A Generic Construction of Integrated Secure-Channel Free PEKS and PKE

Tatsuya Suzuki[1(✉)], Keita Emura[2], and Toshihiro Ohigashi[1]

[1] Tokai University, 2-3-23, Takanawa, Minato-ku, Tokyo 108-8619, Japan
t-suzuki@star.tokai-u.jp, ohigashi@tsc.u-tokai.ac.jp
[2] National Institute of Information and Communications Technology, 4-2-1, Nukui-Kitamachi, Koganei, Tokyo 184-8795, Japan
k-emura@nict.go.jp

Abstract. To provide a search functionality for encrypted data, public key encryption with keyword search (PEKS) has been widely recognized. In actual usage, a PEKS scheme should be employed with a PKE scheme since PEKS itself does not support the decryption of data. Since a naive composition of a PEKS ciphertext and a PKE ciphertext does not provide CCA security, several attempts have been made to integrate PEKS and PKE in a joint CCA manner (PEKS/PKE for short). In this paper, we further extend these works by integrating secure-channel free PEKS (SCF-PEKS) and PKE, which we call SCF-PEKS/PKE, where no secure channel is required to send trapdoors. We give a formal security definition of SCF-PEKS/PKE in a joint CCA manner, and propose a generic construction of SCF-PEKS/PKE based on anonymous identity-based encryption, tag-based encryption, and one-time signature. We also strengthen the current consistency definition according to the secure-channel free property, and show that our construction is strongly consistent if the underlying IBE provides unrestricted strong collision-freeness which is defined in this paper. Finally, we show that such an IBE scheme can be constructed by employing the Abdalla et al. transformations (TCC 2010/JoC 2018).

Keywords: PEKS · Integration of PEKS and PKE
Secure-channel free · Joint CCA security

1 Introduction

Integration of Searchable Encryption and Public Key Encryption: Public key encryption with keyword search (PEKS) [6] has been widely recognized as a cryptographic primitive providing a search functionality for encrypted data. Briefly, a trapdoor t_ω is generated with respect to a keyword ω, and one can search a ciphertext of ω by using t_ω. As defined by Abdalla et al. [1], PEKS should provide (wrong keyword) consistency and keyword privacy. Briefly, the former guarantees that for two distinct keywords ω and ω', a ciphertext of ω

C. Su and H. Kikuchi (Eds.): ISPEC 2018, LNCS 11125, pp. 69–86, 2018.
https://doi.org/10.1007/978-3-319-99807-7_5

is not searched by $t_{\omega'}$. The latter guarantees that no information of keyword is revealed from the ciphertext. Abdalla et al. [1] gave a generic construction of PEKS from anonymous identity-based encryption (IBE), e.g., [7,11,23].

In actual usage, PEKS should be employed with a PKE scheme since PEKS itself does not support the decryption of data. For example, assume that an e-mail is required to be encrypted. Then, a sender encrypts the mail header or title using a PEKS scheme, and encrypts the mail body using a PKE scheme whose public key is managed by the receiver. Then, a mail gateway can forward the encrypted e-mail by using PEKS, and the receiver can decrypt the ciphertext using their own secret key of the PKE scheme. From now on, we denote the integrated PEKS and PKE as PEKS/PKE as in [30]. As a naive composition, for a PEKS ciphertext C_{PEKS} and a PKE ciphertext C_{PKE}, a ciphertext of PEKS/PKE is described as its concatenation $C_{\mathsf{PEKS}}||C_{\mathsf{PKE}}$.

Although indistinguishability against chosen ciphertext attack (IND-CCA) is widely recognized as a standard security definition of PKE, obviously, the naive composition does not provide CCA security even if the underlying PKE scheme is CCA secure. For example, the challenge ciphertext $C^*_{\mathsf{PEKS}}||C^*_{\mathsf{PKE}}$ can be modified such as $C_{\mathsf{PEKS}}||C^*_{\mathsf{PKE}}$ where $C_{\mathsf{PEKS}} \neq C^*_{\mathsf{PEKS}}$, and one can send it to the decryption oracle. This was pointed out by Baek et al. [4] who gave a definition of joint CCA security for PEKS/PKE. Later, Zhang and Imai [30] pointed out that Baek et al.'s definition does not consider keyword privacy. They gave a formal definition of PEKS/PKE that captures both data privacy and keyword privacy, and proposed a generic construction of PEKS/PKE. Abdalla et al. [2,3] further pointed out that there is a room for improvement in the Zhang-Imai model since an adversary is not allowed to access the test oracle in the model. Chen et al. [12] further considered the trapdoor oracle, and proposed a generic construction of PEKS/PKE from (hierarchical) IBE schemes. As concrete constructions, Buccafurri et al. [9] and Saraswat and Sahu [27] proposed PEKS/PKE schemes from (asymmetric) pairings.[1]

Secure-Channel Free PEKS: In typical usage of PEKS, a receiver generates a trapdoor, and sends it to a server (e.g., mail gateway). Then, since anyone can run the test algorithm when they obtain a trapdoor, the trapdoor must be sent to the server via a secure channel. To remove the secure channel, secure-channel free PEKS (SCF-PEKS), which is also called designated tester PEKS, has been proposed [13–15,20,26,28]. Unlike the case of employing SSL/TLS in a naive way, only the designated server can run the test algorithm even if trapdoors are exposed. In SCF-PEKS, the server also has a public key and a secret key, and a keyword is encrypted by using the server pubic key in addition to the receiver pubic key. The test algorithm is run by using the server secret key in addition to a trapdoor.

[1] As a similar primitive, decryptable searchable encryption has been proposed [18,21] where keywords can be recovered from ciphertexts via the decryption procedure. One main difference from PEKS/PKE is that no plaintext space is defined.

Our Contribution: As in PEKS, all PEKS/PKE have assumed that trapdoors are sent to the server via a secure channel. In this paper, to remove this limitation we propose PEKS/PKE supporting secure-channel free property, which we call SCF-PEKS/PKE.

First we give a formal security definition of SCF-PEKS/PKE in a joint CCA manner. Basically, we extend the security definition of SCF-PEKS given by Fang et al. [16].[2] We strengthen their consistency definition as follows. First, an adversary is allowed to access the trapdoor oracle in our model. Owing to the secure-channel free property, this setting is natural since trapdoors are sent via a public channel. Moreover, we give the server secret key to the adversary to guarantee that the server has no way of producing inconsistent ciphertexts. We call this weak consistency. We further strengthen the consistency, which we call strong consistency, where (1) an adversary can obtain trapdoors even for challenge keywords, and (2) an adversary is allowed to produce the challenge ciphertext. The first extension is the same as that of unrestricted strong robustness [17], and the second extension is the same as those of strong robustness [2,3] and strong collision-freeness [25]. For keyword privacy, as in Fang et al., we consider two situations where either an adversary is modeled as the server (then the server secret key is given to the adversary), or an adversary is modeled as a receiver (then the receiver secret key is given to the adversary). In the former, the adversary is allowed to access the trapdoor oracle and the test oracle, and in the latter, the adversary is allowed to access the test oracle. We additionally consider the decryption oracle to integrate SCF-PEKS and PKE in our joint CCA security. We further define data privacy. To guarantee that the server does not obtain information of data via the test procedure, we give the server secret key to the adversary. Moreover, the adversary is allowed to access the decryption oracle.

Second, we propose a generic construction of SCF-PEKS/PKE with weak consistency from anonymous IBE, tag-based encryption (TBE) [24], and a one-time signature (OTS). We also show that our construction is strongly consistent if the underlying anonymous IBE provides unrestricted strong collision-freeness which is implied by unrestricted strong robustness [17]. We will show how to construct these ingredients in Sect. 5. Our construction can be seen as an extension of a generic construction of SCF-PEKS from the same ingredients as above, proposed by Emura et al. [14], by considering an observation given by Abdalla et al. [2,3]. Namely, Abdalla et al. mentioned that if PEKS and PKE support tags, then these can be combined via the Canetti-Halevi-Katz (CHK) transformation [10], leading to a PEKS/PKE scheme secure in the joint CCA manner. That is, by introducing an OTS scheme, a verification key is regarded as a tag of both ciphertexts, and a signature is produced on them. We point out that the Emura et al. construction yields a "tag-based" SCF-PEKS scheme. By introducing a TBE scheme as the underlying PKE scheme supporting tags, we can construct SCF-PEKS/PKE secure in the joint CCA manner. We further modify the construction to protect against re-encryption attacks (See Sect. 4: High-level

[2] Remark that we do not consider security against keyword guessing attacks which is considered by Fang et al. [16], and leave it as a future work of this paper.

Description of Our Construction for details) by preparing an IBE plaintext to be correlated to a verification key.

2 Preliminaries

We denote that $x \xleftarrow{\$} S$ when 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.

First, we introduce the definition of TBE [24] as follows. Let $\mathcal{TAG}$ and $\mathcal{M}_{\mathsf{TBE}}$ be a tag space of TBE and a plaintext space of TBE, respectively.

