a relational database management system; that is, when the user wishes to search the data, he can input the keywords according to the fields that the system sets. In another hand, the variable keyword field schemes [6, 30] can be applied to more than the relational database. The advantage of variable keyword field is that it only needs the less amount of the storage space for the server to store the ciphertext. If the storage space is an on-demand storage service, the user can only purchase the minimal storage space. On the contrast, the fixed keyword field has the advantage of security and convenience because it reveals the least amount of information to the server [26].

However, in order to design a conjunctive keyword searchable scheme which is suitable for most of the applications, Park *et al.* [21] presented a new scheme based on a public key cryptosystem, which is named Public key Encryption with Conjunctive fields Keyword Search (PECKS) scheme. They used the fixed keyword fields assumption in [9] and adopted the bilinear pairing to construct their scheme. However, Byun *et al.* [4] pointed out that PEKS scheme [2] may be attacked easily by off-line keyword-guessing attacks since the keyword space is much smaller than the password. Besides, the trapdoors are transferred via a public network in the scheme based on the public key cryptosystme [1, 2, 21, 22], the attackers have much probability to eavesdrop the trapdoors and derive the keywords from them. Therefore, most of the existing keyword searchable schemes pay more attentions on

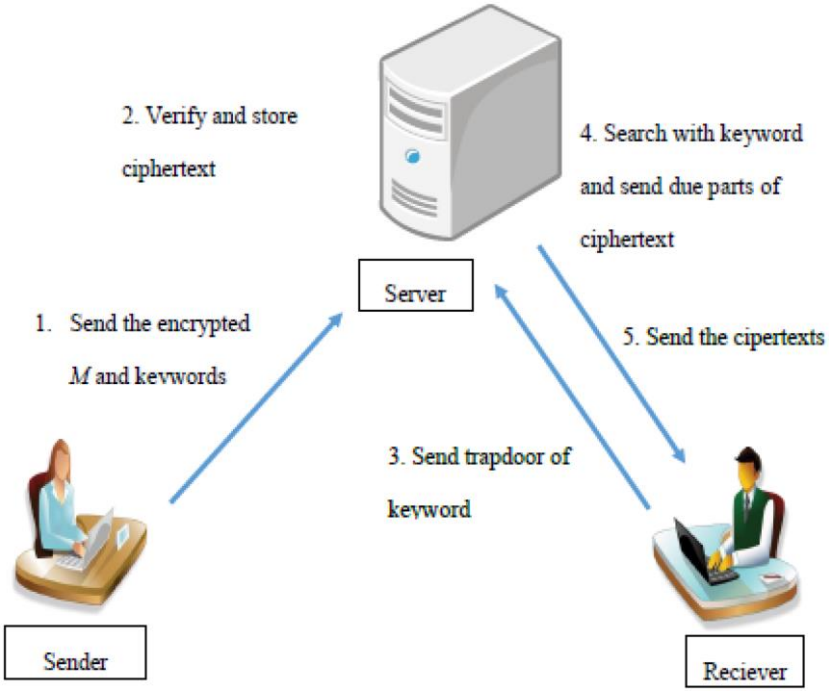

Figure 1. The PEKS

enhancing the security of their schemes [3, 5, 6, 23, 26, 30]. Unfortunately, some of schemes need a large amount of computing time or produce long keyword ciphertexts and trapdoors which are inefficient for users [15]. In this paper, we construct a new PECKS scheme based on bilinear paring and discuss the requirements as follows:

1. **Unforgeability of the trapdoor** [30]: Since the keyword ciphertexts contain the receiver's public key, only the trapdoor which is generated by the corresponding private key can complete the queries. Therefore, the proposed scheme should ensure that no one can forge the legal trapdoor without the authorized receiver's private key.
2. **Anonymousness of the ciphertext** [30]: After encrypting, the keywords are changed into a sequential of characters that cannot be distinguished. This requirement means that nobody can gain the embedded keywords from the keyword ciphertexts.
3. **Practicability** [15]: For users, it is burdensome to remember too much extra information to encrypt the keywords and search the encrypted data. Therefore, the proposed scheme should be adopted easily in the reality.
4. **Efficiency**: Most of the existing conjunctive keyword searchable schemes are still inefficient for users. In order to apply the conjunctive keyword searchable scheme with weak devices, the proposed scheme should perform efficiently.
5. **Against off-line keyword-guessing attacks**: Since the trapdoors are transferred in the public network, the adversaries can easily capture the trapdoors. However, the trapdoors should secure enough that can stand against the inside and outside off-line keyword-guessing attacks.

1.3. Our Contributions

In this paper, we study the literatures that search the encrypted data with conjunctive keywords and the security definitions for a conjunctive keyword searchable scheme. Furthermore, we point out that two of the recent PECKS schemes fail to resist the off-line keyword-guessing attack from the outside attacker. We further define a new and efficient public key encryption with a conjunctive field keyword search scheme based on bilinear pairing and prove that our scheme is semantically security under DDH [13] assumption without random oracle model [18, 19] and can stand against the off-line keyword-guessing attacks. Furthermore, our scheme does not need the secure channel and has the less computational costs for users, and can be further adopted in the weak devices.

1.4. Organization

This paper is organized as follows: In Section 2, we give some definitions and notions that we adopt in this paper. In Section 3, we review two PECKS schemes based on a bilinear form and perform the off-line keyword-guessing attack on them. In Section 4, we present our new SCF-PECKS scheme and analyze the consistency. Then we analyze the security and performance of the proposed scheme in Section 5. The conclusion is in Section 6.

2. Preliminaries

In this section, we first define the hard assumption that supports our schemes, and we introduce what is the Secure Channel Free Public key Encryption with Conjunctive field Keyword Search (SCF-PECKS) scheme and how the bilinear pairing is for ElGamal public key system (ElGamal BP) works.

2.1. Hardness Assumption

Definition 1 (DDH). *Let G be a cyclic group of prime order p with a generator g. The decisional Diffie-Hellman problem is to distinguish the triplets of the form* (g^a, g^b, g^{ab}) *and* (g^a, g^b, g^c). *We assume the DDH problem is* (ϵ, t)*-hard in G. For every adversary* $\mathcal{A}$ *with a polynomial time t, he/she has an adversary* ϵ *in solving DDH if*

$$|Pr[\mathcal{A}(g^a, g^b, g^{ab}) = true] - Pr[\mathcal{A}(g^a, g^b, g^c) = true]| > \epsilon.$$

2.2. Definitions of SCF-PECKS

After Golle *et al.* [2] proposed the PEKS scheme, Baek *et al.* [1] presented a Secure Channel Free PEKS (SCF-PEKS) scheme to remove the assumption of secure channel in 2008 [2], so the trapdoor can be sent via the public network. In order to develop a more efficient conjunctive keyword searchable scheme that can be widespreadly used, we integrate the concept of SCF-PEKS scheme with the conjunctive keyword searchable security model. Firstly, we describe the definition of PECKS scheme that Park *et al.* [21] presented in 2004. Secondly, we give the definition of SCF-PECKS which has been described in [8]. Furthermore, we define the security games for SCF-PECKS scheme. We identify that there are m keywords in each document and denote $D_i = (w_{i,1}, w_{i,2}, \dots, w_{i,m})$ where D_i is the vector of keywords that the data sender sets for the ith document. Also, we denote the queries $Q = (w'_1, w'_2, \dots, w'_t, I_1, I_2, \dots, I_t)$, where $I_1, I_2, \dots, I_t$ are the locations of the keyword fields.

