Generic Constructions of Secure-Channel Free Searchable Encryption with Adaptive Security *

Keita Emura[†] Atsuko Miyaji[‡] Mohammad Shahriar Rahman[§]
Kazumasa Omote[¶]

January 11, 2017

Abstract

For searching keywords against encrypted data, the public key encryption scheme with keyword search (PEKS), and its an extension called secure-channel free PEKS (SCF-PEKS) have been proposed. In SCF-PEKS, a receiver makes a trapdoor for a keyword, and uploads it on a server. A sender computes an encrypted keyword, and sends it to the server. The server executes the searching procedure (called the test algorithm, which takes as inputs an encrypted keyword, trapdoor, and secret key of the server). In this paper, we extend the security of SCF-PEKS, calling it adaptive SCF-PEKS, wherein an adversary (modeled as a "malicious-but-legitimate" receiver) is allowed to issue test queries *adaptively*, and show that adaptive SCF-PEKS can be generically constructed by anonymous identity-based encryption (anonymous IBE) only. That is, for constructing adaptive SCF-PEKS we need not require any additional cryptographic primitive when compared to the Abdalla et al. PEKS construction (J. Cryptology 2008), even though adaptive SCF-PEKS requires additional functionalities. Note that our generic construction needs to apply the KEM/DEM framework (a.k.a. hybrid encryption), where KEM stands for key encapsulation mechanism, and DEM stands for data encapsulation mechanism. We also show that there is a class of anonymous IBE that can be applied for constructing adaptive SCF-PEKS without using hybrid encryption, and propose an adaptive SCF-PEKS construction based on this IBE. Although our second construction is not fully generic, it is efficient compared to the first, since we can exclude the DEM part. Finally, we instantiate an adaptive SCF-PEKS scheme (via our second construction) that achieves a similar level of efficiency for the costs of the test procedure and encryption, compared to the (non-adaptive secure) SCF-PEKS scheme by Fang et al. (CANS2009).

Keywords : Public-key Encryption with Keyword Search, Adaptive Security, Anonymous Identity-Based Encryption

1 Introduction

1.1 Public Key Encryption Scheme with Keyword Search (PEKS)

PEKS was proposed by Boneh et al. [9], and considers searching keywords from encrypted data. Briefly, the flow of PEKS is as follows: A receiver makes a trapdoor t_ω for a keyword ω, and uploads it on a server. A

*Preliminary versions of this paper appear in the 14th Information Security Conference, ISC 2011 [20], and 7th International Conference on Security and Cryptography, SECRYPT 2012 [23]. The first half (the generic construction part) is based on [20], and the second half (not fully generic but more efficient construction part) is based on [23]. This is the full and the merged version. We also corrected some typos appeared in the journal version [21].

[†]National Institute of Information and Communications Technology (NICT), Japan. k-emura@nict.go.jp
[‡]School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), Japan. miyaji@jaist.ac.jp
[§]Department of Computer Science and Engineering, University of Asia Pacific, Bangladesh. shahriar.rahman@uap-bd.edu
[¶]School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), Japan. omote@jaist.ac.jp

sender makes a ciphertext of a keyword ω' by using the receiver public key, and sends it to the server. The server outputs 1 if $\omega = \omega'$, by using t_ω, and 0 otherwise.

PEKS was investigated from both theoretical and practical perspectives. For example, several PEKS schemes with additional functionality have been proposed thus far: PEKS schemes treating plural keywords [11, 33, 41, 46], PEKS with public key encryption (PEKS/PKE) [3, 50], a decryptable PEKS scheme [26], where a receiver can decrypt an encrypted keyword, and a public key encryption with an oblivious keyword search scheme [14], wherein the server can obtain trapdoors without revealing the keywords. From the theoretical perspective, a PEKS scheme based on Jacobi symbols has been proposed [17] (though almost all PEKS schemes are constructed by using bilinear maps). The off-line keyword guessing attack was also introduced in [13, 34, 43, 48, 31], wherein an adversary can guess what keywords were used for computing trapdoors. The notion of interactive PEKS, wherein the trapdoor is generated interactively by the sender and the receiver, has also been proposed [16]. Moreover, PEKS with perfect keyword privacy has been considered in [40] which treats the leakage of keywords from trapdoors. PEKS with trapdoor revocation has also been considered in [22, 38, 49].

As a feasibility result of PEKS, Abdalla et al. [1] showed that a generic construction of PEKS based on anonymous IBE is sufficient.

1.2 Security Conditions of Previous Secure-Channel Free PEKS (SCF-PEKS) Schemes and the Theoretic Extension

PEKS schemes ensure that the server (or an outsider) learns nothing about keywords chosen by the sender without trapdoor information. Namely, if trapdoors are revealed, then anyone can execute the test procedure. Therefore, trapdoors cannot be sent via public (i.e., insecure) channels. So, in PEKS schemes, a secure channel (such as secure socket layer (SSL) and transport layer security (TLS)) between a receiver and a server is required, and establishing the channel requires additional setup costs. To solve this problem, secure-channel free PEKS (SCF-PEKS) has been proposed [4, 24, 29, 30, 35], wherein the server has a public/secret key pair, and the sender makes a ciphertext of a keyword ω' (which is encrypted by the server public key and the receiver public key), and sends it to the server. The server outputs 1 if $\omega = \omega'$ by using the trapdoor t_ω and its own secret key, and 0 otherwise. Even if t_ω is sent via an insecure channel, no entity (except the server) can run the test procedure.

Next, we discuss the security conditions of the previous SCF-PEKS. The security models of the previous SCF-PEKS schemes [4, 24, 29, 30, 35] do not capture the test queries (i.e., no adversary can issue test queries in security games). We point out that this definition does not capture the real environment as follows. Fig.1 illustrates how to instantiate test queries in the real world.

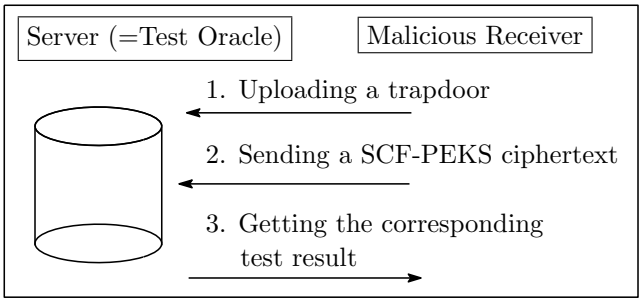

Figure 1: Instantiation of Test Queries in the Real World

1. A malicious receiver computes (or eavesdrops on) a trapdoor, and uploads it to the server.
   - From the viewpoint of the server, this is the same as uploading a trapdoor from a valid receiver.

2. The malicious receiver computes (or eavesdrops on) an SCF-PEKS ciphertext, and sends it to the server.
    - This is the same as sending a ciphertext from a valid sender.
3. The malicious receiver can obtain the result of the test algorithm.

Through the above procedure, the malicious receiver can obtain the result of the test algorithm. In other words, the malicious receiver can use the server as the test oracle. SCF-PEKS therefore has to be secure, even if the malicious receiver can be admitted to issue test queries. The test queries were considered in [42], but this definition is still weak (i.e., "Unquoted CCA-like" security [39]), since test queries cannot be repeated adaptively as follows: Let $\lambda^* = [A_1^*, A_2^*, \ldots, A_n^*]$ be the challenge ciphertext and $t_{\omega_0^*}$ (resp. $t_{\omega_1^*}$) be the trapdoors corresponding to the challenge keyword ω_0^* (resp. ω_1^*). In the definition of [42], $\mathcal{A}$ is allowed to issue (λ, t_ω) such that $t_\omega \notin \{t_{\omega_0^*}, t_{\omega_1^*}\}$ and for all $i \in [1, n]$, $A_i \neq A_i^*$. This is not natural, since $\mathcal{A}$ may compute a ciphertext λ and replace a part of λ with a part of λ^*. This scenario can easily be handled in the above real world example by sending such a "replaced ciphertext" in the guess phase. By considering the CCA2 security, SCF-PEKS must be secure even if a "malicious-but-legitimate" receiver can be admitted to issue test queries *adaptively*. We insist that this adaptive (i.e., "CCA2-like") security is theoretically the natural extension of the SCF-PEKS security, which is called adaptive SCF-PEKS.

1.3 Our Contribution

In this paper, we propose a generic construction of adaptive SCF-PEKS based on anonymous IBE, selective-tag chosen-ciphertext (IND-stag-CCA) secure tag-based encryption (TBE), and strongly existentially unforgeable (sUF) OTS. This is the first generic construction of SCF-PEKS. Note that IND-stag-CCA-secure TBE can be constructed by selective-ID chosen plaintext (sID-CPA) secure IBE [36], and the digital signature can be constructed by IBE [18]. Therefore, our result shows that adaptive SCF-PEKS can be constructed by anonymous IBE only. That is, we show that for constructing adaptive SCF-PEKS no additional cryptographic primitive is required when compared to the Abdalla et al. PEKS construction [1], even though adaptive SCF-PEKS requires additional functionalities. This construction uses double encryption, wherein TBE encrypts a ciphertext of an anonymous IBE. Since the ciphertext space of IBE is usually not equal to the plaintext space of TBE, we apply the KEM/DEM framework [45] (a.k.a. hybrid encryption), where KEM stands for key encapsulation mechanism, and DEM stands for data encapsulation mechanism.

Next, we show that there is a class of anonymous IBE that can be applied for constructing adaptive SCF-PEKS without using hybrid encryption, and propose an adaptive SCF-PEKS construction based on such an IBE. Although it is not fully generic, our second construction is efficient compared to the first one since we can exclude the DEM part.

Finally, we instantiate an adaptive SCF-PEKS scheme based on the Gentry anonymous IBE scheme [27], the Kiltz IND-stag-CCA-secure TBE scheme [36], and the Bellare-Shoup sUF OTS scheme [7], by using our second adaptive SCF-PEKS construction. Our concrete adaptive SCF-PEKS construction (called the GKBS construction due to the author's name) achieves a similar level of efficiency for the costs of the test procedure and encryption, compared to the (non-adaptive secure) SCF-PEKS scheme without random oracles proposed by Fang et al [24] (see the comparison table (Table 2) in section 6.3).

Remark. Independent of our result, Fang et al. [25] proposed a concrete SCF-PEKS scheme with keyword guessing attack resilience. They also considered the test oracle, and give a formal security definition. The efficiency of our concrete instantiation (GKBS) and that of the Fang et al. SCF-PEKS scheme is almost similar. Later, Guo and Yan [31] proposed more efficient SCF-PEKS scheme secure in the Fang et al. model. Note that no generic construction is given in [25, 31], and therefore proposing a generic construction of adaptive SCF-PEKS with keyword guessing attack resilience is an interesting future work.

Figure 2: TBE Experiment

IND-stag-CCA

$Exp^{\text{IND-stag-CCA}}_{\Pi,\mathcal{A}}(\kappa) := \big[(t^*, State) \leftarrow \mathcal{A}(1^\kappa);\ (pk, sk) \leftarrow \mathsf{TBE.KeyGen}(1^\kappa);$
$(M_0^*, M_1^*, State) \leftarrow \mathcal{A}^{\mathcal{DEC}}(\mathsf{find}, pk, State);\ \mu \xleftarrow{\$} \{0,1\};$
$C^*_{TBE} \leftarrow \mathsf{TBE.Enc}(pk, t^*, M^*_\mu);\ \mu' \leftarrow \mathcal{A}^{\mathcal{DEC}}(\mathsf{guess}, C^*, State);$
If $\mu = \mu'$ then output 1, and 0 otherwise]

2 Preliminaries

This section, we define the building tools for our generic adaptive SCF-PEKS construction. $x \xleftarrow{\$} S$ means that x is chosen uniformly from a set S. $y \leftarrow A(x)$ means that y is an output of an algorithm A under an input x. We denote $State$ as the state information transmitted by the adversary to himself across stages of the attack in experiments.

2.1 Definitions of IND-stag-CCA Secure TBE

In the following, $\mathcal{TAG}$ and $\mathcal{M}_{TBE}$ are a tag space of TBE and a plaintext space of TBE, respectively.

