Time-Specific Encryption with Constant-Size Secret-Keys Secure under Standard Assumption

Masahito Ishizaka and Shinsaku Kiyomoto

KDDI Research, Inc.
{ma-ishizaka, kiyomoto}@kddi-research.jp

Abstract. In Time-Specific Encryption (TSE) [Paterson&Quaglia, SCN'10] system, each secret-key (resp. ciphertext) is associated with a time period $t \in [0, T-1]$ (resp. a time interval $[L, R]$ where $L, R \in [0, T-1]$). A ciphertext under $[L, R]$ is correctly decrypted by any secret-key for any time t included in the interval, i.e., $t \in [L, R]$. TSE's generic construction from identity-based encryption (IBE) (resp. hierarchical IBE (HIBE)) from which we obtain a concrete TSE scheme with secret-keys of size $O(\log T)|g|$ (resp. $O(\log^2 T)|g|$) and ciphertexts of size $O(\log T)|g|$ (resp. $O(1)|g|$) has been proposed in [Paterson&Quaglia, SCN'10] (resp. [Kasamatsu et al., SCN'12]), where $|g|$ denotes bit length of an element in a bilinear group $\mathbb{G}$. In this paper, we propose another TSE's generic construction from wildcarded identity-based encryption (WIBE). Differently from the original WIBE ([Abdalla et al., ICALP'06]), we consider WIBE w/o *(hierarchical) key-delegatability*. By instantiating the TSE's generic construction, we obtain the first concrete scheme with constant size secret-keys secure under a standard (static) assumption. Specifically, it has secret-keys of size $O(1)|g|$ and ciphertexts of size $O(\log^2 T)|g|$, and achieves security under the decisional bilinear Diffie-Hellman (DBDH) assumption.

Keywords: Time-specific encryption (TSE), Constant-sized secret-keys, The decisional bilinear Diffie-Hellman (DBDH) assumption, Wildcarded identity-based encryption (WIBE).

1 Introduction

Time-Specific Encryption (TSE) [15]. In a TSE system with time periods $T \in \mathbb{N}$, each secret-key (resp. ciphertext) is associated with a time period $t \in [0, T-1]$ (resp. a time interval $[L, R]$, where $L, R \in [0, T-1]$). Any ciphertext for $[L, R]$ can be correctly decrypted by any secret-key for any t such that $t \in [L, R]$. If we say that a TSE scheme is secure, that informally means that any probabilistic polynomial time (PPT) algorithm, given a ciphertext C^* of a plaintext m^* under an interval $[L^*, R^*]$, cannot get any information about m^*. This type of TSE, on whom we mainly focus in this paper, is called *plain setting* in [15].

Paterson and Quaglia [15] proposed a generic construction of TSE from identity-based encryption (IBE) [18]. Kasamatsu et al. [12, 13] proposed a generic construction of TSE from forward secure encryption (FE) [3, 9] and a more efficient concrete construction based on Boneh-Boyen-Goh hierarchical identity-based encryption (HIBE) [6]. Secret-key size of the IBE-based TSE is $O\left(\log T \cdot K\left(\Sigma_{\mathrm{IBE}}^{2T-1}\right)\right)$, where $\Sigma_{\mathrm{IBE}}^{n}$ denotes an IBE scheme with $n \in \mathbb{N}$ identities and $K\left(\Sigma_{\mathrm{IBE}}^{n}\right)$ denotes its size of secret-keys. This means that secret-key size of any concrete scheme obtained from the IBE-based generic construction of TSE cannot be constant (even if the secret-key size of the underlying IBE scheme is constant). By a similar reason, either secret-key size of any concrete TSE scheme obtained by the FE-based generic TSE construction or that of the concrete BBG HIBE-based TSE construction cannot be constant.

As pointed out by [15], broadcast encryption (BE) [10] is conceptually broader than TSE. As far as we know, master public-key size (and encryption/decryption cost) of currently-known BE schemes increase linearly with total number of users (e.g., one in Subsect. 3.1 in [7], one in [20], one in Subsect. 3.1 in [11]), square root of total number of users (e.g., one in Subsect. in [7]), or maximum cardinality of a set of users associated with a ciphertext (which determines users who can decrypt the ciphertext) (e.g., one in Subsect.

3.3 in [11]). Because of that, from currently-known BE schemes, we obtain TSE schemes whose master public-key size (and encryption/decryption cost) increase linearly with $O(\sqrt{T})$ at least.

It is rational that, based on the fact that TSE is more functionally limited than some existing primitives such as BE, attribute-based encryption [17] and functional encryption [8], we require TSE to be more (asymptotically) efficient than them. Precisely, in this paper, we focus on *(asymptotically) efficient* TSE schemes, whose master public-key/(user's) secret-key/ciphertext size and encryption/decryption cost are $O(\log^2 T)$ at most.

In this paper, we affirmatively solve the following open problem: *Can we construct a TSE scheme with total time periods T, whose secret-key size is $O(1)$, whose master public-key/ciphertext size and encryption/decryption cost are $O(\log^2 T)$ at most, and which is secure under standard (static) computational assumptions?*

Wildcarded Identity-Based Encryption (WIBE) [1, 2, 4]. WIBE is a generalization of HIBE, where each ciphertext is associated with a *wildcarded* identity $wID \in \{0, 1, *\}^L$ which can include some wildcard symbols $*$ and such ciphertext can be correctly decrypted by any secret-key associated with any identity $ID \in \{0, 1\}^L$ s.t. for every $i \in [1, L]$ s.t. $wID[i] \neq *$, it holds $wID[i] = ID[i]$. Abdalla and Birkett et al. [1, 2, 4] showed that by partially modifying existing HIBE schemes [5, 6, 19], we obtain WIBE schemes. Since they defined WIBE as a generalization of HIBE, their WIBE schemes are assumed to have *(hierarchical) key-delegatability*, which guarantees that there exists a polynomial time algorithm to derive a secret-key for an identity from any secret-key for any ancestor identity of the identity. In this paper, meanwhile, we define WIBE as a generalization of IBE, which implies that they lack (hierarchical) key-delegatability.

Our Contribution. We propose a generic construction of *Range Encryption (RE)* from WIBE w/o *key-delegatability*. RE is a generalization of TSE. Differently from TSE, RE handles not only a range $[L, R] \subseteq [0, T-1]$ s.t. $L \leq R$, but also a range $[L, R] \subseteq [0, T-1]$ s.t. $L > R$. From a normal TSE scheme, we can easily construct a RE scheme by making an encryptor divide a range $[L, R] \subseteq [0, T-1]$ s.t. $L > R$ into two subranges $[L, T-1]$ and $[0, R]$ and independently generate a ciphertext for each subrange. Secret-key size of the generic construction is described as $O\left(K\left(\Sigma_{\mathrm{WIBE}}^{\log T}\right)\right)$, where $\Sigma_{\mathrm{WIBE}}^{n}$ is a WIBE scheme whose bit length of an identity is $n \in \mathbb{N}$. Moreover, we show that by modifying Waters IBE scheme [19], we can construct a WIBE scheme (w/o key-delegatability), whose secret-keys consists of constant number of group elements, i.e., two group elements, and which is secure under standard assumptions, i.e., the decisional bilinear Diffie-Hellman (DBDH) assumption. By using the WIBE scheme to instantiate the generic construction of RE from WIBE, we obtain a RE (or TSE) scheme which can justify our claim that we certainly solve the open problem mentioned earlier.

Our Approach. It might be a surprise that it is not hard to obtain a RE scheme with constant-sized secret-keys. However, since the naive methodology has some disadvantages, we propose another improved methodology. Let us explain the details below.

Let PQ-IBE-TSE denote the IBE-based generic construction of TSE [15]. Let PQ-WIBE-TSE denote PQ-IBE-TSE, where we substitute WIBE for IBE. Let PQ-WIBE-RE denote the range encryption constructed by PQ-WIBE-TSE. We briefly explain PQ-WIBE-RE.

We consider a binary tree with depth $\log T$ (as shown in Fig. 3 in Sect. 5). A time period $t \in [0, T-1]$ corresponds to a leaf node with a bit string $t \in \{0, 1\}^{\log T}$ (which is the binary value of t). A secret-key for $t \in [0, T-1]$ is produced as a secret-key for the bit string (or identity) $t \in \{0, 1\}^{\log T}$ by using the key-generation algorithm of the WIBE scheme. Given a time interval $[L, R]$ where $L, R \in [0, T-1]$, we consider some wildcarded strings $\mathbb{T}_{[L,R]}$ which *covers* $[L, R]$ and whose cardinality is the minimum[1]. For instance, when $T = 8$ in Fig. 3, $\mathbb{T}_{[1,6]} = \{001, 01*, 10*, 110\}$. When we encrypt a plaintext under $[L, R]$, we encrypt the plaintext under each (wildcarded) bit string in $\mathbb{T}_{[L,R]}$ by using the encryption algorithm of the WIBE scheme.

[1]For the formal algorithm deriving such a set $\mathbb{T}_{[L,R]}$, we recommend the reader to refer to the original paper [15] or Sect. 5 of this paper.

We can easily prove that if the underlying WIBE scheme is secure, then PQ-WIBE-RE is secure. Secret-key size of PQ-WIBE-RE is $O\left(K\left(\Sigma_{\mathrm{WIBE}}^{\log T}\right)\right)$. There have already existed WIBE schemes (with key-delegatability) whose secret-key size is constant, e.g., Abdalla and Birkett et al.'s WIBE scheme [1, 2, 4] based on Boneh-Boyen-Goh HIBE [6]. Thus, by using such a WIBE scheme to instantiate PQ-WIBE-RE, we can obtain a RE scheme with constant-sized secret-keys.

However, such a naive methodology has the following disadvantages.

1. Ciphertext size of PQ-WIBE-RE can be smaller. In other words, there exists another WIBE-based generic construction of RE, from which we obtain a concrete RE scheme with smaller ciphertext size. We denote it by IK-WIBE-RE.
2. To the best of our knowledge, no WIBE scheme (with or without key-delegatability) with constant-sized secret-keys secure under standard (or static) assumptions has been proposed. For instance, Boneh-Boyen-Goh HIBE-based WIBE scheme [1, 2, 4] with constant-sized secret-keys has been proven to be secure under non-standard (or non-static) assumption.

Let us provide the details in the following 2 paragraphs.

IK-WIBE-RE. Roughly speaking, encryption process adopted in IK-WIBE-RE is the same as that adopted in PQ-WIBE-RE. Namely, that is a process, where we, given a plaintext m and an range $[L, R]$, derive a set of wildcarded IDs $\mathbb{T}_{[L,R]}$, encrypt m under each $wID \in \mathbb{T}_{[L,R]}$ (by using the encryption algorithm of $\Sigma_{\mathrm{WIBE}}^{\log T}$) to produce a ciphertext C_{wID}, and construct $C_{[L,R]}$ as $\{C_{wID} \mid wID \in \mathbb{T}_{[L,R]}\}$. Thus, size of the ciphertext is described as $\sum_{wID \in \mathbb{T}_{[L,R]}} |C_{wID}|$. If we adopt our WIBE scheme based on Waters IBE scheme [19], that is described as $\sum_{wID \in \mathbb{T}_{[L,R]}} \{|g_T| + (2 + |wID|_*)|g|\}$[2][3]. This implies that $|C_{[L,R]}|$ is determined solely by $\mathbb{T}_{[L,R]}$. Let us explain how we derive $\mathbb{T}_{[L,R]}$.

We introduce a binarizing algorithm $\mathtt{Binarize}_{\log T}$ which takes a numerical value (or time period) $t \in [0, T-1]$ and outputs a bit string $b \in \{0,1\}^{\log T}$. For instance, when $T = 32$, the relation between t and b is defined as shown in Fig. 5 in Subsect. 6.1. Then, we introduce a classifying algorithm $\mathtt{Classify}_{\log T}$ which takes a value $t \in [0, T-1]$ and outputs a class index in $[0, \log T]$. Specifically, it takes $t \in [0, T-1]$, and outputs $\log T$ if $t = 0$, or outputs $i \in [0, \log T - 1]$ if $t \bmod 2^{i+1} = 2^i$. (See Fig. 7.)

Given a range (or time interval) $[L, R]$, we firstly determine a value (called *divider*) $D \in [L, R]$ which divides the range into $[L, D-1]$ and $[D, R]$. Informally, D is the value which is classified as the class whose index is the largest among the values in $[L, R]$. Let $\mathtt{Divide}_{\log T}$ denote the algorithm which takes $[L, R]$ and outputs D.

Next, we derive sets of wildcarded IDs $\mathbb{T}_{[D,R]}$ and $\mathbb{T}_{[L,D-1]}$ for $[D, R]$ and $[L, D-1]$, respectively. Firstly, let us explain how to derive $\mathbb{T}_{[D,R]}$. We formally prove that $\forall L, R \in [0, T-1]$ with $D \leftarrow \mathtt{Divide}_{\log T}(L, R)$, $\exists k_1 \in [0, \mathtt{Classify}_{\log T}(D)]$, $\exists k_2 \in [0, k_1 - 1]$, $\cdots$, $\exists k_n \in [0, k_{n-1} - 1]$ s.t. $D - 1 + \sum_{l=1}^{n} 2^{k_l} = R$. Moreover, we prove that $\forall i \in [1, n]$, $\exists wID_i \in \{0, 1, *\}^{\log T}$ which covers a subrange[4] $[D + \sum_{l=1}^{i-1} 2^{k_l}, D - 1 + \sum_{l=1}^{i} 2^{k_l}]$ and satisfies $|wID_i|_* = k_i$. We set $\mathbb{T}_{[D,R]}$ as $\{wID_i \mid i \in [1, n]\}$. Likewise, we derive $\mathbb{T}_{[L,D-1]}$. We prove that $\forall L, R \in [0, T-1]$ with $D \leftarrow \mathtt{Divide}_{\log T}(L, R)$, $\exists k'_1 \in [0, \mathtt{Classify}_{\log T}(D) - 1]$, $\exists k'_2 \in [0, k'_1 - 1]$, $\cdots$, $\exists k'_{n'} \in [0, k'_{n'-1} - 1]$ s.t. $D - \sum_{l=1}^{n} 2^{k'_l} \bmod T = L$. Moreover, we prove that $\forall i \in [1, n']$, $\exists wID'_i \in \{0, 1, *\}^{\log T}$ which covers $[D - \sum_{l=1}^{i} 2^{k'_l} \bmod T, D - 1 - \sum_{l=1}^{i-1} 2^{k'_l} \bmod T]$ and satisfies $|wID'_i|_* = k'_i$. We set $\mathbb{T}_{[L,D-1]}$ as $\{wID'_i \mid i \in [1, n']\}$.

Finally, we derive $\mathbb{T}_{[L,R]}$ from $\mathbb{T}_{[L,D-1]}$ and $\mathbb{T}_{[D,R]}$. Although the most simple way is deriving a union set of the two sets, i.e., $\mathbb{T}_{[L,R]} := \mathbb{T}_{[L,D-1]} \bigcup \mathbb{T}_{[D,R]}$, we adopt the following another way. Let n^* denote the integer s.t. $[0 \leq n^* \leq \min(n, n')] \bigwedge_{i=1}^{n^*} \left[k_i = k'_i\right] \bigwedge \left[n^* < \min(n, n') \implies k_{n^*+1} \neq k'_{n^*+1}\right]$. We formally prove that $\forall i \in [1, n^*]$, wID_i and wID'_i can be merged into a new wID^*_i which covers both of the ranges covered by wID_i

[2] $|g|$ (resp. $|g_T|$) denotes bit length of an element in a bilinear group $\mathbb{G}$ (resp. $\mathbb{G}_T$) for a (symmetric) bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$.

[3] $|wID|_*$ denotes number of wildcard symbol $*$ in $wID \in \{0, 1, *\}^L$.

[4] $wID \in \{0, 1, *\}^L$ covers a range $[a, b]$ means that any value included in the range matches the wildcarded ID and any value excluded from the range does not match it.

and wID'_i, i.e., $[D + \sum_{l=1}^{i-1} 2^{k_l}, D - 1 + \sum_{l=1}^{i} 2^{k_l}] \bigcup [D - \sum_{l=1}^{i} 2^{k'_l} \bmod T, D - 1 - \sum_{l=1}^{i-1} 2^{k'_l} \bmod T]$, and satisfies $|wID^*_i|_* = k_i + 1$. In conclusion, we set $\mathbb{T}_{[L,R]}$ as $\mathbb{T}_{[L,D-1]} \bigcup \mathbb{T}_{[D,R]} \setminus_{i \in [1,n^*]} \{wID_i, wID'_i\} \bigcup_{i \in [1,n^*]} \{wID^*_i\}$.

Let us compare size of ciphertexts generated based on $\mathbb{T}_{[L,R]}$ with that on $\mathbb{T}_{[L,D-1]} \bigcup \mathbb{T}_{[D,R]}$. Let $|C_{L,R}|$ (resp. $|C_{L,D,R}|$) denote size of ciphertext generated based on $\mathbb{T}_{[L,R]}$ (resp. $\mathbb{T}_{[L,D-1]} \bigcup \mathbb{T}_{[D,R]}$). If $\mathbb{T}_{[L,R]} = \mathbb{T}_{[L,D-1]} \bigcup \mathbb{T}_{[D,R]}$, then $|C_{L,R}| = |C_{L,D,R}|$. Else if $\mathbb{T}_{[L,R]} \neq \mathbb{T}_{[L,D-1]} \bigcup \mathbb{T}_{[D,R]}$, then $|C_{L,R}| < |C_{L,D,R}|$. Especially, if $n^* = n = n'$, $|C_{L,R}|$ becomes approximately the half of $|C_{L,D,R}|$. For instance, when $[L, R]$ is $[1, T-2]$ (or $[2^{\log T-1}+1, 2^{\log T-1}-2]$) with $n^* = n = n' = \log T - 1$, $|C_{L,D,R}| = 2(\log T - 1)|g_T| + (\log^2 T + \log T - 2)|g|$ and $|C_{L,R}| = (\log T - 1)|g_T| + \frac{1}{2}(\log^2 T + 3\log T - 4)|g| = \frac{1}{2}|C_{L,D,R}| + (\log T - 1)|g|$. For instance, when $T = 2^{10}$ (resp. $T = 2^{20}$), $|C_{L,D,R}|$ becomes $18|g_T| + 108|g|$ (resp. $38|g_T| + 418|g|$) and $|C_{L,R}|$ becomes $9|g_T| + 63|g|$ (resp. $19|g_T| + 228|g|$). Note that the maximum $|C_{L,D,R}|$ is $2(\log T - 1)|g_T| + (\log^2 T + \log T - 2)|g|$ when $[L, R]$ is $[1, T-2]$ (or $[2^{\log T-1}+1, 2^{\log T-1}-2]$). Thus, in an asymptotic sense, $|C_{L,D,R}|$ is $O(\log T)|g_T| + O(\log^2 T)|g|$. Neither the maximum of $|C_{L,R}|$ nor the range $[L, R]$ maximizing $|C_{L,R}|$ is unknown. However, since for every $[L, R]$, $|C_{L,R}|$ becomes equivalent to or smaller than $|C_{L,D,R}|$, $|C_{L,R}|$ is asymptotically (at most) $O(\log T)|g_T| + O(\log^2 T)|g|$.

