Generic Construction of Forward Secure Public Key Authenticated Encryption with Keyword Search

Keita Emura[§]

[§]National Institute of Information and Communications Technology (NICT), Japan.

March 22, 2023

Abstract

Forward security is a fundamental requirement in searchable encryption, where a newly generated ciphertext is not allowed to be searched by previously generated trapdoors. However, forward security is somewhat overlooked in the public key encryption with keyword search (PEKS) context and there are few proposals, whereas forward security has been stated as a default security notion in the (dynamic) symmetric searchable encryption (SSE) context. In the PEKS context, forward secure PEKS (FS-PEKS) is essentially the same as public key encryption with temporary keyword search (PETKS) proposed by Abdalla et al. (JoC 2016) which can be constructed generically from hierarchical identity-based encryption (HIBE) with level-1 anonymity. Alternatively, Zeng et al. (IEEE Transactions on Cloud Computing 2022) also proposed a generic construction of FS-PEKS from attribute-based searchable encryption supporting OR gates. In the public key authenticated encryption with keyword search (PAEKS) context, a concrete forward secure PAEKS (FS-PAEKS) construction has been proposed by Jiang et al. (The Computer Journal 2022), and no generic construction has been proposed to date. In this paper, we propose a generic construction of FS-PAEKS from PAEKS. In addition to PAEKS, we employ 0/1 encodings proposed by Lin et al. (ACNS 2005). We also show that the Jiang et al. FS-PAEKS scheme does not provide forward security, and thus our generic construction yields the first secure FS-PAEKS schemes. Our generic construction is quite simple, and it can also be applied to construct FS-PEKS. Our generic construction yields a comparably efficient FS-PEKS scheme compared to the previous scheme. Moreover, it eliminates the hierarchical structure or attribute-based feature of the previous generic constructions which is meaningful from a feasibility perspective.

1 Introduction

Searchable encryption is a fundamental tool to provide data confidentiality and data searchability simultaneously. In searchable encryption, forward security is a fundamental requirement, where a newly generated ciphertext is not allowed to be searched by previously generated trapdoors. In the (dynamic) symmetric searchable encryption (SSE) context [46], forward security, which is also referred to as forward privacy, has been a default security notion since the seminal work by Stefanov et al. [47]. However, forward security is somewhat overlooked in the public key encryption with keyword search (PEKS) context [7].

Currently, three forward secure PEKS (FS-PEKS) schemes have been proposed [30, 49, 50], to the best of our knowledge. Kim et al. [30] constructed FS-PEKS from hierarchical identity-based encryption (HIBE). In their construction, a fixed message (ind_i in their paper) is encrypted by

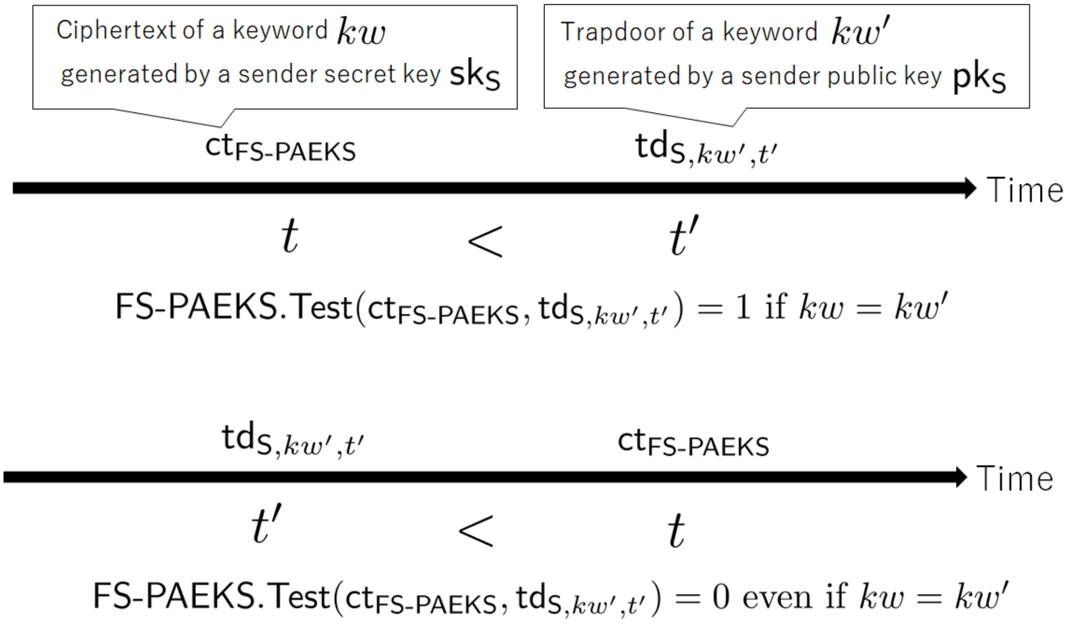
Ciphertext of a keyword kw generated by a sender secret key $\mathsf{sk_S}$
Trapdoor of a keyword kw' generated by a sender public key $\mathsf{pk_S}$
$\mathsf{ct_{FS\text{-}PAEKS}}$
$\mathsf{td}_{\mathsf{S},kw',t'}$
Time
$t < t'$
$\mathsf{FS\text{-}PAEKS.Test}(\mathsf{ct_{FS\text{-}PAEKS}}, \mathsf{td}_{\mathsf{S},kw',t'}) = 1$ if $kw = kw'$
$\mathsf{td}_{\mathsf{S},kw',t'}$
$\mathsf{ct_{FS\text{-}PAEKS}}$
Time
$t' < t$
$\mathsf{FS\text{-}PAEKS.Test}(\mathsf{ct_{FS\text{-}PAEKS}}, \mathsf{td}_{\mathsf{S},kw',t'}) = 0$ even if $kw = kw'$

Figure 1: FS-PAEKS

the underlying HIBE scheme. However, this construction does not provide consistency due to the observation of Abdalla et al. [1]. Zhang et al. [50] proposed an FS-PEKS scheme from lattices. Their construction employs a secret key update algorithm, and an adversary is allowed to obtain secret keys under some restrictions, which is reminiscent of forward secure public key encryption [11] that considers other scenario to the trapdoor leakage. Zeng et al. [49] proposed an FS-PEKS scheme in bilinear groups. They also mentioned that FS-PEKS can be constructed from attribute-based searchable encryption supporting OR gates, and FS-PEKS is essentially the same as public key encryption with temporary keyword search (PETKS) [1], which can be constructed generically from HIBE with level-1 anonymity.

In PEKS, anyone can generate a ciphertext of a keyword. Thus, if one obtains a trapdoor, then information about which keyword is associated with the trapdoor is leaked by running the test algorithm with self-made ciphertexts. To prevent this keyword guessing attack, public key authenticated encryption with keyword search (PAEKS) has been proposed [10, 14–17, 22, 37–39, 41, 42] where a sender secret key is required for encryption. As in PEKS, forward secure PAEKS (FS-PAEKS) can be defined where the encryption algorithm takes a time period t and the trapdoor generation algorithm takes a time period t'. In addition to the search condition defined in PAEKS, a trapdoor works if $t < t'$, that is, a newly generated ciphertext is not allowed to be searched by previously generated trapdoors. See Fig. 1. To date, a concrete FS-PAEKS scheme has been proposed by Jiang et al. [24], and no generic construction of FS-PAEKS has been proposed so far.

Our Contribution. In this paper, we propose a generic construction of FS-PAEKS from PAEKS. We employ 0/1 encodings, which were originally proposed for solving the Millionaires' problem by Lin et al. [35]. We focus on the fact that the encodings are effective way to translate an inequality condition $t < t'$ to an equality condition, and PAEKS originally supports keyword equality matching. Our generic construction yields FS-PAEKS schemes under several complexity assumptions. For example, a lattice-based FS-PAEKS scheme by employing the Cheng-Meng PAEKS scheme [15], a pairing-based FS-PAEKS scheme by employing the Qin et al. PAEKS scheme [42], or other FS-PAEKS schemes by employing PAEKS schemes instantiated by a generic construction of PAEKS [17].[1]

We remark that Jiang et al. [24] employed symmetric pairings which can be seen as a DDH solver (where DDH stands for decisional Diffie-Hellman), but they assumed that the DDH problem is hard. Actually, their FS-PAEKS scheme does not provide forward security. We give a concrete attack in Section 6. Thus, no secure FS-PAEKS scheme has been proposed so far, and our generic construction yields the first secure FS-PAEKS schemes.

Our generic construction is quite simple, and it can also be applied to construct FS-PEKS. This eliminates the hierarchical structure or attribute-based feature of the previous generic constructions which is meaningful from a feasibility perspective. In addition, since PEKS can be constructed from anonymous IBE [1], efficient FS-PEKS constructions can be obtained easily. For example, if we employ the Boneh-Franklin (BF) IBE scheme [8] as the component of the underlying PEKS scheme, then an efficient paring-based FS-PEKS scheme in the random oracle model can be constructed. If the Gentry-Peikert-Vaikuntanathan (GPV) IBE scheme [20] is employed, then an efficient lattice-based FS-PEKS scheme in the quantum random oracle model can be constructed.[2] Moreover, FS-PEKS schemes that are secure in the standard model also can be obtained from the Gentry IBE scheme [19], the Lewko IBE scheme [34], the Chen-Wei-Ling-Wang-Wee IBE (CLLWW) scheme [13], the Kurosawa-Phong (KP) IBE scheme [31], the Jutla-Roy (JR) IBE scheme [25], the Yamada IBE scheme [48], the Katsumata IBE scheme [28], and the Jager-Kurek-Niehues (JKN) IBE scheme [23].

Application: As an application of FS-PEKS, Zeng et al. [49] introduced a secure cloud storage. A sender encrypts a file and stores the ciphertext in a cloud storage together with a ciphertext of keywords. A receiver retrieves an encrypted file by sending a search token (trapdoor). If no forward security is provided, then the cloud server can run the search procedure by using a previously sent trapdoor, which will lead to leakage-abuse attacks. Due to this motivation, Zeng et al. proposed FS-PEKS. This scenario matches FS-PAEKS when a receiver considers who stored a ciphertext. Jiang et al. [24] considered an e-mail routing system as an application of FS-PAEKS. An encrypted e-mail is sent to a gateway. The gateway forwards an encrypted e-mail owing to the search result, e.g., the e-mail receiver specifies a keyword "urgent" and send its trapdoor to the gateway. If no forward security is provided, then the gateway can run the search procedure by using a previously sent trapdoor. Jiang et al. also introduced a case of electronic medical records where a disease name such as "cancer" is set as a keyword for searching clinical records. If no forward security is provided, then a newly added encrypted clinical record could be searched if this patient had a previously searched disease.

2 Preliminaries

Notation. For a positive integer $n \in \mathbb{N}$, we write $[1, n] = \{1, 2, \dots, n\}$. $x \xleftarrow{\$} S$ denotes choosing an element x from a finite set S uniformly at random. For a security parameter λ, $\mathsf{negl}(\lambda)$ is a negligible function where for any $c > 0$, there exists an integer I such that $\mathsf{negl}(\lambda) < 1/\lambda^c$ for all $\lambda > I$. PPT stands for probabilistic polynomial-time.

2.1 PAEKS

In this section, we define PAEKS. We primarily follow the definitions given in [17] because it considers consistency in a multi-sender setting where a trapdoor associated with a sender does not work against ciphertexts generated by the secret key of another sender, even if the same keyword

[1] A flaw in the security proof of the generic construction [37] is identified in [17].

