A Secure Channel Free Public Key Encryption with Keyword Search Scheme without Random Oracle

Liming Fang[1], Willy Susilo[2], Chunpeng Ge[1], and Jiandong Wang[1]

[1] College of Information Science and Technology
Nanjing University of Aeronautics and Astronautics, Nanjing, China
{fangliming,gecp}@nuaa.edu.cn
[2] Centre for Computer and Information Security Research
School of Computer Science and Software Engineering
University of Wollongong, Australia
wsusilo@uow.edu.au

Abstract. The public key encryption with keyword Search (PEKS) scheme, proposed by Boneh, Di Crescenzo, Ostrovsky and Persiano, enables one to search for encrypted keywords without compromising the security of the original data. Baek *et al.* noticed that the original notion of PEKS requires the existence of a secure channel, and they further extended this notion by proposing an efficient secure channel free public key encryption scheme with keyword search in the random oracle model. In this paper, we take one step forward by adopting Baek *et al.*'s model and propose a new and efficient scheme that does not require any secure channels, and furthermore, its security does not use random oracles.

Keywords: public key encryption with keyword search, searchable encryption, without random oracle.

1 Introduction

The public key encryption with keyword search (PEKS) scheme, proposed by Boneh *et al.* [4], enables one to search for encrypted keywords without revealing the security of the original data. PEKS schemes can be widely used and deployed in many practical applications. An interesting application of PEKS is intelligent email routing. Suppose Bob sends an encrypted email to Alice using Alice's public key. The contents of the email comprises a header information, a body of the email and a list of keywords that are encrypted. In this case, the mail gateway cannot observe the header information (as well as the body and the keywords attached) and hence, cannot make routing decisions. Alice uses different electronic devices to read her email, such as an iPhone or a PDA and a desktop computer. Alice may prefer emails to be routed to her devices depending on the associated keywords. For example, she may like to receive emails with the keyword "urgent" on her iPhone, meanwhile all other emails can be sent to her desktop computer instead. In particular, when Alice is away on holiday, she only wants to read emails with the keyword "urgent" that will require her urgent attention, instead of reading all of her work emails.

In short, PEKS provides a mechanism that enables Alice to provide the gateway with the ability to test whether "urgent" is a keyword in the original email, but additionally, the gateway should learn nothing else about the email itself. More generally, the mail gateway can search the keywords required, but learn nothing else.

Waters *et al.* [16] introduced another interesting application of PEKS schemes, which is to let an untrusted logging device to maintain an encrypted audit log of privacy-sensitive data that is efficiently searchable by authorized auditors only. The entries in the audit log are encrypted under the public key of a PEKS scheme, of which the corresponding secret key is unknown to the logging device. If the device is ever confiscated, or if the logbook leaks, privacy of users and their actions is maintained. The secret key is known only to a trusted audit escrow agent, who provides (less trusted) authorized investigators with trapdoors for the keywords they want to search for.

Related Work
Despite a large number of research work related to the privacy of database data, Boneh *et al.* [4] noted that PEKS is different from the previously known solutions. Unlike the private-key setting, data collected by the mail server is from third parties, and can not be "organized" by the user in any convenient way. Furthermore, unlike the publicly available database, the data is not public, and hence the Public Information Retrieval (PIR) solutions will not be applicable. Shortly after Boneh *et al.*'s pioneering work, Waters *et al.* [16] showed that the PEKS scheme based on the bilinear pairing can be applied to build encrypted and searchable audit logs. Golle *et al.* [12] proposed schemes that allow for conjunctive keyword queries on encrypted data. Boneh and Waters [5] extended PEKS to support conjunctive, subset, and range comparisons over the keywords. Furthermore, the subsequent papers [9,18] investigated the secure combination of public key encryption with keyword search (PEKS) with public key data encryption (PKE). Since the fact that keywords are chosen from much smaller space than passwords and users usually use well-known keywords for search, the work in [6,13,15,17] studied the off-line keyword guessing attacks on PEKS.

The drawback of the Boneh *et al.*'s scheme [4] is that it uses a *secure channel* between Alice and the email server, which is usually costly [2]. Baek *et al.* [2] proposed an alternative solution to eliminate the need for a secure channel, by proposing public key encryption with keyword search scheme with a designated server. Throughout this paper, we refer this model as SCF-PEKS, which refers to *Secure Channel-Free PEKS*.

In 2007, Gu *et al.* [11] proposed an interesting construction of PEKS scheme based on pairings, in which there is no pairing operation involved in the encryption procedure. Then, they provided further discussion on removing the secure channel from PEKS, and presented an efficient secure channel free PEKS scheme. Rhee *et al.* [14] enhanced the Baek *et al.*'s security model [2] for SCF-PEKS in which an attacker is also allowed to obtain the relation between non-challenge ciphertexts and a trapdoor. They presented an SCF-PEKS scheme secure in the enhanced security model. The limitation of the existing schemes in the literature is that their security can only be guaranteed in the random oracle model. Unfortunately, a proof in the random oracle model has been shown to possibly lead to insecure schemes when the random oracles are implemented in the standard model [7]. Therefore, it is desirable to have a solution that does not require the random oracle model.