Thus far, we introduced IK-WIBE-RE. Henceforth, we explain how smaller ciphertext size of IK-WIBE-RE is than that of PQ-WIBE-RE. Let $\mathbb{T}^{\mathsf{PQ}}_{[L,R]}$ denote the set of wildcarded IDs (for the range $[L, R]$) in PQ-WIBE-RE. For every $[L, R]$, $\mathbb{T}^{\mathsf{PQ}}_{[L,R]}$ and $\mathbb{T}_{[L,D-1]} \bigcup \mathbb{T}_{[D,R]}$, where $D \leftarrow \mathtt{Divide}_{\log T}(L, R)$, do not become the same, but *resemble*. Precisely, they consist of the same number of wildcarded IDs, and if there exists a wildcarded ID in one of them which covers a subrange of $[L, R]$, then there also exists a wildcarded ID in another one of them which covers the same subrange. This implies that size of ciphertexts generated based on them become the same. Hence, for any $[L, R]$, size of ciphertext of IK-WIBE-RE becomes equivalent to or smaller than that of PQ-WIBE-RE, and for some $[L, R]$, the former approximately becomes the half of the latter.

Our WIBE Scheme with Constant-Sized Secret-Keys Secure under the DBDH Assumption. In IK-WIBE-RE, the WIBE scheme is not required to be (hierarchically) key-delegatable. In other words, from any WIBE scheme with constant-sized secret-keys w/o key-delegatability, we can obtain a RE (or TSE) scheme with constant-sized secret-keys. Influenced by [1, 2, 4], we show that a WIBE scheme w/o key-delegatability and with constant-sized secret-keys is obtained by modifying Waters IBE scheme [19]. Security of the WIBE scheme is reduced to that of Waters IBE scheme, namely the decisional bilinear Diffie-Hellman (DBDH) assumption. By adopting the WIBE scheme to instantiate IK-WIBE-RE, we obtain a RE scheme which can be an evidence that we certainly solve the open problem mentioned earlier.

Another Generic Construction of TSE from BE. As we explained earlier, concrete TSE constructions obtained by using the naive generic construction of TSE from BE have large master public-key and encryption/decryption cost which increase linearly with $O(\sqrt{T})$ at least. We found another generic construction of TSE from BE. Let us denote it by IK-BE-TSE. It adopts the same tree-based technique as the IBE-based TSE construction [15] (PQ-IBE-TSE) and the naive WIBE-based TSE construction (PQ-WIBE-TSE). The details can be seen in Subsect. C.3.

In IK-BE-TSE, we use BE schemes whose maximum cardinality of a set of users associated with a ciphertext is $2\log T - 2$. Because of that, from IK-BE-TSE, we can obtain concrete TSE schemes whose master public-key and encryption/decryption cost are in polylogarithmic order in T. However, each secret-key for a time period t consists of $\log T + 1$ number of secret-keys of the BE scheme. So, secret-key size of any concrete TSE scheme obtained from IK-BE-TSE increase linearly with $\log T + 1$ and thus cannot be constant.

Paper Organization. In Sect. 2 for preliminaries, we introduce some special notations, and give definitions of bilinear groups and DBDH assumption. In Sect. 3 (resp. Sect. 4), we provide syntax and security definition for WIBE w/o key-delegatability (resp. RE). In Subsect. 6.1 and Subsect. 6.2, we explain some algorithms of IK-WIBE-RE. Before that, in Subsect. 5, we explain the IBE-based TSE by [15] and the WIBE-based TSE which replaces the underlying IBE scheme in the IBE-based TSE with an WIBE scheme, since they are closely related to IK-WIBE-RE. In Subsect. 6.3, we compare existing generic RE/TSE constructions

in terms of space efficiency. In Sect. 7, we instantiate IK-WIBE-RE by our original WIBE scheme w/o key-delegatability and compare existing concrete RE/TSE constructions in terms of space/time efficiency, security and required assumptions.

2 Preliminaries

Notations. For an integer $\lambda \in \mathbb{N}$, 1^λ denotes a security parameter. $\mathbb{PPT}_\lambda$ denotes a set of all probabilistic algorithms whose running time is polynomial in λ. We say that a function $f : \mathbb{N} \to \mathbb{R}$ is negligible if for every $c \in \mathbb{N}$, there exists $x_0 \in \mathbb{N}$ such that for every $x \geq x_0$, $f(x) \leq x^{-c}$. $\mathbb{NEG}_\lambda$ denotes a set of all negligible functions for λ. For a bit string $a \in \{0,1\}^N$, $a[i] \in \{0,1\}$ denotes the i-th bit of a. For a wildcarded identity $wID \in \{0,1,*\}^N$, $|wID|_* \in [0,N]$ denotes number of wildcard symbol $*$ in wID, formally $\sum_{i\in[0,N-1] \text{ s.t. } wID[i]=*} 1$. We say that a wildcarded identity $wID \in \{0,1,*\}^N$ covers a subrange $[a,b]$ of a range $[A,B]$ if every value included in the subrange matches (or satisfies) the wID and every value excluded from the subrange does not match the wID.

Bilinear Groups of Prime Order. $\mathcal{G}_{BG}$ generates bilinear groups of prime order. Let $\lambda \in \mathbb{N}$. Specifically, it takes 1^λ and randomly generates and outputs $(p, \mathbb{G}, \mathbb{G}_T, e, g, h)$. First, p is a prime with bit length λ. Second, $(\mathbb{G}, \mathbb{G}_T)$ are multiplicative groups of order p. Third, (g,h) are generators of $\mathbb{G}$. Fourth, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a (symmetric) function computable in polynomial time which satisfies the following conditions: (1) Bilinearity: For every $a, b \in \mathbb{Z}_p$, $e(g^a, h^b) = e(g,h)^{ab}$. (2) Non-degeneracy: $e(g,h) \neq 1_{\mathbb{G}_T}$, where $1_{\mathbb{G}_T}$ denotes the unit element.

Definition 1. *Decisional Bilinear Diffie-Hellman (DBDH) assumption holds if* $\forall\lambda \in \mathbb{N}$, $\forall \mathsf{A} \in \mathbb{PPT}_\lambda$, $\exists\epsilon \in \mathbb{NEG}_\lambda$ *s.t.* $\mathsf{Adv}^{\mathsf{DBDH}}_{\mathsf{A},\lambda}(\lambda) := |\Pr[1 \leftarrow \mathsf{A}(p, \mathbb{G}, g, g^\alpha, g^\beta, g^\gamma, e(g,g)^{\alpha\beta\gamma})] - \Pr[1 \leftarrow \mathsf{A}(p, \mathbb{G}, g, g^\alpha, g^\beta, g^\gamma, e(g,g)^{\omega})]| < \epsilon$, *where* $(p, \mathbb{G}, \mathbb{G}_T, e, g, \cdot) \leftarrow \mathcal{G}_{BG}(1^\lambda)$ *and* $\alpha, \beta, \gamma, \omega \xleftarrow{\mathrm{U}} \mathbb{Z}_p$.

3 Wildcarded Identity-Based Encryption (WIBE)

Wildcarded identity-based encryption (WIBE) [1, 2, 4] was originally introduced as a generalized primitive of hierarchical IBE (HIBE). In WIBE, a plaintext is encrypted under a wildcarded identity which can include some wildcard symbols $*$, and the ciphertext can be correctly decrypted by a secret-key for an identity matching the wildcarded identity. The original WIBE automatically inherits the (hierarchical) key-delegatability. In this paper, we consider WIBE lacking key-delegatability, whose definitions are given in this section. Definitions of the original WIBE are given in Subsect. A.5.

Syntax. Wildcarded identity-based encryption (WIBE) consists of following 4 polynomial time algorithms, where $\mathtt{Dec}$ is deterministic and the others are probabilistic: Let 1^λ, where $\lambda \in \mathbb{N}$, denote a security parameter. Let $L \in \mathbb{N}$ denote bit length of an ID or wildcarded ID (wID). *Setup algorithm* $\mathtt{Setup}$ takes $(1^\lambda, 1^L)$ as input, then outputs a master public-key mpk and a master secret-key msk. We write the procedure as $(mpk, msk) \leftarrow \mathtt{Setup}(1^\lambda, 1^L)$. We assume that space of plaintexts $\mathcal{M}$ is uniquely determined by mpk. Note that the other algorithms implicitly takes mpk as input. *Key-generation algorithm* $\mathtt{KGen}$ takes msk and an $ID \in \{0,1\}^L$, then outputs a secret-key sk_{ID}. We write it as $sk_{ID} \leftarrow \mathtt{KGen}(msk, ID)$. *Encryption algorithm* $\mathtt{Enc}$ takes a plaintext $m \in \mathcal{M}$ and a $wID \in \{0,1,*\}^L$, then outputs a ciphertext C_{wID}. We write it as $C_{wID} \leftarrow \mathtt{Enc}(m, wID)$. *Decryption algorithm* $\mathtt{Dec}$ takes a secret-key sk_{ID} and a ciphertext C_{wID}, then outputs a plaintext $m \in \mathcal{M}$ or a special symbol $\perp$ which means that decryption failed. We write it as $m \,/\, \perp \leftarrow \mathtt{Dec}(sk_{ID}, C_{wID})$.

Additionally, we define *matching algorithm* $\mathtt{Match}_L$. It takes an $ID \in \{0,1\}^L$ and a $wID \in \{0,1,*\}^L$, verifies whether the ID matches the wID, then outputs a Boolean symbol. Formallly, it outputs 1 (if $\forall i \in [0, L-1]$ s.t. $wID[i] \in \{0,1\}$, $ID[i] = wID[i]$) or 0 (otherwise).

We require every WIBE scheme to be correct. A WIBE scheme $\Sigma_{\mathrm{WIBE}} = \{\mathtt{Setup}, \mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec}, \mathtt{Match}\}$ is correct, if $\forall\lambda \in \mathbb{N}$, $\forall L \in \mathbb{N}$, $\forall(mpk, msk) \leftarrow \mathtt{Setup}(1^\lambda, 1^L)$, $\forall ID \in \{0,1\}^L$, $\forall sk_{ID} \leftarrow \mathtt{KGen}(msk, ID)$,

$\forall m \in \mathcal{M}$, $\forall wID \in \{0,1,*\}^L$ s.t. $\mathtt{Match}_L(ID, wID) = 1$, $\forall C_{wID} \leftarrow \mathtt{Enc}(m, wID)$, it holds that $m \leftarrow \mathtt{Dec}(sk_{ID}, C_{wID})$.

IND-CPA Security on Multiple Ciphertexts. For WIBE schemes, we consider a security notion of (adaptive or selective) indistinguishability against adaptive chosen plaintexts attack (IND-CPA) on multiple ciphertexts. In this section, we give the definition of the adaptive security. The one of the selective security is given in Subsect. A.1. For a WIBE scheme Σ_{WIBE}, a probabilistic algorithm A and a bit $b \in \{0,1\}$, we consider a security experiment $\mathbf{Expt}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},b}$ described in Fig. 1.

$\mathbf{Expt}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},b}(1^\lambda, 1^L, 1^n)$:
$(mpk, msk) \leftarrow \mathtt{Setup}(1^\lambda, 1^L)$
$(wID_1^\star, \cdots, wID_n^\star, m_0, m_1, st_1) \leftarrow \mathsf{A}_1^{\mathcal{O}^{\mathtt{KGen}_1}_{msk}}(mpk)$, where
$\triangleright \mathcal{O}^{\mathtt{KGen}_1}_{msk}(ID_j \in \{0,1\}^L)$, where $j \in [1, q_k]$: **Return** $sk_j \leftarrow \mathtt{KGen}(msk, ID_j)$.
Assume that $\bigwedge_{i\in[1,n]} \left[wID_i^\star \in \{0,1,*\}^L \bigwedge_{j\in[1,q_k]} \mathtt{Match}_L(ID_j, wID_i^\star) = 0 \right]$
$\forall i \in [1,n], C_i^\star \leftarrow \mathtt{Enc}(m_b, wID_i^\star)$. **Return** $b' \leftarrow \mathsf{A}_2^{\mathcal{O}^{\mathtt{KGen}_2}_{msk}}(st_1, C_1^\star, \cdots, C_n^\star)$, where
$\triangleright \mathcal{O}^{\mathtt{KGen}_2}_{msk}(ID \in \{0,1\}^L)$: **Return** $sk_{ID} \leftarrow \mathtt{KGen}(msk, ID)$, if $\bigwedge_{i\in[1,n]} \mathtt{Match}_L(ID, wID_i^\star) = 0$. Else, **Return** $\perp$.

Fig. 1. Security experiment for a WIBE scheme Σ_{WIBE}

Definition 2. *Let $n \in \mathbb{N}$. A WIBE scheme Σ_{WIBE} is IND-nWID-CPA*[5]*, if $\forall \lambda \in \mathbb{N}$, $\forall L \in \mathbb{N}$, $\forall \mathsf{A} \in \mathbb{PPT}_\lambda$, $\exists \epsilon \in \mathbb{NEG}_\lambda$, $\mathit{Adv}^{\mathit{IND\text{-}nWID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},\lambda,L,n}(\lambda) := |\sum_{b\in\{0,1\}}(-1)^b \Pr[\mathbf{Expt}^{\mathit{IND\text{-}nWID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},b}(1^\lambda, 1^L, 1^n) = 1]| < \epsilon$.*

The following theorem, whose proof is given in Subsect. D.1, guarantees that the adaptive or selective IND-CPA security on single ciphertext implies the one on multiple ciphertexts.

Theorem 1. *For every WIBE scheme Σ_{WIBE} and every integer $n \in \mathbb{N}$, if Σ_{WIBE} is IND-WID-CPA (resp. IND-sWID-CPA), then it is IND-nWID-CPA (resp. IND-snWID-CPA).*

4 Range Encryption (RE)

Paterson and Quaglia [15] introduced time-specific encryption (TSE) as a generalization of time-release encryption [14, 16]. In this paper, we are interested in one type of TSE whom [15] named *plain setting*[6]. In the TSE system (in plain setting) with time periods T in total, there is a trusted authority which privately generates a secret-key for a time period $t \in [0, T-1]$. A plaintext is encrypted under a time interval $[L, R]$ such that $0 \le L \le R \le T-1$, and the ciphertext can be correctly decrypted by using a secret-key for a time period t in the inverval, i.e., $t \in [L, R]$. In this paper, we consider a generalized primitive named *range encryption (RE)*. In RE, the range $[L, R]$ is allowed to be one s.t. $0 \le R < L \le T-1$. In this case, the range is equivalent to $[0, R] \bigcup [L, T-1]$[7]. In this section, we provide some definitions of RE. (Definitions of TSE can be seen in Subsect. A.3.)

[5] When $n = 1$, we refer to this as IND-WID-CPA.

[6] Other than the plain setting, they considered the other types of TSE, namely *public-key setting* and *identity-based setting*. In the former (resp. latter) setting, each ciphertext is associated with not only a range $[L, R]$ but also a public-key (resp. identity), and the ciphertext can be correctly decrypted only when using not only a secret-key for a time period $t \in [L, R]$ but also a secret-key for the public-key (resp identity). They showed that we can obtain a TSE scheme in public-key setting (resp. identity-based setting) from a TSE scheme in plain setting and a normal public-key encryption (resp. identity-based encryption) scheme.

[7] As you might already notice, there is no big difference between RE and TSE, since TSE also can encrypt a plaintext under a range $[L, R]$ s.t. $0 \le R < L \le T-1$ by encrypting the plaintext under the subrange $[0, R]$ and doing the same under the subrange $[L, T-1]$.

Syntax. Range encryption (RE) consists of following 4 polynomial time algorithms, where $\mathtt{Dec}$ is deterministic and the others are probabilistic: Let 1^λ, where $\lambda \in \mathbb{N}$, denote a security parameter. Let $[0, T-1]$, where $T \in \mathbb{N}$, denote a space of numerical values. *Setup algorithm* $\mathtt{Setup}$ takes $(1^\lambda, 1^T)$ as input then outputs a master public-key mpk and a master secret-key msk. We write the procedure as $(mpk, msk) \leftarrow \mathtt{Setup}(1^\lambda, 1^T)$. We assume that space of plaintexts $\mathcal{M}$ is uniquely determined by mpk. Note that all the other three algorithms implicitly take mpk as input. *Key-generation* algorithm $\mathtt{KGen}$ takes msk and a numerical value $t \in [0, T-1]$, then outputs a secret-key sk_t for the numerical value. We write it as $sk_t \leftarrow \mathtt{KGen}(msk, t)$. *Encryption algorithm* $\mathtt{Enc}$ takes a plaintext $m \in \mathcal{M}$ and a range $[L, R]$, where $L, R \in [0, T-1]$, then outputs a ciphertext $C_{[L,R]}$. Note that if $L \le R$ (resp. $L > R$), $[L, R]$ is equivalent to $\{L, L+1, \cdots, R-1, R\}$ (resp. $\{L, \cdots, T-1\} \cup \{0, \cdots, R\}$). We write it as $C_{[L,R]} \leftarrow \mathtt{Enc}(m, [L, R])$. *Decryption algorithm* $\mathtt{Dec}$ takes a secret-key sk_t and a ciphertext $C_{[L,R]}$, then outputs a plaintext $m \in \mathcal{M}$ or a special symbol $\perp$ which means that the decrypting procedure failed. We write it as $m \,/\, \perp \leftarrow \mathtt{Dec}(sk_t, C_{[L,R]})$.

We require every RE scheme to be correct. A RE scheme $\Sigma_{\mathrm{RE}} = \{\mathtt{Setup}, \mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec}\}$ is correct, if $\forall \lambda \in \mathbb{N}$, $\forall T \in \mathbb{N}$, $\forall (mpk, msk) \leftarrow \mathtt{Setup}(1^\lambda, 1^T)$, $\forall t \in [0, T-1]$, $\forall sk_t \leftarrow \mathtt{KGen}(msk, t)$, $\forall m \in \mathcal{M}$, $\forall L, R \in [0, T-1]$ s.t. $t \in [L, R]$, $\forall C_{[L,R]} \leftarrow \mathtt{Enc}(m, [L, R])$, $m \leftarrow \mathtt{Dec}(sk_t, C_{[L,R]})$.

IND-CPA Security. As the security notion whom RE schemes should satisfy, we consider IND-CPA, which intuitively means that no PPT adversary who is given a ciphertext for a plaintext under a range $[L, R]$ cannot get any information about the plaintext even if he can acquire any secret-keys for any numerical value t s.t. $t \notin [L, R]$. As the usual TSE schemes, e.g. [15], we consider two types of the security notion, namely *adaptive* one (IND-R-CPA) and *selective* one (IND-sR-CPA). In this section, we give the definition of the adaptive security. The one of the selective security is given in Subsect. A.2. For a RE scheme Σ_{RE}, a probabilistic algorithm A, and a bit $b \in \{0, 1\}$, we consider a security experiment $\mathbf{Expt}^{\mathtt{IND\text{-}R\text{-}CPA}}_{\Sigma_{\mathrm{RE}},\mathsf{A},b}$ in Fig. 2.