Definition 2.1 (Syntax of TBE). *A TBE scheme [36]* Π *consists of the following three algorithms,* $\mathsf{TBE.KeyGen}$, $\mathsf{TBE.Enc}$ *and* $\mathsf{TBE.Dec}$*:*

$\mathsf{TBE.KeyGen}(1^\kappa)$ *: This algorithm takes as an input the security parameter* $\kappa \in \mathbb{N}$*, and returns a public key* pk *and a secret key* sk*.*

$\mathsf{TBE.Enc}(pk, t, M)$ *: This algorithm takes as inputs* pk*, a message* $M \in \mathcal{M}_{TBE}$ *with a tag* $t \in \mathcal{TAG}$*, and returns a ciphertext* C_{TBE}*.*

$\mathsf{TBE.Dec}(sk, t, C_{TBE})$ *: This algorithms takes as inputs* sk*,* t*, and* C_{TBE}*, and returns* M *or* $\perp$*.*

Correctness is defined as follows: For all $(pk, sk) \leftarrow \mathsf{TBE.KeyGen}(1^\kappa)$, all $M \in \mathcal{M}_{TBE}$, and all $t \in \mathcal{TAG}$, $\mathsf{TBE.Dec}(sk, t, C_{TBE}) = M$ holds, where $C_{TBE} \leftarrow \mathsf{TBE.Enc}(pk, t, M)$.

Next, we define the security requirement of TBE under selective-tag CCA (IND-stag-CCA) as follows.

Definition 2.2 (IND-stag-CCA). *For any PPT adversary* $\mathcal{A}$ *and the security parameter* $\kappa \in \mathbb{N}$*, we define the experiment* $Exp^{IND\text{-}stag\text{-}CCA}_{\Pi,\mathcal{A}}(\kappa)$ *in Figure 2, and define the advantage of* $\mathcal{A}$ $Adv^{IND\text{-}stag\text{-}CCA}_{\Pi,\mathcal{A}}(\kappa)$ *as follows.*

$$Adv^{IND\text{-}stag\text{-}CCA}_{\Pi,\mathcal{A}}(\kappa) := \big|\Pr\big[Exp^{IND\text{-}stag\text{-}CCA}_{\Pi,\mathcal{A}}(\kappa) = 1\big] - \frac{1}{2}\big|$$

Here, $\mathcal{DEC}$ *is the* decryption *oracle for any tag* $t \neq t^*$*, where for input of a ciphertext* $(C_{TBE}, t) \neq (C^*_{TBE}, t^*)$*,*
it returns the corresponding plaintext M*. Note that* (C^*_{TBE}, t^*) *is not allowed as input to* $\mathcal{DEC}$*.*

We say that a TBE scheme Π *is IND-stag-CCA secure if the advantage* $Adv^{IND\text{-}stag\text{-}CCA}_{\Pi,\mathcal{A}}(1^\kappa)$ *is negligible.*

2.2 Definitions of Anonymous IBE

In the following, $\mathcal{ID}$ and $\mathcal{M}_{IBE}$ are an identity space and a plaintext space of IBE, respectively.

Definition 2.3 (Syntax of IBE). *IBE scheme* Π *consists of the following four algorithms,* $\mathsf{IBE.Setup}$, $\mathsf{IBE.Extract}$, $\mathsf{IBE.Enc}$ *and* $\mathsf{IBE.Dec}$*:*

Figure 3: IBE Experiments

IBE-IND-CPA

$Exp^{\text{IBE-IND-CPA}}_{\Pi,\mathcal{A}}(1^\kappa) := \big[(pk, mk) \leftarrow \mathsf{IBE.Setup}(1^\kappa);$
$(M_0^*, M_1^*, ID^*, State) \leftarrow \mathcal{A}^{\mathcal{EXTRACT}}(\mathsf{find}, pk);\ \mu \xleftarrow{\$} \{0,1\};$
$C^*_{IBE} \leftarrow \mathsf{IBE.Enc}(pk, ID^*, M^*_\mu);\ \mu' \leftarrow \mathcal{A}^{\mathcal{EXTRACT}}(\mathsf{guess}, C^*_{IBE}, State);$
If $\mu = \mu'$ then output 1, and 0 otherwise]

IBE-ANO-CPA

$Exp^{\text{IBE-ANO-CPA}}_{\Pi,\mathcal{A}}(1^\kappa) := \big[(pk, mk) \leftarrow \mathsf{IBE.Setup}(1^\kappa);$
$(ID_0^*, ID_1^*, M^*, State) \leftarrow \mathcal{A}^{\mathcal{EXTRACT}}(\mathsf{find}, pk);\ \mu \xleftarrow{\$} \{0,1\};$
$C^*_{IBE} \leftarrow \mathsf{IBE.Enc}(pk, ID^*_\mu, M^*);\ \mu' \leftarrow \mathcal{A}^{\mathcal{EXTRACT}}(\mathsf{guess}, C^*_{IBE}, State);$
If $\mu = \mu'$ then output 1, and 0 otherwise]

$\mathsf{IBE.Setup}(1^\kappa)$ *: This algorithm takes as an input the security parameter* $\kappa \in \mathbb{N}$*, and returns a public key* pk *and a master key* mk*.*

$\mathsf{IBE.Extract}(pk, mk, ID)$ *: This algorithm takes as inputs an identity* $ID \in \mathcal{ID}$*, and* mk*, and returns a secret key corresponding to* ID sk_{ID}*.*

$\mathsf{IBE.Enc}(pk, ID, M)$ *: This algorithm takes as inputs* pk*,* $ID \in \mathcal{ID}$*, and a message* $M \in \mathcal{M}_{IBE}$*, and returns a a ciphertext* C_{IBE}*.*

$\mathsf{IBE.Dec}(sk_{ID}, C_{IBE})$ *: This algorithm takes as inputs* sk_{ID} *and* C_{IBE}*, and returns* M *or* $\perp$*.*

Correctness is defined as follows: For all $(pk, mk) \leftarrow \mathsf{IBE.Setup}(1^\kappa)$, all $M \in \mathcal{M}_{IBE}$, and all $ID \in \mathcal{ID}$, $\mathsf{IBE.Dec}(sk_{ID}, C_{IBE}) = M$ holds, where $C_{IBE} \leftarrow \mathsf{IBE.Enc}(pk, ID, M)$ and $sk_{ID} \leftarrow \mathsf{IBE.Extract}(pk, mk, ID)$.

Next, we define the security requirement of IBE under chosen plaintext attack (IBE-IND-CPA) as follows.

Definition 2.4 (IBE-IND-CPA)**.** *For any PPT adversary* $\mathcal{A}$ *and the security parameter* $\kappa \in \mathbb{N}$*, we define the experiment* $Exp^{\textit{IBE-IND-CPA}}_{\Pi,\mathcal{A}}(\kappa)$ *in Figure 3, and define the advantage of* $\mathcal{A}$ $Adv^{\textit{IBE-IND-CPA}}_{\Pi,\mathcal{A}}(\kappa)$ *as follows.*

$$Adv^{\textit{IBE-IND-CPA}}_{\Pi,\mathcal{A}}(\kappa) := \Big| \Pr\big[Exp^{\textit{IBE-IND-CPA}}_{\Pi,\mathcal{A}}(\kappa) = 1\big] - \frac{1}{2} \Big|$$

Here, $\mathcal{EXTRACT}$ *is the* extraction *oracle for input of an identity* ID *it returns the corresponding secret key* sk_{ID}*. Note that* ID^* *is not allowed as input to* $\mathcal{EXTRACT}$ *in the IBE-IND-CPA experiment.*

We say that an IBE scheme Π *is IBE-IND-CPA secure if the advantage* $Adv^{\textit{IBE-IND-CPA}}_{\Pi,\mathcal{A}}(\kappa)$ *is negligible.*

Next, we define anonymity experiment of IBE under CPA (IBE-ANO-CPA).

Definition 2.5 (IBE-ANO-CPA)**.** *For any PPT adversary* $\mathcal{A}$ *and the security parameter* $\kappa \in \mathbb{N}$*, we define the experiment* $Exp^{\textit{IBE-ANO-CPA}}_{\Pi,\mathcal{A}}(1^\kappa)$ *in Figure 3, and define the advantage of* $\mathcal{A}$ $Adv^{\textit{IBE-ANO-CPA}}_{\Pi,\mathcal{A}}(\kappa)$ *as follows.*

$$Adv^{\textit{IBE-ANO-CPA}}_{\Pi,\mathcal{A}}(\kappa) := \Big| \Pr\big[Exp^{\textit{IBE-ANO-CPA}}_{\Pi,\mathcal{A}}(\kappa) = 1\big] - \frac{1}{2} \Big|$$

ID_0^* *and* ID_1^* *are not allowed as input to* $\mathcal{EXTRACT}$ *in the IBE-ANO-CPA experiment. We say that an IBE scheme* Π *is IBE-ANO-CPA secure if the advantage* $Adv^{\textit{IBE-ANO-CPA}}_{\Pi,\mathcal{A}}(1^\kappa)$ *is negligible.*

Definition 2.6 (Anonymous IBE)**.** *We say that an IBE scheme is anonymous IBE if the IBE scheme is both IBE-IND-CPA secure and IBE-ANO-CPA secure.*

Figure 4: OTS Experiment

one-time sUF-CMA
$Exp^{\text{one-time sUF-CMA}}_{\Pi,\mathcal{A}}(1^\kappa) := \big[(K_s, K_v) \leftarrow \mathsf{Sig.KeyGen}(1^\kappa);\ (M, State) \leftarrow \mathcal{A}(K_v);$ $\sigma \leftarrow \mathsf{Sign}(K_s, M); (M^*, \sigma^*) \leftarrow \mathcal{A}(State, \sigma);$ If $(M^*, \sigma^*) \neq (M, \sigma)$ and $\mathsf{Verify}(K_v, \sigma^*, M^*) = 1$ then output 1, and 0 otherwise$\big]$

Table 1: Oracles used in SCF-PEKS Experiments

$\mathcal{TRAP}$	$\mathcal{TEST}$
This is the *trapdoor* oracle for an input keyword ω, it returns a trapdoor t_ω. Note that $\mathcal{A}$ cannot query the challenge keywords ω_0^* and ω_1^* to $\mathcal{TRAP}$.	This is the *test* oracle for an input (λ, t_ω) which satisfies $(\lambda, t_\omega) \notin \{(\lambda^*, t_{\omega_0^*}), (\lambda^*, t_{\omega_1^*})\}$, it returns the result of the test algorithm.

2.3 Definitions of sUF OTS

In the following, $\mathcal{M}_{Sig}$ is a message space of OTS.

Definition 2.7 (Syntax of OTS)**.** *A strongly existentially unforgeable (sUF) OTS against adaptively chosen message attack (CMA) (e.g., [7]) consists of the following three algorithms,* $\mathsf{Sig.KeyGen}$*,* Sign *and* Verify*:*

$\mathsf{Sig.KeyGen}(1^\kappa)$ *: This algorithm takes as an input a security parameter* 1^κ *(*$\kappa \in \mathbb{N}$*), and returns a signing/verification key pair* (K_s, K_v)*.*

$\mathsf{Sign}(K_s, M)$ *: This algorithm takes as inputs* K_s *and a message* $M \in \mathcal{M}_{Sig}$*, and returns a signature* σ*.*

$\mathsf{Verify}(K_v, \sigma, M)$ *: This algorithm takes as inputs* K_v*,* σ*, and* M*, and returns 1 if* σ *is a valid signature of* M*, and 0 otherwise.*

Correctness is defined as follows: For all $(K_s, K_v) \leftarrow \mathsf{Sig.KeyGen}(1^\kappa)$ and all $M \in \mathcal{M}_{Sig}$, $\mathsf{Verify}(K_v, \sigma, M) = 1$ holds, where $\sigma \leftarrow \mathsf{Sign}(K_s, M)$.

Definition 2.8 (one-time sUF-CMA)**.** *For any PPT adversary* $\mathcal{A}$ *and the security parameter* $\kappa \in \mathbb{N}$*, we define the experiment* $Exp^{\text{one-time sUF-CMA}}_{\Pi,\mathcal{A}}(\kappa)$ *in Figure 4, and define the advantage of* $\mathcal{A}$ $Adv^{\text{one-time sUF-CMA}}_{\Pi,\mathcal{A}}(\kappa)$ *as follows.*

$$Adv^{\text{one-time sUF-CMA}}_{\Pi,\mathcal{A}}(\kappa) := \Pr\left[Exp^{\text{one-time sUF-CMA}}_{\Pi,\mathcal{A}}(\kappa) = 1\right]$$

We say that a signature scheme Π *is one-time sUF-CMA secure if the advantage* $Adv^{\text{one-time sUF-CMA}}_{\Pi,\mathcal{A}}(\kappa)$ *is negligible.*

3 Definitions of Adaptive SCF-PEKS

In this section, we define security requirements of SCF-PEKS. In the following, $\mathcal{K}$ is a keyword space.

