Constructing Secure-channel Free Searchable Encryption from Anonymous IBE with Partitioned Ciphertext Structure

Keita Emura[1] and Mohammad Shahriar Rahman[2]
[1]*Network Security Research Institute, Security Architecture Laboratory, National Institute of Information and Communications Technology (NICT), Tokyo, Japan*
[2]*Department of Computer Science and Engineering, University of Asia Pacific, Dhaka, Bangladesh*

Keywords: Adaptive Secure-channel Free Public Key Encryption Scheme with Keyword Search, IBE with Partitioned Ciphertext Structure.

Abstract: As an extension of public key encryption with keyword search (PEKS), secure channel free PEKS (SCF-PEKS) has been considered. Generic construction of SCF-PEKS (with adaptive security) from strongly existentially unforgeable one-time signature, selective-tag CCA secure tag-based encryption (TBE) and anonymous identity-based encryption (IBE) has been proposed in ISC2011. Since this construction follows the double encryption, where a ciphertext of anonymous IBE is encrypted by TBE, hybrid encryption is applied because usually the ciphertext space of IBE is not equal to the plaintext space of TBE. In this paper, we show that hybrid encryption is not necessary as long as previously-known anonymous IBE schemes are used as a building tool of adaptive SCF-PEKS. Our result leads to a composability of IBE schemes whether they can be applied for constructing adaptive SCF-PEKS or not. Moreover, since we can exclude DEM part, our construction is efficient compared to the original one.

1 INTRODUCTION

1.1 Research Background

Any encryption scheme is required to be secure in the following sense: no information of plaintext is revealed from the corresponding ciphertext. Therefore, it seems hard to achieve to realize a searchable functionality against encrypted data. Due to such requirement, Public key Encryption scheme with Keyword Search (PEKS) has been proposed (Boneh et al., 2004b). In PEKS, a receiver makes a trapdoor t_ω for a keyword ω, and uploads it on a server. A sender makes a ciphertext of a keyword ω' by using the receiver's public key, and sends it to the server. The server outputs 1 if $\omega = \omega'$, by using t_ω, and 0 otherwise. Moreover, Secure-Channel Free PEKS (SCF-PEKS) have been proposed (Baek et al., 2008; Fang et al., 2009; Gu and Zhu, 2010; Gu et al., 2007; Khader, 2007) as an extension of PEKS. In SCF-PEKS, the server has a public/secret key pair, and the sender makes a ciphertext of a keyword ω' (which is encrypted by using both the server's public key and the receiver's 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. Note that a malicious receiver can use the server as the test oracle according to the following way (see Fig.1).

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) a 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.

To capture such circumstance, Emura et al. (Emura et al., 2011) consider a strong security notion of SCF-PEKS, called adaptive SCF-PEKS, where a "malicious-but-legitimate" receiver can be admitted to issue test queries adaptively, and show that adaptive SCF-PEKS implies timed-release encryption (Matsuda et al., 2010). Moreover,

Emura K. and Shahriar Rahman M..
Constructing Secure-channel Free Searchable Encryption from Anonymous IBE with Partitioned Ciphertext Structure.
DOI: 10.5220/0004022700840093
In *Proceedings of the International Conference on Security and Cryptography* (SECRYPT-2012), pages 84-93
ISBN: 978-989-8565-24-2

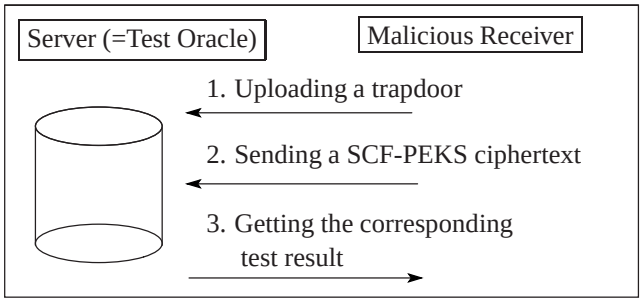

Figure 1: Instantiation of test queries in the real world.

they also gave 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) one-time signature (OTS). Briefly, this construction follows the double encryption, where a ciphertext of anonymous IBE is encrypted by TBE. Since usually the ciphertext space of IBE is not equal to the plaintext space of TBE, they applied the KEM/DEM framework (Shoup, 2000) (a.k.a. hybrid encryption), where KEM stands for key encapsulation mechanism, and DEM stands for data encapsulation mechanism.