Definition 2 (PECKS). *A public key encryption with conjunctive field keyword search scheme consists the following algorithms:*

1. Setup(1^k): Taking a security parameter k as input, this algorithm returns a public/private key pairs (pk, sk).
2. PECKS(pk, D): Taking a public key pk and a document D as inputs, this algorithm outputs the conjunctive keyword ciphertexts C.
3. Trapdoor(sk, Q): Taking a private key sk and a query Q as inputs, this algorithm outputs the trapdoor $T_{w'}$.
4. Test($C, T_{w'}$): Taking a conjunctive keyword ciphertext C and a trapdoor $T_{w'}$ as inputs, this algorithm returns a symbol "Correct" if $\{w_{I_1} = w'_1\}$, $\{w_{I_2} = w'_2\}$, …, $\{w_{I_t} = w'_t\}$ and "Incorrect" otherwise.

Definition 3 (SCF-PECKS). *A secure channel free public key encryption with conjunctive field keyword search scheme consists of the following algorithms:*

- GlobalSetup (λ): Takes a security parameter λ as input and generates a global parameter $\mathcal{GP}$.
- KeyGen$_{Server}$ ($\mathcal{GP}$): Takes the global parameters $\mathcal{GP}$ as input and outputs the public/private key pair (pk_S, sk_S) of the server S.
- KeyGen$_{Receiver}$ ($\mathcal{GP}$): Takes $\mathcal{GP}$ as input and outputs public/private key pair (pk_R, sk_R) of the receiver R.
- dPECKS ($\mathcal{GP}, pk_S, pk_R, D$): Takes $\mathcal{GP}$, a server's public key pk_S, a receiver's public key pk_R, and a document D as inputs. Returns a dPECKS cihpertext C.
- dTrapdoor($\mathcal{GP}, sk_R, Q$): Taking a $\mathcal{GP}$, a receiver's private key sk_R and a query Q as inputs, generates a trapdoor $T_{w'}$.
- dTest($\mathcal{GP}, sk_S, C, T_{w'}$): Taking $\mathcal{GP}$, a server's private key sk_S, a dPECKS ciphertext C, a trapdoor $T_{w'}$ as inputs, outputs a symbol "Correct" if $\{w_{I_1} = w'_1\}$, $\{w_{I_2} = w'_2\}$, ..., $\{w_{I_t} = w'_t\}$ or "Incorrect" otherwise.

Definition 4. *Let* $\mathcal{A}$ *be an adversary with bounded time which is polynomial in a security parameter* λ. *We say a SCF-PECKS is semantically secure according to the security games ICC, ICR and ICLR between the adversary* $\mathcal{A}$ *and a challenger* $\mathcal{B}$, *which are described as follows:*

Security Game ICC (indistinguishability of ciphertext from ciphertext): Let $\mathcal{A}$ be an adversary (the server) with a polynomial time, t, and $\mathcal{B}$ be a challenger. The goal of Game ICC is that $\mathcal{A}$ has to distinguish two encrypted documents, where D_0 and D_1 are chosen by $\mathcal{A}$.

1. An adversary $\mathcal{A}$ adaptively requests the encryption dPECKS(GP, pk_S, pk_R, D_i) of documents D_i where $i \in \{0,1\}^*$, and searches trapdoors.
2. $\mathcal{A}$ chooses two documents D_0, D_1 and sends them to $\mathcal{B}$.
3. $\mathcal{B}$ chooses $b \in \{0,1\}$ randomly and sends $\mathcal{A}$ an encryption of D_b.
4. $\mathcal{A}$ again asks for the encrypted documents and trapdoors, with the restriction that $\mathcal{A}$ may not ask for a trapdoor that is distinguishable for D_0 and D_1.
5. $\mathcal{A}$ outputs b′ ∈ {0, 1} and wins the game ICC if $b' = b$. We say the adversary $\mathcal{A}$ has an ϵ-advantage if the adversary's advantage is

$$Adv_A(1^\lambda) = |Pr[b = b'] - 1/2| > \epsilon.$$

Security Game ICR (indistinguishability of ciphertexts from random): Let $\mathcal{A}$ be an inside adversary with a polynomially bounded time and $\mathcal{B}$ be a challenger (the user). The adversary chooses one

document D_0 and a keyword subset T of D_0. The goal of Game ICR is that $\mathcal{A}$ has to distinguish two encrypted documents, D_0 and D_1, where D_0 is chosen by $\mathcal{A}$ and D_1 is generated by $\mathcal{B}$. $Rand(D,T)$ denotes that the keywords of document D are all replaced by the randomly chosen keywords in T, where T is a set of keywords. The game ICR works as follows:

1. An adversary $\mathcal{A}$ adaptively requests the encryption $\mathsf{dPECKS}(\mathcal{GP}, pk_S, pk_R, D_i)$ of documents D_i where $i \in \{0,1\}^*$, and searches trapdoors.
2. $\mathcal{A}$ chooses a document D_0 and subset $T \subseteq \{1,\dots,m\}$, then sends them to $\mathcal{B}$.
3. $\mathcal{B}$ creates a document $D_1 = Rand(D_0, T)$ and chooses a random bit $b \in \{0,1\}$, then sends the encryption of D_b to $\mathcal{A}$.
4. $\mathcal{A}$ again asks for encrypted documents and trapdoors, with the restriction that $\mathcal{A}$ may not ask for a trapdoor that distinguishes D_0 from D_1.
5. $\mathcal{A}$ outputs $b' \in \{0,1\}$ and wins the game ICR if $b' = b$. We say that the adversary $\mathcal{A}$ has an ϵ-advantage if the adversary's advantage is

$$Adv_A(1^\lambda) = |Pr[b' = b] - 1/2| > \epsilon.$$

Security Game ICLR (indistinguishability of ciphertexts from limited random): Let $\mathcal{A}$ be an adversary (the server) with a polynomial time and $\mathcal{B}$ be a challenger (the user). The adversary chooses one document and a keyword subset T. The goal of Game ICLR is that $\mathcal{A}$ has to distinguish two encrypted documents which are created by $\mathcal{B}$. This security game also gives the notion that an adversary cannot gain the plaintext from the other documents [28].

