# Time-Specific Encryption from Forward-Secure Encryption

Kohei Kasamatsu[1], Takahiro Matsuda[2,⋆], Keita Emura[3], Nuttapong Attrapadung[2], Goichiro Hanaoka[2], and Hideki Imai[1]

[1] Chuo University, Japan
{kasamatsu-kohei,h-imai}@imailab.sakura.ne.jp
[2] National Institute of Advanced Industrial Science and Technology, Japan
{t-matsuda,n.attrapadung,hanaoka-goichiro}@aist.go.jp
[3] National Institute of Information and Communications Technology, Japan
k-emura@nict.go.jp

**Abstract.** Paterson and Quaglia (SCN 2010) proposed the concept of time-specific encryption (TSE) and its efficient constructions. TSE is a type of public key encryption with additional functionality where an encryptor can specify a suitable time interval, meaning that the ciphertexts may only be decrypted within this time interval. In this work, we propose a new methodology for designing efficient TSE scheme by using forward-secure encryption (FSE), and based on this methodology, we present a specific TSE scheme using Boneh-Boyen-Goh FSE, and a generic construction from any FSE. Our proposed TSE schemes are practical in *all* aspects with regard to computational costs and data sizes. The sizes of the ciphertext and the public parameter in our schemes are significantly smaller than those in previous schemes in an asymptotic sense.

## 1 Introduction

In SCN 2010, Paterson and Quaglia proposed the concept of time-specific encryption (TSE), and showed its efficient constructions [22]. TSE is a class of public key encryption (PKE) with additional functionality where an encryptor can specify a suitable time interval such that the ciphertexts may only be decrypted within this time interval. Such an encryption scheme is useful in applications where it is necessary to ensure that the receiver can recover the plaintext within only a specific time interval, e.g. in electronic sealed-bid auctions.

In this paper, we propose a novel methodology for the construction of TSE schemes, and provide practical constructions based on our methodology. We show that forward-secure encryption (FSE) [1,7] is a powerful building block for designing efficient TSE schemes compared to the previous methodologies [22], which were based on identity-based encryption (IBE) [24,5] and broadcast encryption (BE) [15,6]. Based on our methodology, new TSE schemes can be obtained that are practical in terms of computational costs and data sizes.

In the remaining parts of this section, we provide a review of TSE and discuss our results in detail.

⋆ The second author is supported by a JSPS Fellowship for Young Scientists.

I. Visconti and R. De Prisco (Eds.): SCN 2012, LNCS 7485, pp. 184–204, 2012.

### 1.1 Time-Specific Encryption

In a typical scenario where TSE is used, a semi-trusted agent called a *time-server* publishes a global system parameter and periodically issues a time instant key (TIK) that is used by each receiver to decrypt a ciphertext. In TSE, a sender can specify any interval (decryption time interval, DTI) $[t_L, t_R]$ when encrypting a plaintext $M$ to form a ciphertext $c$, where $t_L$ and $t_R$ denote the start and end points of the DTI, respectively, and a receiver can decrypt the ciphertext if the receiver is in possession of the TIK $SK_t$ for some $t$ with $t \in [t_L, t_R]$. This functionality seems to be a natural extension of that of timed-release encryption (TRE) [20,8,9,12,19], but interestingly, it is not easy to construct TSE by straightforward modifications of existing TRE schemes.

Paterson and Quaglia presented elegant methods to efficiently achieve the required functionality of TSE [22]. Specifically, they proposed two generic constructions of TSE where one is based on IBE, and the other is based on BE, and instantiations of these schemes can yield significantly higher efficiencies than those using straightforward modification of existing TRE. However, as shown in Table 1 (in Sect. 5), these schemes are still not very efficient in some aspects, and it is thus necessary to explore other solutions which can overcome their (potential) shortcomings. In particular, if we assume that the lifetime of the global system parameter is divided into $T$ time periods, then the size of a ciphertext in the generic construction from IBE may be linear in $T$, and the size of the global system parameter in the construction from BE may also be linear in $T$. Because $T$ is generally a large value, this could be problematic in certain situations.

### 1.2 Our Contribution

In this work, we propose a new methodology for designing efficient TSE scheme by using FSE. We consider this approach to be more promising than the previous methods because of the similarity between the functionalities of TSE and FSE. In fact, we can immediately produce a TSE scheme that allows only restricted DTIs with $t_L = 0$ by directly using FSE as it is. Based on this observation, we give a specific (i.e. not generic) construction of TSE from an existing FSE scheme[1] by Boneh, Boyen, and Goh [4], and a generic construction that can be used with any FSE scheme. Remarkably, these schemes can yield sufficiently high efficiency in all aspects in terms of computational cost and data size, and in particular, with regard to the evaluation items given in Table 1 (in Sect. 5), the complexities of our proposed schemes are all at most poly-logarithmic in $T$, and are thus significantly smaller than $O(T)$ in the asymptotic sense. Also, our specific construction is more efficient than the best-known instantiation from our generic construction in all aspects (except for the computational cost for decryption). However, our generic construction is still advantageous in the sense that when a new construction is discovered, this then automatically results in a

[1] This FSE scheme is obtained from the hierarchical IBE (HIBE) scheme in [4] via the "HIBE-to-FSE" transformation by Canetti, Halevi, and Katz [7]. See Sect. 2.2.

new TSE scheme via our generic construction, and this scheme may potentially be more efficient than our specific construction.

Here, we give an overview of our basic ideas for construction of our proposed schemes. As noted above, from FSE, we can immediately derive a TSE scheme for restricted DTIs with $t_L = 0$. This is based on the following observation: in FSE, a decryption key is updated periodically, while the corresponding encryption key is fixed, and more specifically, a decryption key for one particular time period can be derived from another decryption key for any previous time period, but *not vise versa*. Therefore, if a decryption key for one time period is publicized, then all other decryption keys for subsequent time periods can be generated from it. *This exactly describes a TSE scheme for restricted DTIs with* $t_L = 0$. Similarly to this construction, we can also derive another TSE scheme for restricted DTIs with $t_R = T - 1$, and it seems that we can also obtain full-fledged TSE by considering multiple encryptions [27,13] of these two restrictive TSE schemes. However, unfortunately, this idea does not immediately work. Because, in this (insecure) TSE scheme, a decryption key for each time period consists of the two independent decryption keys of the two underlying restrictive TSE schemes, a malicious user can thus illegally generate decryption keys for various time periods by combining these components (i.e. the decryption keys of the underlying restrictive TSE schemes) in multiple decryption keys. In our proposed schemes, we overcome this technical hurdle by introducing the following ideas: (1) in our specific construction, the two decryption keys of the underlying restrictive TSE schemes are connected in an inseparable manner (by using the algebraic property of the FSE scheme in [4]), meaning that it consequently becomes impossible to generate an illegal decryption key from the components of multiple decryption keys for the different time periods; and (2) in our generic construction, we set up many underlying restrictive TSE schemes (rather than using only two underlying schemes), and avoid the above attack by using a combinatorial method proposed by Attrapadung et al. [2]. Because our specific construction requires only two underlying restrictive TSE schemes, it is more efficient than our generic construction, which requires many underlying schemes. However, our generic construction does not depend on the algebraic property of the underlying scheme and therefore can be constructed from any FSE scheme.

### 1.3 Related Works

TSE was introduced as an extension of TRE, and thus we briefly describe TRE here. TRE is a type of encryption system introduced by May [20] in 1993. In TRE, a message can be encrypted in such a way that it cannot be decrypted (even by a legitimate receiver who owns the decryption key for the ciphertext) until the time (called the release-time) that was specified by the encryptor. TRE can therefore be interpreted as TSE in which we can only use the most restricted type of DTI $[t_L, t_R]$ with $t_L = t_R$. Many practical applications and situations where TRE schemes can be used have been considered, including sealed-bid auctions, electronic voting, content predelivery systems, and on-line examinations.

There are mainly two major approaches for realizing TRE. One approach is the use of *time-lock puzzles* [23]. In this approach, a sender generates a ciphertext which cannot be completely decrypted until the release-time in a receiver's environment, even if the receiver continues computing to decrypt the ciphertext after it is received. This imposes a heavy computational cost on the receiver, and it is difficult to precisely estimate the required time at which the receiver recovers a message. (This approach is also unsuitable for TSE.)

The other approach uses a semi-trusted agent, called the *time-server*, which periodically generates the time specific information needed to encrypt a message and/or decrypt a ciphertext. Earlier TRE schemes [20,23] adopted a model in which the time-server and the system users needed to interact.

Chan et al. [9] and Cheon et al. [10] proposed TRE schemes in which no interaction between the time-server and users was required. Most of the TRE schemes [8,18,11,21,12,19,14] that were proposed after these schemes [9,10], and the TSE schemes of Paterson and Quaglia [22], follow this approach.

It should be noted that most of the previous TRE schemes that adopted the time-server model of [9,10] are in fact *public-key* (or *identity-based*) TRE schemes, which consider confidentiality against the time-server. In the model, each receiver has its own secret key along with its corresponding public information (either a public key or an identity), and when encrypting a message, the encryptor specifies not only release time but also a receiver's public information to generate a receiver-specific ciphertext. We can also consider a "plain" version of TRE in the time-server model in which each ciphertext is not specific to any receiver and the ciphertext can be decrypted by anyone who receives the time-specific information from the time-server. (In this model, the confidentiality against the time-server is not considered.) This plain TRE can be realized easily from any IBE by regarding an identity as a time. Indeed, most of the previous public key TRE schemes mentioned above were realized by combining an IBE-like primitive with a PKE-like primitive. The difference between these settings (i.e. the "public key" setting and the "plain" setting) were explicitly considered for TSE by Paterson and Quaglia [22]. For a more detailed explanation, see Sect. 2.1.

## 2 Preliminaries

In this section, we formally introduce the definition of TSE, FSE, and bilinear groups, and describe the decisional $\ell$-weak Bilinear Diffie-Hellman Inversion ($\ell$-wBDHI) assumption.