[2] The GPV-IBE scheme is secure in the quantum random oracle model [29].

is associated. As a difference from [17], we introduce the setup algorithm $\mathsf{PAEKS.Setup}$ because it captures most of previous PAEKS syntax whereas a designated-receiver setting is considered in [17] where the $\mathsf{PAEKS.KG_S}$ algorithm takes a receiver public key $\mathsf{pk_R}$ as input, and no setup algorithm is defined. We remark that the following definition can be modified easily to capture the designated-receiver setting.

Definition 1 (Syntax of PAEKS). *A PAEKS scheme* PAEKS *consists of the following six algorithms* $(\mathsf{PAEKS.Setup}, \mathsf{PAEKS.KG_R}, \mathsf{PAEKS.KG_S}, \mathsf{PAEKS.Enc}, \mathsf{PAEKS.Trapdoor}, \mathsf{PAEKS.Test})$ *defined as follows.*

$\mathsf{PAEKS.Setup}$: *The setup algorithm takes a security parameter* λ *as input, and outputs a common parameter* pp*. We assume that* pp *implicitly contains the keyword space* $\mathcal{KS}$*.*

$\mathsf{PAEKS.KG_R}$: *The receiver key generation algorithm takes* pp *as input, and outputs a public key* $\mathsf{pk_R}$ *and secret key* $\mathsf{sk_R}$*.*

$\mathsf{PAEKS.KG_S}$: *The sender key generation algorithm takes* pp *as input, and outputs a public key* $\mathsf{pk_S}$ *and secret key* $\mathsf{sk_S}$*.*

$\mathsf{PAEKS.Enc}$: *The keyword encryption algorithm takes* $\mathsf{pk_R}$*,* $\mathsf{pk_S}$*,* $\mathsf{sk_S}$*, and a keyword* $kw \in \mathcal{KS}$ *as input, and outputs a ciphertext* $\mathsf{ct_{PAEKS}}$*.*

$\mathsf{PAEKS.Trapdoor}$: *The trapdoor algorithm takes* $\mathsf{pk_R}$*,* $\mathsf{pk_S}$*,* $\mathsf{sk_R}$*, and a keyword* $kw' \in \mathcal{KS}$ *as input, and outputs a trapdoor* $\mathsf{td}_{\mathsf{S},kw'}$*.*

$\mathsf{PAEKS.Test}$: *The test algorithm takes* $\mathsf{ct_{PAEKS}}$ *and* $\mathsf{td}_{\mathsf{S},kw'}$ *as input, and outputs 1 or 0.*

Definition 2 (Correctness). *For any security parameter* λ*, any common parameter* $\mathsf{pp} \leftarrow \mathsf{PAEKS.Setup}(1^\lambda)$*, any key pairs* $(\mathsf{pk_R}, \mathsf{sk_R}) \leftarrow \mathsf{PAEKS.KG_R}(\mathsf{pp})$ *and* $(\mathsf{pk_S}, \mathsf{sk_S}) \leftarrow \mathsf{PAEKS.KG_S}(\mathsf{pp})$*, and any keyword* $kw \in \mathcal{KS}$*, let* $\mathsf{ct_{PAEKS}} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk_R}, \mathsf{pk_S}, \mathsf{sk_S}, kw)$ *and* $\mathsf{td}_{\mathsf{S},kw} \leftarrow \mathsf{PAEKS.Trapdoor}(\mathsf{pk_R}, \mathsf{pk_S}, \mathsf{sk_R}, kw)$*. Then* $\Pr[\mathsf{PAEKS.Test}(\mathsf{ct_{PAEKS}}, \mathsf{td}_{\mathsf{S},kw}) = 1] = 1 - \mathsf{negl}(\lambda)$ *holds.*

Next, we define consistency that defines the condition by which the $\mathsf{PAEKS.Test}$ algorithm outputs 0. As in PEKS, essentially, $0 \leftarrow \mathsf{PAEKS.Test}(\mathsf{ct_{PAEKS}}, \mathsf{td}_{\mathsf{S},kw})$ when $\mathsf{ct_{PAEKS}} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk_R}, \mathsf{pk_S}, \mathsf{sk_S}, kw)$, $\mathsf{td}_{\mathsf{S},kw'} \leftarrow \mathsf{PAEKS.Trapdoor}(\mathsf{pk_R}, \mathsf{pk_S}, \mathsf{sk_R}, kw')$, and $kw \neq kw'$. However, due to its authenticity, a trapdoor associated with a sender should not work against ciphertexts generated by the secret key of another sender, even if the same keyword is associated. Thus, we introduce the definition given in [17] that considers this case.

Definition 3 (Computational Consistency). *For all PPT adversaries* $\mathcal{A}$*, we define the following experiment.*

$$
\begin{aligned}
&\mathsf{Exp}^{\mathsf{consist}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda): \\
&\quad \mathsf{pp} \leftarrow \mathsf{PAEKS.Setup}(1^\lambda);\ (\mathsf{pk_R}, \mathsf{sk_R}) \leftarrow \mathsf{PAEKS.KG_R}(\mathsf{pp}) \\
&\quad (\mathsf{pk}_{\mathsf{S}[0]}, \mathsf{sk}_{\mathsf{S}[0]}) \leftarrow \mathsf{PAEKS.KG_S}(\mathsf{pp});\ (\mathsf{pk}_{\mathsf{S}[1]}, \mathsf{sk}_{\mathsf{S}[1]}) \leftarrow \mathsf{PAEKS.KG_S}(\mathsf{pp}) \\
&\quad (kw, kw', i, j) \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{pk_R}, \mathsf{pk}_{\mathsf{S}[0]}, \mathsf{pk}_{\mathsf{S}[1]}) \\
&\qquad \textit{s.t. } kw, kw' \in \mathcal{KS} \wedge i, j \in \{0,1\} \wedge (kw, i) \neq (kw', j) \\
&\quad \mathsf{ct_{PAEKS}} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk_R}, \mathsf{pk}_{\mathsf{S}[i]}, \mathsf{sk}_{\mathsf{S}[i]}, kw) \\
&\quad \mathsf{td}_{\mathsf{S}[j],kw'} \leftarrow \mathsf{PAEKS.Trapdoor}(\mathsf{pk_R}, \mathsf{pk}_{\mathsf{S}[j]}, \mathsf{sk_R}, kw') \\
&\quad \textit{If } \mathsf{PAEKS.Test}(\mathsf{ct_{PAEKS}}, \mathsf{td}_{\mathsf{S}[j],kw'}) = 1,\ \textit{then output } 1,\ \textit{and } 0 \textit{ otherwise.}
\end{aligned}
$$

We say that a PAEKS scheme PAEKS *is consistent if the advantage*

$$\mathsf{Adv}^{\mathsf{consist}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{consist}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda) = 1]$$

is negligible in the security parameter λ.

Next, we define indistinguishability against the chosen keyword attack (IND-CKA) which guarantees that no information about the keyword is leaked from ciphertexts. Qin et al. [41] considered multi-ciphertext indistinguishability (MCI) where in the IND-CKA experiment $\mathcal{A}$ declares two keyword vectors $(kw^*_{0,1}, \dots, kw^*_{0,N})$ and $(kw^*_{1,1}, \dots, kw^*_{1,N})$ for some N, and the challenger returns the challenge ciphertexts of $kw^*_{b,i}$ for $i \in [1, N]$. As mentioned in [42], if the encryption oracle $\mathcal{O}_C$ has no restriction (i.e., any input is allowed), then IND-CKA implies MCI. Thus, the following definition provides MCI security.

Definition 4 (IND-CKA)**.** *For all PPT adversaries* $\mathcal{A}$, *we define the following experiment.*

$$\begin{aligned}
&\mathsf{Exp}^{\mathsf{IND\text{-}CKA}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda, n) :\\
&\quad \mathsf{pp} \leftarrow \mathsf{PAEKS.Setup}(1^\lambda);\ (\mathsf{pk_R}, \mathsf{sk_R}) \leftarrow \mathsf{PAEKS.KG_R}(\mathsf{pp})\\
&\quad \textit{For } i \in [1, n],\ (\mathsf{pk}_{\mathsf{S}[i]}, \mathsf{sk}_{\mathsf{S}[i]}) \leftarrow \mathsf{PAEKS.KG_S}(\mathsf{pp})\\
&\quad (kw^*_0, kw^*_1, i^*, \mathsf{state}) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pp}, \mathsf{pk_R}, \{\mathsf{pk}_{\mathsf{S}[i]}\}_{i\in[1,n]})\\
&\qquad \textit{s.t. } kw^*_0, kw^*_1 \in \mathcal{KS} \wedge\ kw^*_0 \neq kw^*_1 \wedge i^* \in [1, n]\\
&\quad b \xleftarrow{\$} \{0, 1\};\ \mathsf{ct}^*_{\mathsf{PAEKS}} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk_R}, \mathsf{pk}_{\mathsf{S}[i^*]}, \mathsf{sk}_{\mathsf{S}[i^*]}, kw^*_b)\\
&\quad b' \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{state}, \mathsf{ct}^*_{\mathsf{PAEKS}})\\
&\quad \textit{If } b = b' \textit{ then output } 1, \textit{ and } 0 \textit{ otherwise.}
\end{aligned}$$

Here, $\mathcal{O} := \{\mathcal{O}_C(\mathsf{pk_R}, \cdot, \cdot), \mathcal{O}_T(\mathsf{pk_R}, \cdot, \mathsf{sk_R}, \cdot)\}$. $\mathcal{O}_C$ *takes* $kw \in \mathcal{KS}$ *and* $i \in [1, n]$ *as input, and returns the result of* $\mathsf{PAEKS.Enc}(\mathsf{pk_R}, \mathsf{pk}_{\mathsf{S}[i]}, \mathsf{sk}_{\mathsf{S}[i]}, kw)$. *Here, there is no restriction.* $\mathcal{O}_T$ *takes* $kw' \in \mathcal{KS}$ *and* $i \in [1, n]$ *as input, and returns the result of* $\mathsf{PAEKS.Trapdoor}(\mathsf{pk_R}, \mathsf{pk}_{\mathsf{S}[i]}, \mathsf{sk_R}, kw')$. *Here* $(kw', i) \notin \{(kw^*_0, i^*), (kw^*_1, i^*)\}$. *We say that a PAEKS scheme* PAEKS *is IND-CKA secure if the advantage*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CKA}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda, n) := \Pr[\mathsf{Exp}^{\mathsf{IND\text{-}CKA}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda, n) = 1]$$

is negligible in the security parameter λ.

Next, we define indistinguishability against the inside keyword guessing attack (IND-IKGA) which guarantees that no information about the keyword is leaked from trapdoors. Pan and Li [39] considered multi-trapdoor indistinguishability (MTI) where in the IND-IKGA experiment $\mathcal{A}$ declares two keyword vectors $(kw^*_{0,1}, \dots, kw^*_{0,N})$ and $(kw^*_{1,1}, \dots, kw^*_{1,N})$ for some N, and the challenger returns the challenge trapdoors of $kw^*_{b,i}$ for $i \in [1, N]$. Although the following definition does not capture MTI, it can be modified to capture MTI if $\mathcal{A}$ is allowed to send either (kw^*_0, i^*) or (kw^*_1, i^*) to the trapdoor oracle $\mathcal{O}_T$.