Definition 3.1 (Syntax of SCF-PEKS)**.** *An SCF-PEKS scheme* Π *consists of the following five algorithms,* $\mathsf{SCF\text{-}PEKS.KeyGen_S}$*,* $\mathsf{SCF\text{-}PEKS.KeyGen_R}$*,* $\mathsf{SCF\text{-}PEKS.Trapdoor}$*,* $\mathsf{SCF\text{-}PEKS.Enc}$ *and* $\mathsf{SCF\text{-}PEKS.Test}$*:*

Figure 5: SCF-PEKS Experiments

Consistency
$Exp^{\text{SCF-PEKS-CONSIST}}_{\Pi,\mathcal{A}}(1^\kappa) := \big[(pk_S, sk_S) \leftarrow \mathsf{SCF\text{-}PEKS.KeyGen_S}(1^\kappa);\ (pk_R, sk_R) \leftarrow \mathsf{SCF\text{-}PEKS.KeyGen_R}(1^\kappa);$ $(\omega, \omega') \leftarrow \mathcal{A}(pk_S, pk_R); \omega \neq \omega';\ \lambda \leftarrow \mathsf{SCF\text{-}PEKS.Enc}(pk_S, pk_R, \omega);\ t_{\omega'} \leftarrow \mathsf{SCF\text{-}PEKS.Trapdoor}(sk_R, \omega');$ If $\mathsf{SCF\text{-}PEKS.Test}(\lambda, sk_S, t_{\omega'}) = 1$ then output 1, and 0 otherwise$\big]$
IND-CKA-SSK
$Exp^{\text{IND-CKA-SSK}}_{\Pi,\mathcal{A}}(1^\kappa) := \big[(pk_S, State) \leftarrow \mathcal{A}(1^\kappa);\ (pk_R, sk_R) \leftarrow \mathsf{SCF\text{-}PEKS.KeyGen_R}(1^\kappa);$ $(\omega_0^*, \omega_1^*, State) \leftarrow \mathcal{A}^{\mathcal{TRAP}}(\mathsf{find}, pk_R, State);\ \mu \xleftarrow{\$} \{0,1\};\ \lambda^* \leftarrow \mathsf{SCF\text{-}PEKS.Enc}(pk_S, pk_R, \omega_\mu^*);$ $\mu' \leftarrow \mathcal{A}^{\mathcal{TRAP}}(\mathsf{guess}, \lambda^*, State)$; If $\mu = \mu'$, then output 1, and 0 otherwise$\big]$
Adaptive-IND-CKA-AT
$Exp^{\text{Adaptive-IND-CKA-AT}}_{\Pi,\mathcal{A}}(1^\kappa) := \big[(pk_S, sk_S) \leftarrow \mathsf{SCF\text{-}PEKS.KeyGen_S}(1^\kappa);\ (pk_R, State) \leftarrow \mathcal{A}(1^\kappa);$ $(\omega_0^*, \omega_1^*, State) \leftarrow \mathcal{A}^{\mathcal{TEST}}(\mathsf{find}, pk_S, State);\ \mu \xleftarrow{\$} \{0,1\};\ \lambda^* \leftarrow \mathsf{SCF\text{-}PEKS.Enc}(pk_S, pk_R, \omega_\mu^*);$ $\mu' \leftarrow \mathcal{A}^{\mathcal{TEST}}(\mathsf{guess}, \lambda^*, State)$; If $\mu = \mu'$, then output 1, and 0 otherwise$\big]$

$\mathsf{SCF\text{-}PEKS.KeyGen_S}(1^\kappa)$ *: This server key generation algorithm takes as input the security parameter* 1^κ $(\kappa \in \mathbb{N})$*, and returns a server public key* pk_S *and a server secret key* sk_S*.*

$\mathsf{SCF\text{-}PEKS.KeyGen_R}(1^\kappa)$ *: This receiver key generation algorithm takes as input the security parameter* 1^κ $(\kappa \in \mathbb{N})$*, and returns a receiver public key* pk_R *and a receiver secret key* sk_R*.*

$\mathsf{SCF\text{-}PEKS.Trapdoor}(sk_R, \omega)$ *: This trapdoor generation algorithm takes as input* sk_R *and a keyword* $\omega \in \mathcal{K}$*, and returns a trapdoor* t_ω *corresponding to keyword* ω*.*

$\mathsf{SCF\text{-}PEKS.Enc}(pk_S, pk_R, \omega)$ *: This encryption algorithm takes as input* pk_R*,* pk_S*, and* ω*, and returns a ciphertext* λ*.*

$\mathsf{SCF\text{-}PEKS.Test}(\lambda, sk_S, t_\omega)$ *This test algorithm takes as input* λ*,* sk_S*, and* t_ω*, and returns 1 if* $\omega = \omega'$*, where* ω' *is the keyword which was used for computing* λ*, and 0 otherwise.*

A sender makes a ciphertext λ of a keyword ω' using both pk_S and pk_R, and sends λ to the server. The server runs $\mathsf{SCF\text{-}PEKS.Test}(\lambda, sk_S, t_\omega)$, whose output is 1 if $\omega = \omega'$, and 0 otherwise.

We require the correctness property as follows: For all $(pk_S, sk_S) \leftarrow \mathsf{SCF\text{-}PEKS.KeyGen_S}(1^\kappa)$, all $(pk_R, sk_R) \leftarrow \mathsf{SCF\text{-}PEKS.KeyGen_R}(1^\kappa)$, and all $\omega \in \mathcal{K}$, $\mathsf{SCF\text{-}PEKS.Test}(\lambda, sk_S, t_\omega) = 1$ holds, where $\lambda \leftarrow \mathsf{SCF\text{-}PEKS.Enc}(pk_R, pk_S, \omega)$ and $t_\omega \leftarrow \mathsf{SCF\text{-}PEKS.Trapdoor}(sk_R, \omega)$.

Next, we consider two security requirements "consistency" and "keyword privacy".

Definition 3.2 (Consistency)**.** *For any PPT adversary* $\mathcal{A}$ *and the security parameter* $\kappa \in \mathbb{N}$*, we define the experiment* $Exp^{\text{SCF-PEKS-CONSIST}}_{\Pi,\mathcal{A}}(\kappa)$ *in Figure 5, and define the advantage of* $\mathcal{A}$ $Adv^{\text{SCF-PEKS-CONSIST}}_{\Pi,\mathcal{A}}(\kappa)$ *as follows.*

$$Adv^{\text{SCF-PEKS-CONSIST}}_{\Pi,\mathcal{A}}(\kappa) := \Pr\left[Exp^{\text{SCF-PEKS-CONSIST}}_{\Pi,\mathcal{A}}(\kappa) = 1\right]$$

We say that an SCF-PEKS scheme Π *is computationally consistent if the advantage* $Adv^{\text{SCF-PEKS-CONSIST}}_{\Pi,\mathcal{A}}(\kappa)$ *is negligible.*

Next, we define two security notions for keyword privacy, "indistinguishability against chosen keyword attack with the server secret key" (IND-CKA-SSK for short) and "indistinguishability against chosen keyword attack with all trapdoors" (IND-CKA-AT for short). In the IND-CKA-SSK experiment, an adversary $\mathcal{A}$ is assumed to be a malicious server. Therefore, $\mathcal{A}$ is given the server secret key sk_S, whereas $\mathcal{A}$ cannot obtain the receiver secret key sk_R. Instead of obtaining sk_R, $\mathcal{A}$ can issue a query to a trapdoor oracle $\mathcal{TRAP}$, which is defined in Table 1. As in the definition of [42], $\mathcal{A}$ computes (pk_S, sk_S), and gives pk_S to the challenger. So, we omit sk_S in the IND-CKA-SSK experiment.

Definition 3.3 (IND-CKA-SSK)**.** *For any PPT adversary $\mathcal{A}$ and the security parameter $\kappa \in \mathbb{N}$, we define the experiment $Exp_{\Pi,\mathcal{A}}^{IND\text{-}CKA\text{-}SSK}(\kappa)$ in Figure 5, and define $Adv_{\Pi,\mathcal{A}}^{IND\text{-}CKA\text{-}SSK}(\kappa)$ as follows.*

$$Adv_{\Pi,\mathcal{A}}^{IND\text{-}CKA\text{-}SSK}(\kappa) := \left| \Pr\left[Exp\Pi, \mathcal{A}^{IND\text{-}CKA\text{-}SSK}(\kappa) = 1\right] - \frac{1}{2} \right|$$

We say that an SCF-PEKS scheme Π is IND-CKA-SSK-secure if the advantage $Adv_{\Pi,\mathcal{A}}^{IND\text{-}CKA\text{-}SSK}(\kappa)$ is negligible.

Remark: Note that, for our TRE construction, the adversarial server's key generation above is not required. That is, the weaker definition can be used, where $\mathcal{C}$ can run $(pk_S, sk_S) \leftarrow \mathsf{SCF\text{-}PEKS.KeyGen_S}(1^\kappa)$, and sends (pk_S, sk_S) to $\mathcal{A}$ in our proof of Theorem 2.

Next, we define the adaptive-IND-CKA-AT experiment. In this experiment, an adversary $\mathcal{A}$ is assumed to be a malicious-but-legitimate receiver or outsider. Therefore, $\mathcal{A}$ is given the receiver secret key sk_R, whereas $\mathcal{A}$ cannot obtain the server secret key sk_S. This means that $\mathcal{A}$ knows *all* trapdoors. $\mathcal{A}$ can issue a query to a test oracle $\mathcal{TEST}$, which is defined in Table 1. As in the definition of [42], $\mathcal{A}$ computes (pk_R, sk_R), and gives pk_R to the challenger. So, we omit sk_R in the Adaptive-IND-CKA-AT experiment.

Definition 3.4 (Adaptive-IND-CKA-AT)**.** *For any PPT adversary $\mathcal{A}$ and the security parameter $\kappa \in \mathbb{N}$, we define the experiment $Exp_{\Pi,\mathcal{A}}^{Adaptive\text{-}IND\text{-}CKA\text{-}AT}(\kappa)$ in Figure 5, and define $Adv_{\Pi,\mathcal{A}}^{Adaptive\text{-}IND\text{-}CKA\text{-}AT}(\kappa)$ as follows.*

$$Adv_{\Pi,\mathcal{A}}^{Adaptive\text{-}IND\text{-}CKA\text{-}AT}(\kappa) = \left| \Pr\left[Exp_{\Pi,\mathcal{A}}^{Adaptive\text{-}IND\text{-}CKA\text{-}AT}(\kappa) = 1\right] - \frac{1}{2} \right|$$

We say that an SCF-PEKS scheme is adaptive-IND-CKA-AT-secure if the advantage $Adv_{\Pi,\mathcal{A}}^{Adaptive\text{-}IND\text{-}CKA\text{-}AT}(\kappa)$ is negligible for any PPT adversary $\mathcal{A}$ in the following experiment.

Remark: As in the IND-CKA-SSK, for TRE construction, the adversarial receiver's key generation above is not required. That is, we use the weaker definition, where $\mathcal{C}$ can run $(pk_R, sk_R) \leftarrow \mathsf{SCF\text{-}PEKS.KeyGen_R}(1^\kappa)$, and sends (pk_R, sk_R) to $\mathcal{A}$ in our proof of Theorem 1.