*Notation.* Throughout this paper, we will consider time as a discrete set of time periods, regarding these as integers between 0 and $T-1$, where $T$ represents the number of time periods supported by the system. We denote by $[t_L, t_R]$, where $t_L \leq t_R$, the interval containing all time periods from $t_L$ to $t_R$ inclusive. "$x \xleftarrow{\mathrm{U}} y$" denotes that $x$ is chosen uniformly at random from $y$. $x \leftarrow y$ denotes $x$ is output from $y$ if $y$ is an algorithm, or $y$ is assigned to $x$ otherwise. "PPT"

denotes *probabilistic polynomial time*. We say that a function $f(k)$ is negligible (in $k$) if $f(k) < 1/p(k)$ for any positive polynomial $p(k)$ and all sufficiently large $k$.

### 2.1 Time-Specific Encryption

As explained in the introduction, a TSE scheme is an extension of a TRE scheme which supports decryption of ciphertexts with respect to DTI. Paterson et al. [22] defined several settings for TSE, namely, plain TSE, public-key TSE, and identity-based TSE. In the plain TSE setting, a ciphertext is not specified to any user and any entity who obtains a TIK corresponding to the DTI of the ciphertext can decrypt the ciphertext. The plain setting is mainly introduced in order to be used as building blocks for TSE schemes for the other two settings (though a plain TSE scheme itself might have some interesting applications). In public-key and identity-based TSE settings, on the other hand, each user (receiver) has its own secret key and either a public-key or an identity, and a ciphertext is made specific to a particular receiver using these public information. Correspondingly, to decrypt a ciphertext, not only a TIK but also the receiver's secret key is now required. TSE schemes in the latter two settings provide confidentiality even against a curious time-server. In this paper, we will only consider the plain setting, which is because public-key (resp. identity-based) TSE scheme with desirable security can be generically obtained by appropriately combining a plain TSE scheme with a chosen-ciphertext secure PKE (resp. IBE) scheme with the previously known methods for TRE schemes [10,21,19]. From here on, when we just write "TSE", we always mean "plain TSE".

A TSE scheme is defined by the four algorithms ($\mathtt{TSE.Setup}$, $\mathtt{TSE.Ext}$, $\mathtt{TSE.Enc}$, $\mathtt{TSE.Dec}$), which has the associated message space $MSP$. The four algorithms are as follows: The setup algorithm $\mathtt{TSE.Setup}(1^k, T)$ takes a security parameter $1^k$ and $T \in \mathbb{N}$ as input, and outputs a master public key $MPK$ and a master secret key $MSK$, where the TSE system supports time space $\mathsf{T} = [0, T-1]$. The key extraction algorithm $\mathtt{TSE.Ext}(MPK, MSK, t)$ takes $MPK$, $MSK$, and a time $t \in \mathsf{T}$ as input, and outputs a TIK $SK_t$. The encryption algorithm $\mathtt{TSE.Enc}(MPK, [t_L, t_R], M)$ takes $MPK$, a DTI $[t_L, t_R] \subseteq \mathsf{T}$, and a message $M \in MSP$ as input, and outputs a ciphertext $C$. The decryption algorithm $\mathtt{TSE.Dec}(MPK, SK_t, C)$ takes $MPK$, $SK_t$, and $C$ as input, outputs either a message $M$ or the failure symbol $\perp$. We require, for all $k \in \mathbb{N}$, all $T \in \mathbb{N}$, all integers $t_L$, $t_R$, and $t$ satisfying $0 \leq t_L \leq t \leq t_R \leq T-1$, all $(MPK, MSK) \leftarrow \mathtt{TSE.Setup}(1^k, T)$, and all messages $M \in MSP$, that $\mathtt{TSE.Dec}(\mathtt{TSE.Ext}(MSK, t), \mathtt{TSE.Enc}(MPK, [t_L, t_R], M)) = M$.

*Security.* We review the security definition for a TSE scheme by Paterson et al. [22]. This security requires that an adversary cannot gain any useful information from a ciphertext under a DTI $[t_L, t_R]$, if the adversary has no TIKs $SK_t$ for $t \in [t_L, t_R]$.

Formally, we say that a TSE scheme is IND-CPA secure if any PPT adversary $\mathcal{A}$ has at most negligible advantage (in the security parameter $k$) in

the following game between a challenger $\mathcal{C}$ and $\mathcal{A}$ for any polynomial $T$: $\mathcal{C}$ runs $\mathtt{TSE.Setup}(1^k, T)$ to generate a master public/secret key pair $(MPK, MSP)$, and gives $MPK$ to $\mathcal{A}$. $\mathcal{A}$ can adaptively issue TIK extraction queries $t_1, t_2, \ldots$. For each TIK extraction query $t_i$, $\mathcal{C}$ responds by running $\mathtt{TSE.Ext}(MPK, MSK, t_i)$ to generate a TIK $SK_{t_i}$ corresponding to $t_i$, and then returns $SK_{t_i}$ to $\mathcal{A}$. At some point $\mathcal{A}$ selects two challenge messages $M_0, M_1 \in MSP$ and the challenge DTI $[t_L, t_R] \subseteq \mathsf{T}$ with the restriction that $t_i \notin [t_L, t_R]$ for all of the previous TIK extraction queries $t_i$ that $\mathcal{A}$ made before the challenge. $\mathcal{A}$ then sends $(M_0, M_1, [t_L, t_R])$ to $\mathcal{C}$. $\mathcal{C}$ chooses a random bit $b$ and computes $C^* \leftarrow \mathtt{TSE.Enc}(MPK, [t_L, t_R], M_b)$, and returns $C^*$ to $\mathcal{A}$. $\mathcal{A}$ can continue to make TIK extraction queries $t_i$ under the restriction $t_i \notin [t_L, t_R]$, and $\mathcal{C}$ responds to those as before. Finally, $\mathcal{A}$ outputs its guess $b' \in \{0, 1\}$ for $b$. The adversary $\mathcal{A}$'s advantage in the above game is defined as $Adv^{CPA}_{TSE,\mathcal{A}}(k) = |\Pr[b' = b] - \frac{1}{2}|$.

### 2.2 Forward-Secure Encryption

An FSE scheme has the property that the threat of key exposure is confined to some span by updating the secret key at each time unit. This scheme realizes the property by using the functionality that a receiver can update the previous secret key $d_{t-1}$ to the next secret key $d_t$ without interacting with any outside entity. We provide a formal definition of FSE by following [7] but slightly customized for our purpose.

An FSE scheme is defined by the four algorithms ($\mathtt{FSE.Gen}$, $\mathtt{FSE.Upd}$, $\mathtt{FSE.Enc}$, $\mathtt{FSE.Dec}$), which has the associated message space $MSP$. The key generation algorithm $\mathtt{FSE.Gen}(1^k, N)$ takes a security parameter $1^k$ and the total number of time periods $N$ as input, and outputs a public key $pk$ and an initial secret key $d_0$. The key update algorithm $\mathtt{FSE.Upd}(pk, i, j, d_i)$ takes $pk$, an index $i < N$ of a previous time period, an index $j > i$ for the current time period, and a secret key $d_i$ (corresponding to the period $i$) as input, and outputs a secret key $d_j$ for the time period $j$. The encryption algorithm $\mathtt{FSE.Enc}(pk, i, M)$ takes $pk$, $i < N$, and a message $M \in MSP$ as input, and outputs a ciphertext $c$. The decryption algorithm $\mathtt{FSE.Dec}(pk, d_{i'}, c)$ takes $pk$, $d_{i'}$, and $c$ as input, and outputs either $M$ or a failure symbol $\perp$. We require, for all $k \in \mathbb{N}$, all $N \in \mathbb{N}$, all $(pk, d_0) \leftarrow \mathtt{FSE.Gen}(1^k, N)$, all indices $i \in [0, N-1]$ (for specifying time periods), and all messages $M \in MSP$, that $\mathtt{FSE.Dec}(pk, \mathtt{FSE.Upd}(pk, 0, i, d_0), \mathtt{FSE.Enc}(pk, i, M)) = M$.

We note that Canetti et al. [7] defined only the "sequential update" algorithm. That is, in their syntax, the key update algorithm only allows an update from a secret key $d_i$ for the time period $i$ to a key $d_{i+1}$ for the next time period. However, for the sake of simplicity, we use the syntax in which the update algorithm allows the "direct update", so that $\mathtt{FSE.Upd}$ takes a key $d_i$ for the time period $i$ as input and outputs the secret key $d_j$ as long as $i < j$. It is straightforward to see that the direct update functionality can be generally achieved by the sequential update algorithm of [7]. In addition, there are FSE schemes which support efficient direct update algorithm (compared to running "sequential update algorithms many times), such as the FSE scheme instantiated with the HIBE scheme by Boneh et al. [4] via the HIBE-to-FSE transformation shown in [7].

*Security.* We say that an FSE scheme is IND-CPA secure if any PPT algorithm $\mathcal{A}$ has at most negligible advantage (in the security parameter $k$) in the following game between a challenger $\mathcal{C}$ and $\mathcal{A}$ for any polynomial $N$: At the beginning of the game $\mathcal{A}(1^k, N)$ outputs the challenge time period $j^*$. $\mathcal{C}$ runs $\mathtt{FSE.Gen}(1^k, N)$ to generate a pair of a public key $pk$ and an initial secret key $d_0$, runs $d_{j^*+1} \leftarrow \mathtt{FSE.Upd}(pk, 0, j^*+1, d_0)$, and then gives $pk$ and $d_{j^*+1}$ to $\mathcal{A}$. $\mathcal{A}$ selects two challenge messages $m_0, m_1 \in MSP$, and sends $m_0, m_1$ to $\mathcal{C}$. $\mathcal{C}$ chooses a random bit $b$, computes $c^* = \mathtt{FSE.Enc}(pk, j^*, m_b)$, and returns $c^*$ to $\mathcal{A}$. Finally $\mathcal{A}$ outputs its guess $b' \in \{0, 1\}$ for $b$. The adversary $\mathcal{A}$'s advantage in the above game is defined as $Adv^{CPA}_{FSE,\mathcal{A}}(k) = |\Pr[b' = b] - \frac{1}{2}|$.