$\mathbf{Expt}^{\mathtt{IND\text{-}R\text{-}CPA}}_{\Sigma_{\mathrm{RE}},\mathsf{A},b}(1^\lambda, 1^T)$:
$(mpk, msk) \leftarrow \mathtt{Setup}(1^\lambda, 1^T)$
$(L^*, R^*, m_0, m_1, st_1) \leftarrow \mathsf{A}_1^{\mathcal{O}^{\mathtt{KGen}}_{msk}}(mpk)$, where
$\triangleright \mathcal{O}^{\mathtt{KGen}_1}_{msk}(t_i \in [0, T-1])$, where $i \in [1, q_k]$: **Return** $sk_i \leftarrow \mathtt{KGen}(msk, t_i)$.
Assume that $L^* \in [0, T-1] \wedge R^* \in [0, T-1] \wedge_{i \in [1, q_k]} t_i \notin [L^*, R^*]$
$C^* \leftarrow \mathtt{Enc}(m_b, [L^*, R^*])$. **Return** $b' \leftarrow \mathsf{A}_2^{\mathcal{O}^{\mathtt{KGen}_2}_{msk}}(st_1, C^*)$, where
$\triangleright \mathcal{O}^{\mathtt{KGen}_2}_{msk}(t \in [0, T-1])$: If $t \notin [L^*, R^*]$, **Return** $sk_t \leftarrow \mathtt{KGen}(msk, t)$. Else, **Return** $\perp$.

Fig. 2. Security experiment for a RE scheme Σ_{RE}.

Definition 3. *A RE scheme Σ_{RE} is IND-R-CPA if* $\forall \lambda \in \mathbb{N}$, $\forall T \in \mathbb{N}$, $\forall \mathsf{A} \in \mathbb{PPT}_\lambda$, $\exists \epsilon \in \mathbb{NEG}_\lambda$, $\mathit{Adv}^{\mathit{IND\text{-}R\text{-}CPA}}_{\Sigma_{\mathrm{RE}},\mathsf{A},\lambda,T}(\lambda) := |\sum_{b \in \{0,1\}} (-1)^b \Pr[\mathbf{Expt}^{\mathit{IND\text{-}R\text{-}CPA}}_{\Sigma_{\mathrm{RE}},\mathsf{A},b}(1^\lambda, 1^T) = 1]| < \epsilon$.

5 PQ-WIBE-TSE

PQ-IBE-TSE. Paterson and Quaglia [15] presented a generic construction of TSE scheme with $T \in \mathbb{N}$ time periods from IBE scheme with $2T-1$ identities. Let us denote the underlying IBE scheme by $\Sigma^{2T-1}_{\mathrm{IBE}}$. Additionally, for the IBE scheme, we denote a secret-key associated with an ID and a ciphertext associated with an ID and a plaintext m by sk_{ID} and C^M_{ID}, respectively. In the TSE scheme, we consider a binary tree whose depth is $\log T$ like the one in Fig. 3. The tree has T leaf nodes. Each leaf node is correlated with each time period. Precisely, a leaf node associated with a bit string b is correlated with a time period which is the decimal value of b. A time instant key (TIK) sk_t for a time period $t \in [0, T-1]$ is composed of $1+\log T$ secret-keys of $\Sigma^{2T-1}_{\mathrm{IBE}}$, namely $sk_t = (sk_{ID=\emptyset}, sk_{ID=b[0]}, sk_{ID=b[0]\|b[1]}, \cdots, sk_{ID=b[0]\|b[1]\|\cdots\|b[\log T-1]})$, where $b \in \{0,1\}^{\log T}$

denotes the binary value of t. In the case of $T = 8$ in Fig. 3, for instance, a TIK for $t = 3$ is $sk_{t=3} = (sk_{\emptyset}, sk_0, sk_{01}, sk_{011})$.

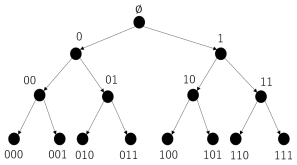

Fig. 3. A binary tree with depth 3

$\mathtt{Cover}_d(L, R)$, where $0 \le L \le R \le 2^d - 1$:
$l := L, r := R, \mathbb{T}_{[L,R]} := \emptyset$. While $l < r$, do:
If $l = 0 \bmod 2$, $l := \mathtt{Parent}(l)$.
Else, $\mathbb{T}_{[L,R]} := \mathbb{T}_{[L,R]} \bigcup \{l\}$, $l := \mathtt{Parent}(l) + 1$.
If $r = 0 \bmod 2$, $\mathbb{T}_{[L,R]} := \mathbb{T}_{[L,R]} \bigcup \{r\}$, $r := \mathtt{Parent}(r) - 1$.
Else, $r := \mathtt{Parent}(r)$.
If $l = r$, $\mathbb{T}_{[L,R]} := \mathbb{T}_{[L,R]} \bigcup \{l\}$. **Return** $\mathbb{T}_{[L,R]}$.

Fig. 4. Algorithm $\mathtt{Cover}_d$, which appeared as *Algorithm 1* in [15], where $d \in \mathbb{N}$ and `Parent` takes a node and returns its parental node.

Next, let us *informally* explain how a ciphertext for a plaintext m under a decryption time interval (DTI) $[L, R]$, where $0 \le L \le R \le T - 1$, is generated. For a node with binary value b in depth $\mathtt{Depth}_{\log T}(b) \in [0, \log T]$, let $\mathbb{S}_b$ denote the set which consists of every leaf node whose one of ancestors is the node b, i.e., $\left\{ b \|_{i \in [0, \log T - \mathtt{Depth}(b) - 1]} \beta \mid \beta \in \{0, 1\} \right\}$. If $T = 8$, for example, $\mathbb{S}_1 = \{100, 101, 110, 111\}$, $\mathbb{S}_{10} = \{100, 101\}$ and $\mathbb{S}_{110} = \{110\}$. To construct the ciphertext for $[L, R]$, we firstly deterministically find a set of nodes $\mathbb{T}_{[L,R]}$ according to the following rule: the union set $\bigcup_{b \in \mathbb{T}_{[L,R]}} \mathbb{S}_b$ is *equivalent* to the set $[L, R]$, and cardinality of the set $|\mathbb{T}_{[L,R]}| = \sum_{b \in \mathbb{T}_{[L,R]}} 1$ is the *smallest*. For instance, $\mathbb{T}_{[2,6]} = \{01, 10, 110\}$, $\mathbb{T}_{[0,6]} = \{0, 10, 110\}$, and $\mathbb{T}_{[7,7]} = \{111\}$. An algorithm `Cover` deriving $\mathbb{T}_{[L,R]}$ is formally described in Fig. 4. Finally, the ciphertext $C^m_{[L,R]}$ is composed of $(\{C^m_{ID=b} \mid b \in \mathbb{T}_{[L,R]}\})$.

Four algorithms of PQ-IBE-TSE are formally described in Fig. 19 in Subsect. C.1.

PQ-WIBE-TSE. The idea on whom PQ-WIBE-TSE is based is similar with the one on whom PQ-IBE-TSE is based on. As PQ-IBE-TSE, PQ-WIBE-TSE defines a binary tree with depth $\log T$. One major difference between the two is that PQ-WIBE-TSE uses a WIBE scheme $\Sigma^{\log T}_{\mathrm{WIBE}}$ (whose bit length of an identity is $\log T$). A TIK for $t \in [0, T - 1]$ is $sk_t = sk_{ID=b}$, where b is the binary value of t. A ciphertext for a plaintext m under a DTI $[L, R]$ satisfying $0 \le L \le R \le T - 1$ is $C^m_{[L,R]} = (\{C^m_{wID=b} \mid b \in \mathbb{T}^*_{[L,R]}\})$, where the set of wildcarded IDs $\mathbb{T}^*_{[L,R]}$ is defined as $\{b \|*^{\log T - |b|} \mid b \in \mathbb{T}_{[L,R]}\}$. For instance, $sk_3 = sk_{011}$, $\mathbb{T}^*_{[2,6]} = \{01*, 10*, 110\}$ and $C^m_{[2,6]} = (C^m_{01*}, C^m_{10*}, C^m_{110})$.

Four formal algorithms of PQ-WIBE-TSE are in Fig. 20 in Subsect. C.2.

6 Our Generic Construction of RE from WIBE (IK-WIBE-RE)

As described in Fig. 6, IK-WIBE-RE has 4 main algorithms $\{\mathrm{RE.}\mathtt{Setup}, \mathrm{RE.}\mathtt{KGen}, \mathrm{RE.}\mathtt{Enc}, \mathrm{RE.}\mathtt{Dec}\}$ and 6 sub algorithms $\{\mathtt{Binarize}_{\log T}, \mathtt{Classify}_{\log T}, \mathtt{Divide}_{\log T}, \mathtt{LatterWID}_{\log T}, \mathtt{FormerWID}_{\log T}, \mathtt{Merge}_{\log T}\}$. In the first subsection, we introduce the sub algorithms. In the second subsection, we introduce the main algorithms. Hereafter, let PQ-WIBE-RE denote the RE scheme constructed from PQ-WIBE-TSE which was explained in Sect. 5.

6.1 Six Sub Algorithms of IK-WIBE-RE

5 sub algorithms other than $\mathtt{Binarize}_{\log T}$ are used to determine a set of wildcarded IDs $\mathbb{T}_{[L,R]}$ for a range $[L, R]$ in an encryption procedure. $\mathtt{Binarize}_{\log T}$ is a general algorithm used in the whole system.

$b[4]$	0																1														
$b[3]$	0							1								1								0							
$b[2]$	0				1				1				0				0				1				1				0		
$b[1]$	0		1		1		0		0		1		1		0		0		1		1		0		0		1		1		0
$b[0]$	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
t	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**

Fig. 5. Correspondence between a numerical value $t \in [0, 31]$ and a binary value $b \in \{0, 1\}^5$.

$\texttt{Binarize}_{\log T}$ binarizing $t \in [0, T-1]$. For PQ-WIBE-RE, the correspondence between a time period and a bit string was defined in a way that b is the binary value of t. For IK-WIBE-RE, the correspondence is defined as shown in Fig. 5. The formal algorithm $\texttt{Binarize}_{\log T}$ which takes a numerical value t and returns its binary value is given in Fig. 6. Here, we provide two theorems. We prove Theorem 2 in Subsect. D.2. We can analogously prove Theorem 3.

Theorem 2. *$\forall k \in [0, \log T]$, $\forall t \in [0, T-1]$ s.t. $t \bmod 2^k = 0$, if we set $ID \coloneqq \texttt{Binarize}_{\log T}(t)$ and $\forall j \in [0, \log T - 1]$, $wID[j] \coloneqq ID[j]$ (if $j \in [k, \log T - 1]$) or $wID[j] \coloneqq *$ (otherwise), then $wID \in \{0, 1, *\}^{\log T}$ covers a range $[t, t + 2^k - 1]$.*

Theorem 3. *$\forall k \in [0, \log T]$, $\forall t \in [0, T-1]$ s.t. $t \bmod 2^k = 0$, if we set $ID \coloneqq \texttt{Binarize}_{\log T}(t-1 \bmod T)$ and $\forall j \in [0, \log T - 1]$, $wID[j] \coloneqq ID[j]$ (if $j \in [k, \log T - 1]$) or $wID[j] \coloneqq *$ (otherwise), then $wID \in \{0, 1, *\}^{\log T}$ covers a range $[t - 2^k \bmod T, t - 1 \bmod T]$. Note that modulo operation with T, i.e., $\bmod T$, should be considered only when $t = 0$.*

$\texttt{Classify}_{\log T}$ classifying $t \in [0, T-1]$ into $\log T + 1$ classes. We classify each value $t \in [0, T-1]$ into $\log T + 1$ classes. If $T = 32$, as shown in Fig. 7, there are 6 classes $i \in \{0, 1, 2, 3, 4, 5\}$. Firstly, 0 is classified as class 5. The other value $t \in [1, T-1]$ is classified as class $i \in \{0, 1, 2, 3, 4\}$ s.t. $t \bmod 2^i = 0 \wedge t \bmod 2^{i+1} = 2^i$. Generally, in case of $T \in \mathbb{N}$ (s.t. $\log T \in \mathbb{N}$), there are $\log T + 1$ classes $i \in [0, \log T]$. 0 is classified as class $\log T$. $t \in [1, T-1]$ is classified as class $i \in [0, \log T - 1]$ s.t. $t \bmod 2^i = 0 \wedge t \bmod 2^{i+1} = 2^i$. The formal algorithm $\texttt{Classify}_{\log T}$ is given in Fig. 6.

Related to $\texttt{Classify}_d$, we provide five theorems below. Intuitively, Theorem 6 says that for every $t \in [0, T-1]$ s.t. $c \coloneqq \texttt{Classify}_{\log T}(t)$ and every $\delta \in [1, 2^c]$, the identities for $t + \delta - 1$ and for $t - \delta$ are identical except for the c-th bit (or the $(\log T - 1)$-th bit, if $t = 0$). Theorem 7 says that for every t and every $k \in [0, \texttt{Classify}_{\log T}(t) - 1]$, the class indexes for $t + 2^k$ and for $t - 2^k$ are (identically) k. Theorem 8 says that for every t w. $c \coloneqq \texttt{Classify}_{\log T}(t)$, the class indexes for $t + 2^c$ and for $t - 2^c$ are greater than or equal to $c + 1$ (or equal to $\log T + 1$ if $t = 0$). Short proofs of the former two (theorems) are given below, but long ones of the latter three are given in Subsect. D.3, D.4 and D.5, respectively.

Theorem 4. *$\forall t \in [0, T-1]$ with $ID \leftarrow \texttt{Binarize}_{\log T}(t)$, $\forall k \in [0, \texttt{Classify}_{\log T}(t)]$, if we set $\forall j \in [0, \log T - 1]$, $wID[j] \coloneqq ID[j]$ (if $j \in [k, \log T - 1]$) or $wID[j] \coloneqq *$ (otherwise), then $wID \in \{0, 1, *\}^{\log T}$ covers a subrange $[t, t + 2^k - 1]$.*

Proof. Obviously, $\forall t \in [0, T-1]$, $\forall k \in [0, \texttt{Classify}_{\log T}(t)]$, $t \bmod 2^k = 0$. Hence, by Theorem 2, the proof is completed. □

Theorem 5. *$\forall t \in [0, T-1]$ with $ID \leftarrow \texttt{Binarize}_{\log T}(t - 1 \bmod T)$, $\forall k \in [0, \texttt{Classify}_{\log T}(t)]$, if we set $\forall j \in [0, \log T - 1]$, $wID[j] \coloneqq ID[j]$ (if $j \in [k, \log T - 1]$) or $wID[j] \coloneqq *$ (otherwise), then $wID \in \{0, 1, *\}^{\log T}$ covers a subrange $[t - 2^k \bmod T, t - 1 \bmod T]$. Note that any modulo operation with T, i.e., $\bmod T$, is considered only when $t = 0$.*

Proof. Obviously, $\forall t \in [0, T-1]$, $\forall k \in [0, \texttt{Classify}_{\log T}(t)]$, $t \bmod 2^k = 0$. Hence, by Theorem 3, the proof is completed. □

RE.$\mathtt{Setup}(1^\lambda, 1^T)$:
$(mpk, msk) \coloneqq \mathrm{WIBE}.\mathtt{Setup}(1^\lambda, 1^{\log T})$.
Return (mpk, msk)

RE.$\mathtt{KGen}(msk, t)$: $ID \coloneqq \mathtt{Binarize}_{\log T}(t)$.
Return $sk_t \coloneqq \mathrm{WIBE}.\mathtt{KGen}(msk, ID)$.

RE.$\mathtt{Enc}(L, R, m)$:
$D \coloneqq \mathtt{Divide}_{\log T}(L, R)$.
$\mathbb{T}_{[D,R]} \coloneqq \mathtt{LatterWID}_{\log T}(D, R)$.
$\mathbb{T}_{[L,D-1]} \coloneqq \mathtt{FormerWID}_{\log T}(L, D-1)$.
$\mathbb{T}_{[L,R]} \coloneqq \mathtt{Merge}_{\log T}(\mathbb{T}_{[D,R]}, \mathbb{T}_{[L,D-1]})$.
Parse $\mathbb{T}_{[L,R]}$ as $\{wID_i \mid i \in [1,n]\}$.
$\forall i \in [1,n]$, $C_i \coloneqq \mathrm{WIBE}.\mathtt{Enc}(wID_i, m)$.
Return $C_{[L,R]} \coloneqq \{C_i \mid i \in [1,n]\}$.

RE.$\mathtt{Dec}(sk_t, C_{[L,R]})$:
Generate $\mathbb{T}_{[L,R]}$ in the same way as RE.Enc.
Parse $\mathbb{T}_{[L,R]}$ as $\{wID_i \mid i \in [1,n]\}$.
Parse $C_{[L,R]}$ as $\{C_i \mid i \in [1,n]\}$.
$ID \coloneqq \mathtt{Binarize}_{\log T}(t)$.
$t \in [L,R] \implies \exists i \in [1,n]$ s.t.
$\mathrm{WIBE}.\mathtt{Match}_{\log T}(ID, wID_i) = 1$.
If such i exists, **Return** $\mathrm{WIBE}.\mathtt{Dec}(sk_t, C_i)$.
Return $\perp$.

$\star\mathtt{Binarize}_{\log T}(t)$:
$flag \coloneqq 0$, $left \coloneqq 0$, $right \coloneqq 2^d - 1$.
$i \coloneqq \log T - 1$. While $i \geq 0$, do:
If $t \in [left, right - 2^i]$, do:
If $flag = 0$, $b[i] \coloneqq 0$.
Else, $b[i] \coloneqq 1$ and $flag \coloneqq 0$.
$right \coloneqq right - 2^i$. $i \coloneqq i - 1$.
Else, do:
If $flag = 1$, $b[i] \coloneqq 0$.
Else, $b[i] \coloneqq 1$ and $flag \coloneqq 1$.
$left \coloneqq left + 2^i$. $i \coloneqq i - 1$
Return $b \in \{0,1\}^{\log T}$

$\star\mathtt{Classify}_{\log T}(t)$: If $t = 0$, **Return** $\log T$.
$i \coloneqq \log T - 1$. While $i \geq 0$, do:
Return i if $t \bmod 2^{i+1} = 2^i$. $i \coloneqq i - 1$.

$\star\mathtt{Divide}_{\log T}(L, R)$:
If $L = 0 \vee L > R$, **Return** $D \coloneqq 0$.
$D \coloneqq 2^{\log T - 1}$, $i \coloneqq \log T - 1$. While $i \geq 0$, do:
If $L \leq D \leq R$, **Return** D.
Else if $R < D$, $D \coloneqq D - 2^{i-1}$ and $i \coloneqq i - 1$.
Else, $D \coloneqq D + 2^{i-1}$ and $i \coloneqq i - 1$.

$\star\mathtt{LatterWID}_{\log T}(D, R)$:
$n \coloneqq 0$, $Q \coloneqq D - 1$, $i \coloneqq \mathtt{Classify}_{\log T}(D)$.
While $i \geq 0$, do:
If $Q + 2^i \leq R$, do:
$n \coloneqq n + 1$, $wID_n \coloneqq \mathtt{Binarize}_{\log T}(Q)$.
$\forall j \in [0, i-1]$, $wID_n[j] \coloneqq *$. $Q \coloneqq Q + 2^i$.
If $Q = R$, **Return** $\mathbb{T}_{[D,R]} \coloneqq \{wID_j \mid j \in [1,n]\}$.
$i \coloneqq i - 1$.

$\star\mathtt{FormerWID}_{\log T}(L, D - 1 \bmod 2^{\log T})$:
$n \coloneqq 0$, $Q \coloneqq D$, $i \coloneqq \mathtt{Classify}_{\log T}(D) - 1$.
While $i \geq 0$, do:
If $Q - 2^i \bmod 2^{\log T} \geq L$, do:
$n \coloneqq n + 1$, $wID_n \coloneqq \mathtt{Binarize}_{\log T}(Q - 1 \bmod 2^{\log T})$.
$\forall j \in [0, i-1]$, $wID_n[j] \coloneqq *$. $Q \coloneqq Q - 2^i \bmod 2^d$.
If $Q = L$, **Return** $\mathbb{T}_{[L,D-1]} \coloneqq \{wID_j \mid j \in [1,n]\}$
$i \coloneqq i - 1$.