1. An adversary $\mathcal{A}$ requests the encryption $\mathsf{dPECKS}(\mathcal{GP}, pk_S, pk_R, D_i)$ of any documents D_i where $i \in \{0,1\}^*$, and any search trapdoors.
2. $\mathcal{A}$ chooses a document D and a subset $T \subseteq \{1,\dots,m\}$, and then sends them to the challenger $\mathcal{B}$.
3. $\mathcal{B}$ creates two documents $D_0 = Rand(D, T - \{t\})$ and $D_1 = Rand(D,T)$, where a value $t \in T$, and chooses a random bit $b \in \{0,1\}$, and then gives $\mathsf{dPECKS}(GP, pkS, pkR, Db)$ to $\mathcal{A}$.
4. $\mathcal{A}$ again asks for encrypted documents and trapdoors, with the restriction that $\mathcal{A}$ may not ask for a trapdoor that is distinguishable for D_0 and D_1.
5. $\mathcal{A}$ outputs $b' \in \{0,1\}$ and wins the game ICLR if $b' = b$. We say that the adversary $\mathcal{A}$ has an ϵ-advantage if the adversary's advantage is

$$Adv_A(1^\lambda) = |Pr[b' = b] - 1/2| > \epsilon.$$

Theorem 1. *[9] If there exists an adversary $\mathcal{A}$ that wins game ICC with advantage ϵ, then there exists an adversary $\mathcal{A}'$ that wins game ICLR with advantage $\epsilon/2m^2$.*

2.3. Bilinear Pairing for ElGamal Public Key System

In order to construct an efficient SCF-PECKS scheme, we adopt the public key system which is named Bilinear Pairing for ElGamal public key system (ElGamal BP) and proposed by Nguyen in 2004 [20].

Definition 5 (Bilinear Pairing). *Let G_1, G_2 and G_t be three addition groups of prime order p, and g_1 is a generator of G_1, g_2 is a generator of G_2. We say $e: G_1 \times G_2 \to G_t$ is a bilinear map if the following properties hold:*

1. Bilinearity: $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$ for all $a, b \in Z_p$.
2. Non-degeneracy: $e(g_1, g_2) \neq 1$.
3. Computability: There exists a polynomial time algorithm to compute $e(g_1, g_2)$.

Definition 6 (ElGamal BP). *A bilinear pairing version for ElGamal public key system consists of the following algorithms:*

- Key generation: Let G_1 and G_2 be two cyclic groups of prime order p, and g be a generator of G_1. Let $e: G_1 \times G_1 \to G_2$ be a bilinear map. Select a random value $x \in Z_p^*$ and compute $\theta = e(g,g)^x$. Output the public key $pk = (g, \theta)$ and the private key $sk = x$.
- Encryption: To encrypt the message $M \in G_2$, this algorithm chooses a random value $r \in Z_p^*$ and computes the ciphertext $C = (C_1, C_2) = (rg, M\theta^r)$.
- Decryption: The message M can be regained by computing $M = C_2 / e(C_1, g)^x$.

3. Attacks on Two PECKS Schemes

In this section, we review two PECKS schemes including one fixed keyword field scheme and one variable keyword field scheme, so we perform the outside off-line keyword-guessing attack on them.

3.1. Review on Chen and Horng's Scheme

In 2009, Chen and Horng [5] proposed a PECKS scheme based on bilinear pairing. In order to improve the efficiency of the server's running time in the Test algorithm, Horng and Chen add the timestamp to classify the ciphertexts. After receiving the encrypted document, the server will create "the encrypted timing data" and store them. The server publishes a value in the encrypted timing data, and then the receiver can take this value as a part of trapdoor. Finally, the server can accelerate the test time of finding the corresponding encrypted documents.

3.1.1. Construction

This scheme uses a hash function $: \{0,1\}^* \to Z_p^*$, three groups G_1, G_2 and G_t of prime order p, and a

bilinear map $e: G_1 \times G_2 \to G_t$. The global parameter $\mathcal{GP} = (p, G_1, G_2, G_t, e, H)$.

1. KeyGen($\mathcal{GP}$): Picks a random value $\alpha \in Z_p^*$, a generator $P_1 \in G_1$ and a generator $P_2 \in G_2$. It returns public key $pk_R = [P_1, P_2, Y = \alpha P_1]$ and private key $sk_R = \alpha$.
2. PECKS($\mathcal{GP}, pk_R, D$): Chooses a random value $r \in Z_p^*$ and computes $V_i = rH(W_i)Y$. It outputs the keyword ciphertext $S = [V_1, V_2, \dots, V_m, rP_1]$.
3. Timestamp(S, k): When the server receives the ciphtertext S, it chooses a random value $k \in Z_{p-1}^*$. Then, it outputs "the encrypted timing data" $S_k = [V_1 k, V_2 k, \dots, V_m k, rP_1, kP_2]$ and publishes the timestamp value kP_2.
4. Trapdoor($\mathcal{GP}, sk_R, D, kP_2$): Takes a timestamp value kP_2 from the public information server and selects a random value $s \in Z_p^*$. Computes $T_{w'} = [T_1, T_2, T_3, I_1, I_2, \dots, I_t]$ where $I_1, I_2, \dots, I_t$ are the keyword fields which the receiver wishes to search, and

$$T_1 = \sum_{i=1}^{t} \big(H(W_i')\big) s\alpha(kP_2)$$

$$T_2 = sP_2$$

$$T_3 = kP_2$$

in which T_3 is a label for searching the corresponding groups of document for the server.

5. Test($\mathcal{GP}, S_k, T_{w'}$): Let $S_k = [A_1, A_2, \dots, A_m, B]$ and the server use T_3 to find the corresponding encrypted documents. Check if $e(B, T_1) = e(A_{l_1} + A_{l_2} + \cdots + A_{l_t}, T_2)$. If so, output "yes", and "no" otherwise.

3.1.2. Off-Line Keyword-Guessing Attack on Chen and Horng's Scheme

Assume that there is an outside adversary $\mathcal{A}$ which tends to perform the off-line keyword-guessing attack on Chen and Horng's scheme. The public parameter is a bilinear map e, a hash function H and the receiver's public key $pk_R = [P_1, P_2, Y = \alpha P_1]$ which can be obtained from the public network. An adversary $\mathcal{A}$ performs the attack as the following steps:

- **Step 1.** $\mathcal{A}$ eavesdrops the trapdoor

  $T_{w'} = (T_1, T_2, T_3, I_1, \dots, I_t)$.
- **Step 2.** $\mathcal{A}$ guesses the keywords

  $w_1^*, w_2^*, \dots, w_t^*$ and computes
  $V^* = \prod_{i=1}^{t} H(w_i^*)$.
- **Step 3.** Check if

  $e(T_1, P_1 \cdot P_2) = e(V^*, Y\, T_2 T_3)$. If the equation holds, the attack successes.
  Otherwise, go to Step 2.