4 Anonymous IBE Implies Adaptive SCF-PEKS

4.1 A Generic Construction of Adaptive SCF-PEKS

This section gives a generic construction of adaptive SCF-PEKS based on anonymous IBE, IND-stag-CCA TBE, and sUF OTS. In our construction, a ciphertext of an anonymous IBE scheme (say C_{IBE}) is used as a "plaintext" of a TBE scheme to hide keyword information from an adversary. From the result of the decryption of the TBE scheme, the ciphertext C_{IBE} must be obtained. In addition, usually, $C_{IBE} \notin \mathcal{M}_{TBE}$. To handle this condition, we apply the KEM/DEM framework [45] (a.k.a. hybrid encryption), where KEM stands for key encapsulation mechanism, and DEM stands for data encapsulation mechanism. By using TBE KEM (see section 6 of [36]), compute $(K, C_{TBE}) \leftarrow \mathsf{TBE.Enc}(pk, t)$, and encrypt C_{IBE} as a plaintext of the CCA secure DEM such that $e = \mathsf{E}_K(C_{IBE})$. Note that a CCA-secure DEM can be generically constructed from any pseudorandom functions without redundancy [37]. So, even if we assume that a CCA secure DEM

exists, we need no additional cryptographic primitive, except anonymous IBE, for constructing adaptive SCF-PEKS. From here, we assume that $C_{IBE} \in \mathcal{M}_{TBE}$ and $e = \mathsf{E}_K(C_{IBE})$ is implicitly included in C_{TBE} (i.e., C_{IBE} is obtained from the decryption of C_{TBE}). Note that section 5 gives an extended construction (which does not require hybrid encryption).

In the following construction, as in the Abdalla et al. PEKS construction [1], a keyword ω is regarded as an "identity" of IBE. As in the Kiltz CCA-secure PKE construction [36] based on IND-stag-CCA TBE, a verification key K_v is regarded as a "tag" of TBE. We use a target collision-resistant (TCR) hash function [6] $H_{tag} : \{0,1\}^* \to \mathcal{TAG}$. We set $\mathcal{M}_{Sig} = \mathcal{C}_{TBE} \times \mathcal{M}_{IBE}$, where $\mathcal{C}_{TBE}$ is a ciphertext space of the underlying TBE.

Our First Adaptive SCF-PEKS Construction

$\mathsf{SCF\text{-}PEKS.KeyGen_S}(1^\kappa)$**:** Run $(pk_S, sk_S) \leftarrow \mathsf{TBE.KeyGen}(1^\kappa)$, and output (pk_S, sk_S).

$\mathsf{SCF\text{-}PEKS.KeyGen_R}(1^\kappa)$**:** Run $(pk_R, sk_R) \leftarrow \mathsf{IBE.KeyGen}(1^\kappa)$, and output (pk_R, sk_R).

$\mathsf{SCF\text{-}PEKS.Trapdoor}(sk_R, \omega)$**:** Run $t_\omega \leftarrow \mathsf{IBE.Extract}(sk_R, \omega)$, and output t_ω.

$\mathsf{SCF\text{-}PEKS.Enc}(pk_S, pk_R, \omega)$**:** Generate $(K_s, K_v) \xleftarrow{\$} \mathsf{Sig.KeyGen}$, compute $t = H_{tag}(K_v)$, choose $R \xleftarrow{\$} \mathcal{M}_{IBE}$, run $C_{IBE} \leftarrow \mathsf{IBE.Enc}(pk_R, \omega, R)$, $C_{TBE} \leftarrow \mathsf{TBE.Enc}(pk_S, t, C_{IBE})$, and $\sigma \leftarrow \mathsf{Sign}(K_s, (C_{TBE}, R))$, and output $\lambda = (C_{TBE}, K_v, \sigma)$.

$\mathsf{SCF\text{-}PEKS.Test}(\lambda, sk_S, t_\omega)$**:** Let $\lambda = (C_{TBE}, K_v, \sigma)$. Compute $t = H_{tag}(K_v)$, run $C'_{IBE} \leftarrow \mathsf{TBE.Dec}(sk_S, t, C_{TBE})$ and $R' \leftarrow \mathsf{IBE.Dec}(t_\omega, C'_{IBE})$. Output 1 if $1{=}\mathsf{Verify}(K_v, \sigma, (C_{TBE}, R'))$, and 0 otherwise.

Obviously, correctness holds if the underlying TBE, IBE, and OTS satisfy correctness.

Intuitively, an adversary (in the IND-CKA-SSK experiment) that has the server secret key (i.e., a decryption key of TBE) can compute C_{IBE} from C_{TBE}. However, since such an adversary does not have trapdoors to the challenge keywords, no information about keywords leaks from a ciphertext of an anonymous IBE, even if R is revealed from σ (without contradicting unforgeable property). In other words, we can reduce from the IND-CKA-SSK experiment to the IBE-ANO-CPA experiment. TBE is applied to hide information about keywords from an adversary (in the adaptive-IND-CKA-AT experiment) that has all the trapdoors. In other words, the adversary loses the opportunity to apply trapdoors to challenge keywords to the challenge ciphertext. In addition, due to the sUF property of the underlying signature scheme and the TCR property of H_{tag}, the adversary cannot issue a test query that the simulator cannot answer.

Non-adaptive SCF-PEKS construction: By observing our construction, a non-adaptive SCF-PEKS (i.e., IND-CKA-AT without test queries, which has the same security requirement as Fang et al. [24]) can be constructed by reducing the one-time signature part and replacing the TBE part with CPA-secure PKE (i.e., chosen plaintext security is enough). A ciphertext is (C_{PKE}, R), where $C_{IBE} \leftarrow \mathsf{IBE.Enc}(pk_R, \omega, R)$ and $C_{PKE} \leftarrow \mathsf{PKE.Enc}(pk_S, C_{IBE})$. As in our adaptive SCF-PEKS construction, we assume that $C_{IBE} \in \mathcal{M}_{PKE}$, where $\mathcal{M}_{PKE}$ is the message space of the underlying PKE scheme. The test procedure is described as follows. Compute $C'_{IBE} \leftarrow \mathsf{PKE.Dec}(sk_S, C_{PKE})$ and $R' \leftarrow \mathsf{IBE.Dec}(t_\omega, C'_{IBE})$. Output 1 if $R' = R$, and 0 otherwise.

4.2 Security Analysis of our First Adaptive SCF-PEKS construction

Theorem 4.1. *The SCF-PEKS scheme constructed by our method is computationally consistent if the underlying IBE scheme is IBE-IND-CPA secure.*

Proof. Let $\mathcal{A}$ be an adversary who breaks the computational consistency of SCF-PEKS constructed by the protocol 1, and $\mathcal{C}$ be the challenger of the IBE-IND-CPA experiment. Then, we can construct an algorithm $\mathcal{B}$ that breaks the IBE-IND-CPA security of the IBE scheme. First, $\mathcal{C}$ runs $\mathsf{IBE.Setup}(1^\kappa)$, and gives pk to $\mathcal{B}$. $\mathcal{B}$ sets pk as pk_R, runs $(pk_S, sk_S) \leftarrow \mathsf{TBE.KeyGen}(1^\kappa)$, and gives (pk_R, pk_S) to $\mathcal{A}$. $\mathcal{B}$ obtains keywords ω

and ω' from $\mathcal{A}$. $\mathcal{B}$ chooses $R_0, R_1 \xleftarrow{\$} \mathcal{M}_{IBE}$ as the challenge messages, and sends (ω, R_0, R_1) to $\mathcal{C}$. $\mathcal{C}$ gives $C^*_{IBE} \leftarrow \mathsf{IBE.Enc}(pk_R, \omega, R_\mu)$ to $\mathcal{B}$, where $\mu \in \{0,1\}$. $\mathcal{B}$ gets a trapdoor $t_{\omega'}$ by issuing an $\mathcal{EXTRACT}$ query. If $\mathsf{IBE.Dec}(t_{\omega'}, C^*_{IBE}) = R_1$, then $\mathcal{B}$ outputs 1, otherwise $\mathcal{B}$ outputs 0. □

Theorem 4.2. *The SCF-PEKS scheme constructed by our method is IND-CKA-SSK secure if the underlying IBE scheme is IBE-ANO-CPA secure.*

Proof. Let $\mathcal{A}$ be an adversary who breaks the IND-CKA-SSK security of SCF-PEKS constructed by the protocol 1, and $\mathcal{C}$ be the challenger of the IBE-ANO-CPA experiment. Then we can construct an algorithm $\mathcal{B}$ that breaks the IBE-ANO-CPA security of the underlying IBE scheme. First, $\mathcal{C}$ runs $\mathsf{IBE.Setup}(1^\kappa)$, and gives pk to $\mathcal{B}$. $\mathcal{B}$ sets pk as pk_R. $\mathcal{A}$ runs $(pk_S, sk_S) \leftarrow \mathsf{TBE.KeyGen}(1^\kappa)$, and gives pk_S to $\mathcal{B}$. For a $\mathcal{TRAP}$ query ω_i, $\mathcal{B}$ forwards ω_i to $\mathcal{C}$ as an $\mathcal{EXTRACT}$ query of the IBE scheme, gets t_{ω_i}, and answers t_{ω_i} to $\mathcal{A}$.

In the $\mathsf{Challenge}$ phase, $\mathcal{A}$ sends the challenge keywords ω^*_0 and ω^*_1 to $\mathcal{B}$. $\mathcal{B}$ chooses $R^* \xleftarrow{\$} \mathcal{M}_{IBE}$, and computes the challenge ciphertext as follows:

1. $\mathcal{B}$ sends $(R^*, \omega^*_0, \omega^*_1)$ to $\mathcal{C}$.
2. $\mathcal{C}$ gives $C^*_{IBE} \leftarrow \mathsf{IBE.Enc}(pk_R, \omega^*_\mu, R^*)$ to $\mathcal{B}$, where $\mu \in \{0,1\}$.
3. $\mathcal{B}$ generates $(K^*_s, K^*_v) \xleftarrow{\$} \mathsf{Sig.KeyGen}$, and computes $t^* = H_{tag}(K^*_v)$, $C^*_3 \leftarrow \mathsf{TBE.Enc}(pk_S, t^*, C^*_{IBE})$, and $\sigma^* \leftarrow \mathsf{Sign}(K^*_s, (C^*_{TBE}, R^*))$.
4. $\mathcal{B}$ sends $\lambda^* = (C^*_{TBE}, K^*_v, \sigma^*)$ to $\mathcal{A}$.

Note that $\mathcal{A}$ can compute $C^*_{IBE} \leftarrow \mathsf{TBE.Dec}(sk_S, H_{tag}(K^*_v), C^*_{TBE})$. In addition, R^* may be revealed from σ^* without contradicting unforgeability property. However, this situation is the same as in the IBE-ANO-CPA experiment, where $\mathcal{A}$ inputs $ID^*_0 := \omega^*_0$, $ID^*_1 := \omega^*_1$, and $M^* := R^*$, and gets the challenge ciphertext C^*_{IBE}. Finally, $\mathcal{B}$ outputs μ', where $\mu' \in \{0,1\}$ is the output of $\mathcal{A}$. □

Theorem 4.3. *The SCF-PEKS scheme constructed by our method is adaptive-IND-CKA-AT secure if the underlying TBE scheme is IND-stag-CCA secure, the underlying signature is one-time sUF-CMA secure, and H_{tag} is a TCR hash function.*

Proof. Let $\mathcal{A}$ be an adversary who breaks the adaptive-IND-CKA-AT security of SCF-PEKS constructed by the protocol 1, and $\mathcal{C}$ be the challenger of the IND-stag-CCA experiment. Then, we can construct an algorithm $\mathcal{B}$ that breaks the IND-stag-CCA security of the underlying TBE scheme. First, $\mathcal{B}$ runs $(K^*_s, K^*_v) \leftarrow \mathsf{Sig.KeyGen}(1^\kappa)$, and sends $t^* := H_{tag}(K^*_v)$ to $\mathcal{C}$ as the challenge tag. $\mathcal{C}$ runs $\mathsf{TBE.KeyGen}(1^\kappa)$, and gives pk to $\mathcal{B}$. $\mathcal{B}$ sets pk as pk_S. $\mathcal{A}$ runs $(pk_R, sk_R) \leftarrow \mathsf{IBE.Setup}(1^\kappa)$, and gives pk_R to $\mathcal{B}$. Let $(\mathsf{SCF\text{-}PEKS.Enc}(pk_S, pk_R, \omega_j) := (C_{TBE}, K_v, \sigma), t_{\omega_j})$ be a $\mathcal{TEST}$ query, where $\omega_j \in \mathcal{ID}$. $\mathcal{B}$ computes $t = H_{tag}(K_v)$, and answers as follows:

$t \neq t^*$: $\mathcal{B}$ can use the $\mathcal{DEC}$ oracle of the underlying TBE scheme as follows.