Definition 1 (Syntax of TBE)**.** *A TBE scheme* TBE *consists of the following three algorithms,* TBE.KeyGen*,* TBE.Enc *and* TBE.Dec*:*

$\mathsf{TBE.KeyGen}(1^\kappa)$: *This key generation algorithm takes as an input the security parameter* $\kappa \in \mathrm{N}$*, and return a public key* pk_{TBE} *and a secret key* sk_{TBE}*.*

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

$\mathsf{TBE.Dec}(sk_{\mathsf{TBE}}, t, C_{\mathsf{TBE}})$: *This decryption algorithm takes as inputs* sk_{TBE}*,* t*, and* C_{TBE}*, and returns a message* M *or a reject symbol* $\perp$*.*

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

Next, we define selective-tag weakly secure against chosen ciphertext attack (IND-stag-CCA) as follows.

Definition 2 (IND-stag-CCA)**.** *For any probabilistic polynomial-time (PPT) adversary* $\mathcal{A}$ *and the security parameter* $\kappa \in \mathrm{N}$*, we define the experiment* $\mathrm{Exp}^{\text{IND-stag-CCA}}_{\mathsf{TBE},\mathcal{A}}(\kappa)$ *as follows.*

$\mathrm{Exp}^{\text{IND-stag-CCA}}_{\mathsf{TBE},\mathcal{A}}(\kappa)$:

$(t^*, State) \leftarrow \mathcal{A}(1^\kappa)$; $(pk_{\mathsf{TBE}}, sk_{\mathsf{TBE}}) \leftarrow \mathsf{TBE.KeyGen}(1^\kappa)$

$(M_0^*, M_1^*, State) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{TBE.DEC}}}(\mathsf{find}, pk_{\mathsf{TBE}})$; $\mu \xleftarrow{\$} \{0,1\}$

$C^*_{\mathsf{TBE}} \leftarrow \mathsf{TBE.Enc}(pk_{\mathsf{TBE}}, t^*, M^*_\mu)$; $\mu' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{TBE.DEC}}}(\mathsf{guess}, C^*_{\mathsf{TBE}}, State)$

If $\mu = \mu'$ *then output* 1*, and* 0 *otherwise*

– $\mathcal{O}_{\mathsf{TBE.DEC}}$*: This decryption oracle takes as input a tag and a ciphertext* $(t, C_{\mathsf{TBE}}) \neq (t^*, C^*_{\mathsf{TBE}})$ *and returns the result of* $\mathsf{TBE.Dec}(sk_{\mathsf{TBE}}, t, C_{\mathsf{TBE}})$*.*

We say that TBE *is IND-stag-CCA secure if the advantage*

$$\mathrm{Adv}^{\text{IND-stag-CCA}}_{\mathsf{TBE},\mathcal{A}}(\kappa) := \mid \Pr[\mathrm{Exp}^{\text{IND-stag-CCA}}_{\mathsf{TBE},\mathcal{A}}(\kappa) = 1] - 1/2 \mid$$

is negligible for any PPT adversary $\mathcal{A}$*.*

Next, we introduce definition of anonymous IBE with CCA security [19] as follows. Let $\mathcal{ID}$ and $\mathcal{M}_{\mathsf{IBE}}$ be an identity space and a plaintext space of IBE, respectively.

Definition 3 (Syntax of IBE)**.** *An IBE scheme* IBE *consists of the following four algorithms,* IBE.Setup*,* IBE.Extract*,* IBE.Enc *and* IBE.Dec*:*

IBE.Setup(1^κ): *This setup algorithm takes as an input the security parameter* $\kappa \in \mathrm{N}$*, and return a public key params and a master key mk.*
IBE.Extract$(params, mk, ID)$: *This extract algorithm takes as input an identity* $ID \in \mathcal{ID}$ *and mk, and returns a secret key* sk_{ID} *corresponding to* ID*.*
IBE.Enc$(params, ID, M)$: *This encryption algorithm takes as input params,* $ID \in \mathcal{ID}$*, a message* $M \in \mathcal{M}_{\mathsf{IBE}}$*, and returns a ciphertext* C_{IBE}*.*
IBE.Dec$(params, sk_{ID}, C_{\mathsf{IBE}})$: *This decryption algorithm takes as inputs* sk_{ID} *and* C_{IBE}*, and returns a message* M *or a reject symbol* $\perp$*.*

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

Next, we define indistinguishability against chosen ciphertext attack (IBE-IND-CCA) as follows.

Definition 4 (IBE-IND-CCA)**.** *For any PPT adversary* $\mathcal{A}$ *and the security parameter* $\kappa \in \mathrm{N}$*, we define the experiment* $\mathrm{Exp}^{\text{IBE-IND-CCA}}_{\mathsf{IBE},\mathcal{A}}(\kappa)$ *as follows.*

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

- $\mathcal{O}_{\mathsf{IBE.DEC}}$*: This decryption oracle takes as input* $(ID, C_{\mathsf{IBE}}) \neq (ID^*, C^*_{\mathsf{IBE}})$ *and returns the result of* $\mathsf{IBE.Dec}(params, sk_{ID}, C_{\mathsf{IBE}})$ *where* $sk_{ID} \leftarrow \mathsf{IBE.Extract}(params, mk, ID)$*.*
- $\mathcal{O}_{\mathsf{IBE.EXTRACT}}$*: This extract oracle takes as input an identity* $ID \neq ID^*$ *and returns the corresponding secret key* $sk_{ID} \leftarrow \mathsf{IBE.Extract}$*(params, mk, ID).*

We say that IBE *is IBE-IND-CCA secure if the advantage*

$$\mathrm{Adv}^{\text{IBE-IND-CCA}}_{\mathsf{IBE},\mathcal{A}}(\kappa) := \mid \Pr[\mathrm{Exp}^{\text{IBE-IND-CCA}}_{\mathsf{IBE},\mathcal{A}}(\kappa) = 1] - 1/2 \mid$$

is negligible for any PPT adversary.

Next, we define anonymity against chosen-ciphertext attack (IBE-ANO-CCA).

Definition 5 (IBE-ANO-CCA)**.** *For any PPT adversary $\mathcal{A}$ and the security parameter $\kappa \in \mathrm{N}$, we define the experiment* $\mathrm{Exp}^{\text{IBE-ANO-CCA}}_{\mathsf{IBE},\mathcal{A}}(\kappa)$ *as follows.*

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

- $\mathcal{O}_{\mathsf{IBE.DEC}}$*: This decryption oracle takes as input* $(ID, C_{\mathsf{IBE}}) \notin \{(ID_0^*, C^*_{\mathsf{IBE}}), (ID_1^*, C^*_{\mathsf{IBE}})\}$ *and returns the result of* $\mathsf{IBE.Dec}(params, sk_{ID}, C_{\mathsf{IBE}})$ *where* $sk_{ID} \leftarrow \mathsf{IBE.Extract}(params, mk, ID)$.
- $\mathcal{O}_{\mathsf{IBE.EXTRACT}}$*: This extract oracle takes as input* $ID \notin \{ID_0^*, ID_1^*\}$ *and returns the corresponding secret key* $sk_{ID} \leftarrow \mathsf{IBE.Extract}(params, mk, ID)$.

We say that IBE *is IBE-ANO-CCA secure if the advantage*

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

is negligible for any PPT adversary.

Next, we define unrestricted strong collision-freeness where strong means that an adversary is allowed to produce the challenge ciphertext C^*_{IBE}. This is an extension of strong collision-freeness [25]. Informally, strong collision-freeness guarantees that no adversary can produce a ciphertext whose decryption result for two decryption keys are the same, i.e., $M_0^* = M_1^*$. In addition, in our unrestricted strong collision-freeness definition, the trapdoor oracle has no restriction as in unrestricted strong robustness [17]. Informally, unrestricted strong robustness guarantees that no adversary can produce a ciphertext whose decryption result for two decryption keys are both non-$\perp$. Since the condition $M_0^* = M_1^*$ is not required, our unrestricted strong collision-freeness is an intermediate notion where it is weaker than unrestricted strong robustness and is stronger than strong collision-freeness. How to construct an IBE scheme with unrestricted strong collision-freeness is explained in Sect. 5.