  $e(T_1, P_1 \cdot P_2)$

$$= e\left(\prod_{i=1}^{t}\big(H(w_i')\big) s\alpha(kP_2), P_1 \cdot P_2\right)$$

$$= e\left(\prod_{i=1}^{t}\big(H(w_i')\big) s\alpha(kP_2), P_1 P_2\right)$$

$$= e\left(\prod_{i=1}^{t}\big(H(w_i')\big), (\alpha P_1)(sP_2)(kP_2)\right)$$

$$= e(V^*, Y\, T_1 T_2).$$

3.2. Review on Zhang and Zhang's Scheme

Zhang and Zhang [30] pointed out that the assumptions in [9] make conjunctive keywords regarded as one keyword and limit the location of keywords since the keyword fields are fixed and inflexible for users. If a user wishes to search five keywords, he has to identify the exactly field that he wishes to query. It burdens the users with extra information needed to remember. Therefore, Zhang and Zhang presented the following two concepts: (1) Keywords should be listed in any order. (2) The repetition of one keyword has nothing wrong about the performance. Zhang and Zhang assumed that each document has l keywords (l is fixed). Although users do not need to define m keyword fields for each document, they still have to build up a l-degree polynomial with l keywords. If the number of keywords is less than l, users can just add some useless keywords to realize the algorithm. Zhang and Zhang's scheme works as follows:

3.2.1. Construction

Assume that there are l keywords in the PECKS algorithm (l is fixed). This scheme uses three groups G_1, G_2 and G_t of prime order p, two collision resistant hash functions: $H: \{0,1\}^* \to Z_p^*$ and $H': G_t \to Z_p^*$. The bilinear group is $\mathcal{GP} = (p, G_1, G_2, G_t, e, H, H')$.

1. Setup($1^k, l$): First choose $l + 1$ parameters: $b_0, b_1, \dots, b_l \in G_1$. Select two random generators $g_1, g_2 \in G_1$, a random generator $h \in G_2$ and a random value $\alpha \in Z_p^*$. Set $h_1 = h^\alpha$. Output the public key $pk_R = (g_1, g_2, h, h_1, b_0, b_1, \dots, b_l)$ and the private key $sk_R = \alpha$.
2. PECKS($\mathcal{GP}, pk_R, w_1, w_2, \dots, w_l$): Choose random elements $a, k \in Z_p$, then construct a l-degree polynomial:

$$f(x) = a \cdot \big(x - H(w_1)\big)\big(x - H(w_2)\big) \dots \big(x - H(w_l)\big) + k = a_l x^l \dots + a_1 x + a_0.$$

Select a random element $r' \in Z_p$, then compute and output the keyword ciphertext S:

$$S = (h^{r'k}, H'(e(g_2, h)^{(a_0 + a_1 + \cdots + a_l) \cdot r'});$$

$$h_1^{a_0 r'}, h_1^{a_1 r'}, \dots, h_1^{a_l r'};\ b_0^{a_0 r'}, b_1^{a_1 r'}, \dots, b_l^{a_l r'}\ ).$$

3. Trapdoor($sk_R, w_1', w_2', \dots, w_s'$): Choose a random value $r \in Z_p$, then compute and output

$$T_{w'} = [g_2^{1/\alpha} \cdot \left(g_1^{H(w_1')^0 + H(w_2')^0 + \cdots + H(w_s')^{0/\alpha \cdot s}} \cdot b_0\right)^r = g_2^{1/\alpha} \cdot (b_0)^r;$$

$$g_2^{1/\alpha} \cdot \left(g_1^{H(w_1')^1 + H(w_2')^1 + \cdots + H(w_s')^{1/\alpha \cdot s}} \cdot b_1\right)^r;$$

$$\vdots$$

$$g_2^{1/\alpha} \cdot \left(g_1^{H(w_1')^l + H(w_2')^l + \cdots + H(w_s')^{l/\alpha \cdot s}} \cdot b_l\right)^r; g_1^r; h_1^r].$$

4. Test(GP, pkR, S, Tw′): Set

$T_{w'} = (T_0, T_1, \ldots, T_l;\ g_1^r, h_1^r)$ and $S = (C_0, C_1; H_0, H_1, \ldots, H_l; B_0, B_1, \ldots, B_l)$.

Then compute the following parameters:

$A_1 = \prod_{i=0}^{l} e(T_i, H_i)$;

$A_2 = e(g_1^r, C_0) = e(g_1^r, h^{r'l})$;

$A_3 = \prod_{i=0}^{l} e(B_i, h_i^r)$

$= \prod_{i=0}^{l} e(b_i^{\alpha_i \cdot \alpha \cdot r'}, h^r)$.

Check if $H'(A_1/(A_2 \cdot A_3)) = C_1$. If so, output "yes", and "no" otherwise.

3.2.2. Off-Line Keyword-Guessing Attack on Zhang and Zhang's Scheme

Assume that there is an outside adversary $\mathcal{A}$ which tends to perform the off-line keyword-guessing attack on Zheng and Zheng's scheme. The public parameters are $\mathcal{GP}$ and the receiver's public key $pk_R = (g_1, g_2, h, h_1, b_0, b_1, \ldots, b_l)$ which can be obtained from the public network. An adversary $\mathcal{A}$ performs the attack as the following steps:

- **Step 1.** $\mathcal{A}$ eavesdrops the trapdoor

  $T_{w'} = [T_0, T_1, \ldots, T_l, g_1^r, h_1^r]$.
- **Step 2.** $\mathcal{A}$ guesses the t keywords

  $w_1^*, w_2^*, \ldots, w_t^*$.
- **Step 3.** Check if

  $e(T_k, h_1) = e(g_2, h) \cdot e(g_1^r, b_k h) \cdot e(g_1^{\left(H(w_1')^k + H(w_2')^k + \cdots + H(w_t')^k\right)/t}, b_1 \cdot h)$,

  where $0 <= k <= l$. If the equation holds, the attack successes. Otherwise, go to Step 2.

  For example, we set $k = 1$ and compute as follows:

  $e(T_1, h_1)$

  $= e(g_2^{\frac{1}{\alpha}} \cdot (g_1^{H(w_1')^1 + H(w_2')^1 + \cdots + H(w_s')^{1/\alpha \cdot s}} \cdot b_1)^r, h_1)$

  $= e(g_2^{\frac{1}{\alpha}}, h_1) \cdot e((g_1^{\left(H(w_1')^1 + H(w_2')^1 + \cdots H(w_s')^1\right)/\alpha \cdot 1} \cdot b_1)^r, h_1)$

  $= e(g_2, h_1^{\frac{1}{\alpha}}) \cdot e(g_1^{\left(H(w_1')^1 + H(w_2')^1 + ldots H(w_s')^1\right)/\alpha \cdot s} \cdot b_1, h^{\alpha})^r$

  $= e(g_2, h) \cdot e(g_1^{\left(H(w_1')^1 + H(w_2')^1 + \cdots + H(w_s')^1\right)/\alpha s}, b_1 \cdot h)^{\alpha r}$

  $= e(g_2, h) \cdot e(g_1^{\left(H(w_1)^1 + H(w_2')^1 + \cdots + H(w_s')^1\right)/s}, b_1 \cdot h)^r$

  $= e(g_2, h) \cdot e(g_1^{r \cdot (H(w_1')^1 + H(w_2')^1 + \cdots + H(w_s')^1/s}), b_1 \cdot h)$

  $= e(g_2, h) \cdot e(g_1^r, b_1 \cdot h) \cdot e(g_1^{\left(H(w_1')^1 + H(w_2')^1 + \cdots + H(w_s')^1\right)/s}, b_1 \cdot h)$

  $= e(g_2, h) \cdot e(g_1^r, b_1 \cdot h) \cdot e(g_1^{\left(H(w_1')^1 + H(w_2')^1 + \cdots H(w_t')^1\right)/t}, b_1 \cdot h)$

4. The Proposed Scheme

In this section, we construct a secure channel free public key encryption with conjunctive field keyword search scheme which can prevent an outside keyword guessing attack. The notations used in the proposed scheme are shown in Table 1. Our scheme consists of the following algorithms: Setup, KeyGen*Server*, KeyGen*Receiver*, dPECKS, dTrapdoor, dTest.