$\star\mathtt{Merge}_{\log T}(\mathbb{T}_{[L,D-1]}, \mathbb{T}_{[D,R]})$:
Parse $\mathbb{T}_{[D,R]}$ as $\{wID_j \mid j \in [1,n_1]\}$.
Parse $\mathbb{T}_{[L,D-1]}$ as $\{wID_j \mid j \in [n_1+1, n_2]\}$.
$\mathbb{T}_{[L,R]} \coloneqq \mathbb{T}_{[L,D-1]} \bigcup \mathbb{T}_{[D,R]} = \{wID_j \mid j \in [1,n_2]\}$.
$class \coloneqq \mathtt{Classify}_{\log T}(D)$. $i \coloneqq 1$.
While $i \leq \min(n_1, n_2 - n_1) \wedge |wID_i|_* = |wID_{i+n_1}|_*$, do:
$wID \coloneqq wID_i$. $wID[class] \coloneqq *$.
$\mathbb{T}_{[L,R]} \coloneqq \mathbb{T}_{[L,R]} \setminus \{wID_i, wID_{i+n_1}\} \cup \{wID\}$. $i \coloneqq i + 1$.
Return $\mathbb{T}_{[L,R]}$.

Fig. 6. Four main algorithms and six sub algorithms (with a symbol $\star$) of IK-WIBE-RE, where $T \in \mathbb{N}$ s.t. $\log T \in \mathbb{N}$, $\lambda \in \mathbb{N}$, $t, L, R, D, D-1 \in [0, T-1]$, $n \in [0, \log T]$ and $wID \in \{0,1,*\}^{\log T}$. Every algorithm runs in polynomial time. RE.Setup, RE.KGen and RE.Enc are probabilistic. The others are deterministic.

T \ t	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
32	5	0	1	0	2	0	1	0	3	0	1	0	2	0	1	0	4	0	1	0	2	0	1	0	3	0	1	0	2	0	1	0
16	4	0	1	0	2	0	1	0	3	0	1	0	2	0	1	0																
8	3	0	1	0	2	0	1	0																								

Fig. 7. Correspondence between a numerical value $t \in [0, T-1]$ and a class $i \in [0, \log T]$. 0 is classifed as class $\log T$. $t \in [1, T-1]$ is classified as class $i \in [0, \log T - 1]$ where $t \bmod 2^{i+1} = 2^i$.

Theorem 6. $\forall t \in [0, T-1]$ *with* $c \coloneqq \texttt{Classify}_{\log T}(t)$, $\forall \delta \in [1, 2^c]$, *if we set* $ID \leftarrow \texttt{Binarize}_{\log T}(t + \delta - 1)$ *and* $ID' \leftarrow \texttt{Binarize}_{\log T}(t - \delta \bmod T)$, *then* $\forall j \in [0, \log T - 1]$, $ID[j]$ *is equal to* $\neg ID'[j]$ *(if* $[t = 0 \wedge j = \log T - 1] \vee [t \in [1, T-1] \wedge j = c]$*) or equal to* $ID'[j]$ *(otherwise).*

Theorem 7. $\forall t \in [0, T-1]$, $\forall k \in [0, \texttt{Classify}_{\log T}(t) - 1]$, $\texttt{Classify}_{\log T}(t + 2^k) = \texttt{Classify}_{\log T}(t - 2^k \bmod T) = k$. *Note that any modulo operation with* T, *i.e.,* mod T, *is considered only when* $t = 0$.

Theorem 8. $\forall t \in [0, T-1]$, $\forall \delta \in \{2^{\texttt{Classify}_{\log T}(t)}, -2^{\texttt{Classify}_{\log T}(t)}\}$, $\texttt{Classify}_{\log T}(t + \delta \bmod T)$ *is equal to* $\log T$ *(if* $t \in \{0, 2^{\log T - 1}\}$*) or greater than or equal to* $\texttt{Classify}_{\log T}(t) + 1$ *(otherwise).*

A Process where a set of wildcarded IDs $\mathbb{T}_{[L,R]}$ for a range $[L, R]$ is determined. As examples, four ranges are described in Fig. 8. In the figure, $D \in [0, 31]$ denotes a divider D dividing each range into two subranges $[L, D-1]$ and $[D, R]$. How we choose a divider D from a range $[L, R]$ is explained later. For each range, all numerical values with the same integer are associated with (or covered by) a single $wID \in \{0, 1, *\}^5$. For instance, in the third example from the top, i.e., $[1, 30]$, all numerical values given 1 (resp. 2, 3, 4) are associated with a wID $[*1 * **]$ (resp. $[*01 * *]$, $[*001*]$, $[*0001]$).

$[L, R]$ \ t	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	D
$[0, 30]$	1																2								3				4		5	-	0
$[9, 30]$	-	-	-	-	-	-	-	-	-	7	6			5			1								2				3		4	-	16
$[1, 30]$	-	4	3		2				1																2				3		4	-	16
$[2, 0]$	1	-	5		4				3								2																0

Fig. 8. A divider D and a set of wildcarded IDs $\mathbb{T}_{[L,R]}$ for four ranges.

Step \ t	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
1	-	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	$\bullet_D$	•	•	•	•	•	•	•	•	•	•	•	•	•	•	-
2	-	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	1								2				3		4	-
3	-	8	7		6				5								1								2				3		4	-
4	-	4	3		2				1																2				3		4	-

Fig. 9. A process where wIDs $\mathbb{T}_{[1,30]}$ are chosen. For $i \in \{1, 2, 3, 4\}$, i-th row from the top describes the state after each algorithm $\texttt{Divide}_5(1, 30)$, $\texttt{LatterWID}_5(16, 30)$, $\texttt{FormerWID}_5(1, 15)$ or $\texttt{Merge}_5(\mathbb{T}_{[16,30]}, \mathbb{T}_{[1,15]})$ is performed, respectively. For each row, all values with the same integer are associated with a single $wID \in \{0, 1, *\}^5$.

A process where we, given a range $[L, R]$, choose such a set of wIDs $\mathbb{T}_{[L,R]}$ proceeds as follows. First, we determine a divider $D \in [L, R]$ for the range. Second, we determine a set of wIDs $\mathbb{T}_{[D,R]}$ for the latter subrange. Third, we determine a set of wIDs $\mathbb{T}_{[L,D-1]}$ for the former subrange. Fourth, we merge the two sets of wIDs into a set of wIDs $\mathbb{T}_{[L,R]}$. For instance, the process where $\mathbb{T}_{[1,30]}$ is determined is described in Fig. 9. We provide the details in the following 4 paragraphs with star symbol ($\star$) in title.

$\star$ $\texttt{Divide}_{\log T}$ *determining a divider* D *for a range* $[L, R]$. Informally, a divider is the numerical value which is classified as the class whose index is the largest among the values in $[L, R]$. Formally, D for $[L, R]$ is

$D = \arg\max_{t\in[L,R]} \texttt{Classify}_{\log T}(t)$[8]. For instance, as Fig. 9 indicates, the divider for $[1, 30]$ is 16, since the class index 4 of the value 16 is the largest in the range and 16 is the *only* value which is classified as the class. A formal algorithm $\texttt{Divide}_{\log T}$ is described in Fig. 6.

$\star$ $\texttt{LatterWID}_{\log T}$ *deriving a set of wID(s) for the latter subrange, i.e.,* $\mathbb{T}_{[D,R]}$. According to Theorem 9 (whose proof is given in Subsect. D.6), for any L, R, D, there exist integers $k_1, \cdots, k_n$ s.t. $D-1+\sum_{l=1}^{n} 2^{k_l} = R$ and $n \leq \texttt{Classify}_{\log T}(D)$, and if we generate $\{wID_i \mid i \in [1, n]\}$ as shown in the theorem, each wID_i covers a subrange $[D + \sum_{l=1}^{i-1} 2^{k_l}, D - 1 + \sum_{l=1}^{i} 2^{k_l}]$ of $[D, R]$. Note that $\bigcup_{i=1}^{n}[D + \sum_{l=1}^{i-1} 2^{k_l}, D - 1 + \sum_{l=1}^{i} 2^{k_l}] = [D, R]$ and $\forall i, j \in [1, n]$ s.t. $i \neq j$, $[D + \sum_{l=1}^{i-1} 2^{k_l}, D - 1 + \sum_{l=1}^{i} 2^{k_l}] \cap [D + \sum_{l=1}^{j-1} 2^{k_l}, D - 1 + \sum_{l=1}^{j} 2^{k_l}] = \emptyset$. Thus, $\mathbb{T}_{[D,R]} = \{wID_i \mid i \in [1, n]\}$ is an adequate set of wIDs for $[D, R]$.

For instance, as shown in Fig. 9, there are 4 wIDs in $\mathbb{T}_{[16,30]} = \{wID_1, wID_2, wID_3, wID_4\}$, and $wID_1 = [11 * **]$ covers $[16, 23]$, $wID_2 = [101 * *]$ covers $[24, 27]$, $wID_3 = [1001*]$ covers $[28, 29]$ and $wID_4 = [10001]$ covers $[30, 30]$. A formal algorithm for this procedure, i.e., $\texttt{LatterWID}_{\log T}$, which takes (D, R) and outputs $\mathbb{T}_{[D,R]}$, is described in Fig. 6.

Theorem 9. $\forall L, R \in [0, T-1]$ *with* $D \leftarrow \texttt{Divide}_{\log T}(L, R)$, $\exists k_1 \in [0, \texttt{Classify}_{\log T}(D)]$, $\exists k_2 \in [0, k_1 - 1]$, $\cdots$, $\exists k_n \in [0, k_{n-1} - 1]$ *s.t.* $D-1+\sum_{l=1}^{n} 2^{k_l} = R$. *Moreover,* $\forall i \in [1, n]$ *with* $ID_i \leftarrow \texttt{Binarize}_{\log T}(D+\sum_{l=1}^{i-1} 2^{k_l})$, *if we set* $\forall j \in [0, \log T - 1]$, $wID_i[j] \coloneqq ID_i[j]$ *(if* $j \in [k_i, \log T - 1]$*) or* $wID_i[j] \coloneqq *$ *(otherwise), then* $wID_i \in \{0, 1, *\}^{\log T}$ *covers a subrange* $[D + \sum_{l=1}^{i-1} 2^{k_l}, D - 1 + \sum_{l=1}^{i} 2^{k_l}]$ *of* $[D, R]$.

$\star$ $\texttt{FormerWID}_{\log T}$ *deriving a set of wID(s) for the former subrange, i.e.,* $\mathbb{T}_{[L,D-1]}$. This is analogous to 6.1. According to Theorem 10 (whose proof is given in Subsect. D.7), for any L, R, D, there exist integers $k_1, \cdots, k_n$ s.t. $D - \sum_{l=1}^{n} 2^{k_l} \bmod T = L$ and $n \leq \texttt{Classify}_{\log T}(D)$, and if we generate $\{wID_i \mid i \in [1, n]\}$ as shown in the theorem, each wID_i covers a subrange $[D - \sum_{l=1}^{i} 2^{k_l} \bmod T, D - 1 - \sum_{l=1}^{i-1} 2^{k_l} \bmod T]$ of $[L, D-1]$. Note that $\bigcup_{i=1}^{n}[D - \sum_{l=1}^{i} 2^{k_l} \bmod T, D - 1 - \sum_{l=1}^{i-1} 2^{k_l} \bmod T] = [L, D-1]$ and $\forall i, j \in [1, n]$ s.t. $i \neq j$, $[D - \sum_{l=1}^{i} 2^{k_l} \bmod T, D - 1 - \sum_{l=1}^{i-1} 2^{k_l} \bmod T] \cap [D - \sum_{l=1}^{j} 2^{k_l} \bmod T, D - 1 - \sum_{l=1}^{j-1} 2^{k_l} \bmod T] = \emptyset$. Thus, $\mathbb{T}_{[L,D-1]} = \{wID_i \mid i \in [1, n]\}$ is an adequate set of wIDs for $[L, D-1]$.

For instance, as shown in Fig. 9, there are 4 wIDs in $\mathbb{T}_{[1,15]} = \{wID_1, wID_2, wID_3, wID_4\}$, and $wID_1 = [01 * **]$ covers $[8, 15]$, $wID_2 = [001 * *]$ covers $[4, 7]$, $wID_3 = [0001*]$ covers $[2, 3]$ and $wID_4 = [00001]$ covers $[1, 1]$. A formal algorithm for this procedure, i.e., $\texttt{FormerWID}_{\log T}$, which takes $(L, D-1)$ and outputs $\mathbb{T}_{[L,D-1]}$, is described in Fig. 6.

Theorem 10. $\forall L, R \in [0, T-1]$ *with* $D \leftarrow \texttt{Divide}_{\log T}(L, R)$, $\exists k_1 \in [0, \texttt{Classify}_{\log T}(D) - 1]$, $\exists k_2 \in [0, k_1 - 1]$, $\cdots$, $\exists k_n \in [0, k_{n-1} - 1]$ *s.t.* $D - \sum_{l=1}^{n} 2^{k_l} \bmod T = L$. *Moreover,* $\forall i \in [1, n]$ *with* $ID_i \leftarrow \texttt{Binarize}_{\log T}(D - 1 - \sum_{l=1}^{i-1} 2^{k_l} \bmod T)$, *if we set* $\forall j \in [0, \log T - 1]$, $wID_i[j] \coloneqq ID_i[j]$ *(if* $j \in [k_i, \log T - 1]$*) or* $wID_i[j] \coloneqq *$ *(otherwise), then* $wID_i \in \{0, 1, *\}^{\log T}$ *covers a subrange* $[D - \sum_{l=1}^{i} 2^{k_l} \bmod T, D - 1 - \sum_{l=1}^{i-1} 2^{k_l} \bmod T]$ *of* $[L, D-1]$. *Note that every modulo operation with* T, *i.e.,* mod T, *should be considered only when* $D = 0$.

$\star$ $\texttt{Merge}_{\log T}$ *merging* $\mathbb{T}_{[L,D-1]}$ *and* $\mathbb{T}_{[D,R]}$ *into* $\mathbb{T}_{[L,R]}$. Let us parse $\mathbb{T}_{[D,R]}$ (resp. $\mathbb{T}_{[L,D-1]}$) as $\{wID_i \mid i \in [1, n]\}$ (resp. $\{wID'_i \mid i \in [1, n']\}$) where $n, n' \leq \texttt{Classify}_{\log T}(D)$. The most simple way for us to merge the two sets is adding all wIDs in one of the sets into the other one, which means $\mathbb{T}_{[L,R]} \coloneqq \{wID_i, wID'_j \mid i \in [1, n], j \in [1, n']\}$. Number of wIDs in $\mathbb{T}_{[L,R]}$ simply becomes $n + n'$. However, as we explain below, there exists another way to merge the two sets into a set with less number of wIDs. Specifically, the number of wIDs can be half of $n + n'$.

We use Corollary 1 which is directly proven by Theorem 6, Theorem 9 and Theorem 10. According to the corollary, for every $i \in [1, n^*]$, wID_i and wID'_i are the same except for 1 bit. Specifically, at the position $j \coloneqq \texttt{Classify}_{\log T}(D)$, one of $wID_i[j]$ and $wID'_i[j]$ is 1 and the other one is 0. This means that wID_i (or wID'_i) whose $\texttt{Classify}_{\log T}(D)$-th bit is changed to wildcard symbol $*$ becomes a wID which covers both of

[8] In general, arg max operation can output a set (of values). In our case, however, for every $T \in \mathbb{N}$ and every $L, R \in [0, T-1]$, $\arg\max_{t\in[L,R]} \texttt{Classify}_{\log T}(t)$ outputs a single value $t' \in [L, R]$.

the subranges covered by wID_i and wID'_i, namely $[D + \sum_{l=1}^{i-1} 2^{k_l}, D - 1 + \sum_{l=1}^{i} 2^{k_l}] \cup [D - \sum_{l=1}^{i} 2^{k_l} \bmod T, D - 1 - \sum_{l=1}^{i-1} 2^{k_l} \bmod T]$. For every $i \in [1, n^*]$, wID_i and wID'_i can be combined into a single wID in such a way. Especially, in cases where $n^* = n = n'$, the number of wIDs in $\mathbb{T}_{[L,R]}$ becomes n which is half of $n + n' = 2n$.

For instance, as shown in Fig. 9, each one of the 4 wIDs in $\mathbb{T}_{[16,30]}$ and each one of the 4 wIDs in $\mathbb{T}_{[1,15]}$ are combined into a single wID. Precisely, $wID_1 = [11 * **]$ (resp. $wID_2 = [101 * *]$, $wID_3 = [1001*]$, $wID_4 = [10001]$) and $wID'_1 = [01 * **]$ (resp. $wID'_2 = [001 * *]$, $wID'_3 = [0001*]$, $wID'_4 = [0001]$) are combined into $wID_1 = [*1 * **]$ (resp. $wID_2 = [*01 * *]$, $wID_3 = [*001*]$, $wID_4 = [*0001]$). Finally, we obtain $\mathbb{T}_{[1,30]} \coloneqq \{wID_i \mid i \in [1, 4]\}$. A formal algorithm for this procedure, i.e., $\mathtt{Merge}_{\log T}$, which takes $\mathbb{T}_{[D,R]}$ and $\mathbb{T}_{[L,D-1]}$ and outputs $\mathbb{T}_{[L,R]}$, is in Fig. 6.

Corollary 1 (from Theorems 6, 9, 10). *Given $L, R \in [0, T-1]$ with $D \leftarrow \mathtt{Divide}_{\log T}(L, R)$ and $c \coloneqq \mathtt{Classify}_{\log T}(D)$, we inherit the notations $\{k_i, wID_i \mid i \in [1, n]\}$ from Theorem 9 and the ones $\{k'_i, wID'_i \mid i \in [1, n']\}$ (with apostrophe marks) from Theorem 10. Let n^* denote the integer s.t. $[0 \leq n^* \leq \min(n, n')] \bigwedge_{i=1}^{n^*} \left[k_i = k'_i\right] \bigwedge \left[n^* < \min(n, n') \implies k_{n^*+1} \neq k'_{n^*+1}\right]$. Then, $\forall i \in [1, n^*]$ and $\forall j \in [0, \log T - 1]$, it holds that $wID_i[j]$ is equal to $\neg wID'_i[j] \in \{0, 1\}$ (if $[c = \log T \bigwedge j = \log T - 1] \bigvee [c \neq \log T \bigwedge j = c]$), equal to $wID'_i[j] = *$ (else if $j \in [0, k_i - 1]$), or equal to $wID'_i[i] \in \{0, 1\}$ (otherwise).*

6.2 Four Main Algorithms of IK-WIBE-RE

Main algorithms of IK-WIBE-RE, i.e., $\{\mathtt{Setup}, \mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec}\}$, are formally described in Fig. 6. A pair of keys (mpk, msk) is a randomly generated one of the underlying WIBE scheme $\Sigma_{\mathrm{WIBE}} = \{\mathtt{Setup}, \mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec}, \mathtt{Match}\}$ whose (wildcarded) ID's bit length is $\log T$. A secret-key for a numerical value $t \in [0, T-1]$ is a randomly generated one of the WIBE scheme for the binary value of t determined according to $\mathtt{Binarize}_{\log T}(t)$. A ciphertext for a plaintext m under a range $[L, R]$ is composed of $\{C_i \mid i \in [1, n]\}$ which is generated by firstly deriving a set of wIDs $\mathbb{T}_{[L,R]}$ by properly using $\mathtt{Divide}_{\log T}$, $\mathtt{LatterWID}_{\log T}$, $\mathtt{FormerWID}_{\log T}$ and $\mathtt{Merge}_{\log T}$, then generating a ciphertext C_i for m under each wID_i in $\mathbb{T}_{[L,R]}$. Note that if a numerical value t is in a range $[L, R]$, $\mathbb{T}_{[L,R]}$ includes only one wID_i satisfied by $ID \coloneqq \mathtt{Binarize}_d(t)$, which means that sk_t derives m by decrypting C_i in $C_{[L,R]}$ associated with wID_i. Its security is guaranteed by the following theorem, whose proof is in Subsect. D.8.