Definition 6 (Unrestricted Strong Collision-Freeness)**.** *For any PPT adversary $\mathcal{A}$ and the security parameter $\kappa \in \mathrm{N}$, we define the experiment* $\mathrm{Exp}^{\text{IBE-usCF}}_{\mathsf{IBE},\mathcal{A}}(\kappa)$ *as follows.*

$\mathrm{Exp}^{\text{IBE-usCF}}_{\mathsf{IBE},\mathcal{A}}(\kappa)$:
$(params, mk) \leftarrow \mathsf{IBE.Setup}(1^\kappa)$
$(C^*_{\mathsf{IBE}}, ID_0^*, ID_1^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{IBE.EXTRACT}}}(\mathsf{find}, params)$
$sk_{ID_0^*} \leftarrow \mathsf{IBE.Extract}(params, mk, ID_0^*);\ sk_{ID_1^*} \leftarrow \mathsf{IBE.Extract}(params, mk, ID_1^*)$
$M_0^* \leftarrow \mathsf{IBE.Dec}(params, sk_{ID_0^*}, C^*_{\mathsf{IBE}});\ M_1^* \leftarrow \mathsf{IBE.Dec}(params, sk_{ID_1^*}, C^*_{\mathsf{IBE}})$
If $M_0^* \neq \perp \wedge M_1^* \neq \perp \wedge M_0^* = M_1^*$ *then output* 1, *and* 0 *otherwise*

– $\mathcal{O}_{\mathsf{IBE.EXTRACT}}$*: This extract oracle takes as input* ID *with no restriction, and returns the corresponding secret key* $sk_{ID} \leftarrow \mathsf{IBE.Extract}(params, mk, ID)$.

We say that IBE *is unrestricted strongly collision-free if the advantage*

$$\mathrm{Adv}^{\text{IBE-usCF}}_{\mathsf{IBE},\mathcal{A}}(\kappa) := \Pr[\mathrm{Exp}^{\text{IBE-usCF}}_{\mathsf{IBE},\mathcal{A}}(\kappa) = 1]$$

is negligible for any PPT adversary $\mathcal{A}$.

Next, we introduce OTS [5] as follows. Let $\mathcal{M}_{\mathsf{Sig}}$ be a message space.

Definition 7 (Syntax of OTS)**.** *A OTS scheme* OTS *consists of the following three algorithms,* Sig.KeyGen*,* Sign *and* Verify*:*

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

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

$\mathsf{Verify}(K_v, M, \sigma)$*: This verification algorithm takes as input* K_v*,* M*, and* σ*, and returns 1 (valid) or 0 (invalid).*

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

Next, we define strong existential unforgeability against chosen message attack (sEUF-CMA) of OTS as follows.

Definition 8 (one-time sEUF-CMA)**.** *For any PPT adversary* $\mathcal{A}$ *and the security parameter* $\kappa \in \mathrm{N}$*, we define the experiment* $\mathrm{Exp}^{\text{one-times EUF-CMA}}_{\mathsf{OTS},\mathcal{A}}(\kappa)$ *as follows.*

$\mathrm{Exp}^{\text{one-time sEUF-CMA}}_{\mathsf{OTS},\mathcal{A}}(\kappa)$:

$(K_s, K_v) \leftarrow \mathsf{Sig.KeyGen}(1^\kappa)$; $(M, State) \leftarrow \mathcal{A}(K_v)$; $M \in \mathcal{M}_{\mathsf{Sig}}$

$\sigma \leftarrow \mathsf{Sign}(K_s, M)$; $(M^*, \sigma^*) \leftarrow \mathcal{A}(\sigma, State)$

If $\mathsf{Verify}(K_v, M^*, \sigma^*) = 1$ *and* $(M^*, \sigma^*) \neq (M, \sigma)$ *then output* 1*, and* 0 *otherwise*

We say that OTS *is one-time sEUF-CMA secure if the advantage*

$$\mathrm{Adv}^{\text{one-time sEUF-CMA}}_{\mathsf{OTS},\mathcal{A}}(\kappa) := \Pr[\mathrm{Exp}^{\text{one-time sEUF-CMA}}_{\mathsf{OTS},\mathcal{A}}(\kappa) = 1]$$

is negligible for any PPT adversary.

3 Definitions of SCF-PEKS/PKE

In this section, we define SCF-PEKS/PKE. As in SCF-PEKS, the server and a receiver manage keys separately. A keyword ω and a plaintext M are encrypted by the server public key, pk_{S}, and the receiver public key, pk_{R}. Although a secret key of the receiver, sk_{R}, plays the role of generating trapdoors in SCF-PEKS, we additionally require that sk_{R} plays a role of decrypting a ciphertext. To search for an encrypted keyword, the test algorithm requires both the server secret key, sk_{S}, and the corresponding trapdoor. Let $\mathcal{K}$ be the keyword space and $\mathcal{M}$ be the message space.

Definition 9 (Syntax of SCF-PEKS/PKE)**.** *A SCF-PEKS/PKE scheme* SCF-PEKS/PKE *consists of the following six algorithms,* SCF-PEKS/PKE.KeyGen$_{\mathsf{S}}$, SCF-PEKS/PKE.KeyGen$_{\mathsf{R}}$, SCF-PEKS/PKE.Trapdoor, SCF-PEKS/PKE.Enc, SCF-PEKS/PKE.Dec *and* SCF-PEKS/PKE.Test*:*

SCF-PEKS/PKE.KeyGen$_{\mathsf{S}}(1^\kappa)$: *This server key generation algorithm takes as input the security parameter* 1^κ *(*$\kappa \in \mathrm{N}$*), and returns a server public key* pk_{S} *and a server secret key* sk_{S}*.*

SCF-PEKS/PKE.KeyGen$_{\mathsf{R}}(1^\kappa)$: *This receiver key generation algorithm takes as input the security parameter* 1^κ *(*$\kappa \in \mathrm{N}$*), and returns a receiver public key* pk_{R} *and a receiver secret key* sk_{R}*.*

SCF-PEKS/PKE.Trapdoor$(pk_{\mathrm{R}}, sk_{\mathrm{R}}, \omega)$: *This trapdoor generation algorithm takes as input* pk_{R}*,* sk_{R}*, and a keyword* $\omega \in \mathcal{K}$*, and returns a trapdoor* t_ω *corresponding to keyword* ω*.*

SCF-PEKS/PKE.Enc$(pk_{\mathrm{S}}, pk_{\mathrm{R}}, \omega, M)$: *This encryption algorithm takes as input* pk_{R}*,* pk_{S}*,* ω*, and a message* $M \in \mathcal{M}$*, and returns a ciphertext* λ*.*

SCF-PEKS/PKE.Dec$(pk_{\mathrm{R}}, sk_{\mathrm{R}}, \lambda)$: *This decryption algorithm takes as input* pk_{R}*,* sk_{R}*, and* λ*, and returns a message* M *or a reject symbol* $\perp$*.*

SCF-PEKS/PKE.Test$(pk_{\mathrm{S}}, sk_{\mathrm{S}}, pk_{\mathrm{R}}, t_\omega, \lambda)$: *This test algorithm takes as input* pk_{S}*,* sk_{S}*,* pk_{R}*,* t_ω*, and* λ*, 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 $(pk_{\mathrm{S}}, sk_{\mathrm{S}}) \leftarrow$ SCF-PEKS/PKE.KeyGen$_{\mathsf{S}}(1^\kappa)$, all $(pk_{\mathrm{R}}, sk_{\mathrm{R}}) \leftarrow$ SCF-PEKS/PKE.KeyGen$_{\mathsf{R}}(1^\kappa)$, all $\omega \in \mathcal{K}$ and all $M \in \mathcal{M}$, let $\lambda \leftarrow$ SCF-PEKS/PKE.Enc$(pk_{\mathrm{S}}, pk_{\mathrm{R}}, \omega, M)$ and $t_\omega \leftarrow$ SCF-PEKS/PKE.Trapdoor$(pk_{\mathrm{R}}, sk_{\mathrm{R}}, \omega)$. Then

$$\text{SCF-PEKS/PKE.Test}(pk_{\mathrm{S}}, sk_{\mathrm{S}}, pk_{\mathrm{R}}, t_\omega, \lambda) = 1 \text{ and}$$
$$\text{SCF-PEKS/PKE.Dec}(pk_{\mathrm{R}}, sk_{\mathrm{R}}, \lambda) = M \text{ holds.}$$

Next, we define consistency. Basically, consistency guarantees that for two trapdoors t_{ω^*} and $t_{\hat{\omega}^*}$ where $\omega^* \neq \hat{\omega}^*$, a ciphertext of ω^* is not searched by $t_{\hat{\omega}^*}$. We give two definitions. The former case, which we call weak consistency, is essentially the same as that of Chen et al. [12] where the ciphertext λ^* is honestly generated. Due to the secure-channel free setting, we additionally consider the trapdoor oracle, and give sk_{S} to the adversary.