Note that in the above security game, the adversary is required to commit to the time period to be attacked at the beginning of the game. While this definition is weaker than the definition of [7], it suffices for our construction of a TSE scheme from any FSE scheme that we will show in the following sections.

*Transformations from Hierarchical IBE.* There is a trivial construction of an FSE scheme that supports $N$ time periods from a hierarchical IBE (HIBE) scheme that supports hierarchy with depth $N$, by interpreting a time period $i$ in FSE as a "chain" $(1, \ldots, i)$ of identities in HIBE. More specifically, for a secret key for the time period $t$ in FSE, we use a decryption key for the identity-vector $(1, \cdots, i)$ in HIBE. To update a secret key for time period $j$ to time period $j > i$, one can run the derivation algorithm of the HIBE scheme to obtain a decryption key for the identity-vector $(1, \cdots, j)$.

Another more sophisticated HIBE-to-FSE transformation is the binary tree-based construction due to Canetti, Halevi, and Katz [7]. This construction has the advantage in that to instantiate an FSE scheme with $N$ time periods, a building block HIBE only needs to support a hierarchy with depth $\log N$.

The common feature of these HIBE-to-FSE transformations is that multiple instances of FSE can virtually be instantiated so that they all share the same public parameters, by regarding the top-level identities as the indices for specifying an independent HIBE scheme, and then applying the HIBE-to-FSE transformations to each HIBE scheme instantiated in the second (and lower) level identity space. This trick will be used in our constructions of TSE.

*Concrete Instantiation from the Boneh-Boyen-Goh HIBE Scheme [4].* In Fig. 1, we review the instantiation of an FSE scheme, which we call the *basic BBG-FSE* scheme, using the HIBE scheme by Boneh, Boyen, Goh (BBG HIBE) [4] via the "chain"-style transformation explained above.

Looking ahead, the basic version of our TSE scheme in Sect. 3.2 is obtained from the above basic BBG-FSE scheme, and our full TSE scheme in Appendix A is based on the FSE scheme obtained from the HIBE-to-FSE transformation due to Canetti et al. [7] (we call this FSE scheme *full BBG-FSE*).

| | |
|---|---|
| $\mathtt{FSE.Gen}_{\mathtt{BBG}}(1^k, N)$ : | $\mathtt{FSE.Upd}_{\mathtt{BBG}}(pk, i, j, \sigma, d_i)$: (where $j > i$) |
| $\alpha, \beta \xleftarrow{\mathrm{U}} \mathbb{Z}_p$; $g, g_2, h_0, \cdots, h_N \xleftarrow{\mathrm{U}} \mathbb{G}$ | Parse $pk$ as $(g, g_1, g_2, \boldsymbol{h}, P)$ |
| $g_1 \leftarrow g^\alpha$; $P \leftarrow e(g^\alpha, g^\beta)$; $d_0 \leftarrow g^{\alpha\beta}$ | $r \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ |
| $pk \leftarrow (g, g_1, g_2, \boldsymbol{h} = (h_0, \cdots, h_N), P)$ | If $i = 0$ then |
| Return $(pk, d_0)$ | $d_j \leftarrow (d_i \cdot f(j, \boldsymbol{h}, \sigma, g_2)^r, g^r, h_{j+1}^r, \cdots, h_N^r)$ |
| $\mathtt{FSE.Enc}_{\mathtt{BBG}}(pk, i, \sigma, M)$ : | Else (i.e. $i \neq 0$) |
| Parse $pk$ as $(g, g_1, g_2, \boldsymbol{h}, P)$ | Parse $d_i$ as $(a_0, a_1, b_{i+1}, \cdots, b_N)$ |
| $s \xleftarrow{\mathrm{U}} \mathbb{Z}_p$; $C_1 \leftarrow P^s \cdot M$; $C_2 \leftarrow g^s$ | $D_0 \leftarrow a_0 \cdot \prod_{k=i+1}^{j} b_k^{\sigma N+k} \cdot f(j, \boldsymbol{h}, \sigma, g_2)^r$ |
| $C_3 \leftarrow f(i, \boldsymbol{h}, \sigma, g_2)^s$ | $D_1 \leftarrow a_1 \cdot g^r$ |
| Return $C \leftarrow (C_1, C_2, C_3)$ | $D'_u \leftarrow b_u \cdot h_u^r$ for all $u \in [j+1, T]$ |
| $\mathtt{FSE.Dec}_{\mathtt{BBG}}(d_i, C)$ : | $d_j \leftarrow (D_0, D_1, D'_{j+1}, \cdots, D'_T)$ |
| Parse $C$ as $(C_1, C_2, C_3)$ | End If |
| Parse $d_i$ as $(D_1, D_2, \cdots)$ | Return $d_j$ |
| Return $M \leftarrow \frac{C_1 \cdot e(C_3, D_2)}{e(C_2, D_1)}$ | |

**Fig. 1.** Basic BBG-FSE: The FSE scheme obtained from the BBG HIBE scheme [4], where $f(i, \boldsymbol{h} = (h_0, \cdots, h_N), \sigma, b) = h_0^{2N+1} \cdot \prod_{k=1}^{i} h_k^{\sigma N+k} \cdot b$

For notational convenience, in Fig. 1, we describe the scheme so that the encryption and update algorithms take an additional input $\sigma \in \{0, 1\}$. This bit $\sigma$ is used to instantiate two BBG-FSE schemes with $N$ time periods under the same public parameter: the first scheme uses the "ordinary" interval $[0, N-1]$, and the second scheme uses the "shifted" interval $[N, 2N-1]$.

### 2.3 Decisional $\ell$-wBDHI Assumption

We first recall bilinear groups. Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of order $p$ for some large prime $p$ (we assume that the size of $p$ is implicitly determined by the security parameter $k$), and let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be an efficiently computable mapping. We call a tuple $(\mathbb{G}, \mathbb{G}_T, e)$ *bilinear groups*, and $e$ a *bilinear map*, if the following two conditions hold: (Bilinear:) for all generators $(g, h) \in \mathbb{G} \times \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, we have $e(g^a, h^b) = e(g, h)^{ab}$. (Non-degenerate:) for all generators $g, h \in \mathbb{G}$, we have $e(g, h) \neq 1$.

Now we recall the decisional $\ell$-wBDHI assumption (which is defined via the so-called decisional $\ell$-wBDHI$^*$ problem [4, Sect. 2.3]). Let $\ell \in \mathbb{N}$. We say that the decisional $\ell$-wBDHI assumption holds in $(\mathbb{G}, \mathbb{G}_T, e)$ if for any PPT algorithm $\mathcal{A}$ the following difference is negligible in the security parameter $k$:

$$|\Pr[\mathcal{A}(g, h, y_1, \cdots, y_\ell, e(g, h)^{\alpha^{\ell+1}}) = 0] - \Pr[\mathcal{A}(g, h, y_1, \cdots, y_\ell, W) = 0]|$$

where $g, h \xleftarrow{\mathrm{U}} \mathbb{G}$, $\alpha \xleftarrow{\mathrm{U}} \mathbb{Z}_p$, $y_i \leftarrow g^{(\alpha^i)}$, and $W \xleftarrow{\mathrm{U}} G_T$.

## 3 Concrete Construction from Specific FSE

In this section, we present our proposed TSE scheme based on a specific FSE scheme obtained from the BBG HIBE scheme. Although the construction strongly

depends on the algebraic structure of the underlying BBG-FSE scheme (thus it is not a generic construction), it leads to an efficient TSE scheme compared to TSE schemes derived from our generic construction in the next section.

### 3.1 The Idea of Our Construction

Before going into the description of the scheme, we give an intuitive explanation of our strategy behind the proposed construction. As explained in Sect. 1.2, TSE obtained by multiple encryption of two restrictive TSE schemes (which are derived from FSE) is insecure. In this insecure TSE scheme, a TIK for each time period consists of two independent decryption keys of underlying two restrictive TSE schemes. A malicious user can illegally generate decryption keys for various time period by combining components in multiple decryption keys. Here, observe that such an attack is possible because these two restrictive TSE schemes are instantiated independently.

Our idea for the proposed construction is to "connect" the secret keys from the underlying two restrictive TSE schemes in an "inseparable" manner by using the specific algebraic structure of the BBG-FSE scheme. Specifically, we divide the master secret key information $d_0 = g^{\alpha\beta}$ of the BBG-FSE scheme into two shares $g^{\alpha\beta+\xi}$ and $g^{-\xi}$ in the 2-out-of-2 secret-sharing manner using a "blinding factor" $\xi$. This blinding factor $\xi$ is a randomness generated for each execution of a TIK extraction algorithm. Intuitively, this $\xi$ connects the secret keys from the underlying restrictive TSE schemes, and thus an adversary cannot come up with an illegal "virtual" TIK as above by combining multiple TIKs for time periods that do not include the DTI. In order to make the decryption by the above TIK possible, we appropriately modify the encryption algorithm so that the underlying BBG-FSE-based TSE schemes (one with DTI for $t_L = 0$ and the other with DTI for $t_R = T - 1$) use a common randomness $s$. Such use of a common randomness is possible again due to the algebraic structure of the BBG-FSE scheme.

For the sake of simplicity, in this section we only give the basic version of our proposed construction whose public parameter size and TIK size are $O(T)$ and whose ciphertext size is constant. Our full TSE scheme, in which the public parameter size is $O(\log T)$ and TIKs size is $O(\log^2 T)$ by using binary tree structures inspired by the HIBE-to-FSE transformation of Canetti et al. [7], is given in Appendix A. We stress that those proposed schemes share the same idea as explained above, and we believe that the basic version of our proposed scheme is helpful for understanding the full construction.