Our Contributions
In this paper, we present an efficient and secure channel free public key encryption with keyword search scheme. Based on the DBDH assumption and the truncated q-ABDHE assumption, we prove its indistinguishability of secure channel free PEKS against chosen keyword attack (IND-SCF-CKA) security without random oracle.

To summarize the existing knowledge, the three constructions of SCF-PEKS schemes are due to Baek *et al.* [2], Gu *et al.* [11] and Rhee *et al.*[14] that require random oracle model. Our work fills the gap in the literature by proposing an *efficient* secure channel free PEKS (SCF-PEKS) scheme that does not incorporate random oracle, as outlined in Table 1.

Table 1. Comparison Among Various SCF-PEKS Schemes

Properties	**Baek *et al.*** [2]	**Gu *et al.*** [11]	**Rhee *et al.***[14]	**This paper**
Without ROM	×	×	×	✓
Assumption	BDH	BDH, q-BDHI	BDH, q-BDHI	DBDH, q-ABDHE

Paper Organization
The rest of this paper is organized as follows. In the next section, we will present some definitions and notations that will be used throughout this paper. In Section 3, we present our new and efficient scheme and analyze its security. Finally, Section 4 concludes the paper.

2 Definitions

In this section, we firstly review the complexity assumptions required in our schemes, and then provide the definition and security of a public key encryption with keyword search scheme.

2.1 Negligible Function

A function $\varepsilon(n) : N \rightarrow R$ is negligible in n if $1/\varepsilon(n)$ is a non-polynomially-bounded quantity in n.

2.1 Bilinear Maps

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be multiplicative cyclic groups of prime order p, and g be a generator of $\mathbb{G}_1$. (By ${\mathbb{G}_1}^*$ and ${\mathbb{Z}_p}^*$, we denote $\mathbb{G}_1\backslash\{1\}$ where 1 is the identity element of $\mathbb{G}_1$, and $\mathbb{Z}_p\backslash\{0\}$ respectively). We say $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a bilinear map [3] if the following conditions hold.

1. $e({g_1}^a, {g_2}^b) = e(g_1, g_2)^{ab}$ for all $a, b \in \mathbb{Z}_p$ and $g_1, g_2 \in \mathbb{G}_1$.
2. $e(g, g) \neq 1$.
3. There is an efficient algorithm to compute $e(g_1, g_2)$ for all $g_1, g_2 \in \mathbb{G}_1$.

2.2 The DBDH Assumption

Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a bilinear map. We define the advantage function

$$Adv_{\mathbb{G}_1,\mathcal{B}}{}^{DBDH}(\lambda)$$

of an adversary $\mathcal{B}$ as

$$|Pr[\mathcal{B}(g, g^a, g^b, g^c, e(g,g)^{abc}) = 1] - Pr[\mathcal{B}(g, g^a, g^b, g^c, e(g,g)^r) = 1]|$$

where $a, b, c, r \in \mathbb{Z}_p$ are randomly chosen. We say that the decisional bilinear Diffie Hellman assumption [3] relative to generator $\mathbb{G}_1$ holds if $Adv_{\mathbb{G}_1,\mathcal{B}}{}^{DBDH}(\lambda)$ is negligible for all PPT $\mathcal{B}$.

2.3 The Truncated (Decisional) *q*-ABDHE Assumption

Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a bilinear map. We define the advantage function

$$Adv_{\mathbb{G}_1,\mathcal{B}}{}^{q-ABDHE}(\lambda)$$

of an adversary $\mathcal{B}$ as

$$\begin{aligned}|Pr[\mathcal{B}(g, g^x, \cdots, g^{x^q}, g^z, g^{zx^{q+2}}, e(g,g)^{zx^{q+1}}) = 1] - \\ Pr[\mathcal{B}(g, g^x, \cdots, g^{x^q}, g^z, g^{zx^{q+2}}, e(g,g)^r) = 1]|\end{aligned}$$

where $x, z, r \in \mathbb{Z}_p$ are randomly chosen. We say that the truncated decisional augmented bilinear Diffie-Hellman exponent assumption [10] relative to generator $\mathbb{G}_1$ holds if $Adv_{\mathbb{G}_1,\mathcal{B}}{}^{q-ABDHE}(\lambda)$ is negligible for all PPT $\mathcal{B}$.

2.4 Definition of SCF-PEKS

In the following, we will provide the definition of a SCF-PEKS scheme [2] and the game-based security definition model.