1. $\mathcal{B}$ forwards (C_{TBE}, t) to $\mathcal{C}$ as a $\mathcal{DEC}$ query of the TBE scheme.
2. $\mathcal{C}$ answers $C'_{IBE} \leftarrow \mathsf{TBE.Dec}(sk, t, C_{TBE})$.
   - Note that if t is not the legitimate tag of C_{TBE}, then $\mathcal{C}$ answers $\perp$. In this case, $\mathcal{B}$ answers 0.
3. $\mathcal{B}$ computes $R' \leftarrow \mathsf{IBE.Dec}(t_{\omega_j}, C'_{IBE})$.
4. If $\mathsf{Verify}(K_v, \sigma, (C_{TBE}, R')) = 1$, then $\mathcal{B}$ returns 1, and 0 otherwise.

$t = t^*$: If $K_v \neq K^*_v$, then $\mathcal{B}$ breaks the TCR property of H_{tag}. If $K_v = K^*_v$ (we call this a forge_1 event), then $\mathcal{B}$ gives a random answer in $\mathcal{C}$, and aborts.

In the $\mathsf{Challenge}$ phase, $\mathcal{A}$ sends the challenge keywords ω^*_0 and ω^*_1 to $\mathcal{B}$. $\mathcal{B}$ chooses $R^* \xleftarrow{\$} \mathcal{M}_{IBE}$, and computes the challenge ciphertext as follows:

1. $\mathcal{B}$ computes $C_{IBE,0} \leftarrow \mathsf{IBE.Enc}(pk_R, \omega_0^*, R^*)$ and $C_{IBE,1} \leftarrow \mathsf{IBE.Enc}(pk_R, \omega_1^*, R^*)$.
2. $\mathcal{B}$ sends $(M_0^*, M_1^*) := (C_{IBE,0}, C_{IBE,1})$ to $\mathcal{C}$ as the challenge messages of the IND-stag-CCA experiment of the TBE scheme.
3. $\mathcal{C}$ gives $C_{TBE}^* \leftarrow \mathsf{TBE.Enc}(pk_S, t^*, M_\mu^*)$ to $\mathcal{B}$.
4. $\mathcal{B}$ computes $\sigma^* \leftarrow \mathsf{Sign}(K_s^*, (C_{TBE}^*, R^*))$, and sends $\lambda^* = (C_{TBE}^*, K_v^*, \sigma^*)$ to $\mathcal{A}$.

Again, let $(\mathsf{SCF\text{-}PEKS.Enc}(pk_S, pk_R, \omega_j) := (C_{TBE}, K_v, \sigma), t_{\omega_j})$ be a $\mathcal{TEST}$ query, where $\omega_j \in \mathcal{ID}$. $\mathcal{B}$ computes $t = H_{tag}(K_v)$, and answers as follows:

In the case $t_{\omega_j} \in \{t_{\omega_0^*}, t_{\omega_1^*}\}$:

$t = t^*$: If $K_v \neq K_v^*$, then $\mathcal{B}$ breaks the TCR property of H_{tag}. If $K_v = K_v^*$ (we call this a forge_2 event), then $\mathcal{B}$ gives a random answer in $\mathcal{C}$, and aborts.

$t \neq t^*$: Then $\mathcal{B}$ can use the $\mathcal{DEC}$ oracle of the underlying TBE scheme as follows. .

1. $\mathcal{B}$ forwards (C_{TBE}, t) to $\mathcal{C}$ as a $\mathcal{DEC}$ query of the TBE scheme.
2. $\mathcal{C}$ answers $C'_{IBE} \leftarrow \mathsf{TBE.Dec}(sk, t, C_{TBE})$.
   - Note that if t is not the legitimate tag of C_{TBE}, then $\mathcal{C}$ answers $\perp$. In this case, $\mathcal{B}$ answers 0.
3. $\mathcal{B}$ computes $R' \leftarrow \mathsf{IBE.Dec}(t_{\omega_j}, C'_{IBE})$.
4. If $\mathsf{Verify}(K_v, \sigma, (C_{TBE}, R')) = 1$, then $\mathcal{B}$ returns 1, and 0 otherwise.

In the case $t_{\omega_j} \notin \{t_{\omega_0^*}, t_{\omega_1^*}\}$:

$(C_{TBE}, K_v, \sigma) = (C_{TBE}^*, K_v^*, \sigma^*)$: $\mathcal{B}$ returns 0, since $(C_{TBE}^*, K_v^*, \sigma^*)$ is an SCF-PEKS ciphertext of either ω_0^* or ω_1^*.

$(C_{TBE}, K_v, \sigma) \neq (C_{TBE}^*, K_v^*, \sigma^*)$: $\mathcal{B}$ runs the same simulation as in the find stage.

If $\mathcal{B}$ does not abort, then our simulation is perfect. Finally, $\mathcal{B}$ outputs μ', where $\mu' \in \{0, 1\}$ is the output of $\mathcal{A}$.

Next, we prove that $\Pr[\mathsf{forge}] := \Pr[\mathsf{forge}_1 \vee \mathsf{forge}_2]$ is negligible. We construct an algorithm $\mathcal{B}'$ which can win the sUF game with probability at least $\Pr[\mathsf{forge}]$. $\mathcal{B}'$ obtains K_v^* from the sUF challenger, instead of executing $\mathsf{Sig.KeyGen}(1^\kappa)$. $\mathcal{B}'$ runs $(pk_S, sk_S) \leftarrow \mathsf{TBE.KeyGen}(1^\kappa)$, and gives pk_S to $\mathcal{A}$. $\mathcal{A}$ runs $(pk_R, sk_R) \leftarrow \mathsf{IBE.Setup}(1^\kappa)$, and gives pk_R to $\mathcal{B}$. Since $\mathcal{B}'$ has sk_S, $\mathcal{B}'$ can answer any $\mathcal{TEST}$ queries. In the challenge phase of the adaptive-IND-CKA-AT experiment, $\mathcal{B}'$ computes $t^* = H_{tag}(K_v^*)$, chooses $R^* \xleftarrow{\$} \mathcal{M}_{IBE}$, runs $C_{IBE}^* \leftarrow \mathsf{IBE.Enc}(pk_R, \omega_\mu, R)$, and $C_{TBE}^* \leftarrow \mathsf{TBE.Enc}(pk_S, t^*, C_{IBE}^*)$, sets $M^* := (C_{TBE}^*, R^*)$, sends M^* to the sUF challenger, and obtains σ^* from the sUF challenger. Therefore, $\mathcal{B}'$ makes at most one signature query. Note that we do not have to care about the value $\mu \in \{0, 1\}$, since we only have to guarantee that $\lambda^* = (C_{TBE}^*, K_v^*, \sigma^*)$ is a valid SCF-PEKS ciphertext. In the forge events, $\mathcal{A}$ sends a $\mathcal{TEST}$ query $((C_{TBE}, K_v, \sigma), t_{\omega_j})$ with $K_v = K_v^*$.

forge_1 : In this case, $\mathcal{B}'$ can obtain a signature without issuing the signature query. $\mathcal{B}'$ computes $C_{IBE} \leftarrow \mathsf{TBE.Dec}(sk_S, H_{tag}(K_v), C_{TBE})$ and $R' \leftarrow \mathsf{IBE.Dec}(t_{\omega_j}, C_{IBE})$. If $((C_{TBE}, R'), \sigma)$ is not a valid signature pair, then $\mathcal{B}'$ returns 0 as the answer of this $\mathcal{TEST}$ query. Otherwise, if $((C_{TBE}, R'), \sigma)$ is a valid signature pair, then $\mathcal{B}'$ submits a forged pair $((C_{TBE}, R'), \sigma)$ to the sUF challenger and wins.

forge_2 : Now $t_{\omega_j} \in \{t_{\omega_0^*}, t_{\omega_1^*}\}$. Then $(C_{TBE}, \sigma) \neq (C_{TBE}^*, \sigma^*)$. $\mathcal{B}'$ computes $C_{IBE} \leftarrow \mathsf{TBE.Dec}(sk_S, H_{tag}(K_v), C_{TBE})$ and $R' \leftarrow \mathsf{IBE.Dec}(t_{\omega_j}, C_{IBE})$. If $((C_{TBE}, R'), \sigma)$ is not a valid signature pair, then $\mathcal{B}'$ returns 0 as the answer of this $\mathcal{TEST}$ query. Otherwise, if $((C_{TBE}, R'), \sigma)$ is a valid signature pair, then $\mathcal{B}'$ submits a forged pair $((C_{TBE}, R'), \sigma)$ to the sUF challenger and wins.

Therefore, $\Pr[\mathsf{forge}] := \Pr[\mathsf{forge}_1 \vee \mathsf{forge}_2]$ is negligible, since the underlying signature is sUF. □

4.3 IBE with Partitioned Ciphertext Structure (PCS-IBE)

The role of the KEM/DEM framework in the first adaptive SCF-PEKS construction (presented in section 4) is that an IBE ciphertext is regarded as a TBE plaintext to hide keyword information from an adversary that has mk (this adversary appears in the Adaptive-IND-CKA-AT experiment). In this section, we propose an extension of the first SCF-PEKS construction, which need not require hybrid encryption. Unfortunately, we assume that the underlying IBE belongs a special class, however, while previously known pairing-based anonymous IBE schemes [10, 12, 14, 15, 19, 27, 44] belong to this class. In other words, we do not have to apply hybrid encryption as long as a previously known anonymous IBE scheme (enumerated in the above list) is used as a building tool of adaptive SCF-PEKS.

Here, we show that the KEM/DEM framework can be reduced if the underlying IBE satisfies the following properties (called IBE with partitioned ciphertext structure (PCS-IBE)[1]).

Definition 4.1 (PCS-IBE)**.** *We say that IBE is PCS-IBE if its ciphertext C_{IBE} can be split into two parts $C_{IBE} := (C_{IBE,1}, C_{IBE,2})$ with the following properties.*

- *$C_{IBE,1}$ is a single group element.*
  - *Note that the essential condition is $C_{IBE,1} \in \mathcal{M}_{TBE}$. However, since Kiltz [36] proposed a TBE scheme with $\mathcal{M}_{TBE} = \mathbb{G}$ and $\mathcal{M}_{TBE} = \mathbb{G}_T$, respectively, where $(\mathbb{G}, \mathbb{G}_T)$ is a bilinear group, it is enough to require that $C_{IBE,1}$ is a single group element.*
- *$C_{IBE,1}$ only includes an identity ID (i.e., $C_{IBE,2}$ is independent of ID).*
- *For any common message M and distinct identities ID and ID' ($ID \neq ID'$), $C_{IBE,2}$ can be commonly used for $(C_{IBE,1}, C_{IBE,2}) \leftarrow \mathsf{IBE.Enc}(pk, ID, M; s)$ and $(C'_{IBE,1}, C_{IBE,2}) \leftarrow \mathsf{IBE.Enc}(pk, ID', M; s)$ if the **same random number s** is used for both encryptions.*
  - *That is, both $(C_{IBE,1}, C_{IBE,2})$ and $(C'_{IBE,1}, C_{IBE,2})$ are valid ciphertexts.*

This structure is used for computing the challenge ciphertext in the proof of the adaptive IND-CKA-AT. In the proof, no matter which plaintext $(C_{0,IBE,1}, C_{1,IBE,1})$ is encrypted, both $C_{0,IBE,2}$ and $C_{1,IBE,2}$ can be used as a part of the challenge ciphertext, since $C_{0,IBE,2} = C_{1,IBE,2}$ due to the PCS property.

Here, we explain the above structure in the Boneh-Franklin (BF) IBE scheme [10] case due to its easy-to-understand structure as follows: For a message M and an identity ID, a ciphertext $(C_{IBE,1}, C_{IBE,2})$ is described as $C_{IBE,1} = M \cdot e(Y, Hash(ID)^s)$ and $C_{IBE,2} = g^s$, where Y is a public key of the key authority, and $s \in \mathbb{Z}_p$ (with a prime order p) is a random number of an encryptor's choice. Then, for the common message M, an another identity ID', and the same random number s, $(C'_{IBE,1}, C_{IBE,2})$ is also a valid ciphertext, where $C'_{IBE,1} = M \cdot e(Y, Hash(ID')^s)$.