Theorem 11. *IK-WIBE-RE $= \{\mathrm{RE}.\mathtt{Setup}, \mathrm{RE}.\mathtt{KGen}, \mathrm{RE}.\mathtt{Enc}, \mathrm{RE}.\mathtt{Dec}\}$ is IND-R-CPA (resp. IND-sR-CPA) if the underlying WIBE scheme $\Sigma_{\mathrm{WIBE}} = \{\mathrm{WIBE}.\mathtt{Setup}, \mathrm{WIBE}.\mathtt{KGen}, \mathrm{WIBE}.\mathtt{Enc}, \mathrm{WIBE}.\mathtt{Dec}, \mathrm{WIBE}.\mathtt{Match}\}$ is IND-nWID-CPA (resp. IND-snWID-CPA), where $n \coloneqq 2 \log T - 3$.*

6.3 Efficiency Comparison among Generic RE/TSE Constructions

We use Fig. 1 to compare efficiency of our WIBE-based RE construction with those of IBE/BE-based TSE constructions [15], that of FE-based one [12, 13], that of the generic RE construction obtained by replacing the underlying IBE scheme in the IBE-based one with an WIBE scheme, and that of our BE-based TSE construction, in terms of their underlying building blocks.

In each one of PQ-IBE-TSE, KME-FE-TSE, and IK-BE-TSE, size of secret-keys asymptotically grows linearly with $\log T$, which means that it cannot be constant. On the other hand, size of secret-keys of IK/PQ-WIBE-RE or PQ-BE-TSE becomes that of the underlying building block itself, which means that by adopting a concrete scheme whose size of secret-keys is constant, we can obtain a RE/TSE scheme with constant-sized secret-keys. As we will see in the next section, such WIBE schemes actually exist.

7 Instantiation of Our RE Scheme

Our WIBE scheme w/o key-delegatability in Fig. 10 is obtained by partially modifying Waters IBE scheme [19] in Fig. 18 in Sect. B. We prove Theorem 12 in Subsect. D.9. From Theorem 13, Theorem 12 and Theorem 1, we obtain Corollary 2.

Table 1. Efficiency comparison among some generic RE/TSE constructions in terms of the building blocks

Generic Const.	Building Block	$\lvert mpk\rvert$	$\lvert sk_t\rvert$	$\lvert C_{[L,R]}\rvert$
PQ-IBE-TSE [15]	IBE $\Sigma_{\mathrm{IBE}}^{2T-1}$	$P\left(\Sigma_{\mathrm{IBE}}^{2T-1}\right)$	$(\log T+1)\cdot K\left(\Sigma_{\mathrm{IBE}}^{2T-1}\right)$	$O\left(\log T\cdot C\left(\Sigma_{\mathrm{IBE}}^{2T-1}\right)\right)$
PQ-BE-TSE [15]	BE $\Sigma_{\mathrm{BE}}^{T,T}$	$P\left(\Sigma_{\mathrm{BE}}^{T,T}\right)$	$K\left(\Sigma_{\mathrm{BE}}^{T,T}\right)$	$C\left(\Sigma_{\mathrm{BE}}^{T,T}\right)$
KME-FE-TSE [12, 13]	FE Σ_{FE}^{T}	$O\left(T\cdot P\left(\Sigma_{\mathrm{FE}}^{T}\right)\right)$	$O\left(\log T\cdot K\left(\Sigma_{\mathrm{FE}}^{T}\right)\right)$	$O\left(C\left(\Sigma_{\mathrm{FE}}^{T}\right)\right)$
IK-BE-TSE	BE $\Sigma_{\mathrm{BE}}^{2T-1,2\log T-2}$	$P\left(\Sigma_{\mathrm{BE}}^{2T-1,2\log T-2}\right)$	$(\log T+1)\cdot K\left(\Sigma_{\mathrm{BE}}^{2T-1,2\log T-2}\right)$	$C\left(\Sigma_{\mathrm{BE}}^{2T-1,2\log T-2}\right)$
IK/PQ-WIBE-RE	WIBE $\Sigma_{\mathrm{WIBE}}^{\log T}$	$P\left(\Sigma_{\mathrm{WIBE}}^{\log T}\right)$	$K\left(\Sigma_{\mathrm{WIBE}}^{\log T}\right)$	$O\left(\log T\cdot C\left(\Sigma_{\mathrm{WIBE}}^{\log T}\right)\right)$

$\Sigma_{\mathrm{IBE}}^{n}$ denotes an IBE scheme with $n\in\mathbb{N}$ identities. $\Sigma_{\mathrm{WIBE}}^{n}$ denotes a WIBE scheme whose bit length of an identity is $n\in\mathbb{N}$. Σ_{FE}^{n} denotes an FE scheme with $n\in\mathbb{N}$ time periods. $\Sigma_{\mathrm{BE}}^{n,l}$ denotes a BE scheme, where total number of users is $n\in\mathbb{N}$ and maximum cardinality of a set of users associated with a ciphertext is $l\le n$. For a scheme X, $P(X)$, $C(X)$ and $K(X)$ denote its size of master public-key/ciphertext/secret-key, respectively.

$\mathtt{Setup}(1^\lambda,1^L)$: $(p,\mathbb{G},\mathbb{G}_T,e,g)\leftarrow\mathcal{G}_{BG}(1^\lambda)$; $\alpha\xleftarrow{\mathrm{U}}\mathbb{Z}_p$; $g_1:=g^\alpha$; $g_2,u',u_0,\cdots,u_{L-1}\xleftarrow{\mathrm{U}}\mathbb{G}$; $msk:=g_2^\alpha$ and $mpk:=(p,\mathbb{G},\mathbb{G}_T,e,g,g_1,g_2,u',u_1,\cdots,u_L,e(g_1,g_2))$; **Return** (mpk,msk)
$\mathtt{KGen}(msk,ID\in\{0,1\}^L)$: $r\xleftarrow{\mathrm{U}}\mathbb{Z}_p$; **Return** $sk_{ID}:=\left(g_2^\alpha\cdot\left(u'\prod_{i\ s.t.\ ID[i]=1}u_i\right)^r,g^r\right)$;
$\mathtt{Enc}(wID\in\{0,1,*\}^L,m\in\mathbb{G}_T)$: $s\xleftarrow{\mathrm{U}}\mathbb{Z}_p$; **Return** $C_{wID}:=\left(e(g_1,g_2)^s\cdot m,g^s,\left(u'\prod_{i\ s.t.\ wID[i]=1}u_i\right)^s,\left\{u_i^s\mid i\ s.t.\ wID[i]=*\right\}\right)$;
$\mathtt{Dec}(sk_{ID},C_{wID})$: Parse sk_{ID} as $(d_1,d_2)\in\mathbb{G}^2$; Parse C_{wID} as $(c_1,c_2,c_3,\{c_{4,i}\mid i\ s.t.\ wID[i]=*\})\in\mathbb{G}_T\times\mathbb{G}^{2+\lvert wID\rvert_*}$; **Return** $c_1\cdot e\left(d_2,c_3\cdot\prod_{i\ s.t.\ wID[i]=*\wedge ID[i]=1}c_{4,i}\right)/e(d_1,c_2)$;

Fig. 10. Our WIBE scheme Π_{WIBE}, based on Waters IBE scheme [19]

Theorem 12. *Our WIBE scheme Π_{WIBE} is IND-WID-CPA (resp. IND-sWID-CPA) if Waters IBE scheme [19] is IND-ID-CPA (resp. IND-sID-CPA).*

Corollary 2. *Π_{WIBE} is IND-nWID-CPA, where $n:=2\log T-3$, under the DBDH assumption.*

Comparing Some Concrete TSE/RE Constructions. Let us compare some concrete TSE/RE schemes in Table 2. There are 5 TSE/RE schemes which have poly-logarithmic size/cost in all measures in the table, namely the PQ-IBE-TSE scheme instantiated by Waters' IBE scheme [19], the TSE scheme by Kasamatsu et al. [12], the IK-BE-TSE scheme instantiated by 2nd Gentry and Waters' BE scheme [11], the IK/PQ-WIBE-RE scheme instantiated by our WIBE scheme Π_{WIBE}, and the IK/PQ-WIBE-RE scheme instantiated by Abdalla et al.'s WIBE scheme (w. key-delegatability) based on Boneh-Boyen-Goh HIBE scheme [1, 6]. Among them, only the latter two achieves the constant-size secret-keys. IK/PQ-WIBE-RE scheme instantiated by Π_{WIBE} is superior to IK/PQ-WIBE-RE instantiated by Abdalla et al.'s WIBE scheme, since it achieves the adaptive security under standard (static) assumption.

In an asymptotic sense, IK-WIBE-RE by Π_{WIBE} and PQ-WIBE-RE by Π_{WIBE} achieve the equivalent size/cost in all measures. However, the actual value of ciphertext length can be greatly different between them. Let $\lvert C_{[L,R]}^{\mathrm{IK}}\rvert$ (resp. $\lvert C_{[L,R]}^{\mathrm{PQ}}\rvert$) denote size of a ciphertext under $[L,R]$ for IK-WIBE-RE (resp. PQ-WIBE-RE). Precisely, for every range $[L,R]$, $\lvert C_{[L,R]}^{\mathrm{IK}}\rvert$ is equivalent to or smaller than $\lvert C_{[L,R]}^{\mathrm{PQ}}\rvert$, and for some ranges $[L,R]$, the former is almost the half of the latter. The details are given in Sect. E.

Table 2. Comparison among some existing and our concrete TSE/RE constructions secure in the standard model

Generic Construction	Building Blocks	$\|mpk\|$ [bit]	$\|sk_t\|$ [bit]	$\|C_{[L,R]}\|$ [bit]	Enc. Cost	Dec. Cost	Sec.	Assumption(s)
PQ-IBE-TSE [15]	Waters [19]	$(\log T + 5)\|g\|$	$(2\log T + 2)\|g\|$	$O(\log T)$ $(\|g_T\| + \|g\|)$	$[0, O(\log T),$ $O(\log^2 T)]$	[2, 0, 2]	Ada.	DBDH
PQ-BE-TSE [15]	BGW1 [7]	$\|g_T\| + (2T+1)\|g\|$	$\|g\|$	$2\|g\|$	$[0, 2, O(T)]$	$[2, 0, O(T)]$	Sel.	T-DBDHE
	BGW2 [7]	$\|g_T\| + 3\sqrt{T}\|g\|$	$\|g\|$	$\|g_T\| + (\sqrt{T}+1)\|g\|$	$[0, \sqrt{T}+1, O(T)]$	$[2, 0, O(\sqrt{T})]$	Sel.	$\sqrt{T}$-DBDHE
	Waters [20]	$\|g_T\| + (T+10)\|g\|$	$(T+7)\|g\|$	$10\|g\|$	$[0, 12, O(T)]$	$[9, 0, O(T)]$	Ada.	DLIN, DBDH
	GW1 [11]	$2\|g_T\| + (2T+2)\|g\|$	$(2T+2)\|g\|$	$4\|g\|$	$[0, 4, O(T)]$	$[2, 0, O(T)]$	Ada.	$2T$-DBDHE, IND-CPA SE
	GW2 [11]	$(3T-7)\|g\|$	$\|g\|$	$2\|g_T\| + 6\|g\|$ $+O(T)$	$[2, O(T), O(T)]$	$[2, O(T), O(T)]$	Ada.	$(2T,T)$-DBDHES, IND-CPA SE, PRF
KME-TSE [12, 13]	-	$\|g_T\| + O(\log T)\|g\|$	$O(\log^2 T)\|g\|$	$\|g_T\| + 3\|g\|$	$[0, O(\log T),$ $O(\log T)]$	$[3, O(\log T),$ $O(\log T)]$	Sel.	$(\log T + 1)$-DBDHI
IK-BE-TSE	BGW1 [7]	$\|g_T\|$ $+(4T-1)\|g\|$	$(\log T + 1)\|g_T\|$	$\|g_T\| + 2\|g\|$	$[0, 2, O(\log T)]$	$[2, 0,$ $O(\log T)]$	Sel.	$(2T-1)$-DBDHE
	BGW2 [7]	$\|g_T\|$ $+3\sqrt{2T-1}\|g\|$	$(\log T + 1)\|g\|$	$\|g_T\|,$ $+O(\sqrt{\log T})\|g\|$	$[0, O(\sqrt{\log T}),$ $O(\log T)]$	$[2, 0,$ $O(\sqrt{\log T})]$	Sel.	$\sqrt{2T-1}$-DBDHE
	Waters [20]	$\|g_T\| + (2T+9)\|g\|$	$O(T \cdot \log T)\|g\|$	$10\|g\|$	$[0, 12, O(\log T)]$	$[9, 0, O(\log T)]$	Ada.	DLIN, DBDH
	GW1 [11]	$\|g_T\| + (4T-2)\|g\|$	$O(T \cdot \log T)\|g\|$	$4\|g\|$	$[0,4, O(\log T)]$	$[2, 0, O(\log T)]$	Ada.	$(4T-2)$-DBDHE, IND-CPA SE
	GW2 [11]	$(6\log T + 1)\|g\|$	$(\log T + 1)\|g\|$	$2\|g_T\| + 6\|g\|$ $+O(\log T)$	$[2, O(\log T),$ $O(\log T)]$	$[2, O(\log T),$ $O(\log T)]$	Ada.	PRF, $(4T-2, 2\log T - 2)$ -DBDHES, IND-CPA SE
IK/PQ-WIBE-RE	Our Π_{WIBE}	$(\log T + 4)\|g\| + \|g_T\|$	$2\|g\|$	$O(\log T)\|g_T\|$ $+O(\log^2 T)\|g\|$	$[0, O(\log^2 T),$ $O(\log^2 T)]$	$[2, 0, O(\log T)]$	Ada.	DBDH
	ABC [1]	$(\log T + 4)\|g\| + \|g_T\|$	$2\|g\|$	$O(\log T)\|g_T\|$ $+O(\log^2 T)\|g\|$	$[0, O(\log^2 T),$ $O(\log^2 T)]$	$[2, 0, O(\log T)]$	Sel.	$\log T$-DBDHI

KME-TSE denotes a concrete (non-generic) TSE construction proposed in [12, 13]. BGW1 and BGW 2 are BE scheme introduced in Subsect. 3.1 and Subsect. 3.2 in [7], respectively. GW1 and GW2 are adaptively secure BE scheme obtained by applying a methodology (which converts semi-statically secure BE scheme into adaptively secure one) to semi-statically secure BE scheme introduced in Subsect. 3.1 and Subsect. 3.3 in [11], respectively. $|mpk|$, $|sk_t|$ and $|C_{[L,R]}|$ denote bit length of the master public-key, that of a secret-key for a time period t, and that of a ciphertext for a range $[L,R]$, respectively. $|g|$ and $|g_T|$ denote bit length of an element in bilinear group $\mathbb{G}$ and $\mathbb{G}_T$, respectively. In each column for encryption/decryption costs, $[\alpha,\beta,\gamma]$ denote number of pairings, that of exponentiations, and that of multiplications, respectively. Ada. and Sel. denote adaptive security and selective security, respectively. DBDHE (resp. DBDHES, DLIN) is the decisional bilinear Diffie-Hellman exponent (resp. the decisional bilinear Diffie-Hellman exponent sum, the decisional linear) assumption. PRF denotes pseudo random function. IND-CPA SE denotes IND-CPA secure symmetric encryption.

References

1. M. Abdalla, J. Birkett, D. Catalano, A.W Dent, J. Malone-Lee, G. Neven, J.C.N. Schuldt, and N.P. Smart. Wildcarded identity-based encryption. *Journal of Cryptology*, 24(1):42–82, 2011.
2. M. Abdalla, D. Catalano, A.W. Dent, J. Malone-Lee, G. Neven, and N.P. Smart. Identity-based encryption gone wild. In *ICALP 2006*, volume 4052 of LNCS, pages 300–311. Springer, 2006.
3. R. Anderson. Two remarks on public key cryptology. http://www.cl.cam.ac.uk/users/rja14, 1997.
4. J. Birkett, A.W. Dent, G. Neven, and J.C.N. Schuldt. Efficient chosen-ciphertext secure identity-based encryption with wildcards. In *ACISP 2007*, volume 4586 of LNCS, pages 274–292. Springer, 2007.
5. D. Boneh and X. Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In *EUROCRYPT 2004*, volume 3027 of LNCS, pages 223–238. Springer, 2004.
6. D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *EUROCRYPT 2005*, volume 3494 of LNCS, pages 440–456. Springer, 2005.
7. D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *CRYPTO 2005*, volume 3621 of LNCS, pages 258–275. Springer, 2005.
8. D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In *TCC 2011*, volume 6597 of LNCS, pages 253–273. Springer, 2011.
9. R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. In *EUROCRYPT 2003*, volume 2656 of LNCS, pages 255–271. Springer, 2003.
10. A. Fiat and M. Naor. Broadcast encryption. In *CRYPTO 1993*, volume 773 of LNCS, pages 480–491. Springer, 1993.
11. C. Gentry and B. Waters. Adaptive security in broadcast encryption systems (with short ciphertexts). In *EUROCRYPT 2009*, volume 5479 of LNCS, pages 171–188. Springer, 2009.
12. K. Kasamatsu, T. Matsuda, K. Emura, N. Attrapadung, G. Hanaoka, and H. Imai. Time-specific encryption from forward-secure encryption. In *SCN 2012*.
13. K. Kasamatsu, T. Matsuda, K. Emura, N. Attrapadung, G. Hanaoka, and H. Imai. Time-specific encryption from forward-secure encryption: generic and direct constructions. *International Journal of Information Security*, 15(5):549–571, 2016.
14. T. May. Time-release crypto. http://www.cyphernet.org/cyphernomicon/chapter14/14.5.html, 1993.
15. K. G. Paterson and E. A. Quaglia. Time-specific encryption. In *SCN 2010*, volume 6280 of LNCS, pages 1–16. Springer, 2010.
16. R. L. Rivest, A. Shamir, and D. A. Wagner. Time-lock puzzles and timed-release crypto. 1996.
17. A. Sahai and B. Waters. Fuzzy identity-based encryption. In *EUROCRYPT 2005*, volume 3494 of LNCS, pages 457–473. Springer, 2005.
18. A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO 1984*, volume 196 of LNCS, pages 47–53. Springer, 1984.
19. B. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT 2005*, volume 3494 of LNCS, pages 114–127. Springer, 2005.
20. B. Waters. Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions. In *CRYPTO 2009*, volume 5677 of LNCS, pages 619–639. Springer, 2009.

A Some Definitions

A.1 Selective IND-CPA Security on Multiple Ciphertexts for a WIBE Scheme

For a WIBE scheme Σ_{WIBE}, a probabilistic algorithm A, a bit $b \in \{0,1\}$ and an integer $n \in \mathbb{N}$, we consider a security experiment $\mathbf{Expt}^{\mathtt{IND\text{-}snWID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},b}$ described in Fig. 11.