- Setup: Let G_1 be an additive group of prime order p with a generator g and G_2 be a multiplicative group of prime order p. We use a bilinear map $e: G_1 \times G_1 \to G_2$ and a secure one-way hash function $H: \{0,1\}^* \to Z_p^*$. Let KS_w denotes the keyword space. The global parameter $\mathcal{GP} = (G_1, G_2, e, g, H, KS_w)$.
- KeyGen*Server*($\mathcal{GP}$): Select a randomly value $\alpha \in Z_p$ and set $sk_S = \alpha$. Then compute $pk_S = (pk_{S_1}, pk_{S_2}) = (e(g,g)^{\alpha}, g^{\alpha})$. Output the server's public/private key pair (pk_S, sk_S).
- KeyGen*Receiver*($\mathcal{GP}$): Select a randomly value $\beta \in Z_p$ and set $sk_R = \beta$. Then compute $pk_R = (pk_{R_1}, pk_{R_2}) = (e(g,g)^{\beta}, g^{\beta})$. Output the receiver's public/private key pair (pk_R, sk_R).

  dPECKS($\mathcal{GP}, pk_S, pk_R, D$): In this algorithm, the data sender defines and encrypts m keywords for the document, and stores the output to the server. First, this algorithm chooses a random value $r \in Z_p^*$ and computes $C_1 = H(w_1) \cdot pk_{S1}^r$, $C_2 = H(w_2) \cdot pk_{S_1}^r, \ldots, C_m = H(w_m) \cdot pk_{S_1}^r, C_{m+1} = r \cdot g$ and $C_{m+2} = pk_{R_2} \cdot pk_{S1}^r$. It returns the keyword ciphertexts $C = (C_1, C_2, \ldots, C_m, C_{m+1}, C_{m+2})$.
- dTrapdoor($\mathcal{GP}, pk_S, sk_R, Q$): In this algorithm, an authorized receiver produces a trapdoor for keywords $w_1', w_2', \ldots, w_t'$ in Q. This algorithm first selects a random value $k \in Z_p^*$ and computes $V = \prod_{i=1}^{t} H(w_i')$. Then, it computes $T_1 = k \cdot g$ and $T_2 = (sk_R + V)^{-1} \cdot (pk_{S_1})^{kt}$. It returns the trapdoor $T_{w'} = (T_1, T_2, I_1, I_2, \ldots, I_t)$ and sends $T_{w'}$ to the server.

Table 1. Notations

Symbol	Description
G_1, G_2	Two cyclic group of prime order p
g	The generator of G_1
e	A bilinear map
H	A secure one-way hash function
KS_w	The keyword space
pk_S, sk_S	The server's public/private key pairs
pk_R, sk_R	The receiver's public/private key pairs
m	The number of keywords that defined for each document
t	The number of keywords that the receiver searches
D	The vector of m keywords that data sender set for the document
Q	The format of the query that the receiver produces
$I_1, I_2, \dots, I_t$	The locations of keywords
$w_1, \dots, w_i$	The keywords that the data sender sets where $i <= m$
$w_1', \dots, w_i'$	The keywords that the receiver searches
C	The ciphertext of keyword that data sender produces
$T_{w'}$	The trapdoor which contains w'

- $\mathsf{dTest}(\mathcal{GP}, T_{w'}, C, sk_S)$: Whenever the server receives the trapdoor $T_{w'}$ from the receiver, it executes this algorithm. First, this algorithm computes $(u, \tilde{u})$ from ciphertext C as follows:

  $u = \left(C_{I_1} \times C_{I_2} \times \dots C_{I_t}\right) / e(C_{m+1}, g)^{sk_S \cdot t}$

  $= \prod_{i=1}^{t} H(w_i)$

  $\tilde{u} = (C_{m+2}) / e(C_{m+1}, g)^{sk_S}$

  $= pk_{R_2}$.

  Then, it computes

  $z = \dfrac{T_2}{e(T_1, g)^{sk_S \cdot t}}$

  $= (\beta + V)^{-1}$.

  Check if $e(\tilde{u} \cdot g^u, g^z) = e(g, g)$. If so, output "yes" and "no" otherwise.

5. Analysis of The Proposed Scheme

In this section, we first prove that our scheme satisfies the computational consistency, which means our scheme can perform accurately. Then we analyze the security of our scheme under the DDH assumption without random oracle model and make a comparison with other schemes for the security requirements. Moreover, we prove that our scheme can stand against the outside off-line keyword-guessing attacks.

5.1. Consistency

Theorem 2. *Our SCF-PECKS scheme satisfies computational consistency.*

Proof: In order to conform the computational consistency of our scheme, first we compute $u, \tilde{u}$ and z as follows:

$$u = \frac{C_{I_1} \times C_{I_2} \dots C_{I_t}}{e(C_{m+1}, g)^{sk_S \cdot t}}$$

$$= \frac{\prod_{i=1}^{t} H(w_i') \cdot \left(pk_{S_1}\right)^r}{e(rg, g)^{\alpha \cdot t}}$$

$$= \frac{e(g, g)^{\alpha r t} \cdot \prod_{i=1}^{t} H(w_i')}{e(g, g)^{\alpha r t}}$$

$$= \prod_{i=1}^{t} H(w_i);$$

$$\tilde{u} = \frac{C_{m+2}}{e(C_{m+1}, g)^{sk_S}}$$

$$= \frac{pk_{R_2} \cdot \left(pk_{S_1}\right)^r}{e(rg, g)^{sk_S}}$$

$$= \frac{pk_{R_2} \cdot e(g, g)^{\alpha \cdot r}}{e(g, g)^{\alpha r}}$$

$$= pk_{R_2};$$

$$z = \frac{T_2}{e(T_1, g)^{sk_S \cdot t}}$$

$$= \frac{(sk_R + V)^{-1} \cdot \left(pk_{S_1}\right)^{kt}}{e(kg, g)^{\alpha \cdot t}}$$

$$= \frac{(\beta + V)^{-1} \cdot e(g, g)^{\alpha k t}}{e(g, g)^{\alpha k t}}$$

$$= (\beta + V)^{-1}.$$

Then, we perform the following steps:

$$e(\tilde{u} \cdot g^u, g^z) = e(g^\beta g^u, g^{(\beta+V)^{-1}})$$
$$= e(g^{\beta+u}, g^{(\beta+V)^{-1}})$$
$$= e(g,g)^{(\beta+u)\cdot(\beta+V)^{-1}}.$$

If $(\beta+u)\cdot(\beta+V)^{-1} = 1$ which means $V = u = \prod_{i=1}^{t} H(w_i)$, the equation will compute as $e(g,g)^{(\beta+u)\cdot(\beta+V)^{-1}} = e(g,g)$. Finally, the computational consistency is complete.