In the Gentry IBE case (which is used for our instantiations), for a message M and an identity ID, a ciphertext $(C_{IBE,1}, C_{IBE,2})$ is described as $C_{IBE,1} = (g' g^{-ID})^s$ and $C_{IBE,2} = \big(e(g,g)^s, M \cdot e(g,h)^{-s}\big)$. So, for the common message M, an another identity ID', and the same random number s, $(C'_{IBE,1}, C_{IBE,2})$ is also a valid ciphertext, where $C'_{IBE,1} = (g' g^{-ID'})^s$.

4.4 The Second Adaptive SCF-PEKS Construction based on PCS-IBE

Although $\mathsf{SCF\text{-}PEKS.KeyGen_S}$, $\mathsf{SCF\text{-}PEKS.KeyGen_R}$, and $\mathsf{SCF\text{-}PEKS.Trapdoor}$ are the same as these of the first adaptive SCF-PEKS construction (protocol 2), for the sake of clarity, we descrbe these algorithms in the following. Let a ciphertext space of the underlying PCS-IBE be $\mathcal{C}_{IBE} = \mathcal{C}_{IBE,1} \times \mathcal{C}_{IBE,2}$. We set $\mathcal{C}_{IBE,1} = \mathcal{M}_{TBE}$ and $\mathcal{M}_{Sig} = \mathcal{C}_{IBE,2} \times \mathcal{C}_{TBE} \times \mathcal{M}_{IBE}$.

[1] Note that our partitioned requirement is different from that of partitioned IBKEM [2].

Our Second Adaptive SCF-PEKS Construction w/o Hybrid Encryption

$\mathsf{SCF\text{-}PEKS.KeyGen_S}(1^\kappa)$**:** Run $(pk_S, sk_S) \leftarrow \mathsf{TBE.KeyGen}(1^\kappa)$, and output (pk_S, sk_S).

$\mathsf{SCF\text{-}PEKS.KeyGen_R}(1^\kappa)$**:** Run $(pk_R, sk_R) \leftarrow \mathsf{IBE.KeyGen}(1^\kappa)$, and output (pk_R, sk_R).

$\mathsf{SCF\text{-}PEKS.Trapdoor}(sk_R, \omega)$**:** Run $t_\omega \leftarrow \mathsf{IBE.Extract}(sk_R, \omega)$, and output t_ω.

$\mathsf{SCF\text{-}PEKS.Enc}(pk_S, pk_R, \omega)$**:** Generate $(K_s, K_v) \xleftarrow{\$} \mathsf{Sig.KeyGen}$, compute $t = H_{tag}(K_v)$, choose $R \xleftarrow{\$} \mathcal{M}_{IBE}$, run $(C_{IBE,1}, C_{IBE,2}) \leftarrow \mathsf{IBE.Enc}(pk_R, \omega, R)$, $C_{TBE} \leftarrow \mathsf{TBE.Enc}(pk_S, t, C_{IBE,1})$, and $\sigma \leftarrow \mathsf{Sign}(K_s, (C_{IBE,2}, C_{TBE}, R))$, and output $\lambda = (C_{IBE,2}, C_{TBE}, K_v, \sigma)$.

$\mathsf{SCF\text{-}PEKS.Test}(\lambda, sk_S, t_\omega)$**:** Let $\lambda = (C_{IBE,2}, C_{TBE}, K_v, \sigma)$. Compute $t = H_{tag}(K_v)$, and run $C'_{IBE,1} \leftarrow \mathsf{TBE.Dec}(sk_S, t, C_{TBE})$ and $R' \leftarrow \mathsf{IBE.Dec}(t_\omega, (C'_{IBE,1}, C_{IBE,2}))$. Output 1 if $1{=}\mathsf{Verify}(K_v, \sigma, (C_{IBE,2}, C_{TBE}, R'))$, and 0 otherwise.

The security proofs are the same as those of the first ones, except the construction of the challenge ciphertext in the adaptive-IND-CKA-AT experiment as follows: In the $\mathsf{Challenge}$ phase, $\mathcal{B}$ needs to compute the SCF-PEKS challenge ciphertext λ^*, although $\mathcal{B}$ does not know $\mu \in \{0,1\}$ chosen by the TBE challenger $\mathcal{C}$. So, our PCS property comes into effect to compute λ^*, since $\mathcal{B}$ can use both $C_{0,IBE,2}$ and $C_{1,IBE,2}$ (so we set $C^*_{IBE,2} = C_{0,IBE,2}$). More concretely, in the $\mathsf{Challenge}$ phase, $\mathcal{A}$ sends the challenge keywords ω_0^* and ω_1^* to $\mathcal{B}$, $\mathcal{B}$ chooses $R^* \xleftarrow{\$} \mathcal{M}_{IBE}$, and computes the challenge ciphertext as follows:

1. $\mathcal{B}$ computes $(C_{0,IBE,1}, C_{0,IBE,2}) \leftarrow \mathsf{IBE.Enc}(pk_R, \omega_0^*, R^*)$ and $(C_{1,IBE,1}, C_{1,IBE,2}) \leftarrow \mathsf{IBE.Enc}(pk_R, \omega_1^*, R^*)$ **using the same random number** (i.e., $C_{0,IBE,2} = C_{1,IBE,2}$). $\mathcal{B}$ sets $C^*_{IBE,2} := C_{0,IBE,2}$.
   - Note that both $(C_{0,IBE,1}, C^*_{IBE,2})$ and $(C_{1,IBE,1}, C^*_{IBE,2})$ are valid ciphertexts of the underlying IBE scheme. This is the reason we require anonymous "PCS"-IBE.
2. $\mathcal{B}$ sends $(M_0^*, M_1^*) := (C_{0,IBE,1}, C_{1,IBE,1})$ to $\mathcal{C}$ as the challenge messages.
3. $\mathcal{C}$ gives $C^*_{TBE} \leftarrow \mathsf{TBE.Enc}(pk_S, t^*, M^*_\mu)$ to $\mathcal{B}$, where $\mu \in \{0,1\}$ is the challenge bit.
4. $\mathcal{B}$ computes $\sigma^* \leftarrow \mathsf{Sign}(K_s^*, (C^*_{IBE,2}, C^*_{TBE}, R^*))$, and sends $\lambda^* = (C^*_{IBE,2}, C^*_{TBE}, K_v^*, \sigma^*)$ to $\mathcal{A}$.

Then, λ^* is a valid ciphertext due to the PCS property. Since $\mathcal{B}$ does not have to consider the bit μ chosen by $\mathcal{C}$, $\mathcal{B}$ can use $C^*_{IBE,2}$.

As in the first one, non-adaptive SCF-PEKS can be constructed by reducing the one-time signature part and replacing the TBE part with CPA-secure PKE. Let the underlying IBE be PCS, then a ciphertext is $(C_{IBE,2}, C_{PKE}, R)$, where $(C_{IBE,1}, C_{IBE,2}) \leftarrow \mathsf{IBE.Enc}(pk_R, \omega, R)$ and $C_{PKE} \leftarrow \mathsf{PKE.Enc}(pk_S, C_{IBE,1})$.

4.5 Comparison Between Our First/Second Adaptive SCF-PEKS Constructions

In the first adaptive SCF-PEKS construction (protocol 1, section 4), the DEM part $e = \mathsf{E}_k(C_{IBE})$ is implicitly included in C_{TBE}. Here, we explicitly include e in a SCF-PEKS ciphertext as follows: $\lambda_1 = (e, C_{TBE}, K_v, \sigma)$ (subscript 1 means that it is a ciphertext of the first adaptive SCF-PEKS construction). On the contrary, $\lambda_2 = (C_{IBE,2}, C_{TBE}, K_v, \sigma)$ in the second SCF-PEKS construction (proposed in this section, protocol 2, subscript 2 means that it is a ciphertext of the second adaptive SCF-PEKS construction). Since the size of e is at least the same size of C_{IBE}, by excluding the DEM part, the size of the ciphertext of the second construction (say $|\lambda_2|$) is smaller than that of the first one (say $|\lambda_1|$): i.e., $|\lambda_1| \geq |C_{IBE,1}| + |\lambda_2|$. Since the ciphertext size is the most bottlenecked point of our adaptive SCF-PEKS construction compared to the concrete constructions, we can say that the second adaptive SCF-PEKS construction is more efficient than the first, though it is not fully generic.

Table 2: Comparison between our constructions and the Fang et al. SCF-PEKS

	Comp. λ	Comp. Test	Length of λ	Adaptive Security
Fang et al. [24]	$2ME(\mathbb{G}) + 3ME(\mathbb{G}_T)$ $(11ME(\mathbb{G}))$	$ME(\mathbb{G}) + 2ME(\mathbb{G}_T) + 2BM$ $(7ME(\mathbb{G}) + 2BM)$	$2\|\mathbb{G}\| + 2\|\mathbb{G}_T\|$ (2382 bits)	No
GBBS construction	$4ME(\mathbb{G}) + 2ME(\mathbb{G}_T)$ $(10ME(\mathbb{G}))$	$ME(\mathbb{G}) + ME(\mathbb{G}_T) + BM$ $(4ME(\mathbb{G}) + BM)$	$3\|\mathbb{G}\| + 3\|\mathbb{G}_T\|$ (3573 bits)	No
GKBS construction	$8ME(\mathbb{G}) + 2ME(\mathbb{G}_T)$ $(14ME(\mathbb{G}))$	$5ME(\mathbb{G}) + ME(\mathbb{G}_T) + BM$ $(8ME(\mathbb{G}) + BM)$	$7\|\mathbb{G}\| + 2\|\mathbb{G}_T\| + \|\mathbb{Z}_p\| + \kappa$ (3577 bits)	Yes

5 A Concrete Instantiation of Adaptive SCF-PEKS

5.1 The GKBS Construction

Here, by using the extended version of our adaptive SCF-PEKS construction, we instantiate an adaptive SCF-PEKS scheme based on the Gentry (PCS) anonymous IBE [27], the Kiltz IND-stag-CCA-secure TBE [36], and the Bellare-Shoup sUF one-time signature [7]. We call it the GKBS construction by picking up authors' name. Let $\mathbb{G}$ and $\mathbb{G}_T$ be cyclic groups of prime order p, e be an efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, and $H_{sig} : \{0,1\}^\kappa \times \{0,1\}^* \to \mathbb{Z}_p$ be a CR hash function, where each κ-bit key K specifies a particular hash function $H(K, \cdot)$ with domain $\{0,1\}^*$. We assume that $e(g,g)$ and $e(g,h)$ are included in public keys to reduce the number of pairing computations.

An adaptive SCF-PEKS scheme without random oracles (the GKBS construction)

$\mathsf{SCF\text{-}PEKS.KeyGen_S}(1^\kappa)$**:** Choose $g_1 \xleftarrow{\$} \mathbb{G}$ and $x_1, x_2, y_1, y_2 \xleftarrow{\$} \mathbb{Z}_p$. Choose $g_2, z \in \mathbb{G}$ with $g_1^{x_1} = g_2^{x_2} = z$. Compute $u_1 = g_1^{y_1}$ and $u_2 = g_2^{y_2}$. Output $(pk_S, sk_S) = \big((g_1, g_2, z, u_1, u_2), (x_1, x_2, y_1, y_2)\big)$.

$\mathsf{SCF\text{-}PEKS.KeyGen_R}(1^\kappa)$**:** Choose $g, h \xleftarrow{\$} \mathbb{G}$ and $\alpha \xleftarrow{\$} \mathbb{Z}_p$, compute $g' = g^\alpha$, and output $(pk_R, sk_R) = \big((g', h, e(g,g), e(g,h)), \alpha\big)$.

$\mathsf{SCF\text{-}PEKS.Trapdoor}(sk_R, \omega)$**:** For a keyword $\omega \in \mathbb{Z}_p$, choose $r_\omega \xleftarrow{\$} \mathbb{Z}_p$, compute $h_\omega = (hg^{-r_\omega})^{\frac{1}{\alpha-\omega}}$, and output $t_\omega = (r_\omega, h_\omega)$.