### 3.2 Basic Construction

Here, we give the basic version of our proposed TSE scheme. Let $(\mathbb{G}, \mathbb{G}_T, e)$ be bilinear groups, and let $T \in \mathbb{N}$ be the number of time periods. Then we construct the basic version of our TSE scheme as in Fig. 2.

As mentioned in Sect. 3.1, we combine two basic BBG-FSE schemes (from Fig. 1) in which one of the schemes is regarded as a TSE scheme which allows

| $\mathtt{TSE.Setup}(1^k, T)$ : | $\mathtt{TSE.Enc}(MPK, [t_L, t_R], M)$ : |
|---|---|
| $(pk, d_0) \leftarrow \mathtt{FSE.Gen}_{\mathtt{BBG}}(1^k, T)$ | Parse $MPK$ as $(g, g_1, g_{2,F}, g_{2,B}, \boldsymbol{h}, P)$ |
| $g_{2,B} \xleftarrow{\mathrm{U}} \mathbb{G}$ | $s \xleftarrow{\mathrm{U}} \mathbb{Z}_p$; $pk_F \leftarrow (g, g_1, g_{2,F}, \boldsymbol{h}, P)$ |
| Parse $pk$ as $(g, g_1, g_{2,F}, \boldsymbol{h}, P)$ | $pk_B \leftarrow (g, g_1, g_{2,B}, \boldsymbol{h}, P)$ |
| Parse $\boldsymbol{h}$ as $(h_0, \cdots, h_T)$ | $(C_1, C_2, C_3) \leftarrow \mathtt{FSE.Enc}_{\mathtt{BBG}}(pk_F, t_R + 1, 0, M; s)$ |
| $MPK \leftarrow (g, g_1, g_{2,F}, g_{2,B}, \boldsymbol{h}, P)$ | $(C_1, C_2, C_4) \leftarrow \mathtt{FSE.Enc}_{\mathtt{BBG}}(pk_B, T - t_L, 1, M; s)$ |
| $MSK \leftarrow d_0$ | $C_5 \leftarrow [t_L, t_R]$ |
| Return $(MPK, MSK)$ | Return $C \leftarrow (C_1, C_2, C_3, C_4, C_5)$ |
| $\mathtt{TSE.Ext}(MPK, MSK, t)$ : | $\mathtt{TSE.Dec}(MPK, SK_t, C)$ : |
| Parse $MPK$ as $(g, g_1, g_{2,F}, g_{2,B}, \boldsymbol{h}, P)$ | Parse $MPK$ as $(g, g_1, g_{2,F}, g_{2,B}, \boldsymbol{h}, P)$ |
| $\xi \xleftarrow{\mathrm{U}} \mathbb{Z}_p$ | Parse $C$ as $(C_1, C_2, C_3, C_4, C_5)$ |
| $pk_F \leftarrow (g, g_1, g_{2,F}, \boldsymbol{h}, P)$ | Parse $SK_t$ as $(d_{t+1,F}, d_{T-t,B}, t)$ |
| $pk_B \leftarrow (g, g_1, g_{2,B}, \boldsymbol{h}, P)$ | $pk_F \leftarrow (g, g_1, g_{2,F}, \boldsymbol{h}, P)$ |
| $d_{0,F} \leftarrow MSK \cdot g^{\xi} = g^{\alpha\beta+\xi}$ | $pk_B \leftarrow (g, g_1, g_{2,B}, \boldsymbol{h}, P)$ |
| $d_{0,B} \leftarrow g^{-\xi}$ | $C_F \leftarrow (C_1, C_2, C_3)$ |
| $d_{t+1,F}$ | If $t \notin C_5$ then return $\perp$ |
| $\leftarrow \mathtt{FSE.Upd}_{\mathtt{BBG}}(pk_F, 0, t+1, 0, d_{0,F})$ | $d_{t_R+1,F}$ |
| $d_{T-t,B}$ | $\leftarrow \mathtt{FSE.Upd}_{\mathtt{BBG}}(pk_F, t+1, t_R+1, 0, d_{t+1,F})$ |
| $\leftarrow \mathtt{FSE.Upd}_{\mathtt{BBG}}(pk_B, 0, T-t, 1, d_{0,B})$ | $d_{T-t_L,B}$ |
| $SK_t \leftarrow (d_{t+1,F}, d_{T-t,B}, t)$ | $\leftarrow \mathtt{FSE.Upd}_{\mathtt{BBG}}(pk_B, T-t, T-t_L, 1, d_{T-t,B})$ |
| Return $SK_t$ | $M' \leftarrow \mathtt{FSE.Dec}_{\mathtt{BBG}}(d_{t_R+1,F}, C_F)$ |
| | Parse $d_{T-t_L,B}$ as $(R_1, R_2, \cdots)$ |
| | Return $M \leftarrow \frac{M' \cdot e(R_2, C_4)}{e(R_1, C_2)}$ |

**Fig. 2.** The basic version of the proposed TSE scheme based on the BBG-FSE scheme

only DTIs with $t_R = T - 1$, by introducing a blinding factor $\xi$ in order to construct a TSE scheme. More specifically, we set the initial key $d_{0,F}$ of the TSE scheme for restricted DTIs with $t_L = 0$ as $g^{\alpha\beta+\xi}$ which includes the blind factor $g^{\xi}$, and the initial key $d_{0,B}$ of the TSE scheme for restricted DTIs with $t_R = T - 1$ as $g^{-\xi}$ which will remove the blinding factor, by using the above mentioned method.

We would like the reader to notice that in Fig. 2, the scheme is described at the cost of efficiency, so that it is easy to see that two basic BBG-FSE schemes are combined by the blinding factor $\xi$ as we explained above. For example, in the encryption scheme, the ciphertext components $C_1$ and $C_2$ are computed twice by running $\mathtt{FSE.Enc}_{\mathtt{BBG}}$ from common randomness $s$. However, in practice, only $C_4$ needs to be computed in the second execution of $\mathtt{FSE.Enc}_{\mathtt{BBG}}$, which can be done without calculating $C_1$ and $C_2$.

The security is guaranteed by the following theorem (the proof is given in the full version of this paper).

**Theorem 1.** *If the decisional $(T+1)$-wBDHI assumption holds in $(\mathbb{G}, \mathbb{G}_T, e)$, then the TSE scheme (which supports $T$ time periods) constructed as in Fig. 2 is IND-CPA secure.*

## 4 Generic Construction from Any FSE

In Sect. 3 (and in Appendix A), we proposed an efficient construction of TSE based on the BBG-FSE scheme, where it directly exploits the algebraic structure of the BBG-FSE scheme. In this section, we describe a generic construction of TSE from any FSE scheme in a black-box manner; nevertheless, it can be shown to be far more efficient compared to trivial generic constructions.

The intuition for our construction is as follows. Recall that we have observed that any FSE scheme already implies a TSE scheme with restricted interval types of the form $[A, *] \subseteq [A, B]$, where $*$ denotes arbitrary value where the encryptor would specify, and $A, B$ are a-priori fixed values. By taking the key derivation in the backward manner, FSE also implies another TSE scheme with restricted interval types of the form $[*, D] \subseteq [C, D]$, where $C, D$ are fixed. Our purpose is to construct a TSE scheme that allows any intervals $[*, *] \subseteq [0, T-1]$. The idea is then to pre-define a collections $\mathcal{S}$ of allowed intervals to only consist of these restricted types in such a way that for any interval we can "cover" it by using these predefined intervals. That is, for any $[x, y] \subseteq [0, T-1]$, there exist some $S_1, \ldots, S_j \in \mathcal{S}$ such that $[x, y] = S_1 \cup \cdots \cup S_j$. This is exactly the idea of subset-cover broadcast encryption, albeit in our case we deal only with sets that are intervals. We will therefore utilize a subset-cover system which permits efficient covering for interval sets. The subset-cover system proposed in [2] allows exactly this: for any interval, we can cover by using at most two predefined sets (*i.e.,* $j \leq 2$ in the above union). Hence, the ciphertext size of the resulting TSE will be only at most twice of that of FSE. In the following subsection, we first capture TSE schemes that allow restricted types of the form $[A, *]$ and $[*, D]$ as future TSE and past TSE, respectively.

### 4.1 Future TSE and Past TSE

In this subsection, we introduce two special classes of TSE, which we call *future time-specific encryption* (FTSE) and *past time-specific encryption* (PTSE), that we will use as "intermediate" building blocks for our generic construction of a TSE scheme from an FSE scheme. Using FTSE and PTSE, the description of our generic construction can be simplified. We also show how to generically construct these schemes from an FSE scheme.

FTSE (resp. PTSE) is a special class of a TSE scheme in which any ciphertext for time $t'$ can be decrypted by using a TIK for time $t$ as long as $t' \geq t$ (resp. $t' \leq t$). FTSE (resp. PTSE) can be viewed as a TSE scheme whose starting time $t_L$ (resp. closing time $t_R$) of a DTI is always fixed to be 0 (resp. $T-1$). An FTSE scheme (resp. a PTSE scheme) consists of the four algorithms ($\mathtt{FTSE.Setup}, \mathtt{FTSE.Ext}, \mathtt{FTSE.Enc}, \mathtt{FTSE.Dec}$) (resp. ($\mathtt{PTSE.Setup}, \mathtt{PTSE.Ext}, \mathtt{PTSE.Enc}, \mathtt{PTSE.Dec}$)) that are defined in the same way as those for a TSE scheme, with the following exceptions: Since the starting time $t_L$ (resp. the closing time $t_R$) of a DTI is always fixed to be 0 (resp. $T-1$), the encryption algorithm $\mathtt{FTSE.Enc}$ (resp. $\mathtt{PTSE.Enc}$) does not need to take $t_L$ (resp. $t_R$) as an input, and thus we denote by "$c \leftarrow \mathtt{FTSE.Enc}(mpk, t_R, M)$" (resp. "$c \leftarrow \mathtt{PTSE.Enc}(mpk, t_L, M)$") the

process of generating a ciphertext $c$ of a plaintext $M$ that can be decrypted using a TIK generated by $\mathtt{FTSE.Ext}(msk, t)$ (resp. $\mathtt{PTSE.Ext}(msk, t)$) with $t \in [0, t_R]$ (resp. $t \in [t_L, T-1]$). Furthermore, in order to stress that a TIK for time $t$ generated by the extraction algorithm can be used to decrypt all ciphertexts corresponding to time later than $t$ (resp. time $t$ or earlier), a TIK generated by $\mathtt{FTSE.Ext}(msk, t)$ (resp. $\mathtt{PTSE.Ext}(msk, t)$) is denoted by "$sk_{\geq t}$" (resp. "$sk_{\leq t}$").