5.2. Security Analysis

Theorem 3. *Our SCF-PECKS scheme is secure according to game ICLR without random oracle assuming DDH is intractable.*

Proof: According to Theorem 1, if there exists an adversary with a non-negligible probability which can win the game ICC, there also exists an advantage which can win the game ICLR. Assume that $\mathcal{A}$ is an inside attacker, which can attack our scheme in a polynomial time and make at most q_k trapdoor queries where $p >= q_k$, and has the advantage ϵ in solving the DDH problem in G_1. Let G_1 and G_2 be two groups of prime order p, and g be a generator of G_1.

We build a simulator $\mathcal{B}$ to be a challenger which has an advantage $\epsilon' = \epsilon/e^m q_k m$, where e is the base of the natural logarithm.

Suppose we are given an instance (g_a, g_b, g_c) of the DDH problem in G_1, where a, b, c are random values in Z_p. The goal of $\mathcal{B}$ is to distinguish $g^c = g^{ab}$ from a random element in G_1. $\mathcal{B}$ picks z uniformly which is independent to the position t that $\mathcal{A}$ chooses in Step 2 of game ICLR.

1. Setup: An adversary $\mathcal{A}$ picks a random value $\alpha \in Z_p$ as the private key sk_S and computes $pk_{S_1} = e(g,g)^\alpha$ and $pk_{S_2} = g^\alpha$. Let (pk_{S_1}, pk_{S_2}) be $\mathcal{A}$'s public key pk_S and publish his public key $pk_S = pk_{\mathcal{A}}$. Let $(pk_R, sk_R) = ((e(g,g)^\beta, g^\beta), \beta)$ be $\mathcal{B}$'s public/private key pair.
2. Encrypt queries: An adversary $\mathcal{A}$ makes queries for the ciphertext of document D_i where $D_i = (w_{i,1}, \dots, w_{i,m})$. To simulate the dPECK($\mathcal{GP}, pk_R, pk_S, D_i, KS_w$) algorithm, $\mathcal{B}$ first picks a random value $x_j \in Z_p$ for every $w_{i,j}$ where $1 <= j <= m$. To respond the encryption queries, $\mathcal{B}$ chooses a random value $r_i \in Z_p$ and returns the ciphtertext C as follows:

$$C = (x_1(pk_{S_1})^{r_i}, \dots, x_z\left[(pk_{S_1})^{br_i}\right], \dots,$$
$$x_m(pk_{S_1})^{r_i}, r_i g, pk_{R_2}(pk_{S_2})^{r_i}).$$

3. Trapdoor queries: To evaluate Test($\mathcal{GP}, T_{w'}, C, sk_S$), $\mathcal{A}$ continues to make trapdoor query for the query $Q = (w'_{i,1}, w'_{i,2}, \dots, w'_{i,t}, I_1, I_2, \dots, I_t)$ to $\mathcal{B}$. After receiving a trapdoor query from $\mathcal{A}$, $\mathcal{B}$ checks if the keywords $w'_{i,I_1}, w'_{i,I_2}, \dots, w'_{i,I_t} \in KS_w$ and then computes $V^* = H(w'_{i,I_1}) \cdot H(w'_{i,I_2}) \dots \cdot H(w'_{i,I_t}) = \prod_{j=1}^{t} H\left(w'_{i,I_j}\right)$. Then, $\mathcal{B}$ picks a random value $k \in Z_p$ and computes $T_1 = kg, T_2 = (pk_R + V^*) \cdot (pk_{S_1})^{kt}$ where t is the number of keywords, and then returns the trapdoor $T_{w'} = (T_1, T_2, I_1, \dots, I_t)$.
4. Challenge: $\mathcal{A}$ sends a document D, a set of indices $T \subseteq \{1, \dots, m\}$ and a value $t \in T$ to $\mathcal{B}$. If $z \neq t$, $\mathcal{B}$ replies a random guess to the DDH challenge. If $z = t$, $\mathcal{B}$ responds as follows: Let $E_t = x_t(pk_{S_1})^c$. For $j \neq t$ and $j \in T$, let $E_j = R_j$, where R_j is a random value. For $j \neq t$ and $j \notin T$, let $E_j = x_j(pk_{S_1})^a$. $\mathcal{B}$ returns $\mathcal{A}$ the ciphertext as follows:

$$(E_1, \dots, E_m, rg, pk_{R_2} \cdot (pk_{S_1})^a).$$

If the number z that $\mathcal{B}$ picks is equal to t, then $\mathcal{B}$ does not failure the security game. The ciphertext for every position $j \notin T$ is the encryption of D and the ciphertext in position t where $c = ab$ is an encryption of D, too; otherwise, it is not.
5. More queries: $\mathcal{A}$ queries the encryptions of other documents and trapdoors that $\mathcal{A}$ has not asked before. $\mathcal{B}$ responds in Encrypt queries and Trapdoor queries.
6. Output: $\mathcal{A}$ outputs the guess $b' \in \{0,1\}$. If $b' = 1$ and $\mathcal{B}$ outputs "yes", it means (g^a, g^b, g^c) is a DDH-tuple. Therefore, the position z is equal to t, we prove that (g^a, g^b, g^c) is a DDH-tuple as the following equation:

$$\frac{x_z(pk_{S_1})^{br_i}}{e(r_i g, g)^x} = \frac{x_t(pk_{S_1})^c}{e(ag,g)^x}$$
$$\Leftrightarrow \frac{x_z e(g;g)^{xbr_i}}{e(g;g)^{xr_i}} = \frac{x_t e(g,g)^{xc}}{e(g,g)^{ax}}$$
$$\Leftrightarrow x_z e(g,g)^x e(g,g)^b e(g,g)^{r_i} e(g,g)^a e(g,g,)^x$$
$$= z_t e(g,g)^x e(g,g)^c e(g,g)^{r_i} e(g,g)^x$$
$$\Leftrightarrow e(g,g)^a e(g,g)^b = e(g,g)^c$$
$$\Leftrightarrow e(g,g)^{ab} = e(g,g)^c$$
$$\Leftrightarrow e(g,g^{ab}) = e(g,g^c)$$
$$\Leftrightarrow g^{ab} = g^c.$$

In another hand, if $b' = 0$, it means the encryption at position i is random which cannot conform the above equation. If and only if the challenge is not a DDH tuple, the encryption at position t is the random. However, $\mathcal{A}$ has the same advantage to win the game ICLR that $\mathcal{B}$ solves the DDH challenge.

Now, we define two conditions and give the description of the advantage of $\mathcal{B}$ as follows:

$\mathcal{S}_1$: $\mathcal{B}$ responds the trapdoor queries for $\mathcal{A}$ for m keywords.

$\mathcal{S}_2$: $\mathcal{B}$ is not aborted in the challenge for $\mathcal{A}$.