1.2 Our Contribution

In this paper, we investigate the usage of hybrid encryption in the original construction, and show that hybrid encryption is not necessary as long as previously-known anonymous IBE schemes (e.g., (Boneh and Franklin, 2003; Boyen and Waters, 2006; Camenisch et al., 2009; Caro et al., 2010; Ducas, 2010; Gentry, 2006; Seo et al., 2009)) are used as its building tools. Our result leads to a composability of IBE schemes whether they can be applied for constructing adaptive SCF-PEKS or not. We define IBE with Partitioned Ciphertext Structure (PCS-IBE), where for any common message M and distinct identities ID and ID' ($ID \neq ID'$), a part of ciphertext can be "commonly" used for both ciphertexts if the "same random number" is used for both encryptions. Technically, this ciphertext shareability is the most significant point of the security proof, and such novel simulation technique has not been pointed out so far. Moreover, since we can exclude the DEM part of previous adaptive SCF-PEKS construction, our construction is efficient compared to the original one. Especially, we can reduce the ciphertext size. Note that the size of DEM part is at least the same size of IBE ciphertext, and the ciphertext size is bottleneck point of adaptive SCF-PEKS constructions compared to the concrete constructions. Finally, we instantiate an adaptive SCF-PEKS scheme which achieves the similar level 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 (See Table 1). Since we do not care about the keyword guessing attacks (Byun et al., 2006; Jeong et al., 2009; Rhee et al., 2009b; Yau et al., 2008), it can be an interesting future work.

2 PRELIMINARIES

In this section, we give the definitions of the building tools and adaptive SCF-PEKS.

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 1 (Syntax of TBE)**.** *A TBE scheme (Kiltz, 2006)* 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 (IND-stag-CCA)**.** *For any PPT adversary* $\mathcal{A}$ *and the security parameter* $\kappa \in \mathbb{N}$*, we define the experiment* $Exp^{IND\text{-}stag\text{-}CCA}_{\Pi,\mathcal{A}}(1^\kappa)$ *in Figure 2, and*

IND-stag-CCA

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

Figure 2: TBE experiment.

IBE-IND-CPA

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

IBE-ANO-CPA

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

Figure 3: IBE experiments.

define the advantage of $\mathcal{A}$ $Adv^{\textit{IND-stag-CCA}}_{\Pi,\mathcal{A}}(1^\kappa)$ *as follows.*

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

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* $\mathcal{DEC}$*.*

A TBE scheme Π *is said to be IND-stag-CCA secure if the advantage* $Adv^{\textit{IND-stag-CCA}}_{\Pi,\mathcal{A}}(1^\kappa)$ *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 3 (Syntax of IBE)**.** *IBE scheme* Π *consists of the following four algorithms,* IBE.Setup, IBE.Extract, IBE.Enc *and* IBE.Dec*:*

$\mathsf{IBE.Setup}(1^\kappa)$ *: 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 4 (IBE-IND-CPA)**.** *For any PPT adversary* $\mathcal{A}$ *and the security parameter* $\kappa \in \mathbb{N}$*, we define the experiment* $Exp^{\textit{IBE-IND-CPA}}_{\Pi,\mathcal{A}}(1^\kappa)$ *in Figure 3, and define the advantage of* $\mathcal{A}$ $Adv^{\textit{IBE-IND-CPA}}_{\Pi,\mathcal{A}}(1^\kappa)$ *as follows.*

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

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.*

An IBE scheme Π *is said to be IBE-IND-CPA secure if the advantage* $Adv^{\textit{IBE-IND-CPA}}_{\Pi,\mathcal{A}}(1^\kappa)$ *is negligible.*

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

Definition 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 Table 3, and*

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

Figure 4: OTS experiment.

define the advantage of $\mathcal{A}$ $Adv^{IBE\text{-}ANO\text{-}CPA}_{\Pi,\mathcal{A}}(1^\kappa)$ as follows.