Definition 10 (Weak Consistency)**.** *For any PPT adversary* $\mathcal{A}$ *and the security parameter* $\kappa \in \mathrm{N}$*, we define the experiment* $\mathrm{Exp}^{\text{WEAK-CONSIST}}_{\text{SCF-PEKS/PKE},\mathcal{A}}(\kappa)$ *as follows.*

$\mathrm{Exp}^{\text{WEAK-CONSIST}}_{\text{SCF-PEKS/PKE},\mathcal{A}}(\kappa)$:
$(pk_{\mathrm{S}}, sk_{\mathrm{S}}) \leftarrow$ SCF-PEKS/PKE.KeyGen$_{\mathsf{S}}(1^\kappa)$
$(pk_{\mathrm{R}}, sk_{\mathrm{R}}) \leftarrow$ SCF-PEKS/PKE.KeyGen$_{\mathsf{R}}(1^\kappa)$
$(M^*, \omega^*, \hat{\omega}^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{SCF-PEKS/PKE.TRAP}}}(pk_{\mathrm{S}}, sk_{\mathrm{S}}, pk_{\mathrm{R}})$

$M^* \in \mathcal{M};\ \omega^*,\ \hat{\omega}^* \in \mathcal{K};\ \omega^* \neq \hat{\omega}^*$
$\lambda^* \leftarrow \mathsf{SCF\text{-}PEKS/PKE.Enc}(pk_{\mathrm{S}}, pk_{\mathrm{R}}, \omega^*, M^*)$
$t_{\hat{\omega}^*} \leftarrow \mathsf{SCF\text{-}PEKS/PKE.Trapdoor}(pk_{\mathrm{R}}, sk_{\mathrm{R}}, \hat{\omega}^*)$
If $\mathsf{SCF\text{-}PEKS/PKE.Test}(pk_{\mathrm{S}}, sk_{\mathrm{S}}, pk_{\mathrm{R}}, t_{\hat{\omega}^*}, \lambda^*) = 1$ *then output* 1, *and* 0 *otherwise*

– $\mathcal{O}_{\mathsf{SCF\text{-}PEKS/PKE.TRAP}}$*: This trapdoor oracle takes as input* ω *where* $\omega \notin \{\omega^*, \hat{\omega}^*\}$ *and returns* $t_\omega \leftarrow \mathsf{SCF\text{-}PEKS/PKE.Trapdoor}(pk_{\mathrm{R}}, sk_{\mathrm{R}}, \omega)$.

We say that SCF-PEKS/PKE *is weakly consistent if the advantage*

$$\mathrm{Adv}^{\text{WEAK-CONSIST}}_{\mathsf{SCF\text{-}PEKS/PKE},\mathcal{A}}(\kappa) := \Pr[\mathrm{Exp}^{\text{WEAK-CONSIST}}_{\mathsf{SCF\text{-}PEKS/PKE},\mathcal{A}}(\kappa) = 1]$$

is negligible for any PPT adversary $\mathcal{A}$.

Next, we strengthen weak consistency, which we call strong consistency. Here, an adversary is allowed to produce the ciphertext λ^*. This situation is the same as those of strong robustness [2,3] and strong collision-freeness [25]. Note that, an adversary is not allowed to obtain decryption keys for challenge identities in these models. In our model, the trapdoor oracle has no restriction, i.e., an adversary can obtain trapdoors of challenge keywords. This situation is the same as that of unrestricted strong robustness [17]. Our strong consistency captures the following situation. Owing to the secure-channel free property, an adversary can observe trapdoors. Let the adversary obtain t_{ω^*} and $t_{\hat{\omega}^*}$. Moreover, assume that the adversary knows keywords ω^* and $\hat{\omega}^*$ associated with t_{ω^*} and $t_{\hat{\omega}^*}$, respectively.[3] Then, the adversary may produce a ciphertext where the test algorithm decides that the ciphertext is associated with both ω^* and $\hat{\omega}^*$. Strong consistency prevents this attack.

Definition 11 (Strong Consistency)**.** *For any PPT adversary* $\mathcal{A}$ *and the security parameter* $\kappa \in \mathrm{N}$, *we define the experiment* $\mathrm{Exp}^{\text{STRONG-CONSIST}}_{\mathsf{SCF\text{-}PEKS/PKE},\mathcal{A}}(\kappa)$ *as follows.*

$\mathrm{Exp}^{\text{STRONG-CONSIST}}_{\mathsf{SCF\text{-}PEKS/PKE},\mathcal{A}}(\kappa)$:
$(pk_{\mathrm{S}}, sk_{\mathrm{S}}) \leftarrow \mathsf{SCF\text{-}PEKS/PKE.KeyGen_S}(1^\kappa)$
$(pk_{\mathrm{R}}, sk_{\mathrm{R}}) \leftarrow \mathsf{SCF\text{-}PEKS/PKE.KeyGen_R}(1^\kappa)$
$(\lambda^*, \omega^*, \hat{\omega}^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{SCF\text{-}PEKS/PKE.TRAP}}}(pk_{\mathrm{S}}, sk_{\mathrm{S}}, pk_{\mathrm{R}});\ \omega^*, \hat{\omega}^* \in \mathcal{K};\ \omega^* \neq \hat{\omega}^*$
$t_{\omega^*} \leftarrow \mathsf{SCF\text{-}PEKS/PKE.Trapdoor}(pk_{\mathrm{R}}, sk_{\mathrm{R}}, \omega^*)$
$t_{\hat{\omega}^*} \leftarrow \mathsf{SCF\text{-}PEKS/PKE.Trapdoor}(pk_{\mathrm{R}}, sk_{\mathrm{R}}, \hat{\omega}^*)$
If $\mathsf{SCF\text{-}PEKS/PKE.Test}(pk_{\mathrm{S}}, sk_{\mathrm{S}}, pk_{\mathrm{R}}, t_{\omega^*}, \lambda^*) = 1$ *and*
$\mathsf{SCF\text{-}PEKS/PKE.Test}(pk_{\mathrm{S}}, sk_{\mathrm{S}}, pk_{\mathrm{R}}, t_{\hat{\omega}^*}, \lambda^*) = 1$
then output 1, *and* 0 *otherwise*

– $\mathcal{O}_{\mathsf{SCF\text{-}PEKS/PKE.TRAP}}$*: This trapdoor oracle takes as input* ω *with no restriction, and returns* $t_\omega \leftarrow \mathsf{SCF\text{-}PEKS/PKE.Trapdoor}(pk_{\mathrm{R}}, sk_{\mathrm{R}}, \omega)$.

[3] This assumption is also natural since we do not consider keyword guessing attacks [16].

We say that SCF-PEKS/PKE *is strongly consistent if the advantage*

$$\mathrm{Adv}^{\text{STRONG-CONSIST}}_{\text{SCF-PEKS/PKE},\mathcal{A}}(\kappa) := \Pr[\mathrm{Exp}^{\text{STRONG-CONSIST}}_{\text{SCF-PEKS/PKE},\mathcal{A}}(\kappa) = 1]$$

is negligible for any PPT adversary $\mathcal{A}$*.*

Next, we define two security notions for keyword privacy, indistinguishability of keywords against chosen keyword attack with the server secret key (IND-CKA-SSK) and indistinguishability of keywords against chosen keyword attack with all trapdoors (IND-CKA-AT). In the IND-CKA-SSK definition, an adversary $\mathcal{A}$ is modeled as the server, and thus sk_{S} is given to $\mathcal{A}$. If $\mathcal{A}$ obtains trapdoors, then $\mathcal{A}$ can run the test algorithm by myself. Thus, trapdoors of challenge keywords (ω_0^*, ω_1^*) are not given to $\mathcal{A}$. Instead, $\mathcal{A}$ is allowed to access the test oracle for $(\lambda, \omega) \notin \{(\lambda^*, \omega_0^*), (\lambda^*, \omega_1^*)\}$. To guarantee that no information of keyword is revealed via the decryption procedure, $\mathcal{A}$ is allowed to access the decryption oracle with no restriction.

Definition 12 (IND-CKA-SSK)**.** *For any PPT adversary* $\mathcal{A}$ *and the security parameter* $\kappa \in \mathrm{N}$*, we define the experiment* $\mathrm{Exp}^{\text{IND-CKA-SSK}}_{\text{SCF-PEKS/PKE},\mathcal{A}}(\kappa)$ *as follows.*

$\mathrm{Exp}^{\text{IND-CKA-SSK}}_{\text{SCF-PEKS/PKE},\mathcal{A}}(\kappa)$:
$(pk_{\mathrm{S}}, sk_{\mathrm{S}}) \leftarrow \mathsf{SCF\text{-}PEKS/PKE.KeyGen}_{\mathrm{S}}(1^\kappa)$
$(pk_{\mathrm{R}}, sk_{\mathrm{R}}) \leftarrow \mathsf{SCF\text{-}PEKS/PKE.KeyGen}_{\mathrm{R}}(1^\kappa)$
$(\omega_0^*, \omega_1^*, M^*, \mathit{State})$
$\leftarrow \mathcal{A}^{\mathcal{O}_{\text{SCF-PEKS/PKE.DEC}}, \mathcal{O}_{\text{SCF-PEKS/PKE.TRAP}}, \mathcal{O}_{\text{SCF-PEKS/PKE.TEST}}}(\mathsf{find}, pk_{\mathrm{S}}, sk_{\mathrm{S}}, pk_{\mathrm{R}})$
$\mu \xleftarrow{\$} \{0,1\};\ \lambda^* \leftarrow \mathsf{SCF\text{-}PEKS/PKE.Enc}(pk_{\mathrm{S}}, pk_{\mathrm{R}}, \omega_\mu^*, M^*)$
$\mu' \leftarrow \mathcal{A}^{\mathcal{O}_{\text{SCF-PEKS/PKE.DEC}}, \mathcal{O}_{\text{SCF-PEKS/PKE.TRAP}}, \mathcal{O}_{\text{SCF-PEKS/PKE.TEST}}}(\mathsf{guess}, \lambda^*, \mathit{State})$
If $\mu = \mu'$ *then output* 1*, and* 0 *otherwise*