Definition 1 (SCF-PEKS). *A secure channel free public key encryption with keyword search scheme comprises the following algorithms:*

- $GlobalSetup(\lambda)$: *Takes a security parameter* λ *generates a global parameter* $\mathcal{GP}$.
- $KeyGen_{Server}(\mathcal{GP})$: *Takes as input the common parameters* $\mathcal{GP}$. *Output the public/secret pair* $(pk_{\mathcal{S}}, sk_{\mathcal{S}})$ *of server* $\mathcal{S}$.
- $KeyGen_{Receiver}(\mathcal{GP})$: *Takes as input* $\mathcal{GP}$, *generates public/secret pair* $(pk_{\mathcal{R}}, sk_{\mathcal{R}})$ *of receiver* $\mathcal{R}$.
- $SCF - PEKS(\mathcal{GP}, pk_{\mathcal{S}}, pk_{\mathcal{R}}, w)$: *Takes as input* $\mathcal{GP}$, *a receiver's public key* $pk_{\mathcal{R}}$, *a server's public key* $pk_{\mathcal{S}}$, *and a keyword* w. *Return a PEKS ciphertext* C *under* w.
- $Trapdoor(\mathcal{GP}, sk_{\mathcal{R}}, w)$: *Takes as input* $\mathcal{GP}$, *a receiver's secret key* $sk_{\mathcal{R}}$ *and a keyword* w. *Generate a trapdoor* T_w.

– $Test(\mathcal{GP}, T_w, sk_{\mathcal{S}}, C)$: *Takes as input a common parameter* $\mathcal{GP}$*, a trapdoor* T_w*, a server's secret key* $sk_{\mathcal{S}}$ *and a PEKS ciphertext* $C = SCF-PEKS(\mathcal{GP}, pk_{\mathcal{S}}, pk_{\mathcal{R}}, w')$*. Output a symbol "Correct" if* $w = w'$ *or "Incorrect" otherwise.*

We define the notion of consistency in a SCF-PEKS scheme, which is similar to the notion of consistency in a PEKS scheme from [1].

Definition 2 (Consistency). *Suppose there exits an adversary* $\mathcal{A}$ *that wants to make consistency fail. The consistency is formally defined as follows:*

```
Experiment $Exp_{\mathcal{A}}{}^{cons}(\lambda)$
    $(pk_{\mathcal{R}}, sk_{\mathcal{R}}) \leftarrow KeyGen_{Receiver}(\lambda)$; $(pk_{\mathcal{S}}, sk_{\mathcal{S}}) \leftarrow KeyGen_{Server}(\lambda)$
    $(w, w') \leftarrow \mathcal{A}(pk_{\mathcal{R}}, pk_{\mathcal{S}})$
    $C \leftarrow SCF - PEKS(\mathcal{GP}, pk_{\mathcal{S}}, pk_{\mathcal{R}}, w)$; $T_{w'} \leftarrow Trapdoor(\mathcal{GP}, sk_{\mathcal{R}}, w')$
    if $w \neq w'$ and $Test(\mathcal{GP}, T_{w'}, sk_{\mathcal{S}}, C) =$ "Correct",
        then return 1,
        else return 0.
```

We define the advantage of $\mathcal{A}$ *as:*

$$Adv_{\mathcal{A}}{}^{cons}(\lambda) = Pr[Exp_{\mathcal{A}}{}^{cons}(\lambda) = 1]$$

The scheme is said to be computationally consistent if it is negligible for polynomial time adversaries $\mathcal{A}$ *to win the above experiment.*

In the following, we introduce the game-based security definition of SCF-PEKS, which we call indistinguishability of secure channel free PEKS against chosen keyword attack (IND-SCF-CKA). Informally, IND-SCF-CKA guarantees that the server that has not obtained the trapdoors for given keywords cannot tell which PEKS ciphertext encrypts which keyword, and the outside attacker that has not obtained the server's private key cannot make any decisions about the PEKS ciphertexts even though the attacker gets all the trapdoors for the keywords that it holds. Our definition is adopted from the definition by Baek *et al.* in [2]. Note that the attack models for these two types of attackers are described as Game 1 and Game 2, respectively, in the following definition.

Definition 3 (IND-SCF-CKA game). *Let* λ *be the security parameter and* $\mathcal{A}$ *be the adversary. We consider the following two games between* $\mathcal{A}$ *and the simulator* $\mathcal{B}$*.*
Game 1: $\mathcal{A}$ *is assumed to be a server.*

1. *Setup: The common parameter generation algorithm* $GlobalSetup(\lambda)$*, the two key generation algorithms* $KeyGen_{Server}(\mathcal{GP})$ *and* $KeyGen_{Receiver}(\mathcal{GP})$ *are executed. A common parameter* $\mathcal{GP}$*, private and public key pairs of the receiver and the server, which we denote by* $(pk_{\mathcal{R}}, sk_{\mathcal{R}})$ *and* $(pk_{\mathcal{S}}, sk_{\mathcal{S}})$ *respectively, are set. Then,* $\mathcal{B}$ *sends* $(pk_{\mathcal{S}}, sk_{\mathcal{S}})$ *and* $pk_{\mathcal{R}}$ *to* $\mathcal{A}$*.*
2. *Query phase 1.* $\mathcal{A}$ *makes the queries a number of keywords, each of which is denoted by* w*:*
   – *Trapdoor query* $\langle w \rangle$*:* $\mathcal{A}$ *can adaptively asks* $\mathcal{B}$ *for the trapdoor* T_w *for any keyword* $w \in KS_w$ *of his choice.* $\mathcal{B}$ *responds the trapdoor* $T_w = Trapdoor(\mathcal{GP}, sk_{\mathcal{R}}, w)$ *to* $\mathcal{A}$*.*