$\mathsf{SCF\text{-}PEKS.Enc}(pk_S, pk_R, \omega)$**:** Choose $R \xleftarrow{\$} \mathbb{G}_T$, $s, r_1, r_2, x, y \xleftarrow{\$} \mathbb{Z}_p$, and $K \xleftarrow{\$} \{0,1\}^\kappa$. Compute $X = g^x$, $Y = g^y$, set $K_v = (K, X, Y)$, and compute $t = H_{tag}(K_v)$, $C_{IBE,1} = (g'g^{-\omega})^s$, $C_{IBE,2} = \big(e(g,g)^s, R \cdot e(g,h)^{-s}\big)$, $C_{TBE} = \big(g_1^{r_1}, g_2^{r_2}, (z^t u_1)^{r_1}, (z^t u_2)^{r_2}, C_{IBE,1} \cdot z^{r_1+r_2}\big)$, $c = H_{sig}\big(K, Y||(C_{IBE,2}, C_{TBE}, R)\big)$, and $\sigma = c + yx \bmod p$. Output $\lambda = (C_{IBE,2}, C_{TBE}, \sigma, K_v)$.

$\mathsf{SCF\text{-}PEKS.Test}(\lambda, sk_S, t_\omega)$**:** Parse $sk_S = (x_1, x_2, y_1, y_2)$, $t_\omega = (r_\omega, h_\omega)$, $C_{IBE,2} = (f_1, f_2)$, $C_{TBE} = (v_1, v_2, v_3, v_4, v_5)$, and $K_v = (K, X, Y)$. Compute $t = H_{tag}(K_v)$, and check $v_1^{tx_1+y_1} \stackrel{?}{=} v_3$ and $v_2^{tx_2+y_2} \stackrel{?}{=} v_4$. If not, then output 0. Otherwise, compute $C'_{IBE,1} = v_5/(v_1^{x_1} \cdot v_2^{x_2})$, $R' = f_1^{r_\omega} \cdot e(C'_{IBE,1}, h_\omega) \cdot f_2$, and $c = H_{sig}\big(K, Y||(C_{IBE,2}, C_{TBE}, R')\big)$, and check $g^z \stackrel{?}{=} YX^c$. If not, then output 0. Otherwise, output 1.

We assume the difficulty of the one-more-discrete-log (omdl) problem [5][2], the decisional augmented bilinear Diffie-Hellman exponent (decisional ABDHE) problem [27], and the gap decision linear (gap DLIN) problem [36], and the collision resistance of H_{tag} and H_{sig}. Then, the above SCF-PEKS instantiation is adaptive secure in the standard model.

[2]We can use a discrete-log-based sUF one-time signature [28, 47] with one more $\mathbb{Z}_p$ element.

Next, we estimate the efficiency of the GKBS construction. Although concrete SCF-PEKS schemes have been proposed [4, 29, 30, 42], these schemes are proved in the random oracle model. As a well-known fact, efficient cryptographic schemes can be constructed easily if the random oracle is assumed. So, we focus on SCF-PEKS schemes proposed by Fang et al. [24] and Khader [35], respectively, which are secure in the standard model. Khader [35] shows that PEKS and SCF-PEKS can be constructed by using k-resilient IBE [32] (which is an IBE scheme, wherein an adversary can obtain at most k private keys of IDs). Since k-resilient IBE [32] is designed by applying a DDH-hard group without pairings, the Khader PEKS/SCF-PEKS also enables pairing-free constructions. Unfortunately, the Khader PEKS/SCF-PEKS schemes require k-dependent large number of public keys and high encryption costs. Therefore, here we compare our GKBS construction to the Fang et al. SCF-PEKS scheme [24] in Table 2 (the Fang et al. SCF-PEKS scheme is introduced in the Appendix). In addition, for comparison, we instantiate a non-adaptive SCF-PEKS scheme (using the second construction). We call this the GBBS construction which is based on the Gentry IBE scheme [27] and the linear encryption scheme presented by Boneh, Boyen, and Shacham [8] (the actual construction of this non-adaptive SCF-PEKS scheme is given in the Appendix). The GBBS construction achieves the same security level as that of the Fang et al. construction.

Let $ME(\mathbb{G})$ and $ME(\mathbb{G}_T)$ be the computational costs of multi-exponentiation in $\mathbb{G}$ and $\mathbb{G}_T$, respectively, BM be that of one bilinear map computation, and $|\mathbb{G}|$, $|\mathbb{G}_T|$, and $|\mathbb{Z}_p|$ be the bit-length of the representation of an element of $\mathbb{G}$, $\mathbb{G}_T$, and $\mathbb{Z}_p$, respectively. More precisely, we assume that the security parameter $\kappa = 170$. Therefore, p is a 170-bit prime, $|\mathbb{G}| = 171$ bits and $|\mathbb{G}_T| = 1020$ bits: i.e., we assume that $\mathbb{G}$ is an elliptic curve defined over finite field $\mathbb{F}_p$ and $\mathbb{G}_T$ is a multiplicative group on finite field $\mathbb{F}_{p^k}^{\times}$ with the embedded degree $k = 6$. In this case, the computational complexity over $\mathbb{G}_T$ is approximately three times higher than that of $\mathbb{G}$. So, we estimate $ME(\mathbb{G}_T) = 3ME(\mathbb{G})$, and write them in Table 2 in parentheses. Although in the GKBS construction the length of the ciphertext is larger than that of the Fang et al. construction, the computation of the Test algorithm is faster (if $BM < ME(\mathbb{G})$ which usually holds). Therefore, there is not much efficiency difference between our GKBS construction and the Fang et al. scheme, although the GKBS construction enables adaptive security.

6 Conclusion

In this paper, from a theoretical perspective, we show that no additional cryptographic primitive is required compared to the Abdalla et al. PEKS construction, even though adaptive SCF-PEKS requires additional functionalities.

From a practical perspective, since malicious receivers can use the server as the test oracle, our adaptive security notion is applicable in practice. In addition, our concrete adaptive SCF-PEKS construction (the GKBS construction) achieves a similar level of efficiency for the costs of the test procedure and encryption, compared to the (non-adaptive secure) SCF-PEKS scheme without random oracles proposed by Fang et al, even though adaptive SCF-PEKS requires additional functionalities.

References

[1] Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. *J. Cryptology*, 21(3):350–391, 2008.

[2] Masayuki Abe, Yang Cui, Hideki Imai, and Eike Kiltz. Efficient hybrid encryption from ID-based encryption. *Des. Codes Cryptography*, 54(3):205–240, 2010.

[3] Joonsang Baek, Reihaneh Safavi-Naini, and Willy Susilo. On the integration of public key data encryption and public key encryption with keyword search. In *Proceedings of ISC 2006*, pages 217–232, Samos Island, Greece, 30 August- 2 September 2006. Springer-Verlag, Berlin.

[4] Joonsang Baek, Reihaneh Safavi-Naini, and Willy Susilo. Public key encryption with keyword search revisited. In *Proceedings of ICCSA 2008*, pages 1249–1259, Perugia, Italy, 30 June - 3 July 2008. Springer-Verlag, Berlin.

[5] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The one-more-RSA-inversion problems and the security of chaum's blind signature scheme. *J. Cryptology*, 16(3):185–215, 2003.

[6] Mihir Bellare and Phillip Rogaway. Collision-resistant hashing: Towards making UOWHFs practical. In *Proceedings of CRYPTO 1997*, pages 470–484, Santa Barbara, California, USA, 17-21 August 1997. Springer-Verlag, Berlin.

[7] Mihir Bellare and Sarah Shoup. Two-tier signatures, strongly unforgeable signatures, and Fiat-Shamir without random oracles. In *Proceedings of PKC 2007*, pages 201–216, Beijing, China, 16-20 April 2007. Springer-Verlag, Berlin.

[8] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *Proceedings of CRYPTO 2004*, pages 41–55, Santa Barbara, California, USA, 15-19 August 2004.

[9] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In *Proceedings of EUROCRYPT 2004*, pages 506–522, Interlaken, Switzerland, 2-6 May 2004. Springer-Verlag, Berlin.

[10] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.

[11] Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In *Proceedings of TCC 2007*, pages 535–554, Amsterdam, The Netherlands, 21-24 February 2007. Springer-Verlag, Berlin.

[12] Xavier Boyen and Brent Waters. Anonymous hierarchical identity-based encryption (without random oracles). In *Proceedings of CRYPTO 2006*, pages 290–307, Santa Barbara, California, USA, 20-24 August 2006. Springer-Verlag, Berlin.

[13] Jin Wook Byun, Hyun Suk Rhee, Hyun-A Park, and Dong Hoon Lee. Off-line keyword guessing attacks on recent keyword search schemes over encrypted data. In *Proceedings of SDM 2006*, pages 75–83, Seoul, Korea, 10-11 September 2006. Springer-Verlag, Berlin.

[14] Jan Camenisch, Markulf Kohlweiss, Alfredo Rial, and Caroline Sheedy. Blind and anonymous identity-based encryption and authorised private searches on public key encrypted data. In *Public Key Cryptography*, pages 196–214, 2009.

[15] Angelo De Caro, Vincenzo Iovino, and Giuseppe Persiano. Fully secure anonymous HIBE and secret-key anonymous IBE with short ciphertexts. In *Proceedings of Pairing 2010*, pages 347–366, Yamanaka Hot Spring, Japan, 13-15 December 2010. Springer-Verlag, Berlin.

[16] Yu-Chi Chen and Gwoboa Horn. Timestamped conjunctive keyword-searchable public key encryption. In *ICICIC2009*, pages 729–732, Kaohsiung, Taiwan, 7-9 December 2009. IEEE Computer Society, Washington, DC, USA.

[17] Giovanni Di Crescenzo and Vishal Saraswat. Public key encryption with searchable keywords based on jacobi symbols. In *Proceedings of INDOCRYPT 2007*, pages 282–296, Chennai, India, 9-13 December 2007. Springer-Verlag, Berlin.

[18] Yang Cui, Eiichiro Fujisaki, Goichiro Hanaoka, Hideki Imai, and Rui Zhang. Formal security treatments for IBE-to-signature transformation: Relations among security notions. *IEICE Transactions*, 92-A(1):53–66, 2009.

[19] Léo Ducas. Anonymity from asymmetry: New constructions for anonymous HIBE. In *Proceedings of CT-RSA 2010*, pages 148–164, San Francisco, CA, USA, 1-5 March 2010. Springer-Verlag, Berlin.

[20] Keita Emura, Atsuko Miyaji, and Kazumasa Omote. Adaptive secure-channel free public-key encryption with keyword search implies timed release encryption. In *Proceedings of ISC 2011*, pages 102–118, Xi'an, China, 26-29 October 2011. Springer-Verlag, Berlin.

[21] Keita Emura, Atsuko Miyaji, Mohammad Shahriar Rahman, and Kazumasa Omote. Generic constructions of secure-channel free searchable encryption with adaptive security. *Security and Communication Networks*, 8(8):1547–1560, 2015.

[22] Keita Emura, Le Trieu Phong, and Yohei Watanabe. Keyword revocable searchable encryption with trapdoor exposure resistance and re-generateability. In *IEEE TrustCom*, pages 167–174, 2015.

[23] Keita Emura and Mohammad Shahriar Rahman. Constructing secure-channel free searchable encryption from anonymous IBE with partitioned ciphertext structure. In *Proceedings of SECRYPT 2012*, pages 84–93, Rome, Italy, 24-27 July 2012. SciTePress, Setubal.

[24] Liming Fang, Willy Susilo, Chunpeng Ge, and Jiandong Wang. A secure channel free public key encryption with keyword search scheme without random oracles. In *Proceedings of CANS 2009*, pages 248–258, Kanazawa, Japan, 12-14 December 2009. Springer-Verlag, Berlin.

[25] Liming Fang, Willy Susilo, Chunpeng Ge, and Jiandong Wang. Public key encryption with keyword search secure against keyword guessing attacks without random oracle. *Inf. Sci.*, 238:221–241, 2013.

[26] Thomas Fuhr and Pascal Paillier. Decryptable searchable encryption. In *Proceedings of ProvSec 2007*, pages 228–236, Wollongong, Australia, 1-2 November 2007. Springer-Verlag, Berlin.

[27] Craig Gentry. Practical identity-based encryption without random oracles. In *Proceedings of EUROCRYPT 2006, St. Petersburg, Russia*, pages 445–464. Springer-Verlag, Berlin, 28 May - 1 June 2006.