Definition 4. *Let $n \in \mathbb{N}$. A WIBE scheme Σ_{WIBE} is IND-snWID-CPA, if $\forall \lambda \in \mathbb{N}$, $\forall L \in \mathbb{N}$, $\forall \mathsf{A} \in \mathbb{PPT}_\lambda$, $\exists \epsilon \in \mathbb{NEG}_\lambda$, $\mathbf{Adv}^{\textit{IND-snWID-CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},\lambda,L,n}(\lambda) := |\sum_{b\in\{0,1\}}(-1)^b \Pr[\mathbf{Expt}^{\textit{IND-snWID-CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},b}(1^\lambda, 1^L, 1^n) = 1]| < \epsilon$.*

A.2 Selective IND-CPA Security for a RE Scheme

For a RE scheme Σ_{RE}, a probabilistic algorithm A, and a bit $b \in \{0,1\}$, we consider a security experiment $\mathbf{Expt}^{\mathtt{IND\text{-}sR\text{-}CPA}}_{\Sigma_{\mathrm{RE}},\mathsf{A},b}$ in Fig. 12.

Definition 5. *A RE scheme Σ_{RE} is IND-sR-CPA, if $\forall \lambda \in \mathbb{N}$, $\forall T \in \mathbb{N}$, $\forall \mathsf{A} \in \mathbb{PPT}_\lambda$, $\exists \epsilon \in \mathbb{NEG}_\lambda$, $\mathbf{Adv}^{\textit{IND-sR-CPA}}_{\Sigma_{\mathrm{RE}},\mathsf{A},\lambda,T}(\lambda) := |\sum_{b\in\{0,1\}}(-1)^b \Pr[\mathbf{Expt}^{\textit{IND-sR-CPA}}_{\Sigma_{\mathrm{RE}},\mathsf{A},b}(1^\lambda, 1^T) = 1]| < \epsilon$.*

$\mathbf{Expt}^{\texttt{IND-snWID-CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},b}(1^\lambda, 1^L, 1^n)$:
$(wID_1^\star, \cdots, wID_n^\star, st_0) \leftarrow \mathsf{A}_0(1^\lambda, 1^L)$, where $\bigwedge_{i\in[1,n]} wID_i^\star \in \{0,1,*\}^L$.
$(mpk, msk) \leftarrow \texttt{Setup}(1^\lambda, 1^L)$
$(m_0, m_1, st_1) \leftarrow \mathsf{A}_1^{\mathcal{O}^{\texttt{KGen}_1}_{msk}}(mpk)$,
$\triangleright \mathcal{O}^{\texttt{KGen}_1}_{msk}(ID \in \{0,1\}^L)$: If $\bigwedge_{i\in[1,n]} \texttt{Match}_L(ID, wID_i^\star) = 0$, **Return** $sk_{ID} \leftarrow \texttt{KGen}(msk, ID)$.
Else, **Return** $\perp$.
$\forall i \in [1,n], C_i^\star \leftarrow \texttt{Enc}(m_b, wID_i^\star)$. $b' \leftarrow \mathsf{A}_2^{\mathcal{O}^{\texttt{KGen}_2}_{msk}}(st_1, C_1^\star, \cdots, C_n^\star)$, where
$\triangleright \mathcal{O}^{\texttt{KGen}_2}_{msk}(ID \in \{0,1\}^L)$: If $\bigwedge_{i\in[1,n]} \texttt{Match}_L(ID, wID_i^\star) = 0$, **Return** $sk_{ID} \leftarrow \texttt{KGen}(msk, ID)$.
Else, **Return** $\perp$.
Return b'.

Fig. 11. Selective security experiment for a WIBE scheme Σ_{WIBE}

$\mathbf{Expt}^{\texttt{IND-sR-CPA}}_{\Sigma_{\mathrm{RE}},\mathsf{A},b}(1^\lambda, 1^T)$:
$(L^*, R^*, st_0) \leftarrow \mathsf{A}_0(1^\lambda, 1^T)$, where $L^* \in [0, T-1] \bigwedge R^* \in [0, T-1]$.
$(mpk, msk) \leftarrow \texttt{Setup}(1^\lambda, 1^T)$. $(m_0, m_1, st_1) \leftarrow \mathsf{A}_1^{\mathcal{O}^{\texttt{KGen}_1}_{msk}}(st_0, mpk)$, where
$\triangleright \mathcal{O}^{\texttt{KGen}}_{msk}(t \in [0, T-1])$: If $t \notin [L^*, R^*]$, **Return** $sk \leftarrow \texttt{KGen}(msk, t)$.
Else, **Return** $\perp$.
$C^* \leftarrow \texttt{Enc}(mpk, m_b, [L^*, R^*])$. $b' \leftarrow \mathsf{A}_2^{\mathcal{O}^{\texttt{KGen}_2}_{msk}}(st_1, C^*)$, where
$\triangleright \mathcal{O}^{\texttt{KGen}_2}_{msk}(t \in [0, T-1])$: If $t \notin [L^*, R^*]$, **Return** $sk_t \leftarrow \texttt{KGen}(msk, t)$.
Else, **Return** $\perp$.
Return b'.

Fig. 12. Selective security experiments for a RE scheme Σ_{RE}.

A.3 Time-Specific Encryption (TSE) [15]

Syntax. Time-specific encryption (TSE) consists of following 4 polynomial time algorithms, where $\texttt{Dec}$ is deterministic and the others are probabilistic:

- Let 1^λ, where $\lambda \in \mathbb{N}$, denote a security parameter. Let $T \in \mathbb{N}$ denote total number of time periods. Setup algorithm $\texttt{Setup}$ takes $(1^\lambda, 1^T)$ as input then outputs a master public-key mpk and a master secret-key msk. Let it be denoted by $(mpk, msk) \leftarrow \texttt{Setup}(1^\lambda, 1^T)$.
- Key-generation algorithm $\texttt{KGen}$ takes msk and a time period $t \in [0, T-1]$, then outputs a secret-key (*time-instant key (TIK)*) sk_t. Let it be denoted by $sk_t \leftarrow \texttt{KGen}(msk, t)$.
- Encryption algorithm $\texttt{Enc}$ takes a plaintext $m \in \mathcal{M}$ and a *decryption time interval (DTI)* $[L, R]$, where $L, R \in [0, T-1]$ and $L \le R$, then outputs a ciphertext $C_{[L,R]}$. Let it be denoted by $C_{[L,R]} \leftarrow \texttt{Enc}(m, [L, R])$.
- Decryption algorithm $\texttt{Dec}$ takes a secret-key sk_t and a ciphertext $C_{[L,R]}$, then outputs a plaintext $m \in \mathcal{M}$ or a symbol $\perp$. Let it be denoted by $m \,/\, \perp \leftarrow \texttt{Dec}(sk_t, C_{[L,R]})$.

We require every TSE scheme to be correct. A TSE scheme $\Sigma_{\mathrm{TSE}} = \{\texttt{Setup}, \texttt{KGen}, \texttt{Enc}, \texttt{Dec}\}$ is correct, if $\forall \lambda, T \in \mathbb{N}$, $\forall (mpk, msk) \leftarrow \texttt{Setup}(1^\lambda, 1^T)$, $\forall t \in [0, T-1]$, $\forall sk_t \leftarrow \texttt{KGen}(msk, t)$, $\forall m \in \mathcal{M}$, $\forall L, R \in [0, T-1]$ s.t. $L \le R \wedge t \in [L, R]$, $\forall C_{[L,R]} \leftarrow \texttt{Enc}(m, [L, R])$, $\Pr[m \leftarrow \texttt{Dec}(sk_t, C_{[L,R]})] = 1$.

IND-CPA Security. For a TSE scheme Σ_{TSE}, a probabilistic algorithm A, and $b \in \{0,1\}$, we consider security experiments $\mathbf{Expt}^{\texttt{IND-DTI-CPA}}_{\Sigma_{\mathrm{TSE}},\mathsf{A},b}$ and $\mathbf{Expt}^{\texttt{IND-sDTI-CPA}}_{\Sigma_{\mathrm{TSE}},\mathsf{A},b}$ in Fig. 13.

Definition 6. *A TSE scheme Σ_{TSE} is IND-DTI-CPA if* $\forall \lambda, T \in \mathbb{N}$, $\forall \mathsf{A} \in \mathbb{PPT}_\lambda$, $\exists \epsilon \in \mathbb{NEG}_\lambda$, $\mathit{Adv}^{\mathit{IND\text{-}DTI\text{-}CPA}}_{\Sigma_{\mathrm{TSE}},\mathsf{A},\lambda,T}(\lambda) := |\sum_{b\in\{0,1\}}(-1)^b \Pr[\mathbf{Expt}^{\mathit{IND\text{-}DTI\text{-}CPA}}_{\Sigma_{\mathrm{TSE}},\mathsf{A},b}(1^\lambda, 1^T) = 1]| < \epsilon$.

Definition 7. *A TSE scheme Σ_{TSE} is IND-sDTI-CPA if* $\forall \lambda, T \in \mathbb{N}$, $\forall \mathsf{A} \in \mathbb{PPT}_\lambda$, $\exists \epsilon \in \mathbb{NEG}_\lambda$, $\mathit{Adv}^{\mathit{IND\text{-}sDTI\text{-}CPA}}_{\Sigma_{\mathrm{TSE}},\mathsf{A},\lambda,T}(\lambda) := |\sum_{b\in\{0,1\}}(-1)^b \Pr[\mathbf{Expt}^{\mathit{IND\text{-}sDTI\text{-}CPA}}_{\Sigma_{\mathrm{TSE}},\mathsf{A},b}(1^\lambda, 1^T) = 1]| < \epsilon$.

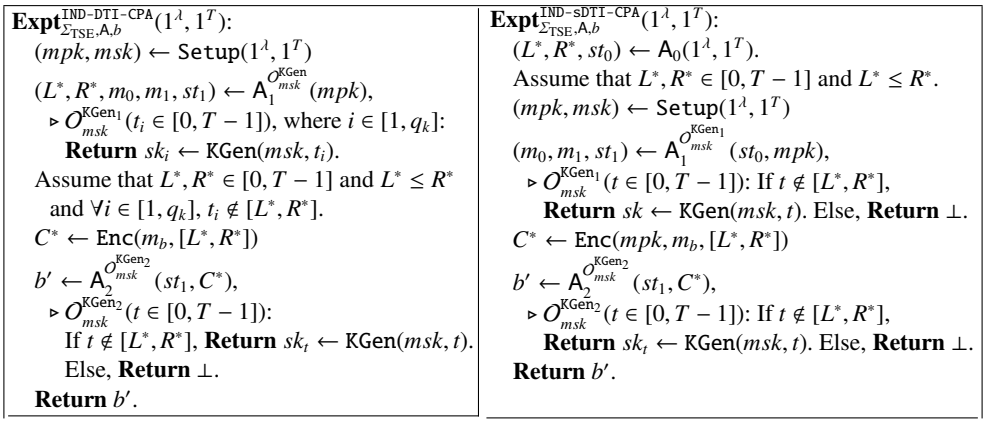

Fig. 13. Security experiments for a TSE scheme Σ_{TSE}

A.4 Identity-Based Encryption (IBE) [18]

Syntax. Identity-based encryption (IBE) consists of the following 4 polynomial time algorithms, where $\mathtt{Dec}$ is deterministic and the others are probabilistic:

- Let 1^λ, where $\lambda \in \mathbb{N}$, denote a security parameter. Let $\mathcal{I}$ denote the ID space. Let $N \in \mathbb{N}$ denote total number of IDs, which implies $|\mathcal{I}| = N$. Setup algorithm $\mathtt{Setup}$ takes $(1^\lambda, 1^N)$ as input, then outputs a master public-key mpk and a master secret-key msk. We write the procedure as $(mpk, msk) \leftarrow \mathtt{Setup}(1^\lambda, 1^N)$.
- Key-generation algorithm $\mathtt{KGen}$ takes msk and an $ID \in \mathcal{I}$, then outputs a secret-key sk_{ID}. We write it as $sk_{ID} \leftarrow \mathtt{KGen}(msk, ID)$.
- Encryption algorithm $\mathtt{Enc}$ takes a plaintext $m \in \mathcal{M}$ and a $ID \in \mathcal{I}$, then outputs a ciphertext C_{ID}. We write it as $C_{ID} \leftarrow \mathtt{Enc}(m, ID)$.
- Decryption algorithm $\mathtt{Dec}$ takes a secret-key sk_{ID} and a ciphertext C_{ID}, then outputs a plaintext $m \in \mathcal{M}$ or a symbol $\perp$. We write it as $m \,/\, \perp \leftarrow \mathtt{Dec}(sk_{ID}, C_{ID})$.

We require every IBE scheme to be correct. An IBE scheme $\Sigma_{\mathrm{IBE}} = \{\mathtt{Setup}, \mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec}\}$ is correct, if $\forall \lambda, N \in \mathbb{N}$, $\forall (mpk, msk) \leftarrow \mathtt{Setup}(1^\lambda, 1^N)$, $\forall ID \in \mathcal{I}$, $\forall sk_{ID} \leftarrow \mathtt{KGen}(msk, ID)$, $\forall m \in \mathcal{M}$, $\forall C_{ID} \leftarrow \mathtt{Enc}(m, ID)$, $\Pr[m \leftarrow \mathtt{Dec}(sk_{ID}, C_{ID})] = 1$.

IND-CPA Security. For an IBE scheme Σ_{IBE}, a probabilistic algorithm A, and $b \in \{0, 1\}$, we consider security experiments $\mathbf{Expt}^{\mathtt{IND\text{-}ID\text{-}CPA}}_{\Sigma_{\mathrm{IBE}},\mathsf{A},b}$ and $\mathbf{Expt}^{\mathtt{IND\text{-}sID\text{-}CPA}}_{\Sigma_{\mathrm{IBE}},\mathsf{A},b}$ given in Fig. 14.

Definition 8. *An IBE scheme Σ_{IBE} is IND-ID-CPA, if* $\forall \lambda, N \in \mathbb{N}$, $\forall \mathsf{A} \in \mathbb{PPT}_\lambda$, $\exists \epsilon \in \mathbb{NEG}_\lambda$, $\mathbf{Adv}^{\mathtt{IND\text{-}ID\text{-}CPA}}_{\Sigma_{\mathrm{IBE}},\mathsf{A},\lambda,N}(\lambda) := |\sum_{b\in\{0,1\}}(-1)^b \Pr[\mathbf{Expt}^{\mathtt{IND\text{-}ID\text{-}CPA}}_{\Sigma_{\mathrm{IBE}},\mathsf{A},b}(1^\lambda, 1^N) = 1]| < \epsilon$.

Definition 9. *An IBE scheme Σ_{IBE} is IND-sID-CPA, if* $\forall \lambda, N \in \mathbb{N}$, $\forall \mathsf{A} \in \mathbb{PPT}_\lambda$, $\exists \epsilon \in \mathbb{NEG}_\lambda$, $\mathbf{Adv}^{\mathtt{IND\text{-}sID\text{-}CPA}}_{\Sigma_{\mathrm{IBE}},\mathsf{A},\lambda,N}(\lambda) := |\sum_{b\in\{0,1\}}(-1)^b \Pr[\mathbf{Expt}^{\mathtt{IND\text{-}sID\text{-}CPA}}_{\Sigma_{\mathrm{IBE}},\mathsf{A},b}(1^\lambda, 1^N) = 1]| < \epsilon$.

A.5 Wildcarded Hierarchical Identity-Based Encryption (WHIBE) [2]

Syntax. Wildcarded hierarchical identity-based encryption (WHIBE) consists of following 4 polynomial time algorithms, where $\mathtt{Dec}$ is deterministic and the others are probabilistic:

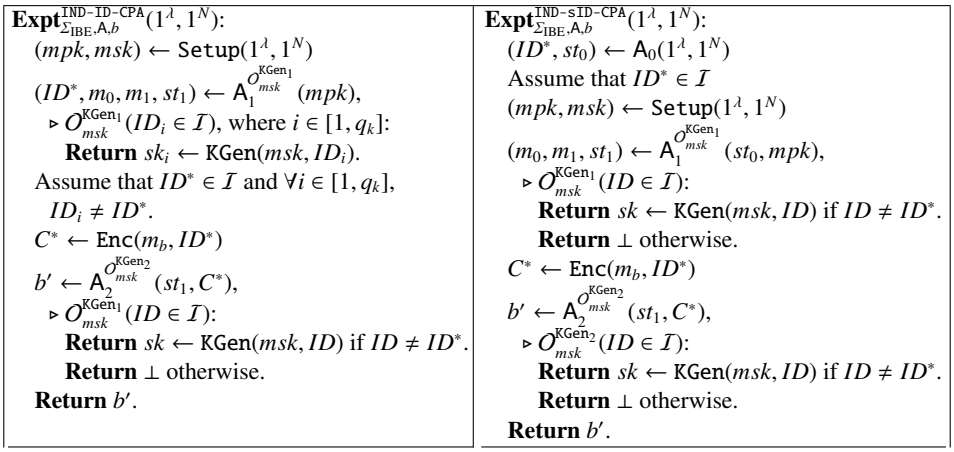

Fig. 14. Security experiments for an IBE scheme Σ_{IBE}

- Let 1^λ, where $\lambda \in \mathbb{N}$, denote a security parameter. Let p denote a prime whose bit length is λ. Let $L \in \mathbb{N}$ denote maximum number of elements in ID or wildcarded ID. Let $\mathcal{I}$ denote space for each element in ID or wildcarded ID, which is assumed to be determined after executing this algorithm $\texttt{Setup}$. For instance, $\mathcal{I}$ can be $\{0,1\}$ or $\mathbb{Z}_p^*$ where p is a prime with bit length λ. Setup algorithm $\texttt{Setup}$ takes $(1^\lambda, 1^L)$ as input, then outputs a master public-key mpk and a master secret-key for the empty set $sk_\emptyset$. Let it be denoted by $(mpk, sk_\emptyset) \leftarrow \texttt{Setup}(1^\lambda, 1^L)$.
- Key-generation algorithm $\texttt{KGen}$ takes sk_{ID_l} for an $ID_l \in \mathcal{I}^l$ where $l \in \mathbb{N}$ and $h \in \mathcal{I}$, then outputs a secret-key $sk_{ID_{l+1}}$ where $ID_{l+1} := ID_l \| h$. Let it be denoted by $sk_{ID_{l+1}} \leftarrow \texttt{KGen}(sk_{ID_l}, ID_l, h)$.
- Encryption algorithm $\texttt{Enc}$ takes a plaintext $m \in \mathcal{M}$ and a $wID_l \in \{\mathcal{I} \cup \{*\}\}^l$ where $l \in \mathbb{N}$, then outputs a ciphertext C_{wID_l}. Let it be denoted by $C_{wID_l} \leftarrow \texttt{Enc}(m, wID_l)$.
- Decryption algorithm $\texttt{Dec}$ takes a secret-key sk_{ID_l} and a ciphertext C_{wID_l}, then outputs a plaintext $m \in \mathcal{M}$ or a symbol $\perp$. Let it be denoted by $m / \perp \leftarrow \texttt{Dec}(sk_{ID_l}, C_{wID_l})$.

As for WIBE schemes, we define matching algorithm $\texttt{Match}_l : \mathcal{I}^l \times \{\mathcal{I} \cup \{*\}\}^l \to 1/0$, for WHIBE schemes, where $l \in [0, L]$, as described in Fig. 15.