3. *Challenge. Once $\mathcal{A}$ decides that Phase 1 is over, it outputs a target key pair (w_0, w_1). (Notice that none of w_0 nor w_1 has been queried for obtaining a corresponding trapdoor in Phase 1). Upon receiving this, $\mathcal{B}$ responds by choosing a random $b \in \{0,1\}$, and creates a target PEKS ciphertext $C^* = SCF-PEKS(\mathcal{GP}, pk_{\mathcal{S}}, pk_{\mathcal{R}}, w_b)$ and sends it to $\mathcal{A}$.*
4. *Query phase 2. $\mathcal{A}$ issues a number of trapdoor queries as in Phase 1. The restriction here is that w_0 and w_1 are not allowed to be queried as trapdoor queries.*
5. *Guess. $\mathcal{A}$ outputs the guess b'. The adversary wins if $b' = b$.*

We define $\mathcal{A}$'s advantage in Game 1 by $Adv_{\mathcal{A}}{}^{Game_1}(\lambda) = |Pr[b = b'] - 1/2|$.

Game 2: *$\mathcal{A}$ is assumed to be an outside attacker (including the receiver).*

1. *Setup: The common parameter generation algorithm $GlobalSetup(\lambda)$, the two key generation algorithms $KeyGen_{Server}(\mathcal{GP})$ and $KeyGen_{Receiver}(\mathcal{GP})$ are executed. A common parameter $\mathcal{GP}$, private and public key pairs of the receiver and the server, which we denote by $(pk_{\mathcal{R}}, sk_{\mathcal{R}})$)and $(pk_{\mathcal{S}}, sk_{\mathcal{S}})$ respectively, are set. Then, $\mathcal{B}$ sends $(pk_{\mathcal{R}}, sk_{\mathcal{R}})$ and $pk_{\mathcal{S}}$ to $\mathcal{A}$.*
2. *Challenge. $\mathcal{A}$ outputs a target keyword pair (w_0, w_1). Upon receiving this, $\mathcal{B}$ responds by choosing a random $b \in \{0,1\}$, and creates a target PEKS ciphertext $C^* = SCF - PEKS(\mathcal{GP}, pk_{\mathcal{S}}, pk_{\mathcal{R}}, w_b)$ and sends it to $\mathcal{A}$.*
3. *Guess. $\mathcal{A}$ outputs the guess b'. The adversary wins if $b' = b$.*

We define $\mathcal{A}$'s advantage in Game 2 by $Adv_{\mathcal{A}}{}^{Game_2}(\lambda) = |Pr[b = b'] - 1/2|$. The SCF-PEKS scheme is said to be IND-SCF-CKA secure if $Adv_{\mathcal{A}}{}^{Game_i}(\lambda)$, where i is either 1 or 2, is negligible.

3 New SCF-PEKS Scheme

In this section, we will present our efficient construction of public key encryption with keyword search scheme without random oracle. Our scheme is based on Gentry's IBE in the standard model.

3.1 Our Construction

Our public key encryption with keyword search scheme is described as follows.

- $GlobalSetup(\lambda)$*:* Let λ be the security parameter and $(p, g, \mathbb{G}_1, \mathbb{G}_2, e)$ be the bilinear map parameters. Select a one-way hash function $H : \{0,1\}^* \rightarrow \mathbb{Z}_p{}^*$. The keyword space $KS_w = \mathbb{Z}_p{}^*$. The global parameters are $\mathcal{GP} = (p, g, \mathbb{G}_1, \mathbb{G}_2, e, H, KS_w)$.
- $KeyGen_{Server}(\mathcal{GP})$*:* Choose $x \in \mathbb{Z}_p{}^*$ uniformly at random and compute $X = g^x$. Choose $Q \in \mathbb{G}_1{}^*$ uniformly at random. Return $pk_{\mathcal{S}} = (\mathcal{GP}, Q, X)$ and $sk_{\mathcal{S}} = (pk_{\mathcal{S}}, x)$ as the server's public and private key, respectively.
- $KeyGen_{Receiver}(\mathcal{GP})$*:* Choose $y \in \mathbb{Z}_p{}^*$ uniformly at random and compute $Y = g^y$. Choose $h \in \mathbb{G}_1{}^*$ uniformly at random. Return $pk_{\mathcal{R}} = (pk_{\mathcal{S}}, Y, h)$ and $sk_{\mathcal{R}} = (pk_{\mathcal{R}}, y)$ as the receiver's public and private key, respectively.