Definition 5 (IND-IKGA). *For all PPT adversaries $\mathcal{A}$, we define the following experiment.*

```
$\mathsf{Exp}^{\mathsf{IND\text{-}IKGA}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda, n)$ :
    $\mathsf{pp} \leftarrow \mathsf{PAEKS.Setup}(1^\lambda)$; $(\mathsf{pk_R}, \mathsf{sk_R}) \leftarrow \mathsf{PAEKS.KG_R}(\mathsf{pp})$
    For $i \in [1, n]$, $(\mathsf{pk}_{\mathsf{S}[i]}, \mathsf{sk}_{\mathsf{S}[i]}) \leftarrow \mathsf{PAEKS.KG_S}(\mathsf{pp})$
    $(kw_0^*, kw_1^*, i^*, \mathsf{state}) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pp}, \mathsf{pk_R}, \{\mathsf{pk}_{\mathsf{S}[i]}\}_{i \in [1,n]})$
        s.t. $kw_0^*, kw_1^* \in \mathcal{KS} \wedge kw_0^* \neq kw_1^* \wedge i^* \in [1, n]$
    $b \xleftarrow{\$} \{0, 1\}$; $\mathsf{td}^*_{\mathsf{S}[i^*], kw_b^*} \leftarrow \mathsf{PAEKS.Trapdoor}(\mathsf{pk_R}, \mathsf{pk}_{\mathsf{S}[i^*]}, \mathsf{sk_R}, kw_b^*)$
    $b' \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{state}, \mathsf{td}^*_{\mathsf{S}[i^*], kw_b^*})$
    If $b = b'$ then output $1$, and $0$ otherwise.
```

Here, $\mathcal{O} := \{\mathcal{O}_C(\mathsf{pk_R}, \cdot, \cdot), \mathcal{O}_T(\mathsf{pk_R}, \cdot, \mathsf{sk_R}, \cdot)\}$. $\mathcal{O}_C$ takes $kw \in \mathcal{KS}$ and $i \in [1, n]$ as input, and returns the result of $\mathsf{PAEKS.Enc}(\mathsf{pk_R}, \mathsf{pk}_{\mathsf{S}[i]}, \mathsf{sk}_{\mathsf{S}[i]}, kw)$. Here, $(kw, i) \notin \{(kw_0^*, i^*), (kw_1^*, i^*)\}$. $\mathcal{O}_T$ takes $kw' \in \mathcal{KS}$ and $i \in [1, n]$ as input, and returns the result of $\mathsf{PAEKS.Trapdoor}(\mathsf{pk_R}, \mathsf{pk}_{\mathsf{S}[i]}, \mathsf{sk_R}, kw')$. Here $(kw', i) \notin \{(kw_0^*, i^*), (kw_1^*, i^*)\}$. We say that a PAEKS scheme PAEKS is IND-IKGA secure if the advantage

$$\mathsf{Adv}^{\mathsf{IND\text{-}IKGA}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda, n) := \Pr[\mathsf{Exp}^{\mathsf{IND\text{-}IKGA}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda, n) = 1]$$

is negligible in the security parameter λ.

2.2 0/1 Encodings

Here, we introduce 0/1 encodings [35]. Let $t \in \mathbb{N}$ be a ℓ-bit positive integer, and its binary representation is denoted $t = t_\ell t_{\ell-1} \cdots t_1$ where $t_i \in \{0, 1\}$ for all $i \in [1, \ell]$. The 0-encoding algorithm takes ℓ and t as input, and outputs a set of strings S_t^0 defined as follows.

$$S_t^0 = \{t_\ell t_{\ell-1} \cdots t_{i+1} 1 \mid t_i = 0, i \in [1, \ell]\}$$

We denote $S_t^0 = \{s_{t,1}^0, s_{t,2}^0, \ldots, s_{t,\ell_t^0}^0\}$ where ℓ_t^0 is the number of strings contained in S_t^0 and is at most $O(\log t) = O(\ell)$. Similarly, the 1-encoding algorithm takes ℓ and t as input and outputs a set of strings S_t^1 defined as follows.

$$S_t^1 = \{t_\ell t_{\ell-1} \cdots t_i \mid t_i = 1, i \in [1, \ell_{t'}^1]\}$$

We denote $S_t^1 = \{s_{t,1}^1, s_{t,2}^1, \ldots, s_{t,\ell_t^1}^1\}$ where ℓ_t^1 is the number of strings contained in S_t^1 and is at most $O(\log t) = O(n)$. As an example, $\ell = 4$, $t = 7$ and $t = 12$ define $S_7^0 = \{1\}$, $S_7^1 = \{01, 011, 0111\}$, $S_{12}^0 = \{111, 1101\}$, and $S_{12}^1 = \{1, 11\}$, since $7_{(10)} = (0111)_{(2)}$ and $12_{(10)} = (1100)_{(2)}$. We remark that "1" and "01" are different strings. The encodings are effective to compare two integer values, t and t', because the following holds.

$$S_t^0 \cap S_{t'}^1 \neq \emptyset \iff t < t'$$

In other word, the encodings are effective to translate an inequality condition $t < t'$ to an equality condition, i.e., for all $s_{t,i}^0 \in S_t^0$ and $s_{t',j}^1 \in S_{t'}^1$, check whether $s_{t,i}^0 = s_{t',j}^1$ or not where $i \in [1, \ell_t^0]$ and $j \in [1, \ell_{t'}^1]$. The number of equality checks is at most $\ell_t^0 \cdot \ell_{t'}^1 = O(\log t \cdot \log t') = O(\ell^2)$. Previous FS-PAEKS [24] and FS-PEKS [30, 49] also employed the encodings. Moreover, group signatures with time-bound keys [36] also employed these encodings.

3 Definition of FS-PAEKS

In this section, we define FS-PAEKS. The encryption algorithm takes a time period t and the trapdoor generation algorithm takes a time period t' (in addition to other inputs required in the syntax of PAEKS). In addition to the search condition defined in PAEKS, a trapdoor works if $t < t'$, that is, a newly generated ciphertext is not allowed to be searched by previously generated trapdoors.

Definition 6 (Syntax of FS-PAEKS)**.** *An FS-PAEKS scheme* $\mathsf{FS\text{-}PAEKS}$ *consists of the following six algorithms* ($\mathsf{FS\text{-}PAEKS.Setup}, \mathsf{FS\text{-}PAEKS.KG_R}, \mathsf{FS\text{-}PAEKS.KG_S}, \mathsf{FS\text{-}PAEKS.Enc}, \mathsf{FS\text{-}PAEKS.Trapdoor}$, $\mathsf{FS\text{-}PAEKS.Test}$) *defined as follows.*

$\mathsf{FS\text{-}PAEKS.Setup}$**:** *The setup algorithm takes a security parameter λ as input, and outputs a common parameter* pp*. We assume that* pp *implicitly contains the keyword space $\mathcal{KS}$ and the time space $\mathcal{T}$.*

$\mathsf{FS\text{-}PAEKS.KG_R}$**:** *The receiver key generation algorithm takes* pp *as input, and outputs a public key* $\mathsf{pk_R}$ *and a secret key* $\mathsf{sk_R}$*.*

$\mathsf{FS\text{-}PAEKS.KG_S}$**:** *The sender key generation algorithm takes* pp *as input, and outputs a public key* $\mathsf{pk_S}$ *and a secret key* $\mathsf{sk_S}$*.*

$\mathsf{FS\text{-}PAEKS.Enc}$**:** *The keyword encryption algorithm takes* $\mathsf{pk_R}$*,* $\mathsf{pk_S}$*,* $\mathsf{sk_S}$*, a keyword $kw \in \mathcal{KS}$, and a time period $t \in \mathcal{T}$ as input, and outputs a ciphertext* $\mathsf{ct_{FS\text{-}PAEKS}}$*.*

$\mathsf{FS\text{-}PAEKS.Trapdoor}$**:** *The trapdoor algorithm takes* $\mathsf{pk_R}$*,* $\mathsf{pk_S}$*,* $\mathsf{sk_R}$*, a keyword $kw' \in \mathcal{KS}$, and a time period $t' \in \mathcal{T}$ as input, and outputs a trapdoor* $\mathsf{td}_{\mathsf{S},kw',t'}$*.*

$\mathsf{FS\text{-}PAEKS.Test}$**:** *The test algorithm takes* $\mathsf{ct_{PAEKS}}$ *and* $\mathsf{td}_{\mathsf{S},kw',t'}$ *as input, and outputs 1 or 0.*

Definition 7 (Correctness)**.** *For any security parameter λ, any common parameter* $\mathsf{pp} \leftarrow \mathsf{FS\text{-}PAEKS.Setup}(1^\lambda)$*, any key pairs* $(\mathsf{pk_R}, \mathsf{sk_R}) \leftarrow \mathsf{FS\text{-}PAEKS.KG_R}(\mathsf{pp})$ *and* $(\mathsf{pk_S}, \mathsf{sk_S}) \leftarrow \mathsf{FS\text{-}PAEKS.KG_S}(\mathsf{pp})$*, and any keyword $kw \in \mathcal{KS}$ and any time periods $t', t \in \mathcal{T}$ where $t < t'$, let* $\mathsf{ct_{FS\text{-}PAEKS}} \leftarrow \mathsf{FS\text{-}PAEKS.Enc}(\mathsf{pk_R}, \mathsf{pk_S}, \mathsf{sk_S}, kw, t)$ *and* $\mathsf{td}_{\mathsf{S},kw,t'} \leftarrow \mathsf{FS\text{-}PAEKS.Trapdoor}(\mathsf{pk_R}, \mathsf{pk_S}, \mathsf{sk_R}, kw, t')$*. Then*

$$\Pr[\mathsf{FS\text{-}PAEKS.Test}(\mathsf{ct_{FS\text{-}PAEKS}}, \mathsf{td}_{\mathsf{S},kw,t'}) = 1] = 1 - \mathsf{negl}(\lambda)$$

holds.

Next, we define consistency. As in PAEKS, due to its authenticity, a trapdoor associated with a sender should not work against ciphertexts generated by the secret key of another sender, even if the same keyword is associated. In addition, due to the forward security, a newly generated ciphertext should not be searchable by previously generated trapdoors, even if the same keyword and legitimate sender public key are specified. Thus, we add the condition $(kw, i) = (kw', j) \wedge t > t'$ below.