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

ID^*_0 and ID^*_1 are not allowed as input to $\mathcal{EXTRACT}$ in the IBE-ANO-CPA experiment. An IBE scheme Π is said to be IBE-ANO-CPA secure if the advantage $Adv^{IBE\text{-}ANO\text{-}CPA}_{\Pi,\mathcal{A}}(1^\kappa)$ is negligible.

Definition 6 (Anonymous IBE)**.** *An IBE scheme is said to be anonymous IBE if the IBE scheme is both IBE-IND-CPA secure and IBE-ANO-CPA secure.*

2.3 Definitions of sUF OTS

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

Definition 7 (Syntax of OTS)**.** *A strongly existentially unforgeable (sUF) OTS against adaptively chosen message attack (CMA) (e.g., (Bellare and Shoup, 2007)) consists of the following three algorithms,* 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 8 (one-time sUF-CMA)**.** *For any PPT adversary $\mathcal{A}$ and the security parameter $\kappa \in \mathbb{N}$, we define the experiment $Exp^{\text{one-time sUF-CMA}}_{\Pi,\mathcal{A}}(1^\kappa)$ in Figure 4, and define the advantage of $\mathcal{A}$ $Adv^{\text{one-time sUF-CMA}}_{\Pi,\mathcal{A}}(1^\kappa)$ as follows.*

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

A signature scheme Π is said to be one-time sUF-CMA secure if the advantage $Adv^{\text{one-time sUF-CMA}}_{\Pi,\mathcal{A}}(1^\kappa)$ is negligible.

2.4 Definitions of Adaptive SCF-PEKS

Here, we introduce security requirements of SCF-PEKS defined in (Emura et al., 2011). In the following, $\mathcal{K}$ is a keyword space.

Definition 9 (Syntax of SCF-PEKS.)**.** *An SCF-PEKS scheme Π consists of the following five algorithms,* $\mathsf{SCF\text{-}PEKS.KeyGen_S}$, $\mathsf{SCF\text{-}PEKS.KeyGen_R}$, SCF-PEKS.Trapdoor, SCF-PEKS.Enc *and* SCF-PEKS.Test*:*

$\mathsf{SCF\text{-}PEKS.KeyGen_S}(1^\kappa)$ *: 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 text 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.*

Correctness is defined 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 state two security requirements "consistency" and "keyword privacy".

Definition 10 (Consistency)**.** *For any PPT adversary $\mathcal{A}$ and the security parameter $\kappa \in \mathbb{N}$, we define the experiment $Exp^{SCF\text{-}PEKS\text{-}CONSIST}_{\Pi,\mathcal{A}}(1^\kappa)$ in Figure 5, and define the advantage of $\mathcal{A}$ $Adv^{SCF\text{-}PEKS\text{-}CONSIST}_{\Pi,\mathcal{A}}(1^\kappa)$ as follows.*

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

Consistency

$$Exp^{\text{SCF-PEKS-CONSIST}}_{\Pi,\mathcal{A}}(1^\kappa) := \big[(pk_S, sk_S) \leftarrow \mathsf{SCF\text{-}PEKS.KeyGen_S}(1^\kappa);\ (pk_R, sk_R) \leftarrow \mathsf{SCF\text{-}PEKS.KeyGen_R}(1^\kappa);\ (\omega, \omega') \leftarrow \mathcal{A}(pk_S, pk_R); \omega \neq \omega';\ \lambda \leftarrow \mathsf{SCF\text{-}PEKS.Enc}(pk_S, pk_R, \omega);\ t_{\omega'} \leftarrow \mathsf{SCF\text{-}PEKS.Trapdoor}(sk_R, \omega');\ \mathsf{SCF\text{-}PEKS.Test}(\lambda, sk_S, t_{\omega'}) = 1\big]$$

IND-CKA-SSK

$$Exp^{\text{IND-CKA-SSK}}_{\Pi,\mathcal{A}}(1^\kappa) := \big[(pk_S, State) \leftarrow \mathcal{A}(1^\kappa);\ (pk_R, sk_R) \leftarrow \mathsf{SCF\text{-}PEKS.KeyGen_R}(1^\kappa);\ (\omega_0^*, \omega_1^*, State) \leftarrow \mathcal{A}^{\mathcal{TRAP}}(\mathsf{find}, pk_R, State);\ \mu \xleftarrow{\$} \{0,1\};\ \lambda^* \leftarrow \mathsf{SCF\text{-}PEKS.Enc}(pk_S, pk_R, \omega_\mu^*); \mu' \leftarrow \mathcal{A}^{\mathcal{TRAP}}(\mathsf{guess}, \lambda^*, State);\ \mu = \mu'\big]$$