- $SCF-PEKS(\mathcal{GP}, pk_{\mathcal{S}}, pk_{\mathcal{R}}, w)$: Choose $s, r \in \mathbb{Z}_p{}^*$ and compute $C_1 = g^s$, $t = H(e(X,Q)^s)$, $C_2 = (Yg^{-w})^{r/t}$, $C_3 = e(g,g)^r$, $C_4 = e(g,h)^r$. The PEKS ciphertext is $C = (C_1, C_2, C_3, C_4)$. Return C.
- $Trapdoor(\mathcal{GP}, sk_{\mathcal{R}}, w)$: Choose $s_w \in \mathbb{Z}_p{}^*$ and compute $d_w = (hg^{-s_w})^{1/(y-w)}$. Let the trapdoor $T_w = (d_w, s_w)$. Return T_w.
- $Test(\mathcal{GP}, T_w, sk_{\mathcal{S}}, C)$: Compute $t = H(e(C_1, Q)^x)$, check if $e(C_2{}^t, d_w)C_3{}^{s_w} = C_4$. If the equation holds return "Correct". Otherwise, return "Incorrect".

Correctness. In the following, we show that a correctly generated PEKS ciphertext can be correctly tested by the server who has the correct trapdoor. In the following, let a PEKS ciphertext $C = (C_1, C_2, C_3, C_4)$ associated with keyword w under the public key $pk_{\mathcal{S}}, pk_{\mathcal{R}}$. Let the trapdoor $T_w = (d_w, s_w)$. We have

$$\begin{aligned} t &= H(e(C_1, Q)^x) \\ &= H(e(g^s, Q)^x) \\ &= H(e(g^x, Q)^s) \\ &= H(e(X, Q)^s). \\ e(C_2{}^t, d_w)C_3{}^{s_w} &= e(((Yg^{-w})^{r/t})^t, (hg^{-s_w})^{1/(y-w)})(e(g,g)^r)^{s_w} \\ &= e(g^{(y-w)r}, (hg^{-s_w})^{1/(y-w)})e(g,g)^{rs_w} \\ &= e(g,h)^r e(g^r, g^{-s_w})e(g,g)^{rs_w} \\ &= C_4. \end{aligned}$$ ∎

3.2 Consistency of Our SCF-PEKS

In this subsection, we prove the computational consistency of our scheme.

Theorem 1. *Our SCF-PEKS scheme is computationally consistent.*

Proof. Suppose there exists a polynomial-time adversary, $\mathcal{A}$, that can attack computational consistency of our scheme. Let (w, w') denote the pair of keywords that $\mathcal{A}$ returns in the consistency experiment, and assume without loss of generality that $w \neq w'$.

Let $s, r \in \mathbb{Z}_p{}^*$ denote the value chosen at random by $SCF-PEKS(\mathcal{GP}, pk_S, pk_R, w)$. Let $h = g^z$, $C_1 = g^s$, $t = H(e(X,Q)^s)$, $C_2 = (Yg^{-w})^{r/t}$, $C_3 = e(g,g)^r$, $C_4 = e(g,h)^r$.

Let $T_w = (d_{w'}, s_{w'})$ where $d_{w'} = (hg^{-s_{w'}})^{1/(y-w')} = g^{(z-s_{w'})/(y-w')}$ be the trapdoor of w'.

Note that $\mathcal{A}$ wins exactly when $w \neq w'$ and $e(C_2{}^t, d_{w'})C_3{}^{s_{w'}} = C_4$.

$$\begin{aligned} & e(C_2{}^t, d_{w'})C_3{}^{s_{w'}} = C_4 \\ \iff & e(((Yg^{-w})^{r/t})^t, g^{(z-s_{w'})/(y-w')})e(g,g)^{rs_{w'}} = e(g,g)^{zr} \\ \iff & e(g^{(y-w)r}, g^{(z-s_{w'})/(y-w')})e(g,g)^{rs_{w'}} = e(g,g)^{zr} \\ \iff & e(g,g)^{((y-w)/(y-w'))(z-s_{w'})r}e(g,g)^{rs_{w'}} = e(g,g)^{zr} \\ \iff & e(g,g)^{((y-w)/(y-w'))zr}e(g,g)^{-((y-w)/(y-w'))s_{w'}r}e(g,g)^{rs_{w'}} = e(g,g)^{zr} \\ \iff & ((y-w)/(y-w'))zr - ((y-w)/(y-w'))s_{w'}r + rs_{w'} = zr \\ \iff & ((y-w)/(y-w') - 1)zr - ((y-w)/(y-w') - 1)s_{w'}r = 0 \\ \iff & ((w'-w)/(y-w'))zr - ((w'-w)/(y-w'))s_{w'}r = 0 \\ \iff & ((w'-w)/(y-w'))(z-s_{w'})r = 0. \end{aligned}$$

Since y, z is receiver's unknown secret key in $\mathbb{Z}_p{}^*$. Therefore, $Pr[s_{w'} = z] = 1/(p-1)$ and $Pr[w' = y] = 1/(p-1)$ where $p-1$ is the total element number in $\mathbb{Z}_p{}^*$.