Definition 8 (Computational Consistency)**.** *For all PPT adversaries $\mathcal{A}$, we define the following* *experiment.*

$$
\begin{aligned}
&\mathsf{Exp}^{\mathsf{consist}}_{\mathsf{FS\text{-}PAEKS},\mathcal{A}}(\lambda):\\
&\quad \mathsf{pp} \leftarrow \mathsf{FS\text{-}PAEKS.Setup}(1^\lambda);\ (\mathsf{pk_R}, \mathsf{sk_R}) \leftarrow \mathsf{FS\text{-}PAEKS.KG_R}(\mathsf{pp})\\
&\quad (\mathsf{pk}_{\mathsf{S}[0]}, \mathsf{sk}_{\mathsf{S}[0]}) \leftarrow \mathsf{FS\text{-}PAEKS.KG_S}(\mathsf{pp});\ (\mathsf{pk}_{\mathsf{S}[1]}, \mathsf{sk}_{\mathsf{S}[1]}) \leftarrow \mathsf{FS\text{-}PAEKS.KG_S}(\mathsf{pp})\\
&\quad (kw, kw', t, t', i, j) \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{pk_R}, \mathsf{pk}_{\mathsf{S}[0]}, \mathsf{pk}_{\mathsf{S}[1]})\\
&\qquad s.t.\ kw, kw' \in \mathcal{KS} \wedge i, j \in \{0,1\} \wedge t, t' \in \mathcal{T}\\
&\qquad \wedge \big((kw, i) \neq (kw', j) \vee \big((kw, i) = (kw', j) \wedge t > t'\big)\big)\\
&\quad \mathsf{ct}_{\mathsf{FS\text{-}PAEKS}} \leftarrow \mathsf{FS\text{-}PAEKS.Enc}(\mathsf{pk_R}, \mathsf{pk}_{\mathsf{S}[i]}, \mathsf{sk}_{\mathsf{S}[i]}, kw, t)\\
&\quad \mathsf{td}_{\mathsf{S}[j],kw',t'} \leftarrow \mathsf{FS\text{-}PAEKS.Trapdoor}(\mathsf{pk_R}, \mathsf{pk}_{\mathsf{S}[j]}, \mathsf{sk_R}, kw', t')\\
&\quad \textit{If } \mathsf{FS\text{-}PAEKS.Test}(\mathsf{ct}_{\mathsf{FS\text{-}PAEKS}}, \mathsf{td}_{\mathsf{S}[j],kw',t'}) = 1 \textit{ then output } 1,\ \textit{and } 0 \textit{ otherwise.}
\end{aligned}
$$

We say that an FS-PAEKS scheme FS-PAEKS *is consistent if the advantage*

$$\mathsf{Adv}^{\mathsf{consist}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{consist}}_{\mathsf{FS\text{-}PAEKS},\mathcal{A}}(\lambda) = 1]$$

is negligible in the security parameter λ.

Next, we define indistinguishability against the chosen keyword attack with forward security (IND-FS-CKA) which guarantees that no information about the keyword is leaked from ciphertexts. Due to the forward security, an adversary $\mathcal{A}$ is allowed to obtain trapdoors for the challenge keyword and the challenge sender if the trapdoor is generated at $t' < t^*$ where the challenge ciphertext is generated at t^*. Thus, we add the condition $(kw', i) \in \{(kw_0^*, i^*), (kw_1^*, i^*)\} \wedge t' < t^*$ to the $\mathcal{O}_T$ oracle. We also remark that Jiang et al. [24] introduced selective forward security where an adversary declares t^* prior to the setup phase. We consider adaptive security where an adversary declares t^* in the challenge phase.[3]

Definition 9 (IND-FS-CKA)**.** *For all PPT adversaries $\mathcal{A}$, we define the following experiment.*

$$
\begin{aligned}
&\mathsf{Exp}^{\mathsf{IND\text{-}FS\text{-}CKA}}_{\mathsf{FS\text{-}PAEKS},\mathcal{A}}(\lambda, n):\\
&\quad \mathsf{pp} \leftarrow \mathsf{FS\text{-}PAEKS.Setup}(1^\lambda);\ (\mathsf{pk_R}, \mathsf{sk_R}) \leftarrow \mathsf{FS\text{-}PAEKS.KG_R}(\mathsf{pp})\\
&\quad \textit{For } i \in [1,n],\ (\mathsf{pk}_{\mathsf{S}[i]}, \mathsf{sk}_{\mathsf{S}[i]}) \leftarrow \mathsf{FS\text{-}PAEKS.KG_S}(\mathsf{pp})\\
&\quad (kw_0^*, kw_1^*, i^*, t^*, \mathsf{state}) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pp}, \mathsf{pk_R}, \{\mathsf{pk}_{\mathsf{S}[i]}\}_{i\in[1,n]})\\
&\qquad s.t.\ kw_0^*, kw_1^* \in \mathcal{KS} \wedge\ kw_0^* \neq kw_1^* \wedge i^* \in [1,n] \wedge t^* \in \mathcal{T}\\
&\quad b \xleftarrow{\$} \{0,1\};\ \mathsf{ct}^*_{\mathsf{FS\text{-}PAEKS}} \leftarrow \mathsf{FS\text{-}PAEKS.Enc}(\mathsf{pk_R}, \mathsf{pk}_{\mathsf{S}[i^*]}, \mathsf{sk}_{\mathsf{S}[i^*]}, kw_b^*, t^*)\\
&\quad b' \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{state}, \mathsf{ct}^*_{\mathsf{FS\text{-}PAEKS}})\\
&\quad \textit{If } b = b' \textit{ then output } 1,\ \textit{and } 0 \textit{ otherwise.}
\end{aligned}
$$

Here, $\mathcal{O} := \{\mathcal{O}_C(\mathsf{pk_R}, \cdot, \cdot, \cdot), \mathcal{O}_T(\mathsf{pk_R}, \cdot, \mathsf{sk_R}, \cdot, \cdot)\}$. $\mathcal{O}_C$ takes $kw \in \mathcal{KS}$, $t \in \mathcal{T}$, and $i \in [1,n]$ as input, and returns the result of $\mathsf{FS\text{-}PAEKS.Enc}(\mathsf{pk_R}, \mathsf{pk}_{\mathsf{S}[i]}, \mathsf{sk}_{\mathsf{S}[i]}, kw, t)$. *Here, there is no restriction. $\mathcal{O}_T$ takes $kw' \in \mathcal{KS}$, $t' \in \mathcal{T}$, and $i \in [1,n]$ as input, and returns the result of* $\mathsf{FS\text{-}PAEKS.Trapdoor}(\mathsf{pk_R}, \mathsf{pk}_{\mathsf{S}[i]}, \mathsf{sk_R}, kw', t')$. *Here $(kw', i) \notin \{(kw_0^*, i^*), (kw_1^*, i^*)\}$ or $(kw', i) \in \{(kw_0^*, i^*), (kw_1^*, i^*)\} \wedge t' < t^*$. We say that an FS-PAEKS scheme* FS-PAEKS *is IND-FS-CKA secure if the advantage*

$$\mathsf{Adv}^{\mathsf{IND\text{-}FS\text{-}CKA}}_{\mathsf{FS\text{-}PAEKS},\mathcal{A}}(\lambda, n) := \Pr[\mathsf{Exp}^{\mathsf{IND\text{-}FS\text{-}CKA}}_{\mathsf{FS\text{-}PAEKS},\mathcal{A}}(\lambda, n) = 1]$$

[3] They are equivalent to $|\mathcal{T}|$ reduction and selective forward security is sufficient if $|\mathcal{T}|$ is a polynomial of the security parameter.

is negligible in the security parameter λ.

Next, we define indistinguishability against the inside keyword guessing attack with forward security (IND-FS-IKGA) which guarantees that no information about the keyword is leaked from trapdoors. Due to the forward security, an adversary $\mathcal{A}$ is allowed to obtain ciphertexts for the challenge keyword and the challenge sender if the ciphertext is generated at $t > t^*$ where the challenge trapdoor is generated at t^*. Thus, we add the condition $(kw, i) \in \{(kw_0^*, i^*), (kw_1^*, i^*)\} \wedge t > t^*$ to the $\mathcal{O}_C$ oracle. As in IND-FS-CKA, we consider adaptive security where an adversary declares t^* in the challenge phase, although Jiang et al. [24] introduced selective forward security.

Definition 10 (IND-FS-IKGA)**.** *For all PPT adversaries $\mathcal{A}$, we define the following experiment.*

$$\begin{aligned}
&\mathsf{Exp}^{\mathsf{IND\text{-}FS\text{-}IKGA}}_{\mathsf{FS\text{-}PAEKS},\mathcal{A}}(\lambda, n):\\
&\quad \mathsf{pp} \leftarrow \mathsf{FS\text{-}PAEKS.Setup}(1^\lambda);\ (\mathsf{pk_R}, \mathsf{sk_R}) \leftarrow \mathsf{FS\text{-}PAEKS.KG_R}(\mathsf{pp})\\
&\quad \textit{For } i \in [1, n],\ (\mathsf{pk}_{\mathsf{S}[i]}, \mathsf{sk}_{\mathsf{S}[i]}) \leftarrow \mathsf{FS\text{-}PAEKS.KG_S}(\mathsf{pp})\\
&\quad (kw_0^*, kw_1^*, i^*, t^*, \mathsf{state}) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pp}, \mathsf{pk_R}, \{\mathsf{pk}_{\mathsf{S}[i]}\}_{i\in[1,n]})\\
&\qquad \textit{s.t. } kw_0^*, kw_1^* \in \mathcal{KS} \wedge\ kw_0^* \neq kw_1^* \wedge i^* \in [1, n] \wedge t^* \in \mathcal{T}\\
&\quad b \xleftarrow{\$} \{0, 1\};\ \mathsf{td}^*_{\mathsf{S}[i^*], kw_b^*, t^*} \leftarrow \mathsf{FS\text{-}PAEKS.Trapdoor}(\mathsf{pk_R}, \mathsf{pk}_{\mathsf{S}[i^*]}, \mathsf{sk_R}, kw_b^*, t^*)\\
&\quad b' \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{state}, \mathsf{td}^*_{\mathsf{S}[i^*], kw_b^*, t^*})\\
&\quad \textit{If } b = b' \textit{ then output } 1,\ \textit{and } 0\ \textit{otherwise.}
\end{aligned}$$

Here, $\mathcal{O} := \{\mathcal{O}_C(\mathsf{pk_R}, \cdot, \cdot, \cdot), \mathcal{O}_T(\mathsf{pk_R}, \cdot, \mathsf{sk_R}, \cdot, \cdot)\}$*.* $\mathcal{O}_C$ *takes* $kw \in \mathcal{KS}$*,* $t \in \mathcal{T}$*, and* $i \in [1, n]$ *as input, and returns the result of* $\mathsf{FS\text{-}PAEKS.Enc}(\mathsf{pk_R}, \mathsf{pk}_{\mathsf{S}[i]}, \mathsf{sk}_{\mathsf{S}[i]}, kw, t)$*. Here,* $(kw, i) \notin \{(kw_0^*, i^*), (kw_1^*, i^*)\}$ *or* $(kw, i) \in \{(kw_0^*, i^*), (kw_1^*, i^*)\} \wedge t > t^*$*.* $\mathcal{O}_T$ *takes* $kw' \in \mathcal{KS}$*,* $t' \in \mathcal{T}$*, and* $i \in [1, n]$ *as input, and returns the result of* $\mathsf{FS\text{-}PAEKS.Trapdoor}(\mathsf{pk_R}, \mathsf{pk}_{\mathsf{S}[i]}, \mathsf{sk_R}, kw', t')$*. Here* $(kw', i) \notin \{(kw_0^*, i^*), (kw_1^*, i^*)\}$*. We say that an FS-PAEKS scheme* FS-PAEKS *is IND-FS-IKGA secure if the advantage*

$$\mathsf{Adv}^{\mathsf{IND\text{-}FS\text{-}IKGA}}_{\mathsf{FS\text{-}PAEKS},\mathcal{A}}(\lambda, n) := \Pr[\mathsf{Exp}^{\mathsf{IND\text{-}FS\text{-}IKGA}}_{\mathsf{FS\text{-}PAEKS},\mathcal{A}}(\lambda, n) = 1]$$

is negligible in the security parameter λ.