*Correctness.* As a correctness requirement of an FTSE scheme, we require that for all $k \in \mathbb{N}$, $T \in \mathbb{N}$, all $(mpk, msk) \leftarrow \mathtt{FTSE.Setup}(1^k, T)$, all integers $t$ and $t_R$ such that $0 \leq t \leq t_R \leq T-1$, and all plaintexts $M$, it holds that $\mathtt{FTSE.Dec}(\mathtt{FTSE.Ext}(msk, t), \mathtt{FTSE.Enc}(mpk, t_R, M)) = M$.

In a similar way, as a correctness requirement of a PTSE scheme, we require that for all $k \in \mathbb{N}$, $T \in \mathbb{N}$, all $(mpk, msk) \leftarrow \mathtt{PTSE.Setup}(1^k, T)$, all integers $t$ and $t_L$ satisfying $0 \leq t_L \leq t \leq T-1$, and all plaintexts $M$, it holds that $\mathtt{PTSE.Dec}(\mathtt{PTSE.Ext}(msk, t), \mathtt{PTSE.Enc}(mpk, t_L, M)) = M$.

*Security Definitions.* IND-CPA security of an FTSE scheme and that of a PTSE scheme is defined analogously to that of a TSE scheme.

*Generic Constructions.* We can construct an FTSE scheme by using an FSE scheme ($\mathtt{FSE.Gen}, \mathtt{FSE.Upd}, \mathtt{FSE.Enc}, \mathtt{FSE.Dec}$) as shown in Fig. 3. Since the following theorem is straightforward from the security and the functionality of an FSE scheme, we omit the proof.

**Theorem 2.** *If the building block FSE scheme is IND-CPA secure, then the FTSE scheme constructed as in Fig. 3 is IND-CPA secure.*

We can also easily obtain a PTSE scheme from an FTSE scheme by "reversing" the role of time in FTSE, i.e., regarding a time $t$ in an FTSE scheme as a time $T-t-1$ for a PTSE scheme. This means that we also have a generic construction of a PTSE scheme from an FSE scheme. More specifically, we identity a TIK $sk_{\leq t}$ with a TIK $sk_{\geq T-t-1}$ of an FTSE scheme. Furthermore, $\mathtt{PTSE.Enc}(mpk, t', M)$ internally runs $\mathtt{FTSE.Enc}(mpk, T-t'-1, M)$. Since the construction we explain here is fairly intuitive and straightforward, we omit the detailed description of the construction.

### 4.2 Generic Construction

Here, we show our generic construction of a TSE scheme from an FSE scheme.

*Notation for Binary Trees.* Let $\lambda \in \mathbb{N}$, and let $T = 2^\lambda$. Our generic construction uses a binary tree as its internal structure, and we introduce several notation regarding them. Consider the complete binary tree with $2^{\lambda+1} - 1$ nodes. We number all the internal nodes (i.e. nodes that are not leaves) from the root in the breast first order (from left to right in numerical order), with the root node being 1. Furthermore, we also put 0 to the root node for convenience (and thus the root node has indices 0 and 1 at the same time). We will later put numbers

| $\mathtt{FTSE.Setup}(1^k, T)$ : | $\mathtt{FTSE.Ext}(msk, t)$ : |
|---|---|
| $(pk, d_0) \leftarrow \mathtt{FSE.Gen}(1^k, T)$ | $d_t \leftarrow \mathtt{FSE.Upd}(pk, 0, t, msk)$ |
| Return $(mpk, msk) \leftarrow (pk, d_0)$ | Return $sk_{\geq t} \leftarrow (t, d_t)$ |
| $\mathtt{FTSE.Enc}(mpk, t_R, M)$ : | $\mathtt{FTSE.Dec}(sk_{\geq t}, C)$ : |
| $c \leftarrow \mathtt{FSE.Enc}(mpk, t_R, M)$ | Parse $sk_{\geq t}$ as $(t, d_t)$ and $C$ as $(t_R, c)$ |
| Return $C \leftarrow (t_R, c)$ | If $t_R < t$ then return $\perp$ |
| | $d_{t_R} \leftarrow \mathtt{FSE.Upd}(pk, t, t_R, d_t)$ |
| | Return $\mathtt{FSE.Dec}(d_{t_R}, c)$ |

**Fig. 3.** Generic construction of FTSE from FSE

also for leaves, and thus in order not to mix up with them, we denote by $\mathtt{INT}$ the set of indices for the internal nodes. Namely, $\mathtt{INT} = \{0, 1, \ldots, 2^\lambda - 1\}$.

Let $\mathtt{LEFT}$ and $\mathtt{RIGHT}$ be subsets of $\mathtt{INT}$ defined as follows: $0 \in \mathtt{LEFT}$, $1 \in \mathtt{RIGHT}$, and for any remaining node $v \in \mathtt{INT} \backslash \{0, 1\}$, if $v$ is the left node of its parent node, then $v \in \mathtt{LEFT}$, otherwise $v \in \mathtt{RIGHT}$.

We next number the leaves from left to right in numerical order, with the leftmost node being 0 (and thus the rightmost being $T - 1$). For $v \in \mathtt{INT}$, let $\ell_v$ (resp. $r_v$) be the index of the leftmost (resp. rightmost) leaf node that is a descendant of $v$. That is, we have $\ell_v = (v \mod 2^{\mathsf{depth}(v)}) \cdot 2^{\lambda - \mathsf{depth}(v)}$ and $r_v = \ell_v + 2^{\lambda - \mathsf{depth}(v)} - 1$, where $\mathsf{depth}(v)$ is defined as the "depth of the node with the root node being depth 0".

For $v \in \mathtt{INT}$, we define the corresponding set $S_v$ of indices of leaves by:

$$S_v = \begin{cases} (0 \leftharpoondown (2^\lambda - 1)) & \text{If } v = 0 \\ ((\ell_v + 1) \leftharpoondown r_v) & \text{If } v \in \mathtt{LEFT} \backslash \{0\} \\ (\ell_v \rightharpoonup (r_v - 1)) & \text{If } v \in \mathtt{RIGHT} \end{cases}$$

where we use the followng notations: $(i \rightharpoonup j) := \{i, i+1, \ldots, j\}$, $(i \leftharpoondown j) := \{j, j-1, \ldots, i\}$, $(i \rightharpoonup i) := \{i\}$, and $(i \leftharpoondown i) := \{i\}$.

Finally, for $v \in \mathtt{INT}$, we let $\widetilde{\ell}_v$ and $\widetilde{r}_v$ be the smallest index and the largest index in the set $S_v$, respectively.

*Generic Construction.* For simplicity, our TSE scheme is parameterized by an integer $\lambda \in \mathbb{N}$ and supports the total number of time periods $T = 2^\lambda$. Let $(\mathtt{FTSE.Setup}, \mathtt{FTSE.Ext}, \mathtt{FTSE.Enc}, \mathtt{FTSE.Dec})$ be an FTSE scheme and let $(\mathtt{PTSE.Setup}, \mathtt{PTSE.Ext}, \mathtt{PTSE.Enc}, \mathtt{PTSE.Dec})$ be a PTSE scheme. Using these as building blocks, we construct a TSE scheme as in Fig. 4. In the construction, we use the following notations: For each DTI $[t_L, t_R]$, we define the corresponding left-index $v_L \in \mathtt{LEFT}$ and the right-index $v_R \in \mathtt{RIGHT}$, that determine which instance(s) of the building block FTSE and/or PTSE schemes are used to encrypt a message, by:

$$v_L = \min\{v \in \mathtt{LEFT} : \widetilde{r}_v \in [t_L, t_R]\}$$

$$v_R = \min\{v \in \mathtt{RIGHT} : \widetilde{\ell}_v \in [t_L, t_R]\}$$

$\mathtt{TSE.Setup}(1^k, T)$ :
For $v \in \mathtt{LEFT}$:
  $(mpk_v, msk_v) \leftarrow \mathtt{PTSE.Setup}(1^k, |S_v|)$
For $v \in \mathtt{RIGHT}$:
  $(mpk_v, msk_v) \leftarrow \mathtt{FTSE.Setup}(1^k, |S_v|)$
$MPK \leftarrow \{mpk_v\}_{v \in \mathtt{INT}}$
$MSK \leftarrow \{msk_v\}_{v \in \mathtt{INT}}$
Return $(MPK, MSK)$

$\mathtt{TSE.Enc}(MPK, [t_L, t_R], M)$ :
$v_L \leftarrow \min\{v \in \mathtt{LEFT} : \widetilde{r}_v \in [t_L, t_R]\}$
$v_R \leftarrow \min\{v \in \mathtt{RIGHT} : \widetilde{\ell}_v \in [t_L, t_R]\}$
If $\mathsf{depth}(v_L) = \mathsf{depth}(v_R)$ then
  If $v_L = 0$ then
    $c_L \leftarrow \mathtt{PTSE.Enc}(mpk_{v_L}, t_L - \widetilde{\ell}_{v_L}, M)$
    $c_R \leftarrow \emptyset$
  Else (i.e. $v_L \neq 0$) then
    $c_L \leftarrow \mathtt{PTSE.Enc}(mpk_{v_L}, t_L - \widetilde{\ell}_{v_L}, M)$
    $c_R \leftarrow \mathtt{FTSE.Enc}(mpk_{v_R}, t_R - \widetilde{\ell}_{v_R}, M)$
  End If
Else If $\mathsf{depth}(v_L) < \mathsf{depth}(v_R)$ then
  $c_L \leftarrow \mathtt{PTSE.Enc}(mpk_{v_L}, t_L - \widetilde{\ell}_{v_L}, M)$
  $c_R \leftarrow \emptyset$
Else (i.e. $\mathsf{depth}(v_L) > \mathsf{depth}(v_R)$)
  $c_L \leftarrow \emptyset$
  $c_R \leftarrow \mathtt{FTSE.Enc}(mpk_{v_R}, t_R - \widetilde{\ell}_{v_R}, M)$
End If
Return $C \leftarrow ([t_L, t_R], c_L, c_R)$