As described above, under the condition $w \neq w'$ and $Test(\mathcal{GP}, T_{w'}, sk_{\mathcal{S}}, C) =$ "Correct"

$$Adv_{\mathcal{A}}{}^{cons}(\lambda) = Pr[Exp_{\mathcal{A}}{}^{cons}(\lambda) = 1] = Pr[(s_{w'} = z) \vee (w' = y)] \leq 2/(p-1).$$

3.3 Security of Our SCF-PEKS

In this subsection, we analyze the security of our SCF-PEKS scheme without requiring any random oracle. The analysis of Game 1 and Game 2 is as follows.

Theorem 2. *The above scheme is IND-SCF-CKA secure without random oracle model assuming that the DBDH problem and q-ABDHE problem are intractable.*

Lemma 1. Let $q \geq q_k + 1$, where q_k is the number of trapdoor queries. Our scheme is semantically secure against a chosen keyword attack in Game 1 without random oracle model assuming q-ABDHE problem is intractable.
Proof. Suppose there exists a polynomial-time adversary, $\mathcal{A}$, in Game 1 that can attack our scheme in the standard model. Let q_k is the number of trapdoor queries. We build a simulator $\mathcal{B}$ that can play a q-ABDHE game. The simulation proceeds as follows:

We first let the challenger set the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ with an efficient bilinear map e and a generator g of $\mathbb{G}_1$. Simulator $\mathcal{B}$ 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 $T = e(g,g)^{zx^{q+1}}$ from a random element in $\mathbb{G}_2$.

1. Setup: Let λ be the security parameter and $(p, g, \mathbb{G}_1, \mathbb{G}_2, e)$ be the bilinear map parameters. Specify a one-way hash function $H : \{0,1\}^* \rightarrow \mathbb{Z}_p{}^*$. Let the keyword space be $KS_w = \mathbb{Z}_p{}^*$. The global parameters are $\mathcal{GP} = (p, g, \mathbb{G}_1, \mathbb{G}_2, e, H, KS_w)$. Choose $a \in \mathbb{Z}_p{}^*$ uniformly at random and compute $X = g^a$. Choose $Q \in \mathbb{G}_1{}^*$ uniformly at random. Let $pk_{\mathcal{S}} = (\mathcal{GP}, Q, X)$ and $sk_{\mathcal{S}} = (\mathcal{GP}, a)$ as the server's public and private key, respectively.
   Pick a random degree q polynomials $f(X)$ and define $Y = g^x, h = g^{f(x)}$. The receiver's public is $pk_{\mathcal{R}} = (pk_{\mathcal{S}}, Y, h)$. Then, send $(pk_{\mathcal{R}}, pk_{\mathcal{S}}, sk_{\mathcal{S}})$ to $\mathcal{A}$.
2. Query phase 1. $\mathcal{A}$ makes trapdoor query :
   - Trapdoor query $\langle w \rangle$: If $\mathcal{A}$ queries w to the trapdoor generation oracle, then $\mathcal{B}$ sets $s_w = f(w)$, computes $d_w = g^{(f(x)-f(w))/(x-w)}$, sends the trapdoor $T_w = \{d_w, s_w\}$ to $\mathcal{A}$. When $q \geq q_k + 1$, $s_w = f(w)$ is random value from $\mathcal{A}$'s view, since $f(X)$ is a random degree q polynomial.
3. Challenge. Once $\mathcal{A}$ decides that Phase 1 is over, it outputs a keyword pair (w_0, w_1). $\mathcal{B}$ responds by choosing a random $b \in \{0,1\}$, let $w^* = w_b$ and sets $\{s_{w^*} = f_k(w^*)\}$, then $\mathcal{B}$ computes $d_{w^*} = g^{(f(x)-f(w^*))/(x-w^*)}$. $\mathcal{B}$ randomly chooses $s^* \in \mathbb{Z}_p{}^*$ and computes $C_1{}^* = g^{s^*}$, $t^* = H(e(X,Q)^{s^*})$. Defines the degree $q+1$ polynomial $F^*(X) = (X^{q+2} - (w^*)^{q+2})/(X - w^*) = \sum_{i=0}{}^{q+1}(F_i{}^* X^i)$.
   Computes

$$C_2{}^* = (g^{zx^{q+2}}(g^z)^{-(w^*)^{q+2}})^{1/t^*}$$
$$C_3{}^* = T^{F_{q+1}{}^*} e(g^z, \prod_{i=0}{}^{q}(g^{x^i})^{F_i{}^*})$$
$$C_4{}^* = e((C_2{}^*)^{t^*}, d_{w^*})(C_3{}^*)^{s_{w^*}}.$$

Sends the target PEKS ciphertext $C^* = (C_1{}^*, C_2{}^*, C_3{}^*, C_4{}^*)$ to $\mathcal{A}$.
Let $r^* = zF^*(x)$, if $T = e(g,g)^{zx^{q+1}}$, then
$C_2{}^* = (g^{zx^{q+2}}(g^z)^{-(w^*)^{q+2}})^{1/t^*} = g^{(x-w^*)(z(x^{q+2}-(w^*)^{q+2})/(x-w^*))1/t^*} = g^{(x-w^*)r^*/t^*} = (Yg^{-w^*})^{r^*/t^*}$
$C_3{}^* = T^{F_{q+1}{}^*} e(g^z, \prod_{i=0}{}^q (g^{x^i})^{F_i{}^*}) = e(g,g)^{r^*}$
$C_4{}^* = e(g,h)^{r^*}$.