Adaptive-IND-CKA-AT

$$Exp^{\text{Adaptive-IND-CKA-AT}}_{\Pi,\mathcal{A}}(1^\kappa) := \big[(pk_S, sk_S) \leftarrow \mathsf{SCF\text{-}PEKS.KeyGen_S}(1^\kappa);\ (pk_R, State) \leftarrow \mathcal{A}(1^\kappa);\ (\omega_0^*, \omega_1^*, State) \leftarrow \mathcal{A}^{\mathcal{TEST}}(\mathsf{find}, pk_S, State);\ \mu \xleftarrow{\$} \{0,1\};\ \lambda^* \leftarrow \mathsf{SCF\text{-}PEKS.Enc}(pk_S, pk_R, \omega_\mu^*); \mu' \leftarrow \mathcal{A}^{\mathcal{TEST}}(\mathsf{guess}, \lambda^*, State);\ \mu = \mu'\big]$$

Figure 5: SCF-PEKS experiments.

The SCF-PEKS scheme Π *is said to be computationally consistent if the advantage* $Adv^{\text{SCF-PEKS-CONSIST}}_{\Pi,\mathcal{A}}(1^\kappa)$ *is negligible.*

Next, we state two security notions for keyword privacy, "indistinguishability against chosen keyword attack with the server's 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. Note that $\mathcal{A}$ computes (pk_S, sk_S), and gives pk_S to the challenger. So, we omit sk_S in the IND-CKA-SSK experiment.

Definition 11 (IND-CKA-SSK)**.** *For any PPT adversary* $\mathcal{A}$ *and the security parameter* $\kappa \in \mathbb{N}$*, we define the experiment* $Exp^{\text{IND-CKA-SSK}}_{\Pi,\mathcal{A}}(1^\kappa)$ *in Figure 5, and define the advantage of* $\mathcal{A}$ $Adv^{\text{IND-CKA-SSK}}_{\Pi,\mathcal{A}}(1^\kappa)$ *as follows.*

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

Here, $\mathcal{TRAP}$ *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* $\mathcal{TRAP}$*.*

An SCF-PEKS scheme Π *is said to be IND-CKA-SSK-secure if the advantage* $Adv^{\text{IND-CKA-SSK}}_{\Pi,\mathcal{A}}(1^\kappa)$ *is negligible.*

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. Note that $\mathcal{A}$ 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 12 (Adaptive-IND-CKA-AT)**.** *For any PPT adversary* $\mathcal{A}$ *and the security parameter* $\kappa \in \mathbb{N}$*, we define the experiment* $Exp^{\text{Adaptive-IND-CKA-AT}}_{\Pi,\mathcal{A}}(1^\kappa)$ *in Figure 5, and define the advantage of* $\mathcal{A}$ $Adv^{\text{Adaptive-IND-CKA-AT}}_{\Pi,\mathcal{A}}(1^\kappa)$ *as follows.*

$$Adv^{\text{Adaptive-IND-CKA-AT}}_{\Pi,\mathcal{A}}(1^\kappa) = \Big|\Pr\big[Exp^{\text{Adaptive-IND-CKA-AT}}_{\Pi,\mathcal{A}}(1^\kappa)\big] - \frac{1}{2}\Big|$$

Here, $\mathcal{TEST}$ *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.*

An SCF-PEKS scheme is said to be adaptive-IND-CKA-AT-secure if the advantage $\mathcal{A}$ $Adv^{\text{Adaptive-IND-CKA-AT}}_{\Pi,\mathcal{A}}(1^\kappa)$ *is negligible.*