Assume that q_k is large enough. The probability of S_1 is $Pr[S_1] = (1/e)^m = 1/e^m$, where e is the base of the natural logarithm, and the probability of S_2 is $Pr[S_2] = 1/q_k m$. Therefore, we can obtain the advantage of $\mathcal{B}$ in breaking the DDH problem as follows:

$$\epsilon' = \epsilon \cdot Pr[S_1 \cap S_2] >= \epsilon/e^m q_k m.$$

Theorem 4. *Our SCF-PECKS scheme is secure against outside off-line keyword-guessing attacks.*

Proof: Assume that there is an outside adversary $\mathcal{A}$ which can eavesdrop the trapdoor $T_{w'} = (T_1, T_2, I_1, \dots, I_t)$. Moreover, $\mathcal{A}$ can obtain the public parameter $\mathcal{GP}$, the server's public key $pk_S = (pk_{S_1}, pk_{S_2})$ and the receiver's public key $pk_R = (pk_{R_1}, pk_{R_2})$ from the public network. To obtain the encrypted keywords, $\mathcal{A}$ first guesses the keywords $w_1^*, w_2^*, \dots, w_t^*$ and computes $V^* = \prod_{i=1}^{t} H(w_i^*)$, then executes the off-line keyword-guessing attacks by using T_2 to conform its guess:

$$\begin{aligned} T_2 &= (\beta + V)^{-1}\left(pk_{S_1}\right)^{kt} \\ &= \left(pk_{S_1}\right)^{kt(\beta+V)^{-1}} \\ &= e(g,g)^{\alpha kt(\beta+V)^{-1}} \\ &= e(g^{\alpha t}, kg)^{(\beta+V)^{-1}} \\ &= e\left(\left(pk_{S_2}\right)^t, T_1\right)^{(\beta+V)^{-1}} \\ &= e\left(\left(pk_{S_2}\right)^t, T_1\right)^{(\beta+V^*)^{-1}}. \end{aligned}$$

In the above equations, the receiver's private key β is unknown to $\mathcal{A}$, therefore, the adversary $\mathcal{A}$ cannot attack our scheme successfully by performing off-line keyword-guessing attacks.

Now we describe that our scheme satisfies two requirements that are unforgeable of the trapdoor and anonymous of the ciphertext as follows:

- **Unforgeability of the trapdoor:** To generate a legal trapdoor, the receiver must have the corresponding private key to generate $T_2 = (sk_R + V)^{-1} \cdot \left(pk_{S_1}\right)^{kt}$. Since it chooses a different random value each time the receiver executes the dTrapdoor algorithm in our scheme. Therefore, no one can get the information of the receiver's private key from the old trapdoors.
- **Anonymousness of the ciphertext:** Similar as unforgeability of the trapdoor, the dPECKS algorithm will choose a random value to protect the keywords. Therefore, no one can get the embedded information from the ciphertext.

Now, we make a comparison with Golle *et al*. (GSW in short) [9], Park *et al*. (PKL in short) [21], Chen and Horng (CH in short) [5], and Zhang and Zhang (ZZ in short) [30]. In this part, we analyze whether the conjunctive keyword searchable scheme conforms the requirements in Section 1, and the result is shown in Table 2. Except GSW, the other schemes are constructed in a public key cryptosystem. We define some notations as follows: Unforg Trap is unforgeable of the trapdoor. Anony Cipher is anonymous of the ciphertext. Inside KG is against inside off-line keyword-guessing attack. Outside KG is against outside off-line keyword-guessing attack. In addition, we do not further discuss the Practicability in this paper, and the Efficiency will be analyzed with the performance in the next section.

We find out that every scheme can achieve unforgeable of the trapdoor and anonymous of the ciphertext since using only the authorized server and receiver's key can produce the legal keyword ciphertext and trapdoors. In the other hand, since the server has to possess plentiful information to execute the Test algorithm, it can perform the off-line keywordguessing attack easily. All of the existing conjunctive keyword searchable schemes cannot stand against the inside keyword-guessing from malicious server. In the schemes constructed in public key system, only our scheme can stand against the outside off-line keyword-guessing attack. Therefore, our scheme is more secure than others.

Table 2. Security comparison

	GSW [9]	PKL [21]	CH [5]	ZZ [30]	Ours
Unforg Trap	○	○	○	○	○
Anony Cipher	○	○	○	○	○
Inside KG	×	×	×	×	×
Ouside KG	○	×	×	×	○

5.3. Performance Analysis

In this section, we analyze the size of the out-puts and the computational costs in the algorithms. Let num be the size in Zp and G_t, $|p|$ be the size in G_1 or G_2, m denote the number of the keyword fields and n stand for the number of documents. E denotes the operation of exponentiation. P denotes the operation of MaptoPoint function which maps a value to an element of G_1 [10]. Comparing with the operation of hash function, a MaptoPoint function generates any amount of operation load which cannot ignore. G_e denotes the operation of the exponentiation of elliptic curve. G_m denotes the operation of elliptic curve. The comparison is shown in Table 3.

Only GSW [9] is constructed in a symmetric cryptosystem. PKL [21], CH [5] and our scheme are a fixed keyword field PECKS schemes. ZZ [30] is a variable keyword field scheme that constructs a polynomial to encrypt the keywords. Although our scheme has a larger computational costs than CH [5] in Encryption and Trapdoor algorithms, our scheme is securer than CH in standing against the outside off-line keyword-guessing attacks. The computational cost of Test algorithm of our scheme is much larger than others. But in general, the server is seemed as the powerful entity that has a huge computational resources; it only has a

Table 3. Performance comparison

	GSW [9]	PKL [21]	CH [5]	ZZ [30]	Ours
$\|pk\|$	-	$3\|p\|$	$3\|p\|$	$(m+5)\|p\|$	$2\|p\|$
$\|sk\|$	num	$2num$	num	num	num
$Encryption$	$(m+1)\|p\|$	$2\|p\| + (m)num$	$(m+1)\|p\|$	$(2m+3)\|p\|$	$(m+2)\|p\|$
$\|Trapdoor\|$	$(n+1)\|p\| + \log m$	$\|p\| + num$	$3\|p\| + \log m$	$(m+3)\|p\|$	$3\|p\| + \log m$
Encryption	$(m+1)G_e$	$(m)P + (m+2)G_m + (m)e$	$(m+1)G_m$	$(2m+4)G_m + e$	$G_e + (m+2)G_m$
Trapdoor	$(n)G_e$	$(m)P + G_m$	$2G_m$	$(m+4)G_e + (m+1)G_m + (m^2)E$	$G_e + 2G_m$
Test	$2G_e + G_m$	e	$2e$	$(2m+3)e$	$5G_e + (m)G_m + 5e$

little influence on the overall performance that the server has largely computational costs.

6. Conclusion

We present an efficient SCF-PECKS scheme that can stand against the off-line keyword-guessing attack. Our scheme is constructed in bilinear pairing based on ElGamal system and the security is under decisional Diffie-Hellman assumption without random oracle. Our scheme is more efficient than other conjunctive keyword searchable schemes and is more suitable for the weak devices. In addition, this scheme can be extended into the multi-user conjunctive keyword search scheme in the future.