4. Query phase 2. $\mathcal{A}$ continues making queries as in the Query phase 1.
5. Guess. $\mathcal{A}$ outputs the guess b', if $b' = b$, then output 1 meaning $T = e(g,g)^{zx^{q+1}}$; else output 0 meaning $T = e(g,g)^r$.

Probability Analysis: If $T = e(g,g)^{zx^{q+1}}$, then the simulation is perfect, and $\mathcal{A}$ will guess the bit b correctly with probability $1/2+\varepsilon$. Else, T is uniformly random, and thus $(C_2{}^*, C_3{}^*)$ is a uniformly random and independent element. In this case, the inequality $C_3{}^* \neq e((C_2{}^*)^{t^*}, g)^{1/(x-w^*)}$ holds with probability $1 - 1/p$. When the inequality holds, the value of

$$\begin{aligned} C_4{}^* &= e((C_2{}^*)^{t^*}, d_{w^*})(C_3{}^*)^{s_{w^*}} \\ &= e((C_2{}^*)^{t^*}, (h)^{1/(x-w^*)})((C_3{}^*)/(e((C_2{}^*)^{t^*}, g)^{1/(x-w^*)}))^{s_{w^*}} \end{aligned}$$

is uniformly random and independent from $\mathcal{A}$'s view (except for the value $C_4{}^*$), since s_{w^*} is uniformly random (when $q \geq q_k + 1$, $s_{w^*} = f(w^*)$ are random values from $\mathcal{A}$'s view) and independent from $\mathcal{A}$'s view (except for the value $C_3{}^*$). Thus, $C_4{}^*$ is uniformly random and independent. Since $s^* \in \mathbb{Z}_p{}^*$ is randomly chosen, $C_1{}^* = g^{s^*}$ is uniformly random and independent from $(C_2{}^*, C_3{}^*, C_4{}^*)$ and $(C_1{}^*, C_2{}^*, C_3{}^*, C_4{}^*)$ can reveal no information regarding the bit b. This completes the proof of Game 1. ∎

Lemma 2. Our scheme is semantically secure against a chosen keyword attack in Game 2 without random oracle model assuming DBDH problem is intractable.
Proof. Suppose there exists a polynomial-time adversary, $\mathcal{A}$, in Game 2 that can attack our scheme in the standard model. We build a simulator $\mathcal{B}$ that can play a DBDH game. The simulation proceeds as follows:

We first let the challenger set the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ with an efficient bilinear map e and a generator g of $\mathbb{G}_1$. $\mathcal{B}$ inputs a DBDH instance (g, g^a, g^b, g^c, T), and has to distinguish $T = e(g,g)^{abc}$ from a random element in $\mathbb{G}_2$.

1. Setup: Let λ be the security parameter and $(p, g, \mathbb{G}_1, \mathbb{G}_2, e)$ be the bilinear map parameters. Specify a one-way hash function $H : \{0,1\}^* \rightarrow \mathbb{Z}_p{}^*$. The global parameters are $\mathcal{GP} = (p, g, \mathbb{G}_1, \mathbb{G}_2, e, H, KS_w)$ where KS_w denotes a description of a keyword space.
Let $X = g^a$ and $Q = g^b$, the server's public key is $pk_{\mathcal{S}} = (\mathcal{GP}, Q, X)$.
Choose $y \in \mathbb{Z}_p{}^*$ uniformly at random and compute $Y = g^y$. Choose $h \in \mathbb{G}_1{}^*$ uniformly at random. $pk_{\mathcal{R}} = (pk_{\mathcal{S}}, Y, h)$ and $sk_R = (pk_{\mathcal{S}}, y)$ denote the receiver's public and private key respectively. Then, send $(pk_{\mathcal{R}}, sk_{\mathcal{R}}, pk_{\mathcal{S}})$ to $\mathcal{A}$.
2. Challenge. $\mathcal{A}$ outputs a key pair (w_0, w_1). $\mathcal{B}$ responds by choosing a random $b \in \{0,1\}$, let the target keyword $w^* = w_b$, $C_1{}^* = g^c$ and $t^* = H(T)$, chooses $r \in \mathbb{Z}_p{}^*$, computes $C_2{}^* = (Yg^{-w^*})^{r/t^*}$, $C_3{}^* = e(g,g)^r$, $C_4{}^* = e(g,h)^r$. The PEKS ciphertext is $C^* = (C_1{}^*, C_2{}^*, C_3{}^*, C_4{}^*)$. Sends C^* to $\mathcal{A}$.