3 PREVIOUS ADAPTIVE SCF-PEKS CONSTRUCTION

In this section, we introduce the original generic construction of adaptive SCF-PEKS based on anonymous IBE, IND-stag-CCA TBE, and sUF OTS. In this 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, $C_{IBE} \notin \mathcal{M}_{TBE}$. By using TBE KEM (e.g., Section 6 of (Kiltz, 2006)), compute $(K_{TBE}, C_{TBE}) \leftarrow \mathsf{TBE.Enc}(pk, t)$, and encrypt C_{IBE} as a plaintext of the CCA secure DEM such that $C_{DEM} = \mathsf{E}_K(C_{IBE})$. Therefore, they assumed $C_{IBE} \in \mathcal{M}_{TBE}$, and $C_{DEM} = \mathsf{E}_K(C_{IBE})$ is implicitly included in C_{TBE} (i.e., C_{IBE} is obtained from the decryption of C_{TBE}). However, for the sake of clarity, we explicitly include C_{DEM} into the ciphertext.

Let $H_{tag} : \{0,1\}^* \to \mathcal{TAG}$ be a target collision resistant (TCR) hash function (Bellare and Rogaway, 1997). We set $\mathcal{M}_{Sig} = \mathcal{C}_{TBE} \times \mathcal{M}_{IBE}$, where $\mathcal{C}_{TBE}$ is a ciphertext space of the underlying TBE.

Protocol 1 (Previous Adaptive SCF-PEKS Construction (Emura et al., 2011))**.**

$\mathsf{SCF\text{-}PEKS.KeyGen_S}(1^\kappa)$**:** *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* $C_{IBE} \leftarrow \mathsf{IBE.Enc}(pk_R, \omega, R)$, $(K_{TBE}, C_{TBE}) \leftarrow \mathsf{TBE.Enc}(pk_S, t, C_{IBE})$, $C_{DEM} = \mathsf{E}_{K_{TBE}}(C_{IBE})$*, and* $\sigma \leftarrow \mathsf{Sign}(K_s, (C_{TBE}, C_{DEM}, R))$*, and output* $\lambda = (C_{TBE}, C_{DEM}, K_v, \sigma)$.

$\mathsf{SCF\text{-}PEKS.Test}(\lambda, sk_S, t_\omega)$**:** *Let* $\lambda = (C_{TBE}, C_{DEM}, K_v, \sigma)$*. Compute* $t = H_{tag}(K_v)$*, run* $K_{TBE} \leftarrow \mathsf{TBE.Dec}(sk_S, t, C_{TBE})$, $C'_{IBE} \leftarrow \mathsf{D}_{K_{TBE}}(C_{DEM})$*, 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.*

4 IBE WITH PARTITIONED CIPHERTEXT STRUCTURE (PCS-IBE)

The role of the KEM/DEM framework in the original adaptive SCF-PEKS construction is that an IBE ciphertext is regarded as a TBE plaintext to hide keyword information from an adversary who has *mk* in the Adaptive-IND-CKA-AT experiment. In this section, we define a class of IBE, called IBE with partitioned ciphertext structure (PCS-IBE) to avoid hybrid encryption[1].

Definition 13 (PCS-IBE)**.** *IBE is said to be 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.*

- *$C_{IBE,1} \in \mathcal{M}_{TBE}$.*
  - *Kiltz (Kiltz, 2006) 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. So, 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 Gentry IBE (Gentry, 2006) case as follows: 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 $C_{IBE,2} = \left(e(g,g)^s, M \cdot e(g,h)^{-s}\right)$. 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 $C'_{IBE,1} = (g' g^{-ID'})^s$.

5 OUR ADAPTIVE SCF-PEKS CONSTRUCTION BASED ON PCS-IBE

In this section, we give our adaptive SCF-PEKS construction based on PCS-IBE, IND-stag-CCA secure TBE, and sUF OTS.

5.1 Proposed Construction

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 (Abe et al., 2010).

Protocol 2 (Our 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.*

Note that non-adaptive SCF-PEKS, where no test query is considered in the IND-CKA-AT experiment, can be constructed by reducing the one-time signature part and replacing the TBE part with CPA-secure PKE as follows: Let the underlying IBE be PCS (i.e., $C_{IBE,1} \in \mathcal{M}_{PKE}$), 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 $C_{PKE} \leftarrow \mathsf{PKE.Enc}(pk_S, C_{IBE,1})$.