- $\mathcal{O}_{\text{SCF-PEKS/PKE.DEC}}$*: This decryption oracle takes as input* λ *with no restriction, and returns the result of* $\mathsf{SCF\text{-}PEKS/PKE.Dec}(pk_{\mathrm{R}}, sk_{\mathrm{R}}, \lambda)$*. Remark that* λ^* *is also allowed to input.*
- $\mathcal{O}_{\text{SCF-PEKS/PKE.TRAP}}$*: This trapdoor oracle takes as input* ω *where* $\omega \notin \{\omega_0^*, \omega_1^*\}$ *and returns* $t_\omega \leftarrow \mathsf{SCF\text{-}PEKS/PKE.Trapdoor}(pk_{\mathrm{R}}, sk_{\mathrm{R}}, \omega)$*.*
- $\mathcal{O}_{\text{SCF-PEKS/PKE.TEST}}$*: This test oracle takes as input* (λ, ω) *where* $(\lambda, \omega) \notin \{(\lambda^*, \omega_0^*), (\lambda^*, \omega_1^*)\}$*, compute* $t_\omega \leftarrow \mathsf{SCF\text{-}PEKS/PKE.Trapdoor}(pk_{\mathrm{R}}, sk_{\mathrm{R}}, \omega)$*, and returns result of* $\mathsf{SCF\text{-}PEKS/PKE.Test}(pk_{\mathrm{S}}, sk_{\mathrm{S}}, pk_{\mathrm{R}}, t_\omega, \lambda)$*.*

We say that a SCF-PEKS/PKE scheme SCF-PEKS/PKE *is IND-CKA-SSK secure if the advantage*

$$\mathrm{Adv}^{\text{IND-CKA-SSK}}_{\text{SCF-PEKS/PKE},\mathcal{A}}(\kappa) :=\mid \Pr[\mathrm{Exp}^{\text{IND-CKA-SSK}}_{\text{SCF-PEKS/PKE},\mathcal{A}}(\kappa) = 1] - 1/2 \mid$$

is negligible for any PPT adversary $\mathcal{A}$*.*

Next, we define IND-CKA-AT. In the IND-CKA-AT definition, an adversary $\mathcal{A}$ is modeled as a receiver. Thus, sk_{R} is given to $\mathcal{A}$. Then, $\mathcal{A}$ can generate trapdoors for all keywords. Since $\mathcal{A}$ does not have sk_{S}, $\mathcal{A}$ is not allowed to run the test algorithm. Thus, $\mathcal{A}$ is allowed to access the test oracle for $(\lambda, \omega) \notin \{(\lambda^*, \omega_0^*), (\lambda^*, \omega_1^*)\}$. To guarantee that no information of keyword is revealed via the decryption procedure, $\mathcal{A}$ is allowed to access the decryption oracle with no restriction.

Definition 13 (IND-CKA-AT)**.** *For any PPT adversary $\mathcal{A}$ and the security parameter $\kappa \in \mathrm{N}$, we define the experiment* $\mathrm{Exp}^{\text{IND-CKA-AT}}_{\mathsf{SCF\text{-}PEKS/PKE},\mathcal{A}}(\kappa)$ *as follows.*

$$
\begin{aligned}
&\mathrm{Exp}^{\text{IND-CKA-AT}}_{\mathsf{SCF\text{-}PEKS/PKE},\mathcal{A}}(\kappa):\\
&\quad (pk_{\mathrm{S}}, sk_{\mathrm{S}}) \leftarrow \mathsf{SCF\text{-}PEKS/PKE.KeyGen}_{\mathrm{S}}(1^{\kappa})\\
&\quad (pk_{\mathrm{R}}, sk_{\mathrm{R}}) \leftarrow \mathsf{SCF\text{-}PEKS/PKE.KeyGen}_{\mathrm{R}}(1^{\kappa})\\
&\quad (\omega_0^*, \omega_1^*, M^*, \mathit{State})\\
&\quad \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{SCF\text{-}PEKS/PKE.DEC}}, \mathcal{O}_{\mathsf{SCF\text{-}PEKS/PKE.TEST}}}(\mathsf{find}, pk_{\mathrm{S}}, pk_{\mathrm{R}}, sk_{\mathrm{R}})\\
&\quad \mu \xleftarrow{\$} \{0,1\};\ \lambda^* \leftarrow \mathsf{SCF\text{-}PEKS/PKE.Enc}(pk_{\mathrm{S}}, pk_{\mathrm{R}}, \omega_{\mu}^*, M^*)\\
&\quad \mu' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{SCF\text{-}PEKS/PKE.DEC}}, \mathcal{O}_{\mathsf{SCF\text{-}PEKS/PKE.TEST}}}(\mathsf{guess}, \lambda^*, \mathit{State})\\
&\quad \mathit{If}\ \mu = \mu'\ \mathit{then\ output}\ 1,\ \mathit{and}\ 0\ \mathit{otherwise}
\end{aligned}
$$

- $\mathcal{O}_{\mathsf{SCF\text{-}PEKS/PKE.DEC}}$*: This decryption oracle takes as input λ with no restriction, and returns the result of* $\mathsf{SCF\text{-}PEKS/PKE.Dec}(pk_{\mathrm{R}}, sk_{\mathrm{R}}, \lambda)$*. Remark that λ^* is also allowed to input.*
- $\mathcal{O}_{\mathsf{SCF\text{-}PEKS/PKE.TEST}}$*: This test oracle takes as input $(\lambda, \omega) \notin \{(\lambda^*, \omega_0^*), (\lambda^*, \omega_1^*)\}$, computes $t_{\omega} \leftarrow \mathsf{SCF\text{-}PEKS/PKE.Trapdoor}(pk_{\mathrm{R}}, sk_{\mathrm{R}}, \omega)$, and returns result of* $\mathsf{SCF\text{-}PEKS/PKE.Test}(pk_{\mathrm{S}}, sk_{\mathrm{S}}, pk_{\mathrm{R}}, t_{\omega}, \lambda)$*.*

We say that a SCF-PEKS/PKE scheme SCF-PEKS/PKE *is IND-CKA-AT security if the advantage*

$$\mathrm{Adv}^{\text{IND-CKA-AT}}_{\mathsf{SCF\text{-}PEKS/PKE},\mathcal{A}}(\kappa) :=\mid \Pr[\mathrm{Exp}^{\text{IND-CKA-AT}}_{\mathsf{SCF\text{-}PEKS/PKE},\mathcal{A}}(\kappa) = 1] - 1/2 \mid$$

is negligible for any PPT adversary $\mathcal{A}$.

Next, we define the data privacy for SCF-PEKS/PKE under chosen ciphertext attack with the server secret key and all trapdoors (IND-CCA-SSK/AT) as follows. To guarantee that the server does not obtain any information of plaintext, the adversary $\mathcal{A}$ is given to sk_{S}. Moreover, to guarantee that no information of plaintext is revealed via the text procedure, $\mathcal{A}$ is allowed to access the trapdoor oracle with no restriction.

Definition 14 (IND-CCA-SSK/AT)**.** *For any PPT adversary $\mathcal{A}$ and the security parameter $\kappa \in \mathrm{N}$, we define the experiment* $\mathrm{Exp}^{\text{IND-CCA-SSK/AT}}_{\mathsf{SCF\text{-}PEKS/PKE},\mathcal{A}}(\kappa)$ *as follows.*

$$
\begin{aligned}
&\mathrm{Exp}^{\text{IND-CCA-SSK/AT}}_{\text{SCF-PEKS/PKE},\mathcal{A}}(\kappa):\\
&\quad (pk_{\mathrm{S}}, sk_{\mathrm{S}}) \leftarrow \text{SCF-PEKS/PKE.KeyGen}_{\mathrm{S}}(1^{\kappa})\\
&\quad (pk_{\mathrm{R}}, sk_{\mathrm{R}}) \leftarrow \text{SCF-PEKS/PKE.KeyGen}_{\mathrm{R}}(1^{\kappa})\\
&\quad (\omega^*, M_0^*, M_1^*, \mathit{State})\\
&\quad \leftarrow \mathcal{A}^{\mathcal{O}_{\text{SCF-PEKS/PKE.DEC}}, \mathcal{O}_{\text{SCF-PEKS/PKE.TRAP}}}(\mathsf{find}, pk_{\mathrm{S}}, sk_{\mathrm{S}}, pk_{\mathrm{R}})\\
&\quad \mu \xleftarrow{\$} \{0,1\};\ \lambda^* \leftarrow \text{SCF-PEKS/PKE.Enc}(pk_{\mathrm{S}}, pk_{\mathrm{R}}, \omega^*, M_{\mu}^*)\\
&\quad \mu' \leftarrow \mathcal{A}^{\mathcal{O}_{\text{SCF-PEKS/PKE.DEC}}, \mathcal{O}_{\text{SCF-PEKS/PKE.TRAP}}}(\mathsf{guess}, \lambda^*, \mathit{State})\\
&\quad \mathit{If}\ \mu = \mu'\ \mathit{then\ output}\ 1,\ \mathit{and}\ 0\ \mathit{otherwise}
\end{aligned}
$$