$\mathtt{TSE.Ext}(MSK, t)$ :
Let $\mathtt{NODES}(t) = \{v \in \mathtt{INT} : t \in S_v\}$
For $v \in \mathtt{LEFT} \cap \mathtt{NODES}(t)$:
  $sk^{(v)}_{\leq t-\widetilde{\ell}_v} \leftarrow \mathtt{PTSE.Ext}(msk_v, t - \widetilde{\ell}_v)$
$SK_{t,L} \leftarrow \{sk^{(v)}_{\leq t-\widetilde{\ell}_v}\}_{v \in \mathtt{LEFT} \cap \mathtt{NODES}(t)}$
For $v \in \mathtt{RIGHT} \cap \mathtt{NODES}(t)$:
  $sk^{(v)}_{\geq t-\widetilde{\ell}_v} \leftarrow \mathtt{FTSE.Ext}(msk_v, t - \widetilde{\ell}_v)$
$SK_{t,R} \leftarrow \{sk^{(v)}_{\geq t-\widetilde{\ell}_v}\}_{v \in \mathtt{RIGHT} \cap \mathtt{NODES}(t)}$
$SK_t \leftarrow (t, SK_{t,L}, SK_{t,R})$
Return $SK_t$

$\mathtt{TSE.Dec}(SK_t, C)$ :
Parse $SK_t$ as $(t, SK_{t,L}, SK_{t,R})$
Let $\mathtt{NODES}(t) = \{v \in \mathtt{INT} : t \in S_v\}$
Parse $SK_{t,L}$ as $\{sk^{(v)}_{\leq t-\widetilde{\ell}_v}\}_{v \in \mathtt{LEFT} \cap \mathtt{NODES}(t)}$
Parse $SK_{t,R}$ as $\{sk^{(v)}_{\geq t-\widetilde{\ell}_v}\}_{v \in \mathtt{RIGHT} \cap \mathtt{NODES}(t)}$
Parse $C$ as $([t_L, t_R], c_L, c_R)$
If parsing fails or $t \notin [t_L, t_R]$
  then return $\perp$
$v_L \leftarrow \min\{v \in \mathtt{LEFT} : \widetilde{r}_v \in [t_L, t_R]\}$
$v_R \leftarrow \min\{v \in \mathtt{RIGHT} : \widetilde{\ell}_v \in [t_L, t_R]\}$
$v \leftarrow \min(\mathtt{NODES}(t) \cap \{v_L, v_R\})$
If $v = \emptyset$ then return $\perp$
If $v \in \mathtt{LEFT}$ then
  return $\mathtt{PTSE.Dec}(sk^{(v)}_{\leq t-\widetilde{\ell}_v}, c_L)$
Else (i.e. $v \in \mathtt{RIGHT}$)
  return $\mathtt{FTSE.Dec}(sk^{(v)}_{\geq t-\widetilde{\ell}_v}, c_R)$
End If

**Fig. 4.** Generic construction of TSE from FTSE and PTSE

Furthermore, for each $t \in [0, T-1]$, we define the set $\mathtt{NODES}(t)$ of internal nodes that determines which instance(s) of the building block FTSE and PTSE schemes are used to generate a TIK (for a TSE) in the extraction algorithm, by:

$$\mathtt{NODES}(t) = \{v \in \mathtt{INT} : t \in S_v\}$$

Our scheme is IND-CPA secure assuming that the underlying FTSE and PTSE schemes are both IND-CPA secure (the proof is given in the full version). Since Fig. 4 might look slightly complicated, in Appendix B we show the instantiation of our TSE scheme in case $T = 2^3$ (see also Fig. 5 there).

**Theorem 3.** *If the FTSE scheme and the PTSE scheme are both IND-CPA secure, then the proposed TSE scheme constructed as in Fig. 4 is IND-CPA secure.*

**Table 1.** Efficiency comparison for TSE schemes. $T$ is the size of the time space. $|g|$ denotes the length of a group element. $|g_T|$ denotes the length of an element in $\mathbb{G}_T$. For $[a, b, c]$, $a$ denotes the number of pairings, $b$ denotes the number of exponentiations, $c$ denotes the number of multiplications.

| | Public Param. Size | Ciphertext Overhead | TIK Size | Encryption Cost | Decryption Cost |
|---|---|---|---|---|---|
| **Ours in Appendix A** | $O(\log T)\ \|g\|$ $+\ \|g_T\|$ | $3\|g\|$ | $O(\log^2 T)\|g\|$ $+\ \|\mathbb{Z}_p\|$ | $[0, O(\log T),$ $O(\log T)]$ | $[3, O(\log T),$ $O(\log T)]$ |
| **Ours §4** ([4]+[7]+§4.3) | $O(\log T)\ \|g\|$ $+\ \|g_T\|$ | $4\|g\|$ | $O(\log^3 T)\|g\|$ $+\ \|\mathbb{Z}_p\|$ | $[0, O(\log T),$ $O(\log T)]$ | $[2, O(\log T),$ $O(\log T)]$ |
| PQ-IBE [22] + Waters [25] | $4\|g\| + \|g_T\|$ | $O(T)\|g\|$ | $O(\log T)\|g\|$ $+\ \|\mathbb{Z}_p\|$ | $[0, O(T),$ $O(T)]$ | $[2, 0, 2]$ |
| PQ-IBE [22] + Gentry [16] | $3\|g\| + \|g_T\|$ | $O(T)\|g\|$ $+\ O(T)\|g_T\|$ | $O(\log T)\|g\|$ $+\ O(\log T)\|\mathbb{Z}_p\|$ | $[0, O(T),$ $O(T)]$ | $[1, 0, 2]$ |
| PQ-BE [22] + GW [17] | $O(T)\|g\|$ $+\ \|g_T\|$ | $8\|g\| + 2\|t_T\|$ | $\|g\| + \|\mathbb{Z}_p\|$ | $[0, 5, O(T)]$ $[0, 5, O(T)]$ | $[2, 3, O(T)]$ $[2, 3, O(T)]$ |
| PQ-BE [22] + BGW1 [6] | $O(T)\|g\|$ $+\ \|g_T\|$ | $2\|g\|$ | $\|g\| + \|\mathbb{Z}_p\|$ | $[0, 2, O(T)]$ $[0, 2, O(T)]$ | $[2, 0, O(T)]$ $[2, 0, O(T)]$ |
| PQ-BE [22] + BGW2 [6] | $O(\sqrt{T})\|g\|$ $+\ \|g_T\|$ | $O(\sqrt{T})\|g\|$ | $\|g\| + \|\mathbb{Z}_p\|$ | $[0, O(\sqrt{T}),$ $O(T)]$ | $[2, 0,$ $O(\sqrt{T})]$ |
| PQ-BE [22] + Waters [26] | $O(T)\|g\|$ $+\ \|g_T\|$ | $8\|g\|$ | $O(T)\|g\| + \|\mathbb{Z}_p\|$ | $[0, 12, O(T)]$ $[0, 12, O(T)]$ | $[9, 0, O(T)]$ $[9, 0, O(T)]$ |

### 4.3 Extension Using HIBE

The proposed generic construction shown in Sect. 4.2 uses $T$ independent instances of the underlying FSE scheme (assuming that the building block FTSE and PTSE schemes are instantiated with an FSE scheme), and thus the size of the master public key $MPK$ grows linearly in the total number of time periods $T$. However, if the underlying FSE scheme is furthermore instantiated from an HIBE scheme, we can use the trick of sharing the public parameter (by using the first-level identities for indices for each "independent" FSE scheme) explained in the second last paragraph in Sect. 2.2. In this case, the size of the master public key of the constructed TSE scheme does not depend on the size of the time space $|\mathsf{T}|$, but becomes identical to that of the underlying HIBE scheme. Using this trick with the full BBG-FSE (obtained via the HIBE-to-FSE transformation of [7]), we still obtain a TSE scheme whose parameter size and computational costs are all polylogarithmic in the number of time periods $T$.

## 5 Comparison

Table 1 shows an efficiency comparison among TSE schemes. We compare our scheme in Appendix A and an instantiation obtained from our generic construction in Sect. 4 in which the full BBG-FSE scheme is used together with the extension explained in Sect. 4.3, with the existing TSE schemes. Here, for the existing TSE schemes, we choose the concrete instantiations of TSE schemes obtained from the

generic construction from IBE schemes (denoted "PQ-IBE") and the generic construction from BE schemes (denoted "PQ-BE") both proposed by Paterson and Quaglia [22]. For concrete IBE schemes, we choose the schemes by Waters [25] and by Gentry [16], and for concrete BE schemes we choose the schemes by Boneh, Gentry, and Waters [6] (BGW1 was proposed in Sect. 3.1 of [6], and BGW2 was proposed in Sect. 3.2 of [6]), by Boneh and Gentry [17], and Waters [26].