4 Our Generic Construction

Trivial and Insecure Construction. One trivial construction is to employ a double encryption method. That is, a PAEKS ciphertext is encrypted by a public key encryption scheme supporting time-related functionality, e.g., past time-specific encryption (PTSE) [26,27] which is a special case of time-specific encryption [40]. In PTSE, the encryption and key extraction algorithms take a time t and t' as input, respectively, and the decryption key works when $t < t'$. Thus, a PAEKS ciphertext of a keyword kw is encrypted by PTSE with a time period t, and a trapdoor is a PAEKS trapdoor of a keyword kw' and a PTSE decryption key associated with time period t'. If $t < t'$, then a PTSE ciphertext can be decrypted by the decryption key, and then the test algorithm of the underlying PAEKS scheme determines whether $kw = kw'$ or not using a PAEKS trapdoor. This construction provides correctness and appears to be secure because no information about the keyword is revealed from ciphertexts (owing to the IND-CPA security of PTSE) and trapdoors (owing to the IND-IKGA security of PAEKS). However, this construction does not provide the IND-FS-CKA security because the keyword-related and time-related parts of a trapdoor are generated separately. For example, an adversary obtains a trapdoor for the challenge keyword kw_0^* and a time period $t' < t^*$, and obtains a trapdoor for any keyword $kw \notin \{kw_0^*, kw_1^*\}$ and a time period t^*. Then, the adversary can generate a trapdoor for kw_0^* at t^* which works to distinguish whether the challenge ciphertext is an encryption of kw_0^* or kw_1^*. This insecure construction suggests that we connect the keyword-related and time-related parts in an inseparable manner, and this is the reason behind of our attack works against the Jiang et al. FS-PAEKS scheme.

High-Level Description. A naive way to connect the keyword-related and time-related parts in an inseparable manner is to consider $kw||t$ for encryption and $kw'||t'$ for trapdoor as keywords. However, this construction only provides the equality matching, and does not check the inequality condition $t < t'$. Thus, we employ 0/1 encodings to translate the inequality condition $t < t'$ to an equality condition. Essentially, a ciphertext of FS-PAEKS for a keyword kw and a time period t is a set of PAEKS ciphertexts for the keyword $kw||s_{t,i}^0$ for all $s_{t,i}^0 \in S_t^0$. Similarly, a trapdoor of FS-PAEKS for a keyword kw' and a time period t' is a set of PAEKS trapdoors for the keyword $kw'||s_{t',j}^1$ for all $s_{t',j}^1 \in S_{t'}^1$. $t < t'$ holds if and only if there exists i and j such that $s_{t,i}^0 = s_{t',j}^1$ since $S_t^0 \cap S_{t'}^1 \neq \emptyset$. For such i and j, $kw||s_{t,i}^0 = kw'||s_{t',j}^1$ holds if $kw = kw'$. Thus, by using the test algorithm of the underlying PAEKS scheme, we can check both $t < t'$ and $kw = kw'$ simultaneously. Thus, obviously correctness holds. For consistency, let i and j be selected by the adversary $\mathcal{A}$ in $\mathsf{Exp}^{\mathsf{consist}}_{\mathsf{FS\text{-}PAEKS},\mathcal{A}}$. When $(kw, i) \neq (kw', j)$, our construction provides consistency since the underlying PAEKS scheme is consistent. When $(kw, i) = (kw', j) \wedge t > t'$, since $S_t^0 \cap S_{t'}^1 = \emptyset$, this case is reduced to the case $kw||s_{t,i}^0 \neq kw'||s_{t',j}^1 \wedge i = j$ but the test algorithm outputs 1. Since this contradicts the consistency of the underlying PAEKS scheme, our construction provides consistency. Moreover, intuitively, no information about the keyword is revealed from ciphertexts and trapdoors due to the IND-CKA security and IND-IKGA security of the underlying PAEKS scheme. The size of $\mathsf{ct}_{\mathsf{FS\text{-}PAEKS}}$ (resp. $\mathsf{td}_{\mathsf{S},kw',t'}$) is ℓ_t^0-times (resp. $\ell_{t'}^1$-times) greater than that of $\mathsf{ct}_{\mathsf{PAEKS}}$ (resp. $\mathsf{td}_{\mathsf{S},kw'}$). Since ℓ_t^0 and $\ell_{t'}^1$ are at most the bit length of time period, our construction is scalable. We remark that information of time period could be leaked unless information of keyword is not leaked. Thus, the $\mathsf{FS\text{-}PAEKS.Test}$ algorithm needs to run the $\mathsf{PAEKS.Test}$ algorithm only once by finding i and j such that $s_{t,i}^0 = s_{t',j}^1$. This technique is also employed in the FS-PEKS scheme proposed by Zeng et al. [49].

As a remaining issue, we must consider the following trapdoor/ciphertext re-use cases. For example, $S_7^1 = \{01, 011, 0111\}$ contains $S_6^1 = \{01, 011\}$. That is, $\mathcal{A}$ can obtain a trapdoor at $t' = 6$ when $\mathcal{A}$ obtains a trapdoor at $t' = 7$. However, this trapdoor derivation for previous time period does not affect the IND-FS-CKA security because $\mathcal{A}$ is allowed to obtain trapdoors for a challenge keyword and sender $(kw', i) \in \{(kw_0^*, i^*), (kw_1^*, i^*)\}$ only when the trapdoors are associated with a previous time period $t' < t^*$. That is, if other trapdoor is derived from the trapdoors for a challenge keyword and sender, it does not work for distinguishing which keyword is selected for generating the challenge ciphertext. Towards this direct trapdoor derivation case, we need to guarantee that any combination of trapdoors obtained via the trapdoor oracle does not affect the IND-FS-CKA security. This can be shown by the fact that $t > t'$ if and only if $S_t^0 \cap S_{t'}^1 = \emptyset$. Similarly, $\mathcal{A}$ may obtain ciphertexts associated to a future time period. For example, $S_8^0 = \{11, 101, 1001\}$ contains $S_9^0 = \{11, 101\}$. That is, $\mathcal{A}$ can obtain a ciphertext at $t = 9$ when $\mathcal{A}$ obtains a ciphertext at $t = 8$. However, this situation also does not affect the IND-FS-IKGA security because $\mathcal{A}$ is allowed to obtain ciphertexts for the challenge keyword and sender $(kw, i) \in \{(kw_0^*, i^*), (kw_1^*, i^*)\}$ only when the ciphertexts are associated with a future time period $t > t^*$. That is, if other ciphertext is derived from the ciphertexts for a challenge keyword and sender, it does not work for distinguishing which keyword is selected for generating the challenge trapdoor. Towards this direct ciphertext derivation case, we need to guarantee that any combination of ciphertexts obtained via the encryption oracle does not affect the IND-FS-IKGA security. This can be shown by the fact that $t > t'$ if and only if $S^0_t \cap S^1_{t'} = \emptyset$.[4]

Let $\mathsf{PAEKS} = (\mathsf{PAEKS.Setup}, \mathsf{PAEKS.KG_R}, \mathsf{PAEKS.KG_S}, \mathsf{PAEKS.Enc}, \mathsf{PAEKS.Trapdoor}, \mathsf{PAEKS.Test})$ be a PAEKS scheme. We construct an FS-PAEKS scheme $\mathsf{FS\text{-}PAEKS} = (\mathsf{FS\text{-}PAEKS.Setup}, \mathsf{FS\text{-}PAEKS.KG_R}, \mathsf{FS\text{-}PAEKS.KG_S}, \mathsf{FS\text{-}PAEKS.Enc}, \mathsf{FS\text{-}PAEKS.Trapdoor}, \mathsf{FS\text{-}PAEKS.Test})$ from PAEKS as follows. We assume that the underlying PAEKS scheme supports the keyword space $\{0,1\}^{2\ell}$ where ℓ is a polynomial of λ. Then, our construction supports $\mathcal{KS} = \mathcal{T} = \{0,1\}^\ell$ because we consider $kw||s^0_{t,i}$ or $kw'||s^1_{t',j}$ as keyword.

Generic Construction of FS-PAEKS

$\mathsf{FS\text{-}PAEKS.Setup}(1^\lambda)$**:** Run $\mathsf{pp} \leftarrow \mathsf{PAEKS.Setup}(1^\lambda)$ and output pp that contains $\mathcal{KS} = \{0,1\}^\ell$ and $\mathcal{T} = \{0,1\}^\ell$ where ℓ is a polynomial of λ.

$\mathsf{FS\text{-}PAEKS.KG_R}(\mathsf{pp})$**:** Run $(\mathsf{pk_R}, \mathsf{sk_R}) \leftarrow \mathsf{PAEKS.KG_R}(\mathsf{pp})$ and output $(\mathsf{pk_R}, \mathsf{sk_R})$.

$\mathsf{FS\text{-}PAEKS.KG_S}(\mathsf{pp})$**:** Run $(\mathsf{pk_S}, \mathsf{sk_S}) \leftarrow \mathsf{PAEKS.KG_S}(\mathsf{pp})$ and output $(\mathsf{pk_S}, \mathsf{sk_S})$.

$\mathsf{FS\text{-}PAEKS.Enc}(\mathsf{pk_R}, \mathsf{pk_S}, \mathsf{sk_S}, kw, t)$**:** Define $S^0_t = \{s^0_{t,1}, s^0_{t,2}, \dots, s^0_{t,\ell^0_t}\}$. For all $i \in [1, \ell^0_t]$, run $\mathsf{ct}_{\mathsf{PAEKS}i} \leftarrow \mathsf{PAEKS.Enc}(\mathsf{pk_R}, \mathsf{pk_S}, \mathsf{sk_S}, kw||s^0_{t,i})$. Output $\mathsf{ct}_{\mathsf{FS\text{-}PAEKS}} = (t, \{\mathsf{ct}_{\mathsf{PAEKS}i}\}_{i\in[1,\ell^0_t]})$.

$\mathsf{FS\text{-}PAEKS.Trapdoor}(\mathsf{pk_R}, \mathsf{pk_S}, \mathsf{sk_R}, kw', t')$**:** Define $S^1_{t'} = \{s^1_{t',1}, s^1_{t',2}, \dots, s^1_{t',\ell^1_{t'}}\}$. For all $j \in [1, \ell^1_{t'}]$, run $\mathsf{td}_{\mathsf{S},kw'||s^1_{t',j}} \leftarrow \mathsf{PAEKS.Trapdoor}(\mathsf{pk_R}, \mathsf{pk_S}, \mathsf{sk_R}, kw'||s^1_{t',j})$. Output $\mathsf{td}_{\mathsf{S},kw',t'} = (t', \{\mathsf{td}_{\mathsf{S},kw'||s^1_{t',j}}\}_{j\in[1,\ell^1_{t'}]})$.

$\mathsf{FS\text{-}PAEKS.Test}(\mathsf{ct}_{\mathsf{FS\text{-}PAEKS}}, \mathsf{td}_{\mathsf{S},kw',t'})$**:** Parse $\mathsf{ct}_{\mathsf{FS\text{-}PAEKS}} = (t, \{\mathsf{ct}_{\mathsf{PAEKS}i}\}_{i\in[1,\ell^0_t]})$ and $\mathsf{td}_{\mathsf{S},kw,t'} = (t', \{\mathsf{td}_{\mathsf{S},kw'||s^1_{t',j}}\}_{j\in[1,\ell^1_{t'}]})$. If $t > t'$, then output 0. Otherwise, if $t < t'$, then find i and j such that $s^0_{t,i} = s^1_{t',j}$. If $1 = \mathsf{PAEKS.Test}(\mathsf{ct}_{\mathsf{PAEKS}i}, \mathsf{td}_{\mathsf{S},kw'||s^1_{t',j}})$, then output 1, and 0 otherwise.

As mentioned in the high-level description paragraph, our generic construction is correct if the underlying PAEKS scheme is correct, due to 0/1 encodings.