In the original adaptive SCF-PEKS construction (Emura et al., 2011), the DEM part $C_{DEM} = \mathsf{E}_k(C_{IBE})$ is included in the ciphertext. On the contrary, since the size of C_{DEM} is at least the same size of C_{IBE}, by excluding the DEM part, the size of ciphertext of our construction is smaller than that of the first one. Concretely, let λ_1 be a ciphertext of the original construction, and λ_2 be a ciphertext of our construction. Then, $|\lambda_1| \geq |\lambda_2| + |C_{IBE,1}|$ holds. Since the ciphertext size is bottleneck point of adaptive SCF-PEKS constructions compared to the concrete constructions, we can say that our adaptive SCF-PEKS construction is more efficient than the previous one, although is not fully generic.

5.2 Security Analysis

In this section, we show the security proofs of our construction. Note that the proofs of consistency and IND-CKA-SSK are same as these of the original ones presented in (Emura et al., 2011). So, we omit these proofs.

Theorem 1. *The SCF-PEKS scheme constructed by our method is computationally consistent if the underlying IBE scheme is IBE-IND-CPA secure.*

Theorem 2. *The SCF-PEKS scheme constructed by our method is IND-CKA-SSK secure if the underlying IBE scheme is IBE-ANO-CPA secure.*

Next, we give the proof of the following theorem.

Theorem 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. We show that there exists an algorithm $\mathcal{B}$ that breaks the IND-stag-CCA security of the underlying TBE scheme using an adversary $\mathcal{A}$ who breaks the adaptive-IND-CKA-AT security of SCF-PEKS. Let $\mathcal{C}$ be the challenger of the IND-stag-CCA experiment. $\mathcal{B}$ 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 $(\mathsf{SCF\text{-}PEKS.Enc}(pk_S, pk_R, \omega_j) := (C_{IBE,2}, C_{TBE}, K_v, \sigma), t_{\omega_j})$ 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 $C'_{IBE,1} \leftarrow \mathsf{TBE.Dec}(sk, t, C_{TBE})$.
   - 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 $R' \leftarrow \mathsf{IBE.Dec}(t_{\omega_j}, (C'_{IBE,1}, C_{IBE,2}))$.
4. If $\mathsf{Verify}(K_v, \sigma, (C_{IBE,2}, C_{TBE}, R')) = 1$, 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 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 (by using the PCS property) 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.

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 $R' \leftarrow \mathsf{IBE.Dec}(t_{\omega_j}, (C'_{IBE}, C_{IBE,2}))$.
4. If $\mathsf{Verify}(K_v, \sigma, (C_{IBE,2}, C_{TBE}, R')) = 1$, then $\mathcal{B}$ returns 1, and 0 otherwise.

In the case $t_{\omega_j} \notin \{t_{\omega^*_0}, t_{\omega^*_1}\}$:

$(C_{IBE,2}, C_{TBE}, K_v, \sigma) = (C^*_{IBE,2}, C^*_{TBE}, K^*_v, \sigma^*)$: $\mathcal{B}$ returns 0, since $(C^*_{IBE,2}, C^*_{TBE}, K^*_v, \sigma^*)$ is an SCF-PEKS ciphertext of either ω^*_0 or ω^*_1.

$(C_{IBE,2}, C_{TBE}, K_v, \sigma) \neq (C^*_{IBE,2}, C^*_{TBE}, K^*_v, \sigma^*)$: $\mathcal{B}$ 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 $\mathcal{A}$. Moreover, since we can construct an algorithm $\mathcal{B}'$ which can win the sUF game with probability at least $\Pr[\mathsf{forge}] := \Pr[\mathsf{forge}_1 \vee \mathsf{forge}_2]$, $\Pr[\mathsf{forge}_1 \vee \mathsf{forge}_2]$ is negligible. □

5.3 The GKBS Construction

Here, we instantiate an adaptive SCF-PEKS scheme based on the Gentry (PCS) anonymous IBE (Gentry, 2006), the Kiltz IND-stag-CCA-secure TBE (Kiltz, 2006), and the Bellare-Shoup sUF one-time signature (Bellare and Shoup, 2007). We call it the GKBS construction by picking up the 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 $\{0,1\}^*$.