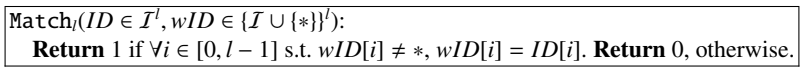

Fig. 15. A formal definition of $\texttt{Match}_l$, where $l \in [0, L]$, for WHIBE schemes

Every WHIBE scheme must be correct. A scheme $\Sigma_{\mathrm{WHIBE}} = \{\texttt{Setup}, \texttt{KGen}, \texttt{Enc}, \texttt{Dec}, \texttt{Match}\}$ is correct, if $\forall \lambda, L \in \mathbb{N}$, $\forall (mpk, sk_\emptyset) \leftarrow \texttt{Setup}(1^\lambda, 1^L)$, $\forall l \in \mathbb{N}$, $\forall ID \in \mathcal{I}^l$, $\forall sk_{ID[0]} \leftarrow \texttt{KGen}(msk, sk_\emptyset, ID[0])$, $\forall sk_{ID[0]\|ID[1]} \leftarrow \texttt{KGen}(msk, sk_{ID[0]}, ID[1])$, $\cdots$, $\forall sk_{ID} \leftarrow \texttt{KGen}(msk, sk_{ID[0]\|\cdots\|ID[l-2]}, ID[l-1])$, $\forall m \in \mathcal{M}$, $\forall wID \in \{\mathcal{I} \cup \{*\}\}^l$ s.t. $\texttt{Match}_l(ID, wID) = 1$, $\forall C_{wID} \leftarrow \texttt{Enc}(m, wID)$, $\Pr[m \leftarrow \texttt{Dec}(sk_{ID}, C_{wID})] = 1$.

IND-CPA Security. For a WHIBE scheme Σ_{WHIBE}, a probabilistic algorithm A and $b \in \{0,1\}$, we consider a security experiments $\mathbf{Expt}^{\texttt{IND-HWID-CPA}}_{\Sigma_{\mathrm{WHIBE}},\mathsf{A},b}$ and $\mathbf{Expt}^{\texttt{IND-sHWID-CPA}}_{\Sigma_{\mathrm{WHIBE}},\mathsf{A},b}$ in Fig. 16.

Definition 10. *A WHIBE scheme Σ_{WHIBE} is IND-HWID-CPA if* $\forall \lambda, L \in \mathbb{N}$, $\forall \mathsf{A} \in \mathbb{PPT}_\lambda$, $\exists \epsilon \in \mathbb{NEG}_\lambda$, $\mathbf{Adv}^{\texttt{IND-HWID-CPA}}_{\Sigma_{\mathrm{WHIBE}},\mathsf{A},\lambda,L}(\lambda) := |\sum_{b\in\{0,1\}}(-1)^b \Pr[\mathbf{Expt}^{\texttt{IND-HWID-CPA}}_{\Sigma_{\mathrm{WHIBE}},\mathsf{A},b}(1^\lambda, 1^L) = 1]| < \epsilon$.

$\mathbf{Expt}^{\mathtt{IND\text{-}HWID\text{-}CPA}}_{\Sigma_{\mathrm{WHIBE}},\mathsf{A},b}(1^\lambda, 1^L)$:
$(mpk, sk_\emptyset) \leftarrow \mathtt{Setup}(1^\lambda, 1^L)$
$(wID^\star, m_0, m_1, st_1) \leftarrow \mathsf{A}_1^{\mathcal{O}^{\mathtt{KGen}_1}_{sk_\emptyset}}(mpk)$,
▷ $\mathcal{O}^{\mathtt{KGen}_1}_{sk_\emptyset}(ID_j \in \mathcal{I}^{l_j})$, where $j \in [1, q]$:
$sk_{0,j} \coloneqq \mathtt{KGen}(msk, sk_\emptyset, ID_j[0]), \cdots, sk_{l_j-1} \coloneqq \mathtt{KGen}(msk, sk_{l_j-2,j}, ID_j[l_j - 1])$.
Return sk_{l_j-1}.
Assume that $wID^\star \in \{\mathcal{I} \cup \{*\}\}^{l^\star}$ for $l^\star \in \mathbb{N}$ and $\forall j \in [1, q]$ s.t. $l_j \leq l^\star$,
$\mathtt{Match}_{l_j}(||_{i\in[0,l_j-1]} ID_j[i], ||_{i\in[0,l_j-1]} wID^\star[i]) = 0$.
$C^\star \leftarrow \mathtt{Enc}(m_b, wID^\star)$.
$b' \leftarrow \mathsf{A}_2^{\mathcal{O}^{\mathtt{KGen}_2}_{sk_\emptyset}}(st_1, C^\star)$,
▷ $\mathcal{O}^{\mathtt{KGen}_2}_{sk_\emptyset}(ID \in \mathcal{I}^l)$:
If $l \leq l^\star$ and $\mathtt{Match}_l(||_{i\in[0,l-1]} ID[i], ||_{i\in[0,l-1]} wID^\star[i]) = 1$, **Return** $\perp$.
$sk_0 \coloneqq \mathtt{KGen}(msk, sk_\emptyset, ID[0]), \cdots, sk_{l-1} \coloneqq \mathtt{KGen}(msk, sk_{l-2}, ID[l - 1])$.
Return sk_{l-1}.
Return b'.

$\mathbf{Expt}^{\mathtt{IND\text{-}sHWID\text{-}CPA}}_{\Sigma_{\mathrm{WHIBE}},\mathsf{A},b}(1^\lambda, 1^L)$:
$(wID^\star, st_0) \leftarrow \mathsf{A}_0(1^\lambda, 1^L)$. Assume that $wID^\star \in \mathcal{I}^{l^\star}$ for $l^\star \in \mathbb{N}$.
$(mpk, sk_\emptyset) \leftarrow \mathtt{Setup}(1^\lambda, 1^L)$
$(m_0, m_1, st_1) \leftarrow \mathsf{A}_1^{\mathcal{O}^{\mathtt{KGen}_1}_{sk_\emptyset}}(mpk)$,
▷ $\mathcal{O}^{\mathtt{KGen}_1}_{sk_\emptyset}(ID \in \mathcal{I}^l)$:
If $l \leq l^\star$ and $\mathtt{Match}_l(||_{i\in[0,l-1]} ID[i], ||_{i\in[0,l-1]} wID^\star[i]) = 1$, **Return** $\perp$.
$sk_0 \coloneqq \mathtt{KGen}(msk, sk_\emptyset, ID[0]), \cdots, sk_{l-1} \coloneqq \mathtt{KGen}(msk, sk_{l-2}, ID[l - 1])$.
Return sk_{l-1}.
$C^\star \leftarrow \mathtt{Enc}(m_b, wID^\star)$.
$b' \leftarrow \mathsf{A}_2^{\mathcal{O}^{\mathtt{KGen}_2}_{sk_\emptyset}}(st_1, C^\star)$,
▷ $\mathcal{O}^{\mathtt{KGen}_2}_{sk_\emptyset}(ID \in \mathcal{I}^\star)$: Same as $\mathcal{O}^{\mathtt{KGen}_1}_{sk_\emptyset}$.
Return b'.

Fig. 16. Security experiments for a WHIBE scheme Σ_{WHIBE}

Definition 11. *A WHIBE scheme* Σ_{WHIBE} *is IND-sHWID-CPA if* $\forall \lambda, L \in \mathbb{N}$, $\forall \mathsf{A} \in \mathbb{PPT}_\lambda$, $\exists \epsilon \in \mathbb{NEG}_\lambda$, $\mathit{Adv}^{\mathit{IND\text{-}sHWID\text{-}CPA}}_{\Sigma_{\mathrm{WHIBE}},\mathsf{A},\lambda,L}(\lambda) := |\sum_{b\in\{0,1\}}(-1)^b \Pr[\mathbf{Expt}^{\mathit{IND\text{-}sHWID\text{-}CPA}}_{\Sigma_{\mathrm{WHIBE}},\mathsf{A},b}(1^\lambda, 1^L) = 1]| < \epsilon$.

A.6 Broadcast Encryption (BE) [10]

Syntax. Broadcast encryption (BE) consists of the following 4 polynomial time algorithms, where $\mathtt{Dec}$ is deterministic and the others are probabilistic:

- Let 1^λ, where $\lambda \in \mathbb{N}$, denote a security parameter. Let $n \in \mathbb{N}$ denote total number of users. Let $l \in \mathbb{N}$ denote the maximum cardinality of a set of users $\mathbb{S}$ associated with a ciphertext. Setup algorithm $\mathtt{Setup}$ takes $(1^\lambda, 1^n, 1^l)$ as input, then outputs a master public-key mpk and a master secret-key msk. We write the procedure as $(mpk, msk) \leftarrow \mathtt{Setup}(1^\lambda, 1^n, 1^l)$.
- Key-generation algorithm $\mathtt{KGen}$ takes msk and a user $i \in \{0, 1, \cdots, n-1\}$, then outputs a secret-key sk_i. We write it as $sk_i \leftarrow \mathtt{KGen}(msk, i)$.
- Encryption algorithm $\mathtt{Enc}$ takes a plaintext $m \in \mathcal{M}$ and a set of users $\mathbb{S} \subseteq \{0, 1, \cdots, n-1\}$ s.t. $|\mathbb{S}| \leq l$, then outputs a ciphertext $C_{\mathbb{S}}$. We write it as $C_{\mathbb{S}} \leftarrow \mathtt{Enc}(m, \mathbb{S})$.
- Decryption algorithm $\mathtt{Dec}$ takes a secret-key sk_i and a ciphertext $C_{\mathbb{S}}$, then outputs a plaintext $m \in \mathcal{M}$ or a symbol $\perp$. We write it as $m \,/\, \perp \leftarrow \mathtt{Dec}(sk_i, C_{\mathbb{S}})$.

We require every BE scheme to be correct. A BE scheme $\Sigma_{\mathrm{BE}} = \{\mathtt{Setup}, \mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec}\}$ is correct, if $\forall \lambda, n, l \in \mathbb{N}$, $\forall (mpk, msk) \leftarrow \mathtt{Setup}(1^\lambda, 1^n, 1^l)$, $\forall i \in \{0, \cdots, n-1\}$, $\forall sk_i \leftarrow \mathtt{KGen}(msk, i)$, $\forall m \in \mathcal{M}$, $\forall \mathbb{S} \subseteq \{0, 1, \cdots, n-1\}$ s.t. $|\mathbb{S}| \leq l$, $\forall C_{\mathbb{S}} \leftarrow \mathtt{Enc}(m, \mathbb{S})$, $\Pr[m \leftarrow \mathtt{Dec}(sk_i, C_{\mathbb{S}})] = 1$.

IND-CPA Security. For a BE scheme Σ_{BE}, a probabilistic algorithm A, and $b \in \{0, 1\}$, we consider security experiments $\mathbf{Expt}^{\mathtt{IND\text{-}\mathbb{S}\text{-}CPA}}_{\Sigma_{\mathrm{BE}},\mathsf{A},b}$ and $\mathbf{Expt}^{\mathtt{IND\text{-}s\mathbb{S}\text{-}CPA}}_{\Sigma_{\mathrm{BE}},\mathsf{A},b}$ given in Fig. 17.

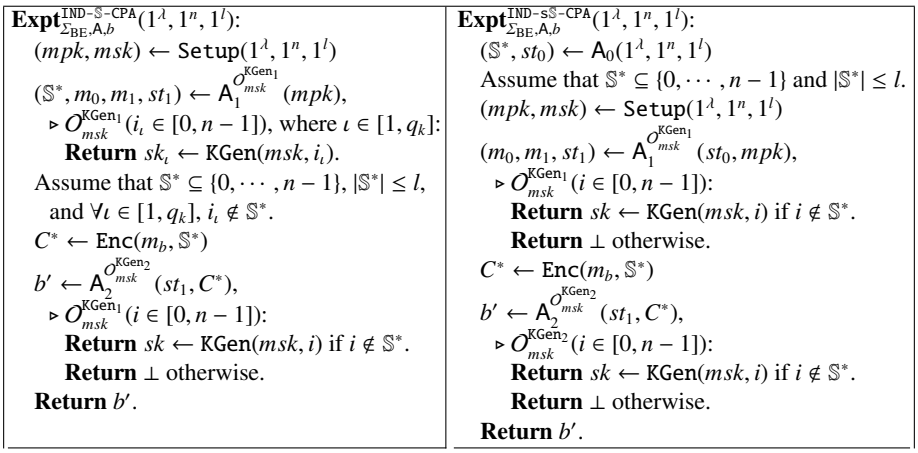

Fig. 17. Security experiments for a BE scheme Σ_{BE}

Definition 12. *A BE scheme* Σ_{BE} *is IND-*$\mathbb{S}$*-CPA, if* $\forall \lambda, n, l \in \mathbb{N}$, $\forall \mathsf{A} \in \mathbb{PPT}_\lambda$, $\exists \epsilon \in \mathbb{NEG}_\lambda$, $\mathit{Adv}^{\mathtt{IND\text{-}\mathbb{S}\text{-}CPA}}_{\Sigma_{\mathrm{BE}},\mathsf{A},\lambda,n,l}(\lambda) := |\sum_{b\in\{0,1\}}(-1)^b \Pr[\mathbf{Expt}^{\mathtt{IND\text{-}\mathbb{S}\text{-}CPA}}_{\Sigma_{\mathrm{BE}},\mathsf{A},b}(1^\lambda, 1^n, 1^l) = 1]| < \epsilon$.

Definition 13. *A BE scheme* Σ_{BE} *is IND-s*$\mathbb{S}$*-CPA, if* $\forall \lambda, n, l \in \mathbb{N}$, $\forall \mathsf{A} \in \mathbb{PPT}_\lambda$, $\exists \epsilon \in \mathbb{NEG}_\lambda$, $\mathit{Adv}^{\mathtt{IND\text{-}\mathbb{S}\text{-}CPA}}_{\Sigma_{\mathrm{BE}},\mathsf{A},\lambda,n,l}(\lambda) := |\sum_{b\in\{0,1\}}(-1)^b \Pr[\mathbf{Expt}^{\mathtt{IND\text{-}\mathbb{S}\text{-}CPA}}_{\Sigma_{\mathrm{BE}},\mathsf{A},b}(1^\lambda, 1^n, 1^l) = 1]| < \epsilon$.

B IBE Scheme by Waters [19]

Waters [19] showed that security of the IBE scheme in Fig. 18 is guaranteed by the following theorem.

$\mathtt{Setup}(1^\lambda, 1^L)$: $(p, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \mathcal{G}_{BG}(1^\lambda)$; $\alpha \xleftarrow{\mathrm{U}} \mathbb{Z}_p$; $g_1 := g^\alpha$; $g_2, u', u_0, \cdots, u_{L-1} \xleftarrow{\mathrm{U}} \mathbb{G}$;   **Return** $msk := g_2^\alpha$ and $mpk := (p, \mathbb{G}, \mathbb{G}_T, e, g, g_1, g_2, u', u_0, \cdots, u_{L-1})$;
$\mathtt{KGen}(msk, ID \in \{0,1\}^L)$: $r \xleftarrow{\mathrm{U}} \mathbb{Z}_p$; **Return** $sk_{ID} := \left(g_2^\alpha \cdot \left(u' \prod_{i \ s.t. \ ID[i]=1} u_i\right)^r, g^r\right)$;
$\mathtt{Enc}(ID \in \{0,1\}^L, m \in \mathbb{G}_T)$: $s \xleftarrow{\mathrm{U}} \mathbb{Z}_p$; **Return** $C_{ID} := \left(e(g_1, g_2)^s \cdot m, g^s, \left(u' \prod_{i \ s.t. \ ID[i]=1} u_i\right)^s\right)$;
$\mathtt{Dec}(sk_{ID}, C_{ID})$:   Parse sk_{ID} as $(d_1, d_2) \in \mathbb{G}^2$; Parse C_{ID} as $(c_1, c_2, c_3) \in \mathbb{G}_T \times \mathbb{G}^2$;   **Return** $c_1 \cdot e(d_2, c_3)/e(d_1, c_2)$;

Fig. 18. Waters' IBE scheme $\Pi_{\mathrm{IBE}}^{\mathtt{Wat}}$ [19]

Theorem 13. *Waters' IBE scheme $\Pi_{\mathrm{IBE}}^{\mathtt{Wat}}$ is IND-ID-CPA under the DBDH assumption.*

C Formal Description of Three Generic TSE Constructions

C.1 PQ-IBE-TSE

Four algorithms of PQ-IBE-TSE are formally described in Fig. 19. [15] proved that its security is guaranteed by Theorem 14.

$\mathtt{Setup}(1^\lambda, 1^T)$:   **Return** $(mpk, msk) :=$ IBE.$\mathtt{Setup}(1^\lambda, 1^{2T-1})$.
$\mathtt{KGen}(msk, t \in [0, T-1])$: $b_{-1,t} := \emptyset$. $\forall i \in [0, \log T - 1]$, $b_{i,t} := t[0] \| \cdots \| t[i]$.   **Return** $sk_t := \{sk_{i,t} \leftarrow$ IBE.$\mathtt{KGen}(msk, b_{i,t}) \mid i \in [-1, \log T - 1]\}$.
$\mathtt{Enc}(m, L, R)$, where $0 \le L \le R \le T-1$: $\mathbb{T}_{[L,R]} := \mathtt{Cover}_{\log T}(L, R)$. **Return** $C_{[L,R]} := \{C_b \leftarrow$ IBE.$\mathtt{Enc}(m, b) \mid b \in \mathbb{T}_{[L,R]}\}$.
$\mathtt{Dec}(sk_t, C_{L,R})$: $b_{-1,t} := \emptyset$. $\forall i \in [0, \log T - 1]$, $b_{i,t} := t[0] \| \cdots \| t[i]$.   Parse sk_t as $\{sk_{i,t} \mid i \in [-1, \log T - 1]\}$. $\mathbb{T}_{[L,R]} := \mathtt{Cover}_{\log T}(L, R)$. Parse $C_{[L,R]}$ as $\{C_b \mid b \in \mathbb{T}_{[L,R]}\}$. $t \in [L, R] \implies \exists l \in [-1, \log T - 1]$ *s.t.* $b_{l,t} \in \mathbb{T}_{[L,R]}$.   **Return** $m \ / \ \perp \leftarrow$ IBE.$\mathtt{Dec}(sk_l, C_{b_{l,t}})$.

Fig. 19. Four algorithms of PQ-IBE-TSE, where IBE.$\mathtt{Setup}$, IBE.$\mathtt{KGen}$, IBE.$\mathtt{Enc}$ and IBE.$\mathtt{Dec}$ denote algorithms of an IBE scheme.