- $\mathcal{O}_{\text{SCF-PEKS/PKE.DEC}}$*: This decryption oracle takes as input a ciphertext* $\lambda \neq \lambda^*$*, and returns the result of* $\text{SCF-PEKS/PKE.Dec}(pk_{\mathrm{R}}, sk_{\mathrm{R}}, \lambda)$*.*
- $\mathcal{O}_{\text{SCF-PEKS/PKE.TRAP}}$*: This trapdoor oracle takes as input* ω *with no restriction, and returns* $t_{\omega} \leftarrow \text{SCF-PEKS/PKE.Trapdoor}(pk_{\mathrm{R}}, sk_{\mathrm{R}}, \omega)$*. Remark that* ω^* *is also allowed to input.*

We say that a SCF-PEKS/PKE scheme SCF-PEKS/PKE *is IND-CCA-SSK/AT secure if the advantage*

$$\mathrm{Adv}^{\text{IND-CCA-SSK/AT}}_{\text{SCF-PEKS/PKE},\mathcal{A}}(\kappa) := \mid \Pr[\mathrm{Exp}^{\text{IND-CCA-SSK/AT}}_{\text{SCF-PEKS/PKE},\mathcal{A}}(\kappa) = 1] - 1/2 \mid$$

is negligible for any PPT adversary $\mathcal{A}$*.*

4 Generic Construction of SCF-PEKS/PKE

In this section, we propose a generic construction of SCF-PEKS/PKE. We construct SCF-PEKS/PKE from IBE = (IBE,Setup,IBE.Extract,IBE.Enc,IBE.Dec), TBE = (TBE.KeyGen,TBE.Enc,TBE.Dec), and OTS = (Sig.KeyGen,Sign,Verify). Our construction can be seen as an extension of a generic construction of PEKS (from anonymous IBE proposed by Abdalla et al. [1]) and a generic construction of SCF-PEKS (from anonymous IBE, TBE, and OTS proposed by Emura et al. [14]).

The Abdalla et al. construction is briefly explained as follows. A receiver has the master key mk as its secret key $sk_{\mathrm{R}}^{\mathsf{IBE}}$. A keyword ω is regarded as an identity, i.e., $\mathcal{K}$ is set to $\mathcal{ID}$, and is encrypted as follows. First, a random plaintext $R \in \mathcal{M}_{\mathsf{IBE}}$ is chosen, and next R is encrypted by IBE such that $C_{\mathsf{IBE}} \leftarrow \mathsf{IBE.Enc}(params, \omega, R)$. Then, the PEKS ciphertext is (C_{IBE}, R). A trapdoor t_{ω} is the decryption key $sk_{\omega} \leftarrow \mathsf{IBE.Extract}(params, sk_{\mathrm{R}}^{\mathsf{IBE}}, \omega)$. The test algorithm outputs 1 if $\mathsf{IBE.Dec}(params, t_{\omega}, C_{\mathsf{IBE}}) = R$ holds. Since the underlying IBE is required to be anonymous, no information of ω is revealed from C_{IBE}. By additionally employing TBE and OTS, Emura et al. [14] added the secure-channel property to the Abdalla et al. construction. In their construction, the server manages a key pair of TBE $(pk_{\mathrm{S}}^{\mathsf{TBE}}, sk_{\mathrm{S}}^{\mathsf{TBE}})$. A random plaintext $R \in \mathcal{M}_{\mathsf{IBE}}$ is encrypted by IBE, and the IBE ciphertext is encrypted by

TBE such that $C_{\mathsf{TBE}} \leftarrow \mathsf{TBE.Enc}(pk_{\mathrm{S}}^{\mathsf{TBE}}, H_{tag}(K_v), C_{\mathsf{IBE}})$, where the verification key K_v is regarded as the tag and $H_{tag} : \{0,1\}^* \rightarrow \mathcal{TAG}$ is a target collision-resistant (TCR) hash function. Finally, a signature is computed such that $\sigma \leftarrow \mathsf{Sign}(K_s, (C_{\mathsf{TBE}}, R))$. The SCF-PEKS ciphertext is $(C_{\mathsf{TBE}}, K_v, \sigma)$. The test algorithm first decrypts C_{TBE} using $sk_{\mathrm{S}}^{\mathsf{TBE}}$, next it decrypts its decryption result using a trapdoor, and then obtains R. The test algorithm outputs 1 if σ is valid on (C_{TBE}, R). Owing to the double encryption, both $sk_{\mathrm{S}}^{\mathsf{TBE}}$ and t_ω are required to run the test algorithm. It is particularly worth noting that the random plaintext R is NOT contained in the ciphertext. Emura et al. mentioned that even if R is contained in a ciphertext, it does not affect the security, and the reason for removing R is to reduce the ciphertext size.

High-Level Description of Our Construction: To integrate SCF-PEKS and PKE, the receiver additionally manages a key pair of TBE $(pk_{\mathrm{R}}^{\mathsf{TBE}}, sk_{\mathrm{R}}^{\mathsf{TBE}})$. Since the Emura et al. construction above can be seen as "tag-based" SCF-PEKS, a plaintext $M \in \mathcal{M}_{\mathsf{TBE}}$ is encrypted by $pk_{\mathrm{R}}^{\mathsf{TBE}}$ with the same tag $H_{tag}(K_v)$ such that

$$C_{\mathsf{TBE,S}} \leftarrow \mathsf{TBE.Enc}(pk_{\mathrm{S}}^{\mathsf{TBE}}, H_{tag}(K_v), C_{\mathsf{IBE}}) \text{ and}$$
$$C_{\mathsf{TBE,R}} \leftarrow \mathsf{TBE.Enc}(pk_{\mathrm{R}}^{\mathsf{TBE}}, H_{tag}(K_v), M)$$

Here, for the sake of clarity, we use subscript S for ciphertexts encrypted by the server pubic key $pk_{\mathrm{S}}^{\mathsf{TBE}}$, and use subscript R for ciphertexts encrypted by the receiver pubic key $pk_{\mathrm{R}}^{\mathsf{TBE}}$. The sender computes the OTS σ on $(C_{\mathsf{TBE,S}}, C_{\mathsf{TBE,R}}, R)$. A SCF-PEKS/PKE ciphertext is described as $\lambda = (C_{\mathsf{TBE,S}}, C_{\mathsf{TBE,R}}, K_v, \sigma, R)$. It is particularly worth noting that the random plaintext R is contained in the ciphertext unlike in the Emura et al. construction. The ciphertext now provides public verifiability since anyone can verify σ. Since the decryption algorithm needs to verify σ, this public verifiability is necessary.

The construction basically works well since TBE+OTS yields CCA-secure PKE [24]. The main difficulty to be handled is explained as follows. Let $\lambda^* = (C^*_{\mathsf{TBE,S}}, C^*_{\mathsf{TBE,R}}, K^*_v, \sigma^*, R^*)$ be the challenge ciphertext in the IND-CKA-SSK game. Now we consider how to reduce the IND-CKA-SSK security to the IBE-ANO-CCA security. Since the adversary $\mathcal{A}$ has $sk_{\mathrm{S}}^{\mathsf{TBE}}$, $\mathcal{A}$ can decrypt $C^*_{\mathsf{TBE,S}}$. Let C^*_{IBE} be the decryption result. Then, $\mathcal{A}$ can compute a valid ciphertext $\lambda \neq \lambda^*$ such that (1) (K_s, K_v) is chosen by $\mathcal{A}$ with the condition $K_v \neq K^*_v$, (2) C^*_{IBE} is re-encrypted with the tag $H_{tag}(K_v)$ such that $C_{\mathsf{TBE,S}} \leftarrow \mathsf{TBE.Enc}(pk_{\mathrm{S}}^{\mathsf{TBE}}, H_{tag}(K_v), C^*_{\mathsf{IBE}})$, (3) $C_{\mathsf{TBE,R}} \leftarrow \mathsf{TBE.Enc}(pk_{\mathrm{R}}^{\mathsf{TBE}}, H_{tag}(K_v), M)$ is computed with arbitrary M, (4) $\sigma \leftarrow \mathsf{Sign}(K_s, (C_{\mathsf{TBE,S}}, C_{\mathsf{TBE,R}}, R^*))$ is computed, and (5) $\lambda = (C_{\mathsf{TBE,S}}, C_{\mathsf{TBE,R}}, K_v, \sigma, R^*)$ is sent to the test oracle with $\omega \in \{\omega^*_0, \omega^*_1\}$. Although the reduction algorithm obtains C^*_{IBE}, the algorithm cannot send the challenge ciphertext C^*_{IBE} with either ω^*_0 or ω^*_1 to the decryption oracle of IBE. Thus, the security proof fails. To protect against this re-encryption attack, we modify the plaintext of C_{IBE} as

$$C_{\mathsf{IBE}} \leftarrow \mathsf{IBE.Enc}(params, \omega, R) \text{ with } R = H_{ibe}(K_v)$$

where $H_{ibe} : \{0,1\}^* \to \mathcal{M}_{\mathsf{IBE}}$ is a TCR hash function, and the test algorithm checks whether or not $R = H_{ibe}(K_v)$. This structure prevents the adversary from employing different K_v and thus, if C^*_{IBE} appears as above, then $K_v = K^*_v$ must hold unless the TCR property is broken. Since this situation contradicts sEUF-CMA security, our simulation works well. Since R can be computed from K_v, we can now remove R from λ without losing public verifiability, and an SCF-PEKS/PKE ciphertext is described as $\lambda = (C_{\mathsf{TBE,S}}, C_{\mathsf{TBE,R}}, K_v, \sigma)$.