5 Security Analysis

Theorem 1. *Our generic construction is consistent if the underlying PAEKS scheme is consistent.*

Proof. Let i and j be chosen by the adversary $\mathcal{A}$ in $\mathsf{Exp}^{\mathsf{consist}}_{\mathsf{FS\text{-}PAEKS},\mathcal{A}}$. If $(kw, i) \neq (kw', j)$, then obviously consistency holds due to the consistency of the underlying PAEKS scheme because the winning conditions of the both experiments are the same. Thus, we consider the case $(kw, i) = (kw', j) \wedge t > t'$ as follows. Let $\mathcal{A}$ be the adversary of FS-PAEKS consistency and $\mathcal{C}$ be the challenger of PAEKS consistency. We construct an algorithm $\mathcal{B}$ that breaks the consistency of the PAEKS scheme as follows. First, $\mathcal{C}$ sends $(\mathsf{pp}, \mathsf{pk_R}, \mathsf{pk_{S[0]}}, \mathsf{pk_{S[1]}})$ to $\mathcal{B}$. $\mathcal{B}$ forwards $(\mathsf{pp}, \mathsf{pk_R}, \mathsf{pk_{S[0]}}, \mathsf{pk_{S[1]}})$ to $\mathcal{A}$. $\mathcal{A}$ declares (kw, kw', t, t', i, j) where $(kw, i) = (kw', j) \wedge t > t'$. $\mathcal{B}$ defines $S^0_t = \{s^0_{t,1}, s^0_{t,2}, \dots, s^0_{t,\ell^0_t}\}$ and $S^1_{t'} = \{s^1_{t',1}, s^1_{t',2}, \dots, s^1_{t',\ell^1_{t'}}\}$. Since $t > t'$, $S^0_t \cap S^1_{t'} = \emptyset$. Now, $\mathsf{FS\text{-}PAEKS.Test}(\mathsf{ct}_{\mathsf{FS\text{-}PAEKS}},$

[4] Although the trapdoor/ciphertext derivation does not affect IND-FS-CKA/IND-FS-IKGA security, the delegatability violates unforgeability of the time period where a trapdoor (resp. ciphertext) associated with a time period is delegated to a trapdoor (resp. ciphertext) associated to a previous (resp. future) time period. Because such unforgeability is not required as a security of FS-PAEKS, we do not consider the derivation anymore. We remark that, in the group signatures with time-bound keys context, such unforgeability is considered [18, 45]. It might be interesting to consider such unforgeability in the FS-P(A)EKS context.

$\mathsf{td}_{\mathsf{S}[j],kw',t'}) = 1$ holds where $\mathsf{ct}_{\mathsf{FS\text{-}PAEKS}} \leftarrow \mathsf{FS\text{-}PAEKS.Enc}(\mathsf{pk}_\mathsf{R}, \mathsf{pk}_{\mathsf{S}[i]}, \mathsf{sk}_{\mathsf{S}[i]}, kw, t)$ and $\mathsf{td}_{\mathsf{S}[j],kw',t'} \leftarrow \mathsf{FS\text{-}PAEKS.Trapdoor}(\mathsf{pk}_\mathsf{R}, \mathsf{pk}_{\mathsf{S}[j]}, \mathsf{sk}_\mathsf{R}, kw', t')$ since $\mathcal{A}$ breaks the consistency. Thus, there exist $i^* \in [1, \ell_t^0]$ and $j^* \in [1, \ell_{t'}^1]$ such that $1 = \mathsf{PAEKS.Test}(\mathsf{ct}_{\mathsf{PAEKS}i^*}, \mathsf{td}_{\mathsf{S}[j],kw'||s^1_{t',j^*}})$ and $kw||s^0_{t,i^*} \neq kw'||s^1_{t',j^*}$ hold. $\mathcal{B}$ randomly guesses such i^* and j^* and sends $(kw||s^0_{t,i^*}, kw'||s^1_{t',j^*}, i, j)$ to $\mathcal{C}$. If the guess is correct (with probability of at least $1/(\ell_t^0 \ell_{t'}^1$ which is non-negligible)), $\mathcal{B}$ breaks the consistency of the underlying PAEKS scheme. This concludes the proof. □

Theorem 2. *Our generic construction is IND-FS-CKA secure if the underlying PAEKS scheme is IND-CKA secure.*

Proof. Let $\mathcal{A}$ be the adversary of IND-FS-CKA and $\mathcal{C}$ be the challenger of IND-CKA. We construct an algorithm $\mathcal{B}$ that breaks the IND-CKA security of the PAEKS scheme as follows. First, $\mathcal{C}$ sends $(\mathsf{pp}, \mathsf{pk}_\mathsf{R}, \{\mathsf{pk}_{\mathsf{S}[i]}\}_{i\in[1,n]})$ to $\mathcal{B}$. $\mathcal{B}$ forwards $(\mathsf{pp}, \mathsf{pk}_\mathsf{R}, \{\mathsf{pk}_{\mathsf{S}[i]}\}_{i\in[1,n]})$ to $\mathcal{A}$.

When $\mathcal{A}$ sends $kw \in \mathcal{KS}$, $t \in \mathcal{T}$, and $i \in [1, n]$ to $\mathcal{O}_C$, $\mathcal{B}$ defines $S_t^0 = \{s^0_{t,1}, s^0_{t,2}, \dots, s^0_{t,\ell_t^0}\}$. Then, for all $k \in [1, \ell_t^0]$, $\mathcal{B}$ sends $kw||s^0_{t,k}$ and i to $\mathcal{C}$ and obtains $\mathsf{ct}_{\mathsf{PAEKS}k}$. $\mathcal{B}$ returns $\mathsf{ct}_{\mathsf{FS\text{-}PAEKS}} = (t, \{\mathsf{ct}_{\mathsf{PAEKS}i}\}_{i\in[1,\ell_t^0]})$ to $\mathcal{A}$. Since there is no restriction, the simulation of $\mathcal{O}_C$ is perfect.

Similarly, when $\mathcal{A}$ sends $kw' \in \mathcal{KS}$, $t' \in \mathcal{T}$, and $i \in [1, n]$ to $\mathcal{O}_T$, $\mathcal{B}$ defines $S^1_{t'} = \{s^1_{t',1}, s^1_{t',2}, \dots, s^1_{t',\ell^1_{t'}}\}$. Then, for all $j \in [1, \ell^1_{t'}]$, $\mathcal{B}$ sends $kw'||s^1_{t',j}$ and i to $\mathcal{C}$ and obtains $\mathsf{td}_{\mathsf{S},kw'||s^1_{t',j}}$. $\mathcal{B}$ returns $\mathsf{td}_{\mathsf{S},kw',t'} = (t', \{\mathsf{td}_{\mathsf{S},kw'||s^1_{t',j}}\}_{j\in[1,\ell^1_{t'}]})$ to $\mathcal{A}$. Here, we need to guarantee that $\mathcal{B}$'s queries do not violate the condition of the $\mathcal{O}_T$ oracle in $\mathsf{Exp}^{\mathsf{IND\text{-}CKA}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda, n)$. In the case of $(kw', i) \notin \{(kw_0^*, i^*), (kw_1^*, i^*)\}$, the simulation is perfect because it does not violate the condition of the $\mathcal{O}_T$ oracle in $\mathsf{Exp}^{\mathsf{IND\text{-}CKA}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda, n)$. In the case of $(kw', i) \in \{(kw_0^*, i^*), (kw_1^*, i^*)\} \wedge t' < t^*$, $S^0_{t^*} \cap S^1_{t'} = \emptyset$. Thus, for all $i \in [1, \ell^0_{t^*}]$ and $j \in [1, \ell^1_{t'}]$, $kw'||s^1_{t',j} \notin \{kw_0^*||s^0_{t^*,i}, kw_1^*||s^0_{t^*,i}\}$ holds. Thus, this case also does not violate the condition of the $\mathcal{O}_T$ oracle in $\mathsf{Exp}^{\mathsf{IND\text{-}CKA}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda, n)$. To sum up, the simulation of $\mathcal{O}_T$ is perfect.

In the challenge phase, $\mathcal{A}$ declares $(kw_0^*, kw_1^*, i^*, t^*)$. $\mathcal{B}$ defines $S^0_{t^*} = \{s^0_{t^*,1}, s^0_{t^*,2}, \dots, s^0_{t,\ell^0_t}\}$. We define sequential of games $\mathsf{Game}_0, \dots, \mathsf{Game}_{\ell^0_{t^*}}$ as follows. In Game_0, the challenge ciphertext is generated by $\mathsf{PAEKS.Enc}(\mathsf{pk}_\mathsf{R}, \mathsf{pk}_{\mathsf{S}[i^*]}, \mathsf{sk}_{\mathsf{S}[i^*]}, kw_0^*||s^0_{t^*,i})$ for all $i = 1, \dots, \ell^0_{t^*}$. In $\mathsf{Game}_{\ell^0_{t^*}}$, the challenge ciphertext is generated by $\mathsf{PAEKS.Enc}(\mathsf{pk}_\mathsf{R}, \mathsf{pk}_{\mathsf{S}[i^*]}, \mathsf{sk}_{\mathsf{S}[i^*]}, kw_1^*||s^0_{t^*,i})$ for all $i = 1, \dots, \ell^0_{t^*}$. In Game_i where $i \in [1, \ell^0_{t^*} - 1]$, the j-th challenge ciphertext is generated by $\mathsf{PAEKS.Enc}(\mathsf{pk}_\mathsf{R}, \mathsf{pk}_{\mathsf{S}[i^*]}, \mathsf{sk}_{\mathsf{S}[i^*]}, kw_0^*||s^0_{t^*,j})$ for all $j = i+1, \dots, \ell^0_{t^*}$ and the k-th challenge ciphertext is generated by $\mathsf{PAEKS.Enc}$ $(\mathsf{pk}_\mathsf{R}, \mathsf{pk}_{\mathsf{S}[i^*]}, \mathsf{sk}_{\mathsf{S}[i^*]}, kw_1^*||s^0_{t^*,k})$ for all $k = 1, \dots, i$. Thus, the difference of the success probability between two neighbor games Game_i and Game_{i+1} are bound by $\mathsf{Adv}^{\mathsf{IND\text{-}CKA}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda, n)$. That is, the ciphertext generated by (kw_0^*, i^*, t^*) and the ciphertext generated by (kw_1^*, i^*, t^*) are indistinguishable with the advantage $\ell^0_{t^*} \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CKA}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda, n)$. This concludes the proof. □

Theorem 3. *Our generic construction is IND-FS-IKGA secure if the underlying PAEKS scheme is IND-IKGA secure.*

The proof of Theorem 3 is very similar to that of Theorem 2. The main difference is: the challenge trapdoor is generated by the $\mathsf{PAEKS.Trapdoor}$ algorithm, and the trapdoor generated by (kw_0^*, i^*, t^*) and the trapdoor generated by (kw_1^*, i^*, t^*) are indistinguishable with the advantage $\ell^1_{t^*} \cdot \mathsf{Adv}^{\mathsf{IND\text{-}IKGA}}_{\mathsf{PAEKS},\mathcal{A}}(\lambda, n)$. Thus, we omit the proof.

Remark. Due to our security proofs above, our construction inherits the security of the underlying PAEKS scheme. For example, several PAEKS schemes do not consider the case that a trapdoor associated with a sender does not work against ciphertexts generated by the secret key of another sender, even if the same keyword is associated. They just consider keywords, i.e., if $kw \neq kw'$

then the test algorithm outputs 0. Even this weaker notion is employed, our generic construction provides the same security level that the underlying PAEKS schemes provide. Similarly, if the underlying PAEKS scheme provides MCI/MTI security, then the FS-PAEKS scheme obtained via our generic construction also provides MCI/MTI security. In this sense, our generic construction can be instantiated by any previous PAEKS scheme.