As seen in Table 1, our schemes yield both better computational cost for encryption and short ciphertext length than those of the PQ-IBE schemes. In particular, our schemes have constant ciphertext overhead, while the PQ-IBE schemes have the ciphertext overhead of $O(T)$ group elements. Compared to the PQ-BE schemes, our schemes are superior in the size of public parameter, i.e. our schemes have the public parameter size of $O(\log T)$, while the PQ-BE schemes have the public parameter size of $O(T)$. Comparing the scheme in Appendix A and the instantiation from the generic construction in Sect. 4, the former scheme has shorter TIK size, i.e. the scheme in Appendix A has the TIK size of $O(\log^2 T)$ and the scheme in Sect. 4 has the TIK size of $O(\log^3 T)$. Furthermore, most notably, both of our schemes have at most poly-logarithmic size/cost in all measures in the table, which has not been achieved by any of the existing TSE schemes. Therefore, we see that our schemes have a feature which has not been achieved by all of the previous TSE schemes, and due to our results, a system designer can choose the parameters regarding TSE schemes that he/she wants to optimize more flexibly. We believe that our results will potentially broaden the applicability of TSE.

Lastly, we remark that both of our TSE schemes in Table 1 have the reduction costs of at least $O(T^2)$, while the PQ-IBE scheme instantiated with Gentry's IBE scheme [16] and the PQ-BE scheme instantiated with the Gentry-Waters BE scheme [17] have tight security reductions to their underlying hardness assumptions. However, all the TSE schemes mentioned here require non-static $\ell$-type assumptions (e.g. the decisional $\ell$-wBDHI in our case). It would be interesting to clarify whether it is possible to construct a TSE scheme whose security can be tightly reduced to more standard "static" assumptions such as decisional linear (DLIN) and decisional bilinear Diffie-Hellman (DBDH), and whose efficiency (parameter sizes and computational costs) is comparable to our schemes.

## References

1. Anderson, R.: Two remarks on public key cryptology. Invited Lecture, ACM CCS 1997 (1997), `http://www.cyphernet.org/cyphernomicon/chapter14/14.5.html`
2. Attrapadung, N., Imai, H.: Graph-Decomposition-Based Frameworks for Subset-Cover Broadcast Encryption and Efficient Instantiations. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 100–120. Springer, Heidelberg (2005)
3. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical Identity Based Encryption with Constant Size Ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
4. Boneh, D., Boyen, X., Goh, E.J.: Hierarchical identity based encryption with constant size ciphertext. Full version of [3]. Cryptology ePrint Archive: Report 2005/015 (2005)

5. 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)
6. Boneh, D., Gentry, C., Waters, B.: Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005)
7. Canetti, R., Halevi, S., Katz, J.: A Forward-secure Public-key Encryption Scheme. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 646–646. Springer, Heidelberg (2003)
8. Cathalo, J., Libert, B., Quisquater, J.J.: Efficient and Non-interactive Timed-Release Encryption. In: Qing, S., Mao, W., López, J., Wang, G. (eds.) ICICS 2005. LNCS, vol. 3783, pp. 291–303. Springer, Heidelberg (2005)
9. Chan, A.C.F., Blake, I.F.: Scalable, server-passive, user-anonymous timed release cryptography. In: Proceedings. 25th IEEE International Conference on ICDCS 2005, pp. 504–513. IEEE (2005)
10. Cheon, J.H., Hopper, N., Kim, Y., Osipkov, I.: Provably secure timed-release public key encryption. ACM Trans. Inf. Syst. Secure. 11(2) (2008)
11. Chow, S., Roth, V., Rieffel, E.: General Certificateless Encryption and Timed-Release Encryption. In: Ostrovsky, R., De Prisco, R., Visconti, I. (eds.) SCN 2008. LNCS, vol. 5229, pp. 126–143. Springer, Heidelberg (2008)
12. Dent, A., Tang, Q.: Revisiting the Security Model for Timed-Release Encryption with Pre-open Capability. In: Garay, J.A., Lenstra, A.K., Mambo, M., Peralta, R. (eds.) ISC 2007. LNCS, vol. 4779, pp. 158–174. Springer, Heidelberg (2007)
13. Dodis, Y., Katz, J.: Chosen-Ciphertext Security of Multiple Encryption. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 188–209. Springer, Heidelberg (2005)
14. Emura, K., Miyaji, A., Omote, K.: Adaptive Secure-Channel Free Public-Key Encryption with Keyword Search Implies Timed Release Encryption. In: Lai, X., Zhou, J., Li, H. (eds.) ISC 2011. LNCS, vol. 7001, pp. 102–118. Springer, Heidelberg (2011)
15. Fiat, A., Naor, M.: Broadcast Encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994)
16. Gentry, C.: Practical Identity-Based Encryption Without Random Oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
17. Gentry, C., Waters, B.: Adaptive Security in Broadcast Encryption Systems (with Short Ciphertexts). In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 171–188. Springer, Heidelberg (2009)
18. Hwang, Y., Yum, D., Lee, P.: Timed-Release Encryption with Pre-open Capability and Its Application to Certified E-mail System. In: Zhou, J., López, J., Deng, R.H., Bao, F. (eds.) ISC 2005. LNCS, vol. 3650, pp. 344–358. Springer, Heidelberg (2005)
19. Matsuda, T., Nakai, Y., Matsuura, K.: Efficient Generic Constructions of Timed-Release Encryption with Pre-open Capability. In: Joye, M., Miyaji, A., Otsuka, A. (eds.) Pairing 2010. LNCS, vol. 6487, pp. 225–245. Springer, Heidelberg (2010)
20. May, T.: Time-release crypto (1993) (manuscript), `http://www.cyphernet.org/cyphernomicon/chapter14/14.5.html`
21. Nakai, Y., Matsuda, T., Kitada, W., Matsuura, K.: A Generic Construction of Timed-Release Encryption with Pre-open Capability. In: Takagi, T., Mambo, M. (eds.) IWSEC 2009. LNCS, vol. 5824, pp. 53–70. Springer, Heidelberg (2009)
22. Paterson, K., Quaglia, E.: Time-Specific Encryption. In: Garay, J.A., De Prisco, R. (eds.) SCN 2010. LNCS, vol. 6280, pp. 1–16. Springer, Heidelberg (2010)
23. Rivest, R.L., Shamir, A., Wagner, D.A.: Time-lock puzzles and timed-release crypto. Technical Report MIT/LCS/TR-684 (1996)

24. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
25. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
26. Waters, B.: Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)
27. Zhang, R., Hanaoka, G., Shikata, J., Imai, H.: On the Security of Multiple Encryption or CCA-security+CCA-security=CCA-security? In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 360–374. Springer, Heidelberg (2004)

## A Main Concrete Construction

Here, we describe the full TSE scheme obtained by using the binary tree structures for the basic version of our scheme presented in Sect. 3. As noted earlier, this construction is obtained by applying the technique from the HIBE-to-FSE transformation by Canetti et al. [7] to the basic version of the proposed scheme for reducing the sizes of the public parameter and TIKs.

Let $\ell \in \mathbb{N}$. Consider two complete binary trees $B_1$ and $B_2$ with $T = 2^\ell - 1$ nodes, where $T$ will be the number of time periods supported by the proposed TSE construction. The nodes in those binary trees are numbered according to a pre-order traversal in an incremental order, with the root node of $B_1$ being 1 and that of $B_2$ being $T + 1$. Then, consider the binary tree $B$ with $2T + 1$ nodes in which the children of the root nodes are the root nodes of $B_1$ and $B_2$, with $B_1$ being left. (That is, $B$ has $B_1$ and $B_2$ as sub trees.) For convenience, we put the number $2T + 1$ to the root node of $B$. Intuitively, each subtree in $B$ will correspond to one instantiation of FSE obtained via the HIBE-to-FSE transformation of Canetti et al. [7] to the BBG HIBE scheme (and will also correspond to one chain in our basic construction shown in Sect. 3.2).

We need to introduce vectors "$TV_t$" and sets "$\mathtt{TVSet}_t$" (for $t \in [1, 2T]$). $TV_t$ is the vector consisting of the indices corresponding to the nodes included in the path from the node $t$ to the root node (of $B$). For $t \in [1, 2T]$, the set $\mathtt{TVSet}_t$ defined as follows: $\mathtt{TVSet}_1 = \{TV_1\}$, $\mathtt{TVSet}_{T+1} = \{TV_{T+1}\}$. Recursively, for $t \in [1, 2T] \backslash \{1, T + 1\}$, $\mathtt{TVSet}_{t+1}$ is defined depending on $\mathtt{TVSet}_t$ as follows: Let $s = \min\{u : TV_u \in \mathtt{TVSet}_t\}$. If $TV_s$ is a leaf node, then $\mathtt{TVSet}_{t+1}$ is obtained by removing the vector $TV_s$ from the set $\mathtt{TVSet}_t$. Otherwise, let $s_F$ (resp. $s_B$) be the index of the left (resp. right) node of the node $s$. $\mathtt{TVSet}_{t+1}$ is the set obtained by removing $TV_s$ from and adding $TV_{s_F}$ and $TV_{s_B}$ to the set $\mathtt{TVSet}_t$.

Let $(\mathbb{G}, \mathbb{G}_T, e)$ be bilinear maps, and let $T = 2^\ell - 1$ be a polynomial that indicates the number of time periods. Using the above notations, We describe our TSE scheme in the following:

$\mathtt{TSE.Setup}(1^k, T = 2^\ell - 1)$**:** Pick $\alpha, \beta \xleftarrow{\mathrm{U}} \mathbb{Z}_p$, $g_{2,F}, g_{2,B}, h_0, \cdots, h_\ell \xleftarrow{\mathrm{U}} \mathbb{G}$. Then compute $MSK \leftarrow g^{\alpha\beta}$ and

$$MPK \leftarrow (g, g_1 \leftarrow g^\alpha, g_{2,F}, g_{2,B}, h_0, \cdots, h_\ell, P \leftarrow e(g^\alpha, g^\beta)),$$

and return $(MPK, MSK)$.

$\mathtt{TSE.Ext}(MSK, t)$**:** Firstly, pick $\xi \xleftarrow{\mathrm{U}} \mathbb{Z}_p$.