We give our construction as follows. Assume that $\mathcal{C}_{\mathsf{IBE}} \subseteq \mathcal{M}_{\mathsf{TBE}}$ and $\mathcal{C}_{\mathsf{TBE}} \times \mathcal{C}_{\mathsf{TBE}} \times \mathcal{M}_{\mathsf{IBE}} \subseteq \mathcal{M}_{\mathsf{Sig}}$, where $\mathcal{C}_{\mathsf{IBE}}$ and $\mathcal{M}_{\mathsf{IBE}}$ are a ciphertext space and plaintext space of IBE respectively, $\mathcal{M}_{\mathsf{TBE}}$ is a plaintext space of TBE, and $\mathcal{M}_{\mathsf{Sig}}$ is a message space of OTS.

The Proposed Construction

$\mathsf{SCF\text{-}PEKS/PKE.KeyGen_S}(1^\kappa)$: Run $(pk_{\mathrm{S}}^{\mathsf{TBE}}, sk_{\mathrm{S}}^{\mathsf{TBE}}) \leftarrow \mathsf{TBE.KeyGen}(1^\kappa)$. Output $pk_{\mathrm{S}} = pk_{\mathrm{S}}^{\mathsf{TBE}}$ and $sk_{\mathrm{S}} = sk_{\mathrm{S}}^{\mathsf{TBE}}$.

$\mathsf{SCF\text{-}PEKS/PKE.KeyGen_R}(1^\kappa)$: Run $(pk_{\mathrm{R}}^{\mathsf{IBE}}, sk_{\mathrm{R}}^{\mathsf{IBE}}) \leftarrow \mathsf{IBE.Setup}(1^\kappa)$ and $(pk_{\mathrm{R}}^{\mathsf{TBE}}, sk_{\mathrm{R}}^{\mathsf{TBE}}) \leftarrow \mathsf{TBE.KeyGen}(1^\kappa)$. Output $pk_{\mathrm{R}} = (pk_{\mathrm{R}}^{\mathsf{IBE}}, pk_{\mathrm{R}}^{\mathsf{TBE}})$ and $sk_{\mathrm{R}} = (sk_{\mathrm{R}}^{\mathsf{IBE}}, sk_{\mathrm{R}}^{\mathsf{TBE}})$. We assume that TCR hash functions $H_{tag} : \{0,1\}^* \to \mathcal{TAG}$ and $H_{ibe} : \{0,1\}^* \to \mathcal{M}_{\mathsf{IBE}}$ are contained in pk_{R}.

$\mathsf{SCF\text{-}PEKS/PKE.Trapdoor}(pk_{\mathrm{R}}, sk_{\mathrm{R}}, \omega)$: Parse $pk_{\mathrm{R}} = (pk_{\mathrm{R}}^{\mathsf{IBE}}, pk_{\mathrm{R}}^{\mathsf{TBE}})$ and $sk_{\mathrm{R}} = (sk_{\mathrm{R}}^{\mathsf{IBE}}, sk_{\mathrm{R}}^{\mathsf{TBE}})$. Run $sk_\omega \leftarrow \mathsf{IBE.Extract}(pk_{\mathrm{R}}^{\mathsf{IBE}}, sk_{\mathrm{R}}^{\mathsf{IBE}}, \omega)$ and output $t_\omega = sk_\omega$.

$\mathsf{SCF\text{-}PEKS/PKE.Enc}(pk_{\mathrm{S}}, pk_{\mathrm{R}}, \omega, M)$: Parse $pk_{\mathrm{S}} = pk_{\mathrm{S}}^{\mathsf{TBE}}$ and $pk_{\mathrm{R}} = (pk_{\mathrm{R}}^{\mathsf{IBE}}, pk_{\mathrm{R}}^{\mathsf{TBE}})$. Run $(K_s, K_v) \leftarrow \mathsf{Sig.KeyGen}(1^\kappa)$ and compute $t = H_{tag}(K_v)$ and $R = H_{ibe}(K_v)$. Run $C_{\mathsf{IBE}} \leftarrow \mathsf{IBE.Enc}(pk_{\mathrm{R}}^{\mathsf{IBE}}, \omega, R)$. Compute $C_{\mathsf{TBE,S}} \leftarrow \mathsf{TBE.Enc}(pk_{\mathrm{S}}^{\mathsf{TBE}}, t, C_{\mathsf{IBE}})$, $C_{\mathsf{TBE,R}} \leftarrow \mathsf{TBE.Enc}(pk_{\mathrm{R}}^{\mathsf{TBE}}, t, M)$, and $\sigma \leftarrow \mathsf{Sign}(K_s, (C_{\mathsf{TBE,S}}, C_{\mathsf{TBE,R}}, R))$, and output $\lambda = (C_{\mathsf{TBE,S}}, C_{\mathsf{TBE,R}}, K_v, \sigma)$.

$\mathsf{SCF\text{-}PEKS/PKE.Dec}(pk_{\mathrm{R}}, sk_{\mathrm{R}}, \lambda)$: Parse $pk_{\mathrm{R}} = (pk_{\mathrm{R}}^{\mathsf{IBE}}, pk_{\mathrm{R}}^{\mathsf{TBE}})$, $sk_{\mathrm{R}} = (sk_{\mathrm{R}}^{\mathsf{IBE}}, sk_{\mathrm{R}}^{\mathsf{TBE}})$ and $\lambda = (C_{\mathsf{TBE,S}}, C_{\mathsf{TBE,R}}, K_v, \sigma)$. Compute $R = H_{ibe}(K_v)$. If $\mathsf{Verify}(K_v, (C_{\mathsf{TBE,S}}, C_{\mathsf{TBE,R}}, R), \sigma) = 0$, then output $\perp$. Otherwise, compute $t = H_{tag}(K_v)$ and output $M \leftarrow \mathsf{TBE.Dec}(pk_{\mathrm{R}}^{\mathsf{TBE}}, sk_{\mathrm{R}}^{\mathsf{TBE}}, t, C_{\mathsf{TBE,R}})$.

$\mathsf{SCF\text{-}PEKS/PKE.Test}(pk_{\mathrm{S}}, sk_{\mathrm{S}}, pk_{\mathrm{R}}, t_\omega, \lambda)$: Parse $pk_{\mathrm{S}} = pk_{\mathrm{S}}^{\mathsf{TBE}}$, $sk_{\mathrm{S}} = sk_{\mathrm{S}}^{\mathsf{TBE}}$, $pk_{\mathrm{R}} = (pk_{\mathrm{R}}^{\mathsf{IBE}}, pk_{\mathrm{R}}^{\mathsf{TBE}})$, and $\lambda = (C_{\mathsf{TBE,S}}, C_{\mathsf{TBE,R}}, K_v, \sigma)$. Compute $t = H_{tag}(K_v)$, and run $C'_{\mathsf{IBE}} \leftarrow \mathsf{TBE.Dec}(pk_{\mathrm{S}}^{\mathsf{TBE}}, sk_{\mathrm{S}}^{\mathsf{TBE}}, t, C_{\mathsf{TBE,S}})$ and $R' \leftarrow \mathsf{IBE.Dec}(pk_{\mathrm{R}}^{\mathsf{IBE}}, t_\omega, C'_{\mathsf{IBE}})$. Output 1 if $R' = H_{ibe}(K_v)$ and $\mathsf{Verify}(K_v, (C_{\mathsf{TBE,S}}, C_{\mathsf{TBE,R}}, R'), \sigma) = 1$ hold, and 0 otherwise.

Obviously, correctness holds if TBE, IBE, and OTS are correct. Due to the page limitation, we omit security proofs of following theorems. We will show the details of proofs in the full version of this paper.