6 Vulnerability of the Jiang et al. FS-PAEKS scheme

In this section, we show that the Jiang et al. FS-PAEKS scheme [24] does not provide forward security. As mentioned in the introduction section, the main problem is their pairing selection where a symmetric paring is employed but the DDH problem is assumed to be held (Theorem 4.2 in [24]). Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a paring where $\mathbb{G}$ and $\mathbb{G}_T$ have the prime order p, and let $g \in \mathbb{G}$ be a generator. For a DDH tuple (g, g^a, g^b, g^c), one can check whether $c = ab$ or not by checking $e(g^a, g^b) = e(g^c, g)$ holds or not. Thus, e can be seen as a DDH solver.

Although the Jiang et al. FS-PAEKS scheme provides conjunctive keyword search, for the sake of simplicity, we consider the single keyword case as follows (but our attack works for conjunctive keyword search). In their scheme, $\mathsf{pk_R} = g^\alpha$, $\mathsf{sk_R} = \alpha$, $\mathsf{pk_S} = g^\beta$, and $\mathsf{sk_S} = \beta$ where $\alpha, \beta \in \mathbb{Z}_p$. Briefly, a ciphertext contains $X = g^{r_1}$ and $\mathsf{CT} = h^{r_1} f^{r_2}$ where $r_1, r_2 \in \mathbb{Z}_p$. Here, h and f are related to the keyword kw to be encrypted and are defined as $h = H(kw, \mathsf{pk_R}^{\mathsf{sk_S}})$ and $f = H'(kw, \mathsf{pk_R}^{\mathsf{sk_S}})$ for some hash functions H and H'. That is, a Diffie-Hellman key $\mathsf{pk_R}^{\mathsf{sk_S}} = \mathsf{pk_S}^{\mathsf{sk_R}} = g^{\alpha\beta}$ is regarded as a key for deriving h and f. Moreover, the ciphertext contains $(R_0, \dots, R_\ell)$ where $R_i = \mathsf{pk_R}^{a_i}$ and a_i is a coefficient of a Lagrange polynomial for all $i \in [0, \ell]$ (here ℓ is the bit-length of a time period t) which is defined by points $(H''(s^0_{t,k}, \mathsf{pk_R}^{\mathsf{sk_S}}), r_2)$ for some hash function H'' and $s^0_{t,k} \in S^0_t$. That is, $\prod_{0 \le i \le \ell} R_i^{\pi^i_{(k)}} = \mathsf{pk_R}^{r_2}$ holds where $\pi_{(k)} := H''(s^0_{t,k}, \mathsf{pk_R}^{\mathsf{sk_S}})$ for any $s^0_{t,k} \in S^0_t$. A trapdoor contains $\pi_1 = g^s$, $\pi_2 = h'^s$, and $\pi_3 = f'^{s/\alpha}$. Here, h' and f' are related to the keyword kw' to be searched and are defined as $h' = H(kw', \mathsf{pk_S}^{\mathsf{sk_R}})$ and $f' = H'(kw', \mathsf{pk_S}^{\mathsf{sk_R}})$. If $kw = kw'$, then $h = h'$ and $f = f'$. Let a ciphertext be generated at t and a trapdoor be generated at t', and assume $t < t'$. Since $S^0_t \cap S^1_{t'} \neq \emptyset$, there exist i and j such that $s^0_{t,i} = s^1_{t',j}$. Because the Lagrange polynomial is defined by points $(H''(s^0_{t,k}, \mathsf{pk_R}^{\mathsf{sk_S}}), r_2)$, $\mu := \prod_{0 \le i \le \ell} R_i^{\pi^i_{(j)}} = \mathsf{pk_R}^{r_2}$ holds where $\pi_{(j)} = H''(s^1_{t',j}, \mathsf{pk_R}^{\mathsf{sk_S}})$ since $(\pi_{(j)}, r_2)$ is a point on the polynomial. The trapdoor contains $\pi_{(j)}$.

Our attack is described as follows. We distinguish whether the challenge trapdoor generated at t^* is for kw^*_0 or kw^*_1. One observation here is that the value CT is related to a keyword, and is independent to a time period, and the values $(R_0, \dots, R_\ell)$ are related to a time period, and are independent to a keyword. Thus, there is room for combining CT for the challenge keyword and $(R_0, \dots, R_\ell)$ for the challenge time period, and our attack below instantiates this observation.

1. An adversary $\mathcal{A}$ issues an encryption query kw^*_0 and $t^* < t$ where kw^*_0 is a challenge keyword. Since a newly generated ciphertext is not allowed to be searched by previously generated trapdoors, this query is not prohibited in the security model. The ciphertext contains $X = g^{r_1}$, $\mathsf{CT} = h^{*r_1}_0 f^{*r_2}_0$, and $(R_0, \dots, R_\ell)$ where $h^*_0 = H(kw^*_0, \mathsf{pk_R}^{\mathsf{sk_S}})$ and $f^*_0 = H'(kw^*_0, \mathsf{pk_R}^{\mathsf{sk_S}})$.

2. $\mathcal{A}$ issues a trapdoor query $kw' \notin \{kw^*_0, kw^*_1\}$ and t' where $t < t'$. Since $kw' \notin \{kw^*_0, kw^*_1\}$, this query is also not prohibited in the security model. Of course, the test algorithm with the ciphertext and the trapdoor outputs 0. However, the trapdoor contains π such that $\mu := \prod_{0 \le i \le \ell} R_i^{\pi^i} = \mathsf{pk_R}^{r_2}$ holds since $t < t'$.

Through the procedure, $\mathcal{A}$ can obtain X, CT, and μ which are used later.

When $\mathcal{A}$ declares kw_0^* and kw_1^*, the challenger generates the challenge trapdoor at t^* for kw_b^* where $b \in \{0,1\}$, and it contains $\pi_1^* = g^s$, $\pi_2^* = h_b^{*s}$, and $\pi_3^* = f_b^{*s/\alpha}$ where $h_b^* = H(kw_b^*, \mathsf{pk_R}^{\mathsf{sk_S}})$ and $f_b^* = H'(kw_b^*, \mathsf{pk_R}^{\mathsf{sk_S}})$. Now

$$\begin{aligned}
e(\pi_1^*, \mathsf{CT}) &= e(g^s, h_0^{*r_1} f_0^{*r_2}) \\
&= e(g^s, h_0^{*r_1}) e(g^s, f_0^{*r_2}) \\
&= e(g^{r_1}, h_0^{*s}) e(g^{r_2}, f_0^{*s}) \\
&= e(X, h_0^{*s}) e(g^{\alpha r_2}, f_0^{*s/\alpha}) \\
&= e(X, h_0^{*s}) e(\mathsf{pk_R}^{r_2}, f_0^{*s/\alpha}) \\
&= e(X, h_0^{*s}) e(\mu, f_0^{*s/\alpha})
\end{aligned}$$

holds. Thus, if $b = 0$, then $e(\pi_1^*, \mathsf{CT}) = e(X, \pi_2^*) e(\mu, \pi_3^*)$ holds, and $b = 1$, otherwise. So $\mathcal{A}$ can distinguish b correctly. We remark that the equation $e(\pi_1, \mathsf{CT}) = e(X, \pi_2) e(\mu, \pi_3)$ is employed in their test algorithm. Thus, it seems not trivial to fix the vulnerability even if DDH-hard asymmetric parings, such as [4, 5], are employed.

7 FS-PEKS

Our technique can also be employed to construct FS-PEKS. The definition of FS-PEKS can be trivially derived from those of FS-PAEKS by eliminating sender key related parts. As in our generic construction of FS-PAEKS, a ciphertext at t is a set of PEKS ciphertexts generated by $kw||s_{t,i}^0$ for all $s_{t,i}^0 \in S_t^0$ and a trapdoor at t' is a set of PEKS trapdoors generated by $kw'||s_{t',i}^1$ for all $s_{t',i}^1 \in S_{t'}^1$. We remark that anyone can generate a ciphertext unlike to (FS-)PAEKS, and thus an encryptor may not follow to employ the 0 encoding and can encrypt any keyword. However, this situation does not affect the security (i.e., still no information about the keyword is revealed from ciphertexts due to the security of the underlying PEKS scheme).

As mentioned by Zeng et al. [49], FS-PEKS is basically the same as PETKS proposed by Abdalla et al. [1]. In PETKS, a trapdoor works in a specific time interval $[s, e]$ (s stands for start and e stands for end). PETKS can be constructed generically from HIBE with level-1 anonymity. Alternatively, Zeng et al. [49] also proposed a generic construction of FS-PEKS from attribute-based searchable encryption supporting OR gates.[5] Since PEKS can be constructed from anonymous IBE [1],[6] our FS-PEKS construction eliminates the hierarchical structure or the attribute-based feature of the previous generic constructions, and is meaningful in the viewpoint of feasibility.

As mentioned in the introduction section, if we employ the BF-IBE scheme [8] as the component of the underlying PEKS scheme, then an efficient paring-based FS-PEKS scheme in the random oracle model can be constructed. If the GPV-IBE scheme [20] is employed, then an efficient lattice-based FS-PEKS scheme in the quantum random oracle model can be constructed. Moreover,

[5] Zeng et al. mentioned that their FS-PEKS scheme is the instantiation of the generic construction from [51].

[6] Briefly, a receiver setups the underlying IBE scheme, generates a master public key and a master secret key, and generates a trapdoor using the master secret key. A trapdoor for a keyword kw' is a decryption key for the identity kw'. A sender encrypts a random plaintext R by the underlying anonymous IBE scheme with the identity kw, and a PEKS ciphertext is the IBE ciphertext and R. The test algorithm outputs 1 if the decryption result of a ciphertext by using a trapdoor is R. Obviously, correctness holds. For consistency, let an adversary produce kw and kw' where $kw \neq kw'$ but the test algorithm for a ciphertext of kw and a trapdoor for kw' outputs 1. Then, an IBE ciphertext encrypted by the identity kw is decryptable by the decryption key for the identity kw', and the IND-CPA security is broken. Thus, this construction provides computational consistency under the IND-CPA security of the underlying anonymous IBE scheme.

Table 1: Comparison among the Zeng et al. FS-PEKS scheme and our two pairing-based instantiations. Frist we construct PEKS schemes from the BF-IBE scheme [8] (over symmetric bilinear groups $(\mathbb{G}, \mathbb{G}_T)$) and the CLLWW IBE scheme [13] (over asymmetric bilinear groups $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$) via the Abdalla et al. transformation [1], and next we construct FS-PEKS schemes from these PEKS schemes. We denote "Ours + BF-IBE" or "Ours + CLLWW-IBE" as these FS-PEKS schemes. Let ℓ be the bit length of time period specified in the encryption and trapdoor generation algorithms, i.e., $\ell = O(\log t)$. We employ the security parameter λ to indicate the output size of the hash function (from $\mathbb{G}_T$ to $\{0,1\}^\lambda$) used in the BF-IBE scheme. ROM stands for random oracle model, STD stands for standard model, GGM stands for generic group model, BDH stands for bilinear Diffie-Hellman, and SXDH stands for symmetric external Diffie-Hellman.