[28] Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In *ASIACRYPT*, pages 444–459, Shanghai, China, 3-7 December 2006. Springer-Verlag, Berlin.

[29] Chunxiang Gu and Yuefei Zhu. New efficient searchable encryption schemes from bilinear pairings. *International Journal of Network Security*, 10(1):25–31, 2010.

[30] Chunxiang Gu, Yuefei Zhu, and Heng Pan. Efficient public key encryption with keyword search schemes from pairings. In *Proceedings of Inscrypt 2007*, pages 372–383, Xining, China, 31 August - 5 September 2007. Springer-Verlag, Berlin.

[31] Lifeng Guo and Wei-Chuen Yau. Efficient secure-channel free public key encryption with keyword search for EMRs in cloud storage. *J. Medical Systems*, 39(2):11, 2015.

[32] Swee-Huay Heng and Kaoru Kurosawa. k-resilient identity-based encryption in the standard model. *IEICE Transactions*, 89-A(1):39–46, 2006.

[33] Yong Ho Hwang and Pil Joong Lee. Public key encryption with conjunctive keyword search and its extension to a multi-user system. In *Proceedings of Pairing 2007*, pages 2–22, Tokyo, Japan, 2-4 July 2007. Springer-Verlag, Berlin.

[34] Ik Rae Jeong, Jeong Ok Kwon, Dowon Hong, and Dong Hoon Lee. Constructing PEKS schemes secure against keyword guessing attacks is possible? *Computer Communications*, 32(2):394–396, 2009.

[35] Dalia Khader. Public key encryption with keyword search based on k-resilient IBE. In *Proceedings of ICCSA 2007*, pages 1086–1095, Kuala Lumpur, Malaysia, 26-29 August 2007. Springer-Verlag, Berlin.

[36] Eike Kiltz. Chosen-ciphertext security from tag-based encryption. In *Proceedings of TCC 2006, New York, NY, USA*, pages 581–600. Springer-Verlag, Berlin, 4-7 March 2006.

[37] Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, 17(2):373–386, 1988.

[38] Yinbin Miao, Jianfeng Ma, and Zhiquan Liu. Revocable and anonymous searchable encryption in multi-user setting. *Concurrency and Computation: Practice and Experience*, 28(4):1204–1218, 2016.

[39] Steven Myers and Abhi Shelat. Bit encryption is complete. In *Proceedings of FOCS 2009*, pages 607–616, Atlanta, Georgia, 25-27 October 2009. IEEE Computer Society, Washington, DC, USA.

[40] Mototsugu Nishioka. Perfect keyword privacy in PEKS systems. In *Proceedings of ProvSec 2012*, pages 175–192, Chengdu, China, 26-28 September 2012. Springer-Verlag, Berlin.

[41] Dong Jin Park, Kihyun Kim, and Pil Joong Lee. Public key encryption with conjunctive field keyword search. In *Proceedings of WISA 2004*, pages 73–86, Jeju Island, Korea, 23-25 August 2004. Springer-Verlag, Berlin.

[42] Hyun Sook Rhee, Jong Hwan Park, Willy Susilo, and Dong Hoon Lee. Improved searchable public key encryption with designated tester. In *Proceedings of ASIACCS 2009*, pages 376–379, Sydney, Australia, 10-12 March 2009. ACM, New York, NY, USA.

[43] Hyun Sook Rhee, Willy Susilo, and Hyun jeong Kim. Secure searchable public key encryption scheme against keyword guessing attacks. In *IEICE Electronics Express Vol 6 (5)*, pages 237–243, 2009.

[44] Jae Hong Seo, Tetsutaro Kobayashi, Miyako Ohkubo, and Koutarou Suzuki. Anonymous hierarchical identity-based encryption with constant size ciphertexts. In *Public Key Cryptography*, pages 215–234, Irvine, CA, USA, 18-20 March 2009. Springer-Verlag, Berlin.

[45] Victor Shoup. Using hash functions as a hedge against chosen ciphertext attack. In *Proceedings of EUROCRYPT 2000*, pages 275–288, Bruges, Belgium, 14-18 May 2000. Springer-Verlag, Berlin.

[46] Qiang Tang. Revisit the concept of PEKS: Problems and a possible solution. In *Technical Report TR-CTIT-08-54, Centre for Telematics and Information Technology University of Twente, Enschede.*, 2008.

[47] Hoeteck Wee. Public key encryption against related key attacks. In *Public Key Cryptography*, pages 262–279, Darmstadt, Germany, 21-23 May 2012. Springer-Verlag, Berlin.

[48] Wei-Chuen Yau, Swee-Huay Heng, and Bok-Min Goi. Off-line keyword guessing attacks on recent public key encryption with keyword search schemes. In *Proceedings of ATC 2008*, pages 100–105, Oslo, Norway, 23-25 June 2008. Springer-Verlag, Berlin.

[49] Yong Yu, Jianbing Ni, Haomiao Yang, Yi Mu, and Willy Susilo. Efficient public key encryption with revocable keyword search. *Security and Communication Networks*, 7(2):466–472, 2014.

[50] Rui Zhang and Hideki Imai. Generic combination of public key encryption with keyword search and public key encryption. In *Proceedings of CANS 2007*, pages 159–174, Singapore, 8-10 December 2007. Springer-Verlag, Berlin.

Appendix.A

Here, we introduce the SCF-PEKS scheme proposed by Fang et al. [24]. Let $\mathbb{G}$ and $\mathbb{G}_T$ be cyclic groups of prime order p, e be an efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ and $g \in \mathbb{G}$ be a generator. We assume that $e(X, Q)$, $e(g, g)$, and $e(g, h)$ are included in public keys to reduce pairing computations.

The Fang et al. SCF-PEKS scheme [24]

SCF-PEKS.KeyGen$_S(1^\kappa)$**:** Choose $x \xleftarrow{\$} \mathbb{Z}_p$ and $Q \xleftarrow{\$} \mathbb{G}$, compute $X = g^x$, and output $(pk_S, sk_S) = ((X, Q, e(X, Q), e(g, g)), x)$.

SCF-PEKS.KeyGen$_R(1^\kappa)$**:** Choose $y \xleftarrow{\$} \mathbb{Z}_p$ and $h \xleftarrow{\$} \mathbb{G}$, compute $Y = g^y$, and output $(pk_R, sk_R) = ((Y, h, e(g, h)), y)$.

SCF-PEKS.Trapdoor(sk_R, ω)**:** For a keyword $\omega \in \mathbb{Z}_p$, choose $r_\omega \xleftarrow{\$} \mathbb{Z}_p$, compute $h_\omega = (hg^{-r_\omega})^{\frac{1}{y-\omega}}$, and output $t_\omega = (r_\omega, h_\omega)$.

SCF-PEKS.Enc(pk_S, pk_R, ω)**:** For a keyword ω, choose $s, r \xleftarrow{\$} \mathbb{Z}_p$, compute $C_1 = g^s$, $t = H(e(X, Q)^s)$, $C_2 = (Yg^{-\omega})^{r/t}$, $C_3 = e(g, g)^r$, and $C_4 = e(g, h)^r$, and output $\lambda = (C_1, C_2, C_3, C_4)$.

SCF-PEKS.Test$(\lambda, sk_S, t_\omega)$**:** Compute $t = H(e(C_1, Q)^x)$, and check $e(C_2^t, h_\omega)C_3^{r_\omega} \stackrel{?}{=} C_4$. If the equation holds, output 1, and 0 otherwise.

The Fang et al. SCF-PEKS scheme is secure (in the sense of the non-adaptive security) if the decisional Bilinear Diffie Hellman (DBDH) assumption holds. This is the first SCF-PEKS scheme without random oracles. Note that the Fang et al. SCF-PEKS scheme is not adaptive secure, since there is a trivial attack as follows. Let a challenge ciphertext be $\lambda^* = (C_1^*, C_2^*, C_3^*, C_4^*) = (g^{s^*}, (Yg^{-W_\mu^*})^{r^*/t^*}, e(g, g)^{r^*}, e(g, h)^{r^*})$, where $t^* = H(e(X, Q,)^{s^*})$. In the adaptive-IND-CKA-AT experiment, $\mathcal{A}$ chooses $r' \xleftarrow{\$} \mathbb{Z}_p$, and computes $C_2' := (C_2^*)^{r'}$, $C_3' := (C_3^*)^{r'}$, and $C_4' := (C_4^*)^{r'}$. Then $\lambda' := (C_1^*, C_2', C_3', C_4')$ is a valid ciphertext for the keyword ω_μ^*, and $\lambda^* \neq \lambda'$. Therefore, $\mathcal{A}$ can issue a test query $(\lambda', t_{\omega_1^*})$, and outputs 1 if the answer to this query is 1, and 0 otherwise. We should notice that this attack is positioned in outside of their security models.

Appendix.B

Here, we instantiate a non-adaptive SCF-PEKS based on the Gentry (PCS) anonymous IBE [27] and linear encryption presented by Boneh, Boyen, and Shacham [8]. We assume that $e(g, g)$ and $e(g, h)$ are included in public keys to reduce pairing computations.

A non-adaptive SCF-PEKS scheme (the GBBS construction)

SCF-PEKS.KeyGen$_S(1^\kappa)$**:** Choose $x, y \in \mathbb{Z}_p$ and $u, v, z \in \mathbb{G}$ with $u^x = v^y = z$. Output $(pk_S, sk_S) = \big((u, v, z), (x, y)\big)$.

SCF-PEKS.KeyGen$_R(1^\kappa)$**:** Choose $g, h \xleftarrow{\$} \mathbb{G}$ and $\alpha \xleftarrow{\$} \mathbb{Z}_p$, compute $g' = g^\alpha$, and output $(pk_R, sk_R) = \big((g', h, e(g, g), e(g, h)), \alpha\big)$.

SCF-PEKS.Trapdoor(sk_R, ω)**:** For a keyword $\omega \in \mathbb{Z}_p$, choose $r_\omega \xleftarrow{\$} \mathbb{Z}_p$, compute $h_\omega = (hg^{-r_\omega})^{\frac{1}{\alpha-\omega}}$, and output $t_\omega = (r_\omega, h_\omega)$.

SCF-PEKS.Enc(pk_S, pk_R, ω)**:** Choose $R \xleftarrow{\$} \mathbb{G}_T$ and $s, r_1, r_2 \xleftarrow{\$} \mathbb{Z}_p$. Compute $C_{IBE,1} = (g'g^{-\omega})^s$, $C_{IBE,2} = \big(e(g, g)^s, R \cdot e(g, h)^{-s}\big)$, and $C_{PKE} = \big(u^{r_1}, v^{r_2}, C_{IBE,1} \cdot z^{r_1+r_2}\big)$. Output $\lambda = (C_{IBE,2}, C_{PKE}, R)$.

SCF-PEKS.Test$(\lambda, sk_S, t_\omega)$**:** Parse $sk_S = (x, y)$, $t_\omega = (r_\omega, h_\omega)$, $C_{IBE,2} = (f_1, f_2)$, and $C_{PKE} = (v_1, v_2, v_3)$. Compute $C'_{IBE,1} = v_3/(v_1^x \cdot v_2^y)$ and $R' = f_1^{r_\omega} \cdot e(C'_{IBE,1}, h_\omega) \cdot f_2$. Check $R' \stackrel{?}{=} R$. If not, then output 0. Otherwise, output 1.

The GBBS construction is secure (in the sense of the non-adaptive security) if the decisional ABDHE assumption and DLIN assumption hold. Note that the GBBS construction is not adaptive secure, since there is a trivial attack as follows. Let $\lambda^* = (e(g,g)^{s^*}, R^* \cdot e(g,h)^{-s^*}, C^*_{PKE}, R^*)$ be the challenge ciphertext. Then, choose $R' \in \mathbb{G}_T$, and compute $R' \cdot \left(R^* \cdot e(g,h)^{-s^*}\right)$ and $R' \cdot R^*$. Then $\lambda' = (e(g,g)^{s^*}, R' \cdot R^* \cdot e(g,h)^{-s^*}, C^*_{PKE}, R' \cdot R^*)$ is a valid ciphertext. Therefore, $\mathcal{A}$ can issue a test query $(\lambda', t_{\omega_1^*})$, and outputs 1 if the answer to this query is 1, and 0 otherwise. To avoid such an attack, TBE and OTS are required in our adaptive SCF-PEKS constructions.