Theorem 1. SCF-PEKS/PKE *is weakly consistent if* IBE *is IBE-IND-CPA secure.*

Theorem 2. SCF-PEKS/PKE *is strongly consistent if* IBE *is unrestricted strong collision-free.*

Theorem 3. SCF-PEKS/PKE *is IND-CKA-SSK secure if* IBE *is IBE-ANO-CCA secure,* OTS *is one-time sEUF-CMA secure, and* H_{ibe} *is a TCR hash function.*

Theorem 4. SCF-PEKS/PKE *is IND-CKA-AT secure if* TBE *is IND-stag-CCA secure,* OTS *is one-time sEUF-CMA secure, and* H_{tag} *is a TCR hash function.*

Theorem 5. SCF-PEKS/PKE *is IND-CCA-SSK/AT secure if* TBE *is IND-stag-CCA secure,* OTS *is one-time sUF-CMA secure, and* H_{tag} *is a TCR hash function.*

5 Instantiation of Our Generic Construction

For TBE, we can simply employ the Kiltz TBE scheme [24], and for the OTS, we can employ any sEUF-CMA secure OTS scheme, e.g., the Wee OTS scheme [29]. We explain how to construct an IBE scheme that matches our requirements, i.e., with unrestricted strong collision-freeness which defined in this paper, and with IBE-ANO-CCA security. To the best of our knowledge, the strongest notion among several robustnesses and collision-freenesses is complete robustness defined by Farshim et al. [17]. They showed that complete robustness implies unrestricted strong robustness. Since unrestricted strong collision-freeness is implied by unrestricted strong robustness, it is enough to construct an IBE scheme with complete robustness for our purpose. Farshim et al. also showed that the transformation from weakly robust IBE (and commitment with the standard hiding and binding properties) to strongly robust IBE, proposed by Abdalla et al. [2,3], is already powerful enough to construct completely robust IBE.[4] Moreover, Abdalla et al. also proposed a transformation from IBE to weakly robust IBE. Since these transformations preserve the anonymity and CCA security of the underlying IBE scheme, we can construct an IBE-ANO-CCA secure IBE scheme with unrestricted strong collision-freeness by applying the two Abdalla et al. transformations (from normal to weakly robust, and from weakly robust to strongly robust).

We have three candidates as the underlying IBE scheme.[5] One candidate is the Gentry IBE scheme [19] which is IBE-ANO-CCA secure in the standard model. As another standard model construction, we can employ a variant of the Boyen-Waters IBE scheme [8] that uses the CHK transform to achieve IBE-ANO-CCA security. Although Abdalla et al. [2,3] mentioned that these schemes

[4] Farshim et al. [17] showed that a transformation proposed by Mohassel [25] is also powerful enough to construct completely robust IBE,al though the transformation requires the random oracle.

[5] Although other anonymous IBE schemes without random oracles based on simple assumptions have been proposed, we cannot employ them. For example, the Chen et al. IBE scheme [11] and the Jutla-Roy IBE scheme [22,23] are IBE-ANO-CPA secure. Although Jutla and Roy gave a CCA version, the scheme is not anonymous due to its public verifiability where one can check whether or not a ciphertext is valid for an identity.

are not robust, we can add unrestricted strong collision-freeness property to them via the Abdalla et al. transformations. Other candidate is the CCA-version of the Boneh-Franklin IBE scheme [7] which is IBE-ANO-CCA secure in the random oracle model. The scheme is also known to provide strong robustness. However, it is not clear whether the scheme provides unrestricted strong collision-freeness. Thus, we need to properly employ the Abdalla et al. transformation.

Since unrestricted strong collision-freeness is weaker than complete robustness, employing the two Abdalla et al. transformations as above may be somewhat excessive. Thus, directly and simply constructing an IBE-ANO-CCA secure IBE scheme with unrestricted strong collision-freeness is left as an interesting open problem.

Acknowledgement. This work was supported in part by the JSPS KAKENHI Grant Number JP16H02808 and the MIC/SCOPE #162108102. We thank Dr. Yohei Watanabe for helpful discussion.

References

1. Abdalla, M., et al.: Searchable encryption revisited: consistency properties, relation to anonymous IBE, and extensions. J. Cryptol. **21**(3), 350–391 (2008)
2. Abdalla, M., Bellare, M., Neven, G.: Robust encryption. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 480–497. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11799-2_28
3. Abdalla, M., Bellare, M., Neven, G.: Robust encryption. J. Cryptol. **31**(2), 307–350 (2018)
4. Baek, J., Safavi-Naini, R., Susilo, W.: On the integration of public key data encryption and public key encryption with keyword search. In: Katsikas, S.K., López, J., Backes, M., Gritzalis, S., Preneel, B. (eds.) ISC 2006. LNCS, vol. 4176, pp. 217–232. Springer, Heidelberg (2006). https://doi.org/10.1007/11836810_16
5. Bellare, M., Shoup, S.: Two-tier signatures, strongly unforgeable signatures, and Fiat-Shamir without random oracles. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 201–216. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71677-8_14
6. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_30
7. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_13
8. Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (without random oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006). https://doi.org/10.1007/11818175_17
9. Buccafurri, F., Lax, G., Sahu, R.A., Saraswat, V.: Practical and secure integrated PKE+PEKS with keyword privacy. In: SECRYPT, pp. 448–453 (2015)
10. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_13

11. Chen, J., Lim, H.W., Ling, S., Wang, H., Wee, H.: Shorter IBE and signatures via asymmetric pairings. In: Abdalla, M., Lange, T. (eds.) Pairing 2012. LNCS, vol. 7708, pp. 122–140. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36334-4_8
12. Chen, Y., Zhang, J., Lin, D., Zhang, Z.: Generic constructions of integrated PKE and PEKS. Des. Codes Cryptogr. **78**(2), 493–526 (2016)
13. Emura, K.: A generic construction of secure-channel free searchable encryption with multiple keywords. In: Yan, Z., Molva, R., Mazurczyk, W., Kantola, R. (eds.) NSS 2017. LNCS, vol. 10394, pp. 3–18. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64701-2_1
14. Emura, K., Miyaji, A., Rahman, M.S., Omote, K.: Generic constructions of secure-channel free searchable encryption with adaptive security. Secur. Commun. Netw. **8**(8), 1547–1560 (2015)
15. Fang, L., Susilo, W., Ge, C., Wang, J.: A secure channel free public key encryption with keyword search scheme without random oracle. In: Garay, J.A., Miyaji, A., Otsuka, A. (eds.) CANS 2009. LNCS, vol. 5888, pp. 248–258. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10433-6_16
16. Fang, L., Susilo, W., Ge, C., Wang, J.: Public key encryption with keyword search secure against keyword guessing attacks without random oracle. Inf. Sci. **238**, 221–241 (2013)
17. Farshim, P., Libert, B., Paterson, K.G., Quaglia, E.A.: Robust encryption, revisited. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 352–368. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36362-7_22
18. Fuhr, T., Paillier, P.: Decryptable searchable encryption. In: Susilo, W., Liu, J.K., Mu, Y. (eds.) ProvSec 2007. LNCS, vol. 4784, pp. 228–236. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75670-5_17
19. Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_27
20. Guo, L., Yau, W.: Efficient secure-channel free public key encryption with keyword search for EMRs in cloud storage. J. Med. Syst. **39**(2), 11 (2015)
21. Hofheinz, D., Weinreb, E.: Searchable encryption with decryption in the standard model. IACR Cryptology ePrint Archive 2008:423 (2008)
22. Jutla, C.S., Roy, A.: Shorter quasi-adaptive NIZK proofs for linear subspaces. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8269, pp. 1–20. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42033-7_1
23. Jutla, C.S., Roy, A.: Shorter quasi-adaptive NIZK proofs for linear subspaces. J. Cryptol. **30**(4), 1116–1156 (2017)
24. Kiltz, E.: Chosen-ciphertext security from tag-based encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_30
25. Mohassel, P.: A closer look at anonymity and robustness in encryption schemes. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 501–518. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17373-8_29
26. Rhee, H.S., Park, J.H., Lee, D.H.: Generic construction of designated tester public-key encryption with keyword search. Inf. Sci. **205**, 93–109 (2012)
27. Saraswat, V., Sahu, R.A.: Short integrated PKE+PEKS in standard model. In: Ali, S.S., Danger, J.-L., Eisenbarth, T. (eds.) SPACE 2017. LNCS, vol. 10662, pp. 226–246. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-71501-8_13

28. Wang, T., Au, M.H., Wu, W.: An efficient secure channel free searchable encryption scheme with multiple keywords. In: Chen, J., Piuri, V., Su, C., Yung, M. (eds.) NSS 2016. LNCS, vol. 9955, pp. 251–265. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46298-1_17
29. Wee, H.: Public key encryption against related key attacks. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 262–279. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30057-8_16
30. Zhang, R., Imai, H.: Combining public key encryption with keyword search and public key encryption. IEICE Trans. **92–D**(5), 888–896 (2009)