For each $TV = (J_0, J_1, \cdots, J_m) \in \mathtt{TVSet}_{t+1}$: pick $r_F \xleftarrow{\mathrm{U}} \mathbb{Z}_p$, and compute

$$d_{TV} \leftarrow (g^{\alpha\beta+\xi} \cdot (\prod_{i=0}^{m} h_i^{J_i} \cdot g_{2,F})^{r_F}, g^{r_F}, h_{m+1}^{r_F}, \cdots, h_\ell^{r_F}).$$

For each $TV' = (K_0, K_1, \cdots, K_n) \in \mathtt{TVSet}_{2T-t}$: pick $r_B \xleftarrow{\mathrm{U}} \mathbb{Z}_p$, and compute

$$d_{TV'} \leftarrow (g^{-\xi} \cdot (\prod_{i=0}^{n} h_i^{K_i} \cdot g_{2,B})^{r_B}, g^{r_B}, h_{n+1}^{r_B}, \cdots, h_\ell^{r_B}).$$

Finally, set $SK_{t,L} \leftarrow \{d_{TV}\}_{TV \in \mathtt{TVSet}_{t+1}}$ and $SK_{t,R} \leftarrow \{d_{TV'}\}_{TV' \in \mathtt{TVSet}_{2T-t}}$, and return $SK_t = (t, SK_{t,L}, SK_{t,R})$.

$\mathtt{TSE.Enc}(MPK, [t_L, t_R], M)$**:** Let $TV_{t_R+1} = (J_0, J_1, \cdots, J_m)$ and $TV_{2T-t_L} = (K_0, K_1, \cdots, K_n)$. Pick $s \xleftarrow{\mathrm{U}} \mathbb{Z}_p$, compute

$$(C_1, C_2, C_3, C_4) \leftarrow (P^s \cdot M, g^s, (\prod_{i=0}^{m} h_i^{J_i} \cdot g_{2,F})^s, (\prod_{i=0}^{n} h_i^{K_i} \cdot g_{2,B})^s)$$

and return $C = (C_1, C_2, C_3, C_4, [t_L, t_R])$.

$\mathtt{TSE.Dec}(SK_t, C)$**:** Let $SK_t = (t, SK_{t,L}, SK_{t,R})$ and $C = (C_1, C_2, C_3, C_4, C_5)$. If $t \notin C_5$, then return $\perp$. Otherwise, retrieve $d_{TV_{t_R+1}} = (L_1, L_2, \cdots)$ and $d_{TV_{2T-t_L}} = (R_1, R_2, \cdots)$ from $SK_{t,L}$ and $SK_{t,R}$, respectively. Compute

$$M = \frac{C_1 \cdot e(L_2, C_3) \cdot e(R_2, C_4)}{e(L_1 \cdot R_1, C_2)}$$

and return $M$.

The security is guaranteed by the following (the proof is given in the full version).

**Theorem 4.** *If the decisional $(\ell+1)$-wBDHI assumption holds in $(\mathbb{G}, \mathbb{G}_T, e)$, then the above TSE scheme (with $T = 2^\ell - 1$ time periods) is IND-CPA secure.*

## B Toy Example of Our Generic Construction

In order to better understand our generic construction in Sect. 4.2, here we describe a toy example of our generic construction in which $T = 2^3$. See also Fig. 5 for the illustration that represents the "directions" (or, "realms" in other words) that the secret keys from the underlying FTSE and PTSE schemes can cover. Note that in this example, $\mathtt{LEFT} = \{0, 2, 4, 6\}$, and $\mathtt{RIGHT} = \{1, 3, 5, 7\}$.

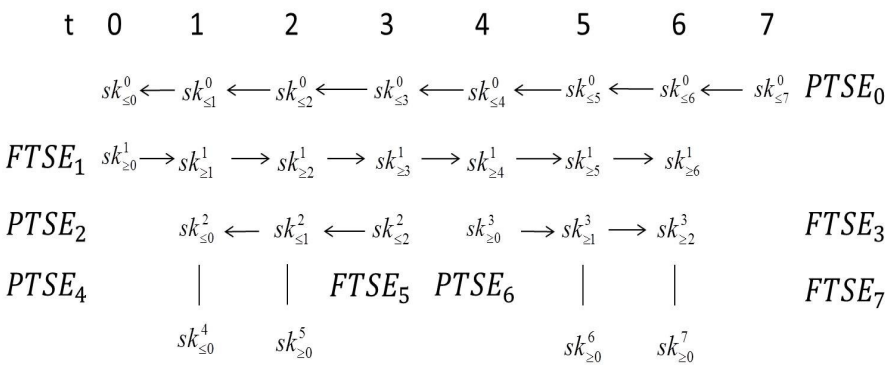


**Fig. 5.** Illustration for our generic construction in case $T = 2^3$

$\mathtt{TSE.Setup}(1^k, T)$**:** Run the setup algorithms of the underlying FTSE and PTSE schemes as follows:

$(mpk_0, msk_0) \leftarrow \mathtt{PTSE.Setup}(1^k, 8)$; $(mpk_1, msk_1) \leftarrow \mathtt{FTSE.Setup}(1^k, 7)$
$(mpk_2, msk_2) \leftarrow \mathtt{PTSE.Setup}(1^k, 3)$; $(mpk_3, msk_3) \leftarrow \mathtt{FTSE.Setup}(1^k, 3)$
$(mpk_4, msk_4) \leftarrow \mathtt{PTSE.Setup}(1^k, 1)$; $(mpk_5, msk_5) \leftarrow \mathtt{FTSE.Setup}(1^k, 1)$
$(mpk_6, msk_6) \leftarrow \mathtt{PTSE.Setup}(1^k, 1)$; $(mpk_7, msk_7) \leftarrow \mathtt{FTSE.Setup}(1^k, 1)$
$MPK \leftarrow (mpk_0, mpk_1, \ldots, mpk_7)$; $MSK \leftarrow (msk_0, msk_1, \ldots, msk_7)$
Return $(MPK, MSK)$.

$\mathtt{TSE.Ext}(msk, t)$**:** The algorithm sets the TIK $SK_t$ corresponding to the column of the time $t$ in Fig. 5 to the secret keys of FTSE and PTSE. For example,

- $SK_0 = (0, SK_{0,L}, SK_{0,R})$ where $SK_{0,L} = sk^{(0)}_{\le 0}$ and $SK_{0,R} = sk^{(1)}_{\ge 0}$.
- $SK_1 = (1, SK_{1,L}, SK_{1,R})$ where $SK_{1,L} = (sk^{(0)}_{\le 1}, sk^{(2)}_{\le 0}, sk^{(4)}_{\le 0})$, and $SK_{1,R} = sk^{(1)}_{\ge 1}$.
- $SK_4 = (4, SK_{4,L}, SK_{4,R})$ where $SK_{4,L} = sk^{(0)}_{\le 4}$ and $SK_{4,R} = (sk^{(1)}_{\ge 4}, sk^{(3)}_{\ge 0})$

Note that $\mathtt{NODES}(0) = \{0, 1\}$, $\mathtt{NODES}(1) = \{0, 1, 2, 4\}$, and $\mathtt{NODES}(4) = \{0, 1, 3\}$.

$\mathtt{TSE.Enc}(mpk, [t_L, t_R], M)$**:** We exemplify the cases in which $[t_L, t_R] = [4, 7]$, $[4, 5]$, and $[2, 6]$ in the following:

- $C = ([4, 7], c_L, c_R)$, where $c_L \leftarrow \mathtt{PTSE.Enc}(mpk_0, 4, M)$ and $c_R \leftarrow \emptyset$. Note that $v_L = \min\{v \in \mathtt{LEFT} : \widetilde{r}_v \in [4, 7]\} = 0$ and thus $\mathsf{depth}(v_L) = 0$, while $v_R = \min\{v \in \mathtt{RIGHT} : \widetilde{\ell}_v \in [4, 7]\} = 3$ and thus $\mathsf{depth}(v_R) = 1$.
- $C = ([4, 5], c_L, c_R)$, where $c_L \leftarrow \emptyset$ and $c_R \leftarrow \mathtt{FTSE.Enc}(mpk_3, 1, M)$. Note that $v_L = \min\{v \in \mathtt{LEFT} : \widetilde{r}_v \in [4, 5]\} = 6$ and thus $\mathsf{depth}(v_L) = 2$, while $v_R = \min\{v \in \mathtt{RIGHT} : \widetilde{\ell}_v \in [4, 5]\} = 3$ and thus $\mathsf{depth}(v_R) = 1$.
- $C = ([2, 6], c_L, c_R)$, where $c_L \leftarrow \mathtt{PTSE.Enc}(mpk_2, 1, M)$ and $c_R \leftarrow \mathtt{FTSE.Enc}(mpk_3, 2, M)$. Note that, $v_L = \min\{v \in \mathtt{LEFT} : \widetilde{r}_v \in [2, 6]\} = 2$ and thus $\mathsf{depth}(v_L) = 1$, while $v_R = \min\{v \in \mathtt{RIGHT} : \widetilde{\ell}_v \in [2, 6]\} = 3$ and thus $\mathsf{depth}(v_R) = 1$.

$\mathtt{TSE.Dec}(SK_t, C)$**:** Using $SK_4 = (4, SK_{4,L}, SK_{4,R})$, we can decrypt the above (correctly generated) ciphertexts:

- If DTI is $[4, 7]$, run $M \leftarrow \mathtt{FTSE.Dec}(sk^{(0)}_{\leq 4}, c_L)$. Note that in this case, $\min(\mathtt{NODES}(4) \cap \{v_L, v_R\}) = 0 \in \mathtt{LEFT}$.
- If DTI is $[4, 5]$ or $[2, 6]$, run $M \leftarrow \mathtt{PTSE.Dec}(sk^{(3)}_{\geq 0}, c_R)$. Note that in both cases, $\min(\mathtt{NODES}(4) \cap \{v_L, v_R\}) = 3 \in \mathtt{RIGHT}$.