Protocol 3. *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 (Bellare et al., 2003), the decisional augmented bilinear Diffie-Hellman exponent (decisional ABDHE) problem (Gentry, 2006), and the gap decision linear (gap DLIN) problem (Kiltz, 2006), and the collision resistance of H_{tag} and H_{sig}. Then,

Table 1: Comparison between our constructions and the Fang et al. SCF-PEKS.

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 a element of $\mathbb{G}$, $\mathbb{G}_T$, and $\mathbb{Z}_p$, respectively. More precisely, we assume that the security parameter $\kappa = 170$. So, p is a 170 bits prime, $|\mathbb{G}| = 171$ bits and $|\mathbb{G}_T| = 1020$ bits, i.e., we assume that $\mathbb{G}$ be an elliptic curve defined over finite field $\mathbb{F}_p$ and $\mathbb{G}_T$ be a multiplicative group on finite field $\mathbb{F}^{\times}_{p^k}$ 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 1 in parentheses.

	Comp. λ	Comp. Test	Length of λ	Adaptive Security
Fang et al. (Fang et al., 2009)	$2ME(\mathbb{G}) + 3ME(\mathbb{G}_T)$ $(11ME(\mathbb{G}))$	$ME(\mathbb{G}) + 2ME(\mathbb{G}_T) + 2BM$ $(7ME(\mathbb{G}) + 2BM)$	$2\|\mathbb{G}\| + 2\|\mathbb{G}_T\|$ (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

the above SCF-PEKS instantiation is adaptive secure in the standard model.

5.4 Comparison

In this section, we estimate the efficiency of the GKBS construction. Although concrete SCF-PEKS schemes have been proposed (Baek et al., 2008; Gu and Zhu, 2010; Gu et al., 2007; Rhee et al., 2009a), these schemes are proved in the random oracle model. So, we focus on SCF-PEKS schemes proposed by Fang et al. (Fang et al., 2009) and Khader (Khader, 2007), respectively, which are secure in the standard model. Khader (Khader, 2007) shows that PEKS and SCF-PEKS can be constructed by using *k*-resilient IBE (Heng and Kurosawa, 2006) (which is an IBE scheme, where an adversary can obtain at most *k* private keys of IDs). Since *k*-resilient IBE (Heng and Kurosawa, 2006) is designed by applying DDH-hard group without pairings, Khader PEKS/SCF-PEKS also enables pairing-free constructions. Unfortunately, Khader PEKS/SCF-PEKS require *k*-dependent large number of public keys and high encryption costs. So, here we compare our GKBS construction to the Fang et al. SCF-PEKS scheme (Fang et al., 2009) in Table 1. Moreover, we instantiate a non-adaptive SCF-PEKS scheme called the GBBS construction which is based on the Gentry IBE (Gentry, 2006) and linear encryption presented by Boneh, Boyen, and Shacham (Boneh et al., 2004a). We give the actual construction of this non-adaptive SCF-PEKS scheme in the Appendix. The GBBS construction achieves the same security level of the Fang et al. construction.

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). So, there is not much difference between our GKBS construction and the Fang et al. scheme in terms of efficiency, even though our construction supports adaptive security.

6 CONCLUSIONS

In this paper, we show that adaptive SCF-PEKS can be constructed without relying on hybrid encryption by using PCS-IBE. Since previously-known anonymous IBE schemes have PCS-IBE property, our adaptive SCF-PEKS construction works as long as previously-known anonymous IBE schemes are used. Since we can exclude the DEM part, our construction is efficient compared to the original one.

REFERENCES

Abe, M., Cui, Y., Imai, H., and Kiltz, E. (2010). Efficient hybrid encryption from ID-based encryption. *Des. Codes Cryptography*, 54(3):205–240.

Baek, J., Safavi-Naini, R., and Susilo, W. (2008). Public key encryption with keyword search revisited. In *ICCSA (1)*, pages 1249–1259.

Bellare, M., Namprempre, C., Pointcheval, D., and Semanko, M. (2003). The one-more-RSA-inversion problems and the security of chaum's blind signature scheme. *J. Cryptology*, 16(3):185–215.

Bellare, M. and Rogaway, P. (1997). Collision-resistant hashing: Towards making UOWHFs practical. In *CRYPTO*, pages 470–484.