FS-PEKS Scheme	Ciphertext Size	Trapdoor Size	Assump.	STD /ROM
Zeng et al. [49]	$(4+\ell)\lvert\mathbb{G}\rvert$	$(3+\ell)\lvert\mathbb{G}\rvert$	GGM	ROM
Abdalla et al. [1] (PETKS)	$\ell\lvert\mathbb{G}\rvert + \lambda$	$\ell((\ell+1)\lvert\mathbb{G}\rvert + \lvert\mathbb{Z}_p\rvert)$	BDH	ROM
Ours + BF-IBE	$\ell(\lvert\mathbb{G}\rvert + 2\lambda)$	$\ell\lvert\mathbb{G}\rvert$	BDH	ROM
Ours + CLLWW-IBE	$\ell(2\lvert\mathbb{G}_T\rvert + 4\lvert\mathbb{G}_1\rvert)$	$4\ell\lvert\mathbb{G}_2\rvert$	SXDH	STD

FS-PEKS schemes that are secure in the standard model also can be obtained from pairings (by PEKS schemes constructed from the Gentry IBE scheme [19], the Lewko IBE scheme [34], the CLLWW IBE scheme [13], the KP IBE scheme [31], or the JR IBE scheme [25]) or lattices (by PEKS schemes constructed from the Yamada IBE scheme [48], the Katsumata IBE scheme [28], or the JKN-IBE scheme [23]).

In the Zeng et al. FS-PEKS scheme [49], the ciphertext size and the trapdoor size depend on the bit length of the time period. Thus, our generic construction yields a comparably efficient FS-PEKS scheme, in terms of ciphertext/trapdoor size and search complexity. In Table 1, we give comparisons among the Zeng et al. FS-PEKS scheme, the Abdalla et al. PETKS scheme instantiated by the Gentry-Silverberg HIBE scheme [21] with a slight modification to provide level-1 anonymity, and our two pairing-based instantiations from PEKS schemes which are instantiations of the Abdalla et al. transformation from the BF-IBE scheme [8] and the CLLWW IBE scheme [13]. We remark that other PETKS schemes can be obtained from other HIBE schemes via the Abdalla et al. generic construction, e.g., anonymous HIBE from parings [6,32,33,43,44] or from lattices [2,3,9,12]. Especially, these pairing-based instantiations provide PETKS schemes secure in the standard model. Nevertheless, our construction is more efficient in terms of the trapdoor size.

8 Conclusion

In this paper, we proposed a generic construction of FS-PAEKS from PAEKS and 0/1 encodings. Since the Jiang et al. FS-PAEKS scheme does not provide forward security, our generic construction yields the first secure FS-PAEKS schemes. Our generic construction is quite simple, and it can be used to construct FS-PEKS. It would be interesting to investigate a generic construction of FS-P(A)EKS without $O(\log t)$-size ciphertext/trapdoor blowup.

Acknowledgment: This work was supported by JSPS KAKENHI Grant Number JP21K11897.

References

[1] Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable encryption

revisited: Consistency properties, relation to anonymous IBE, and extensions. *Journal of Cryptology*, 21(3):350–391, 2008.

[2] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, pages 553–572, 2010.

[3] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In *CRYPTO*, pages 98–115, 2010.

[4] Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. Constructing elliptic curves with prescribed embedding degrees. In *SCN*, pages 257–267, 2002.

[5] Paulo S. L. M. Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In *Selected Areas in Cryptography*, pages 319–331, 2005.

[6] Olivier Blazy, Eike Kiltz, and Jiaxin Pan. (hierarchical) identity-based encryption from affine message authentication. In *CRYPTO*, pages 408–425, 2014.

[7] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In *EUROCRYPT*, pages 506–522, 2004.

[8] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO*, pages 213–229, 2001.

[9] Xavier Boyen and Qinyi Li. Towards tightly secure lattice short signature and id-based encryption. In *ASIACRYPT*, pages 404–434, 2016.

[10] Marco Calderini, Riccardo Longo, Massimiliano Sala, and Irene Villa. Searchable encryption with randomized ciphertext and randomized keyword search. *IACR Cryptol. ePrint Arch.*, page 945, 2022.

[11] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. *Journal of Cryptology*, 20(3):265–294, 2007.

[12] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. *Journal of Cryptology*, 25(4):601–639, 2012.

[13] Jie Chen, Hoon Wei Lim, San Ling, Huaxiong Wang, and Hoeteck Wee. Shorter IBE and signatures via asymmetric pairings. In *Pairing-Based Cryptography*, pages 122–140, 2012.

[14] Leixiao Cheng and Fei Meng. Security analysis of Pan et al.'s "public-key authenticated encryption with keyword search achieving both multi-ciphertext and multi-trapdoor indistinguishability". *Journal of Systems Architecture*, 119:102248, 2021.

[15] Leixiao Cheng and Fei Meng. Public key authenticated encryption with keyword search from LWE. In *ESORICS*, pages 303–324, 2022.

[16] Tianyu Chi, Baodong Qin, and Dong Zheng. An efficient searchable public-key authenticated encryption for cloud-assisted medical internet of things. *Wireless Communications and Mobile Computing*, 2020:8816172:1–8816172:11, 2020.

[17] Keita Emura. Generic construction of public-key authenticated encryption with keyword search revisited: Stronger security and efficient construction. In *ACM APKC*, pages 39–49, 2022.

[18] Keita Emura, Takuya Hayashi, and Ai Ishida. Group signatures with time-bound keys revisited: A new model, an efficient construction, and its implementation. *IEEE Transactions on Dependable and Secure Computing*, 17(2):292–305, 2020.

[19] Craig Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT*, pages 445–464, 2006.

[20] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *ACM STOC*, pages 197–206, 2008.

[21] Craig Gentry and Alice Silverberg. Hierarchical id-based cryptography. In Yuliang Zheng, editor, *ASIACRYPT*, pages 548–566, 2002.

[22] Qiong Huang and Hongbo Li. An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks. *Information Sciences*, 403:1–14, 2017.

[23] Tibor Jager, Rafael Kurek, and David Niehues. Efficient adaptively-secure IB-KEMs and VRFs via near-collision resistance. In *Public-Key Cryptography*, pages 596–626, 2021.

[24] Zhe Jiang, Kai Zhang, Liangliang Wang, and Jianting Ning. Forward secure public-key authenticated encryption with conjunctive keyword search. *The Computer Journal*, 06 2022. `https://doi.org/10.1093/comjnl/bxac075`.

[25] Charanjit S. Jutla and Arnab Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In *ASIACRYPT*, pages 1–20, 2013.

[26] Kohei Kasamatsu, Takahiro Matsuda, Keita Emura, Nuttapong Attrapadung, Goichiro Hanaoka, and Hideki Imai. Time-specific encryption from forward-secure encryption. In *SCN*, pages 184–204, 2012.

[27] Kohei Kasamatsu, Takahiro Matsuda, Keita Emura, Nuttapong Attrapadung, Goichiro Hanaoka, and Hideki Imai. Time-specific encryption from forward-secure encryption: generic and direct constructions. *International Journal of Information Security*, 15(5):549–571, 2016.

[28] Shuichi Katsumata. On the untapped potential of encoding predicates by arithmetic circuits and their applications. In *ASIACRYPT*, pages 95–125, 2017.

[29] Shuichi Katsumata, Shota Yamada, and Takashi Yamakawa. Tighter security proofs for GPV-IBE in the quantum random oracle model. *Journal of Cryptology*, 34(1):5, 2021.

[30] Hyeongseob Kim, Changhee Hahn, and Junbeom Hur. Forward secure public key encryption with keyword search for cloud-assisted IoT. In *IEEE CLOUD*, pages 549–556, 2020.

[31] Kaoru Kurosawa and Le Trieu Phong. Anonymous and leakage resilient IBE and IPE. *Designs, Codes and Cryptography*, 85(2):273–298, 2017.

[32] Roman Langrehr and Jiaxin Pan. Hierarchical identity-based encryption with tight multi-challenge security. In *Public-Key Cryptography*, pages 153–183, 2020.

[33] Kwangsu Lee, Jong Hwan Park, and Dong Hoon Lee. Anonymous HIBE with short ciphertexts: full security in prime order groups. *Designs, Codes and Cryptography*, 74(2):395–425, 2015.

[34] Allison B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In *EUROCRYPT*, pages 318–335, 2012.

[35] Hsiao-Ying Lin and Wen-Guey Tzeng. An efficient solution to the millionaires' problem based on homomorphic encryption. In *ACNS*, pages 456–466, 2005.

[36] Joseph K. Liu, Cheng-Kang Chu, Sherman S. M. Chow, Xinyi Huang, Man Ho Au, and Jianying Zhou. Time-bound anonymous authentication for roaming networks. *IEEE Transactions on Information Forensics and Security*, 10(1):178–189, 2015.

[37] Zi-Yuan Liu, Yi-Fan Tseng, Raylin Tso, Masahiro Mambo, and Yu-Chi Chen. Public-key authenticated encryption with keyword search: Cryptanalysis, enhanced security, and quantum-resistant instantiation. In *ACM ASIACCS*, pages 423–436, 2022.

[38] Mahnaz Noroozi and Ziba Eslami. Public key authenticated encryption with keyword search: revisited. *IET Information Security*, 13(4):336–342, 2019.

[39] Xiangyu Pan and Fagen Li. Public-key authenticated encryption with keyword search achieving both multi-ciphertext and multi-trapdoor indistinguishability. *Journal of Systems Architecture*, 115:102075, 2021.

[40] Kenneth G. Paterson and Elizabeth A. Quaglia. Time-specific encryption. In *SCN*, pages 1–16, 2010.

[41] Baodong Qin, Yu Chen, Qiong Huang, Ximeng Liu, and Dong Zheng. Public-key authenticated encryption with keyword search revisited: Security model and constructions. *Information Sciences*, 516:515–528, 2020.

[42] Baodong Qin, Hui Cui, Xiaokun Zheng, and Dong Zheng. Improved security model for public-key authenticated encryption with keyword search. In *ProvSec*, pages 19–38, 2021.

[43] Somindu C. Ramanna and Palash Sarkar. Anonymous constant-size ciphertext HIBE from asymmetric pairings. In *IMACC*, pages 344–363, 2013.

[44] Somindu C. Ramanna and Palash Sarkar. Efficient (anonymous) compact HIBE from standard assumptions. In *ProvSec*, pages 243–258, 2014.

[45] Olivier Sanders. Improving revocation for group signature with redactable signature. In *Public-Key Cryptography*, pages 301–330, 2021.

[46] Dawn Xiaodong Song, David A. Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *IEEE Symposium on Security and Privacy*, pages 44–55, 2000.

[47] Emil Stefanov, Charalampos Papamanthou, and Elaine Shi. Practical dynamic searchable encryption with small leakage. In *NDSS*, 2014.

[48] Shota Yamada. Asymptotically compact adaptively secure lattice IBEs and verifiable random functions via generalized partitioning techniques. In *CRYPTO*, pages 161–193, 2017.

[49] Ming Zeng, Haifeng Qian, Jie Chen, and Kai Zhang. Forward secure public key encryption with keyword search for outsourced cloud storage. *IEEE Transactions on Cloud Computing*, 10(1):426–438, 2022.

[50] Xiaojun Zhang, Chunxiang Xu, Huaxiong Wang, Yuan Zhang, and Shixiong Wang. FS-PEKS: lattice-based forward secure public-key encryption with keyword search for cloud-assisted industrial internet of things. *IEEE Transactions on Dependable and Secure Computing*, 18(3):1019–1032, 2021.

[51] Qingji Zheng, Shouhuai Xu, and Giuseppe Ateniese. VABKS: verifiable attribute-based keyword search over outsourced encrypted data. In *IEEE INFOCOM*, pages 522–530, 2014.