3. Guess. $\mathcal{A}$ outputs the guess b', if $b' = b$, then output 1 meaning $T = e(g, g)^{abc}$; else output 0 meaning $T = e(g, g)^r$.

Probability Analysis: Suppose there exists a polynomial-time adversary, $\mathcal{A}$, in Game 2 that can attack our scheme in the standard model with an advantage ε. Now we provide the probability of the simulator $\mathcal{B}$:

When $T = e(g, g)^{abc}$ then $\mathcal{A}$ must satisfy $|Pr[b = b'] - 1/2| \geq \varepsilon$. When T is uniform in $\mathbb{G}_2{}^*$ then $Pr[b = b'] = 1/2$. Therefore, when a, b, c are uniform in $\mathbb{Z}_p{}^*$ and T is uniform in $\mathbb{G}_2{}^*$. We have that

$|Pr[\mathcal{B}(g, g^a, g^b, g^c, e(g, g)^{abc}) = 1]$- $Pr[\mathcal{B}(g, g^a, g^b, g^c, e(g, g)^r) = 1]| \geq |(1/2 \pm \varepsilon) - 1/2| = \varepsilon$ as required. This completes the proof of Game 2. ■

4 Conclusion and Future Work

In this paper, we constructed an efficient and secure channel free public key encryption with keyword search scheme without random oracle. This construction fills the gap in the literature that an efficient and secure channel free public key encryption with keyword search can be built without requiring any random oracle model.

References

1. Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., Shi, H.: Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 205–222. Springer, Heidelberg (2005)
2. Baek, J., Safavi-Naini, R., Susilo, W.: Public key encryption with keyword search revisited. In: Gervasi, O., Murgante, B., Laganà, A., Taniar, D., Mun, Y., Gavrilova, M.L. (eds.) ICCSA 2008, Part I. LNCS, vol. 5072, pp. 1249–1259. Springer, Heidelberg (2008)
3. Boneh, D., Boyen, X.: Efficient selective-ID Identity based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
4. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004)
5. Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 535–554. Springer, Heidelberg (2007)
6. Byun, J.W., Rhee, H.S., Park, H.-A., Lee, D.-H.: Off-line keyword guessing attacks on recent keyword search schemes over encrypted data. In: Jonker, W., Petković, M. (eds.) SDM 2006. LNCS, vol. 4165, pp. 75–83. Springer, Heidelberg (2006)
7. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. In: Proc. of 30th ACM STOC, pp. 209–218. ACM Press, New York (1998)
8. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
9. Fuhr, T., Paillier, P.: Decryptable searchable encryption. In: Susilo, W., Liu, J.K., Mu, Y. (eds.) ProvSec 2007. LNCS, vol. 4784, pp. 228–236. Springer, Heidelberg (2007)

10. Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
11. Gu, C., Zhu, Y., Pan, H.: Efficient public key encryption with keyword search schemes from pairings. In: Pei, D., Yung, M., Lin, D., Wu, C. (eds.) Inscrypt 2007. LNCS, vol. 4990, pp. 372–383. Springer, Heidelberg (2008)
12. Golle, P., Staddon, J., Waters, B.: Secure Conjunctive Search over Encrypted Data. In: Jakobsson, M., Yung, M., Zhou, J. (eds.) ACNS 2004. LNCS, vol. 3089, pp. 31–45. Springer, Heidelberg (2004)
13. Jeong, I.R., Kwon, J.O., Hong, D., Lee, D.H.: Constructing PEKS schemes secure against keyword guessing attacks is possible? Computer Communications. Express 32(2), 394–396 (2009)
14. Rhee, H.S., Park, J.H., Susilo, W., Lee, D.H.: Improved searchable public key encryption with designated tester. In: Proc. of the 4th international Symposium on information, Computer, and Communications Security (ASIACCS 2009), pp. 376–379. ACM, New York (2009)
15. Rhee, H.S., Susilo, W., Kim, H.-J.: Secure searchable public key encryption scheme against keyword guessing attacks. IEICE Electron. Express 6(5), 237–243 (2009)
16. Waters, B., Balfanz, D., Durfee, G., Smetters, D.: Building an Encrypted and Searchable Audit Log. In: Proc. of Network and Distributed System Security Symposium (NDSS 2004) (2004)
17. Yau, W.-C., Heng, S.-H., Goi, B.-M.: Off-line keyword guessing attacks on recent public key encryption with keyword search schemes. In: Rong, C., Jaatun, M.G., Sandnes, F.E., Yang, L.T., Ma, J. (eds.) ATC 2008. LNCS, vol. 5060, pp. 100–105. Springer, Heidelberg (2008)
18. Zhang, R., Imai, H.: Generic combination of public key encryption with keyword search and public key encryption. In: Bao, F., Ling, S., Okamoto, T., Wang, H., Xing, C. (eds.) CANS 2007. LNCS, vol. 4856, pp. 159–174. Springer, Heidelberg (2007)