Bellare, M. and Shoup, S. (2007). Two-tier signatures, strongly unforgeable signatures, and Fiat-Shamir without random oracles. In *Public Key Cryptography*, pages 201–216.

Boneh, D., Boyen, X., and Shacham, H. (2004a). Short group signatures. In *CRYPTO*, pages 41–55.

Boneh, D., Crescenzo, G. D., Ostrovsky, R., and Persiano,

G. (2004b). Public key encryption with keyword search. In *EUROCRYPT*, pages 506–522.

Boneh, D. and Franklin, M. K. (2003). Identity-based encryption from the weil pairing. *SIAM J. Comput.*, 32(3):586–615.

Boyen, X. and Waters, B. (2006). Anonymous hierarchical identity-based encryption (without random oracles). In *CRYPTO*, pages 290–307.

Byun, J. W., Rhee, H. S., Park, H.-A., and Lee, D. H. (2006). Off-line keyword guessing attacks on recent keyword search schemes over encrypted data. In *Secure Data Management*, pages 75–83.

Camenisch, J., Kohlweiss, M., Rial, A., and Sheedy, C. (2009). Blind and anonymous identity-based encryption and authorised private searches on public key encrypted data. In *Public Key Cryptography*, pages 196–214.

Caro, A. D., Iovino, V., and Persiano, G. (2010). Fully secure anonymous HIBE and secret-key anonymous IBE with short ciphertexts. In *Pairing*, pages 347–366.

Ducas, L. (2010). Anonymity from asymmetry: New constructions for anonymous HIBE. In *CT-RSA*, pages 148–164.

Emura, K., Miyaji, A., and Omote, K. (2011). Adaptive secure-channel free public-key encryption with keyword search implies timed release encryption. In *ISC*, pages 102–118.

Fang, L., Susilo, W., Ge, C., and Wang, J. (2009). A secure channel free public key encryption with keyword search scheme without random oracles. In *CANS*, pages 248–258.

Gentry, C. (2006). Practical identity-based encryption without random oracles. In *EUROCRYPT*, pages 445–464.

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

Gu, C., Zhu, Y., and Pan, H. (2007). Efficient public key encryption with keyword search schemes from pairings. In *Inscrypt*, pages 372–383.

Heng, S.-H. and Kurosawa, K. (2006). *k*-resilient identity-based encryption in the standard model. *IEICE Transactions*, 89-A(1):39–46.

Jeong, I. R., Kwon, J. O., Hong, D., and Lee, D. H. (2009). Constructing PEKS schemes secure against keyword guessing attacks is possible? *Computer Communications*, 32(2):394–396.

Khader, D. (2007). Public key encryption with keyword search based on *k*-resilient IBE. In *ICCSA (3)*, pages 1086–1095.

Kiltz, E. (2006). Chosen-ciphertext security from tag-based encryption. In *TCC*, pages 581–600.

Matsuda, T., Nakai, Y., and Matsuura, K. (2010). Efficient generic constructions of timed-release encryption with pre-open capability. In *Pairing*, pages 225–245.

Rhee, H. S., Park, J. H., Susilo, W., and Lee, D. H. (2009a). Improved searchable public key encryption with designated tester. In *ASIACCS*, pages 376–379.

Rhee, H. S., Susilo, W., and jeong Kim, H. (2009b). Secure searchable public key encryption scheme against keyword guessing attacks. In *IEICE Electronics Express Vol 6 (5)*, pages 237–243.

Seo, J. H., Kobayashi, T., Ohkubo, M., and Suzuki, K. (2009). Anonymous hierarchical identity-based encryption with constant size ciphertexts. In *Public Key Cryptography*, pages 215–234.

Shoup, V. (2000). Using hash functions as a hedge against chosen ciphertext attack. In *EUROCRYPT*, pages 275–288.

Yau, W.-C., Heng, S.-H., and Goi, B.-M. (2008). Off-line keyword guessing attacks on recent public key encryption with keyword search schemes. In *ATC*, pages 100–105.

Appendix

Protocol 4 (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 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 $R' \cdot \big(R^* \cdot e(g,h)^{-s^*}\big)$ 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.