Acknowledgments

This work was supported in part by Taiwan Information Security Center (TWISC) and National Science Council under the grant NSC102-2221-E-030-003.

References

[1] **J. Baek, R. S. Naini, W. Susilo.** Public key encryption with keyword search revisited. *ICCSA 2008*, Vol. 5072 of *Lecture Notes in Computer Science*, Perugia, Italy, 2008, pp. 1249-1259.

[2] **D. Boneh, G. D. Crescenzom, R. Ostrovsky, G. Rersiano.** Public key encryption with keyword search. In: *Advances in Cryptology - EUROCRYPT 2004*, *Lecture Notes in Computer Science*, Interlaken, Switzerland, 2004, Vol. 3027, pp. 506-522.

[3] **D. Boneh, B. Waters.** Conjunctive, subset, and range queries on encrypted data. In: *4th Theory of Cryptography Conference, TCC 2007*, Vol. 4392 of *Lecture Notes in Computer Science*, 2007, pp. 535-554.

[4] **J. W. Byun, H. S. Rhee, H. Park, D. H. Lee.** Off-line keyword guessing attacks on recent keyword search schemes over encrypted data. In: *Secure Data Management, Lecture Notes in Computer Science*, Seoul, Korea, 2006, Vol. 4165, pp. 75-83.

[5] **Y. C. Chen, G. Horng.** Timestamped conjunctive keyword-searchable public key encryption. In: *Forth International Conference on Innovation Computing Information and Control (ICICIC)*, 2009, pp. 729-732.

[6] **Z. Chen, C. Wu, D. Wang, S. Li.** Conjunctive keywords searchable encryption with efficient pairing, constant ciphertext and short trapdoor. In: *Proceedings of PAISI 2012*, 2012, Vol. 7299 of *Lecture Notes in Computer Science*, pp. 176-189.

[7] **P. S. Chung, C. W. Liu, M. S. Hwang.** A study of attribute-based proxy re-encryption scheme in cloud environments. *International Journal of Network Security*, 2014, Vol. 16, No. 1, 1-13.

[8] **M. Ding, F. Gao, Z. Jin, H. Zhang.** An efficient public key encryption with conjunctive keyword search scheme based on pairings. In: *2012 3rd IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC)*, 2012, pp. 526-530.

[9] **P. Golle, J. Staddon, B. Waters.** Secure conjunctive keyword search over encrypted data. In: *Proceedings of Applied Cryptography and Network Security Conference*, Vol. 3089 of *Lecture Notes in Computer Science*, 2004, pp. 31-45.

[10] **C. Gu, Y. Zhu.** New efficient searchable encryption schemes from bilinear pairings. *International Journal of Network Security*, 2010, Vol. 10, No. 1, 25-31.

[11] **S. T. Hsu, C. C. Yang, M. S. Hwang.** A study of public key encryption with keyword search. *International Journal of Network Security*, 2013, Vol. 15, No. 2, 71-79.

[12] **Y. H. Hwang, P. J. Lee.** Public key encryption with conjunctive keyword search and its extension to a multi-user system. In: *Pairing-Based Cryptography-Pairing 2007*, 2007, Vol. 4575 of *Lecture Notes in Computer Science*, pp. 2-22.

[13] **J. Kar.** ID-based deniable authentication protocol based on Diffie-Hellman problem on elliptic curve. *International Journal of Network Security*, 2013, Vol. 15, No. 5, 357-364.

[14] **C. C. Lee, P. S. Chung, M. S. Hwang.** A survey on attribute-based encryption schemes of access control in cloud environments. *International Journal of Network Security*, 2013, Vol. 15, 231-240.

[15] **C. C. Lee, S. T. Hsu, M. S. Hwang.** A study of conjunctive keyword searchable schemes. *International Journal of Network Security*, 2013, Vol. 15, 321-330.

[16] **C. C. Lee, T. C. Lin, M. S. Hwang.** A key agreement scheme for satellite communications. *Information Technology and Control*, 2010, Vol. 39, No. 1, 43-47.

[17] **C. C. Lee, I. E. Liao, M. S. Hwang.** An extended certificate-based authentication and security protocol for mobile networks. *Information Technology and Control*, 2009, Vol. 38, No. 1, 61-66.

[18] **C. Ma, J. Ao.** Certificateless group oriented signature secure against key replacement attack. *International Journal of Network Security*, 2011, Vol. 12, No. 1, 1-6.

[19] **Y. Ming, Y. Wang.** An efficient verifiably encrypted signature scheme without random oracles. *International Journal of Network Security*, 2009, Vol. 8, No. 2, 125-130.

[20] **L. Nguyen, R. S. Naini.** Efficient and provably secure trapdoor-free group signature schemes from bilinear pairings. In: *Advances in Cryptology - ASIACRYPT 2004*, Lecture Notes in Computer Science, Vol. 3329, 2004, pp. 372-386.

[21] **D. J. Park, K. Kim, P. J. Lee.** Public key encryption with conjunctive field keyword search. In: *5th Interational Workshop on Information Security Applications, WISA 2004, Vol. 3325 of Lecture Notes in Computer Science*, 2005, pp. 73-86.

[22] **H. S. Rhee, J. H. Park, W. Susilo, D. H. Lee.** Improved searchable public key encryption with designated tester. In: *ASIACCS '09 Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, Sydney, NSW, Australia, 2009, pp. 376-379.

[23] **E. K. Ryu, T. Takagi.** Efficient conjunctive keyword-searchable encryption. In: *21st International Conference on Advanced Information Networking and Application, AINAW '07*, Vol. 1, 2007, pp. 409-414.

[24] **D. X. Song, D. Wagner, A. Perrig.** Practical techniques for searches on encrypted data. In: *2000 IEEE Symposium on Security and Privacy*, 2000, pp. 44-55.

[25] **T. T. Tsai, Y. M. Tseng, T. Y. Wu.** Efficient revocable multi-receiver ID-based encryption. *Information Technology and Control*, 2013, Vol. 42, No. 2, 159-169.

[26] **A. H. P. Van Vliet.** Secure Data Storage Outsourcing with Conjunctive Keyword Search. Thesis, Delft University of Technology, 2009.

[27] **A. Venčkauskas, N. Jusas, I. Mikuckiene, S. Maciulevičius.** Generation of the secret encryption key using the signature of the embedded system. *Information Technology and Control*, 2012, Vol. 41, No. 4, 368-375.

[28] **P. Wang, H. Wang, J. Pieprzyk.** Threshold privacy prieserving keyword searches. In: *SOFSEM 2008: Theory and Practice of Computer Science*, Vol. 4910 of *Lecture Note in Computer Science*, 2008, pp. 646-658.

[29] **J. H. Yang, Y. F. Chang, Y. H. Chen.** An efficient authenticated encryption scheme based on ECC and its application for electronic payment. *Information Technology and Control*, 2013, Vol. 42, No. 4, 315-324.

[30] **B. Zhang, F. Zhang.** An efficient public key encryption with conjunctive-subset keywords search. *Journal of Network and Computer Application*, 2011, Vol. 34, No. 1, 262-267.

Received February 2014.