Theorem 14. *PQ-IBE-TSE in Fig. 19 is IND-DTI-CPA (resp. IND-sDTI-CPA) if the underlying IBE scheme Σ_{IBE} is IND-ID-CPA (resp. IND-sID-CPA).*

$\mathtt{Setup}(1^\lambda, 1^T)$: **Return** $(mpk, msk) \coloneqq \text{WIBE.}\mathtt{Setup}(1^\lambda, 1^{\log T})$.
$\mathtt{KGen}(msk, t \in [0, T-1])$: Parse t as $t[0]\|\| \cdots \|\|t[\log T - 1]$. **Return** $sk_t \coloneqq \text{WIBE.}\mathtt{KGen}(msk, t)$.
$\mathtt{Enc}(m, L, R)$, where $0 \le L \le R \le T-1$: $\mathbb{T}_{[L,R]} \coloneqq \mathtt{Cover}_{\log T}(L, R)$. **Return** $C_{[L,R]} \coloneqq \{C_b \leftarrow \text{WIBE.}\mathtt{Enc}(m, b\|\|*^{\log T - \|b\|}) \mid b \in \mathbb{T}_{[L,R]}\}$.
$\mathtt{Dec}(sk_t, C_{L,R})$: $\mathbb{T}_{[L,R]} \coloneqq \mathtt{Cover}_{\log T}(L, R)$. $\mathbb{T} \coloneqq \{b\|\|*^{\log T - \|b\|} \mid b \in \mathbb{T}_{[L,R]}\}$. Parse $C_{[L,R]}$ as $\{C_b \mid b \in \mathbb{T}_{[L,R]}\}$. $t \in [L, R] \implies \exists b_t \in \mathbb{T}_{[L,R]}$ *s.t.* $\text{WIBE.}\mathtt{Match}_{\log T}(t, b_t) = 1$. **Return** $m \,/\, \perp \leftarrow \text{WIBE.}\mathtt{Dec}(sk_t, C_{b_t})$.

Fig. 20. Four algorithms of PQ-WIBE-TSE, where WIBE.Setup, WIBE.KGen, WIBE.Enc, WIBE.Dec and WIBE.Match denote algorithms of a WIBE scheme.

$\mathtt{Setup}(1^\lambda, 1^T)$: **Return** $(mpk, msk) \coloneqq \text{BE.}\mathtt{Setup}(1^\lambda, 1^{2T-1}, 1^{2\log T - 2})$.
$\mathtt{KGen}(msk, t \in [0, T-1])$: $b_{-1,t} \coloneqq \emptyset$. $\forall i \in [0, \log T - 1]$, $b_{i,t} \coloneqq t[0]\|\| \cdots \|\|t[i]$. **Return** $sk_t \coloneqq \{sk_{i,t} \leftarrow \text{BE.}\mathtt{KGen}(msk, b_{i,t}) \mid i \in [-1, \log T - 1]\}$.
$\mathtt{Enc}(m, L, R)$, where $0 \le L \le R \le T-1$: $\mathbb{T}_{[L,R]} \coloneqq \mathtt{Cover}_{\log T}(L, R)$. **Return** $C_{[L,R]} \leftarrow \text{BE.}\mathtt{Enc}(m, \mathbb{T}_{[L,R]})$.
$\mathtt{Dec}(sk_t, C_{L,R})$: $b_{-1,t} \coloneqq \emptyset$. $\forall i \in [0, \log T - 1]$, $b_{i,t} \coloneqq t[0]\|\| \cdots \|\|t[i]$. Parse sk_t as $\{sk_{i,t} \mid i \in [-1, \log T - 1]\}$. $\mathbb{T}_{[L,R]} \coloneqq \mathtt{Cover}_{\log T}(L, R)$. $t \in [L, R] \implies \exists l \in [-1, \log T - 1]$ *s.t.* $b_{l,t} \in \mathbb{T}_{[L,R]}$. **Return** $m \,/\, \perp \leftarrow \text{BE.}\mathtt{Dec}(sk_{l,t}, C_{[L,R]})$.

Fig. 21. Four algorithms of IK-BE-TSE, where BE.Setup, BE.KGen, BE.Enc and BE.Dec denote algorithms of a BE scheme.

C.2 PQ-WIBE-TSE

Four algorithms of PQ-WIBE-TSE are formally described in Fig. 20. Its security is guaranteed by Theorem 15. We omit its proof since it can be proven in the same manner as the security of our RE scheme IK-WIBE-RE.

Theorem 15. *PQ-WIBE-TSE in Fig. 20 is IND-DTI-CPA (resp. IND-sDTI-CPA) if the underlying WIBE scheme Σ_{WIBE} is IND-wID-CPA (resp. IND-swID-CPA).*

C.3 IK-BE-TSE

Four algorithms of IK-BE-TSE are formally described in Fig. 21. Its security is guaranteed by Theorem 16. We omit its proof since it is almost obvious that the theorem is true.

Theorem 16. *IK-BE-TSE in Fig. 21 is IND-DTI-CPA (resp. IND-sDTI-CPA) if the underlying BE scheme Σ_{BE} is IND-$\mathbb{S}$-CPA (resp. IND-s$\mathbb{S}$-CPA).*

D Some Proofs

D.1 Proof of Theorem 1

We only prove the adaptive security since we can analogously prove the selective one.

Let A denote a probabilistic algorithm which behaves as an adversary in the IND-nWID-CPA experiments, namely $\mathbf{Expt}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},b}$ where $b \in \{0,1\}$. For each integer $i \in [0,n]$, we define an experiment $\mathbf{Expt}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},0,i}$ as follows. The experiment is basically the same as $\mathbf{Expt}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},0}$ except for ciphertexts generated in the challenge phase, i.e., $(C^\star_1,\cdots,C^\star_n)$. Precisely, in the experiment, for $j \in [1,n]$, $C^\star_j$ is generated by $\mathtt{Enc}(m_1, wID^\star_j)$ (resp. $\mathtt{Enc}(m_0, wID^\star_j)$) if $j \le i$ (resp. otherwise). Obviously from the definitions, $\mathbf{Expt}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},0,0}$ (resp. $\mathbf{Expt}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},0,n}$) is identical to $\mathbf{Expt}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},0}$ (resp. $\mathbf{Expt}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},1}$). For every integers $\lambda, L, n \in \mathbb{N}$ and $\mathsf{A} \in \mathbb{PPT}_\lambda$ whose running time is t, there exists $\mathsf{B}_i \in \mathbb{PPT}_\lambda$ whose running time is $t + (n-1)t_{enc}$ for $i \in [1,n]$, where t_{enc} denotes computational time to encrypt a plaintext by Σ_{WIBE}, such that

$$\begin{aligned}
&\mathtt{Adv}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},\lambda,L,n}(\lambda)\\
&\le \sum_{i\in[1,n]} \left|\Pr\left[1 \leftarrow \mathbf{Expt}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},0,i-1}\right] - \Pr\left[1 \leftarrow \mathbf{Expt}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},0,i}\right]\right|\\
&= \sum_{i\in[1,n]} \mathtt{Adv}^{\mathtt{IND\text{-}WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{B}_i,\lambda,L,n}(\lambda).
\end{aligned}$$

The first inequality follows the triangle inequality. The second equation follows the following lemma.

Lemma 1. *For any $\lambda, L, n \in \mathbb{N}$, any $i \in [1,n]$, any $\mathsf{A} \in \mathbb{PPT}_\lambda$ which runs in time t, $\exists \mathsf{B} \in \mathbb{PPT}_\lambda$ which runs in time $t + (n-1)t_{enc}$, where t_{enc} denotes computational time to encrypt a plaintext, such that $|\Pr[1 \leftarrow \mathbf{Expt}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},0,i-1}] - \Pr[1 \leftarrow \mathbf{Expt}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},0,i}]| = \mathtt{Adv}^{\mathtt{IND\text{-}WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{B},\lambda,L,n}(\lambda)$.*

Proof. We consider $\mathsf{B} \in \mathbb{PPT}_\lambda$ which behaves as in Fig. 22. Note that for every $j \in \{1,2\}$, when A_j queries an ID to $O^{\mathsf{KGen}_j}_{msk}$, B_j queries it to his own oracle $O^{\mathsf{KGen}_j}_{msk}$ to get a secret-key sk_{ID}, then returns it to A_j. Obviously, B perfectly simulates $\mathbf{Expt}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},0,i-1}$ (resp. $\mathbf{Expt}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},0,i}$) for A when B (unconsciously) is in $\mathbf{Expt}^{\mathtt{IND\text{-}WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{B},0}$ (resp. $\mathbf{Expt}^{\mathtt{IND\text{-}WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{B},1}$). Hence, it holds that $\Pr[1 \leftarrow \mathbf{Expt}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},0,i-1}] = \Pr[1 \leftarrow \mathbf{Expt}^{\mathtt{IND\text{-}WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{B},0}]$ (resp. $\Pr[1 \leftarrow \mathbf{Expt}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},0,i}] = \Pr[1 \leftarrow \mathbf{Expt}^{\mathtt{IND\text{-}WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{B},1}]$).

The reason why the the running time of B is $t + (n-1)t_{enc}$ is that B is given only one ciphertext, i.e., $C^\star_i$, among n ciphertexts and needs to generate the other $n-1$ ciphertexts, i.e., $(C^\star_1,\cdots,C^\star_{i-1},C^\star_{i+1},\cdots,C^\star_n)$, by himself. □

$\mathsf{B}_1^{\mathcal{O}_{msk}^{\mathsf{KGen}_1}}(mpk)$:

1. $(wID_1^\star, \cdots, wID_n^\star, m_0, m_1, st_1) \leftarrow \mathsf{A}_1^{\mathcal{O}_{msk}^{\mathsf{KGen}_1}}(mpk)$.
2. $st_1' := (st_1, wID_1^\star, \cdots, wID_{i-1}^\star, wID_{i+1}^\star, \cdots, wID_n^\star)$.
3. **Return** $(wID_i^\star, m_0, m_1, st_1')$.

$\mathsf{B}_2^{\mathcal{O}_{msk}^{\mathsf{KGen}_2}}(st_1', C^\star)$:

4. Parse st_1' as $(st_1, wID_1^\star, \cdots, wID_{i-1}^\star, wID_{i+1}^\star, \cdots, wID_n^\star)$.
5. For every $j \in [1,n]$, $C_j^\star := \begin{cases} \mathsf{Enc}(m_1, wID_j^\star) & \text{if } j \le i-1, \\ C^\star & \text{if } j = i, \\ \mathsf{Enc}(m_0, wID_j^\star) & \text{otherwise.} \end{cases}$
6. **Return** $b' \leftarrow \mathsf{A}_2^{\mathcal{O}_{msk}^{\mathsf{KGen}_2}}(st_1, C_1^\star, \cdots, C_n^\star)$.

Fig. 22. Algorithm B in the proof of Theorem 1

D.2 Proof of Theorem 2

Let us consider the case of $k = \log T$. Among T values in $[0, T-1]$, only the value $t = 0$ satisfies $t \bmod 2^k = 0$. If we set $wID := *^{\log T}$, it is obviously true that for every $t \in [0, T-1]$, $ID \leftarrow \mathtt{Binarize}_{\log T}(t)$ matches wID.

Consider the case of $k = \log T - 1$. There are two values which satisfy $t \bmod 2^{\log T - 1} = 0$, namely $t = 0$ which satisfies $\mathtt{Binarize}_{\log T}(t)[\log T - 1] = 0 \bigwedge [t, t + 2^k - 1] = [0, 2^{\log T - 1} - 1]$ and $t = 2^{\log T - 1}$ which satisfies $\mathtt{Binarize}_{\log T}(t)[\log T - 1] = 1 \bigwedge [t, t + 2^k - 1] = [2^{\log T - 1}, 2^{\log T} - 1]$. It is obviously true from the definition of $\mathtt{Binarize}_{\log T}$ that every $t' \in [0, 2^{\log T - 1} - 1]$ satisfies $\mathtt{Binarize}_{\log T}(t')[\log T - 1] = 0$ and every $t' \in [2^{\log T - 1}, 2^{\log T} - 1]$ satisfies $\mathtt{Binarize}_{\log T}(t')[\log T - 1] = 1$.

Consider the case of $k = \log T - 2$. There are four values t which satisfy $t \bmod 2^{\log T - 2} = 0$, namely 0 (resp. $2^{\log T - 2}$, $2^{\log T - 1}$, $2^{\log T - 1} + 2^{\log T - 2}$) with $ID := \mathtt{Binarize}_{\log T}(t)$ which satisfies that $(ID[\log T - 1], ID[\log T - 2], [t, t + 2^k - 1])$ is equivalent to $(0, 0, [0, 2^{\log T - 2} - 1])$ (resp. $(0, 1, [2^{\log T - 2}, 2^{\log T - 1} - 1])$, $(1, 1, [2^{\log T - 1} - 1, 2^{\log T - 1} + 2^{\log T - 2} - 1])$, $(1, 0, [2^{\log T - 1} + 2^{\log T - 2}, 2^{\log T} - 1])$). It is obviously true from the definition of $\mathtt{Binarize}_{\log T}$ that every t' in a range $[0, 2^{\log T - 2} - 1]$ (resp. $[2^{\log T - 2}, 2^{\log T - 1} - 1]$, $[2^{\log T - 1}, 2^{\log T - 1} + 2^{\log T - 2} - 1]$, $[2^{\log T - 1} + 2^{\log T - 2}, 2^{\log T} - 1]$) with $ID' := \mathtt{Binarize}_{\log T}(t')$ satisfies that $(ID'[\log T - 1], ID'[\log T - 2])$ is equivalent to $(0, 0)$ (resp. $(0, 1)$, $(1, 1)$, $(1, 0)$).

In the same manner, for the other cases of $k \in [0, \log T - 3]$, we can prove that the theorem holds true.

D.3 Proof of Theorem 6

For $j \in [0, \log T - 1]$, there are 3 cases. Namely, (1) $j \in [\mathtt{Classify}_{\log T}(t) + 1, \log T - 1]$, (2) $j = \mathtt{Classify}_{\log T}(t)$, (3) $j \in [0, \mathtt{Classify}_{\log T} - 1]$. We prove the theorem in each case.

(1) By the definition of $\mathtt{Classify}_{\log T}$, it holds that $t \bmod 2^{\mathtt{Classify}_{\log T}(t)+1} = 2^{\mathtt{Classify}_{\log T}(t)}$. By the property of modulo operation, it holds that

$$
\begin{aligned}
t - 2^{\mathtt{Classify}_{\log T}(t)} &\bmod 2^{\mathtt{Classify}_{\log T}(t)+1} = 0 \\
t - 2^{\mathtt{Classify}_{\log T}(t)} + 1 &\bmod 2^{\mathtt{Classify}_{\log T}(t)+1} = 1 \\
&\vdots \\
t &\bmod 2^{\mathtt{Classify}_{\log T}(t)+1} = 2^{\mathtt{Classify}_{\log T}(t)} \\
&\vdots \\
t + 2^{\mathtt{Classify}_{\log T}(t)} - 1 &\bmod 2^{\mathtt{Classify}_{\log T}(t)+1} = 2^{\mathtt{Classify}_{\log T}(t)+1} - 1 \\
t + 2^{\mathtt{Classify}_{\log T}(t)} &\bmod 2^{\mathtt{Classify}_{\log T}(t)+1} = 0 \\
&\vdots
\end{aligned}
$$

This means that from $t - 2^{\texttt{Classify}_{\log T}(t)}$ to $t + 2^{\texttt{Classify}_{\log T}(t)} - 1$, their IDs have the same bit in position $\texttt{Classify}_{\log T}(t) + 1$. Hence, $\forall \delta \in [1, 2^{\texttt{Classify}_{\log T}(t)}]$, $ID[\texttt{Classify}_{\log T}(t) + 1] = ID'[\texttt{Classify}_{\log T}(t) + 1] = ID_t[\texttt{Classify}_{\log T}(t) + 1]$, where $ID \leftarrow \texttt{Binarize}_{\log T}(t + \delta - 1)$, $ID' \leftarrow \texttt{Binarize}_{\log T}(t - \delta \bmod T)$ and $ID_t \leftarrow \texttt{Binarize}_{\log T}(t)$. By the same logic, we can prove the theorem for the other cases, namely $j \in [\texttt{Classify}_{\log T}(t) + 2, \log T - 1]$.

(2) By the definition of $\texttt{Classify}_{\log T}$, it holds that $t \bmod 2^{\texttt{Classify}_{\log T}(t)} = 0$. By the property of modulo operation, it holds that

$$
\begin{aligned}
t - 2^{\texttt{Classify}_{\log T}(t)} \bmod 2^{\texttt{Classify}_{\log T}(t)} &= 0 \\
t - 2^{\texttt{Classify}_{\log T}(t)} + 1 \bmod 2^{\texttt{Classify}_{\log T}(t)} &= 1 \\
&\vdots \\
t - 1 \bmod 2^{\texttt{Classify}_{\log T}(t)} &= 2^{\texttt{Classify}_{\log T}(t)} - 1 \\
t \bmod 2^{\texttt{Classify}_{\log T}(t)} &= 0 \\
t + 1 \bmod 2^{\texttt{Classify}_{\log T}(t)} &= 1 \\
&\vdots \\
t + 2^{\texttt{Classify}_{\log T}(t)} - 1 \bmod 2^{\texttt{Classify}_{\log T}(t)} &= 2^{\texttt{Classify}_{\log T}(t)} - 1 \\
t + 2^{\texttt{Classify}_{\log T}(t)} \bmod 2^{\texttt{Classify}_{\log T}(t)} &= 0 \\
&\vdots
\end{aligned}
$$

This means that from $t - 2^{\texttt{Classify}_{\log T}(t)}$ to $t-1$, their IDs have the same bit in position $\texttt{Classify}_{\log T}$ with the ID of t, and from t to $t + 2^{\texttt{Classify}_{\log T}(t)} - 1$, their IDs have the distinct bit in position $\texttt{Classify}_{\log T}$ with the ID of t. Hence, $\forall \delta \in [1, 2^{\texttt{Classify}_{\log T}(t)}]$, $ID[\texttt{Classify}_{\log T}(t)] = ID_t[\texttt{Classify}_{\log T}(t)]$ and $ID'[\texttt{Classify}_{\log T}(t)] = \neg ID_t[\texttt{Classify}_{\log T}(t)]$, where $ID \leftarrow \texttt{Binarize}_{\log T}(t + \delta - 1)$, $ID' \leftarrow \texttt{Binarize}_{\log T}(t - \delta \bmod T)$ and $ID_t \leftarrow \texttt{Binarize}_{\log T}(t)$.

(3) Firstly, we focus on $t + \delta - 1$. For $j \in [0, \texttt{Classify}_{\log T}(t) - 1]$, let us define a pair (x_j, t_j) of Boolean variable and integer variable. For instance, $(x_{\texttt{Classify}_{\log T}(t)-1}, t_{\texttt{Classify}_{\log T}(t)-1})$ is defined as follows.

$$
\begin{aligned}
&(x_{\texttt{Classify}_{\log T}(t)-1}, t_{\texttt{Classify}_{\log T}(t)-1}) \\
&:= \begin{cases}
(1, t) \\
\textit{If } t + \delta - 1 \in [t, t + 2^{\texttt{Classify}_{\log T}(t)-1} - 1] \\
(0, t + 2^{\texttt{Classify}_{\log T}(t)-1}) \\
\textit{Else if } t + \delta - 1 \in [t + 2^{\texttt{Classify}_{\log T}(t)-1}, t + 2^{\texttt{Classify}_{\log T}(t)} - 1]
\end{cases}
\end{aligned}
$$

Generally, for each $j \in [1, \texttt{Classify}_{\log T}(t)]$, (x_{j+1}, t_{j+1}) is defined as follows, based on $t_0 := t$ and t_j.

$$
(x_{j+1}, t_{j+1}) := \begin{cases}
(1, t_j) \quad \textit{If } t + \delta - 1 \in [t_j, t_j + 2^{j-1} - 1] \\
(0, t_j + 2^{j-1}) \\
\quad \textit{Else if } t + \delta - 1 \in [t_j + 2^{j-1}, t_j + 2^j - 1]
\end{cases}
$$

In the same manner, for $t - \delta \bmod T$, we define such pairs $\{(x'_j, t'_j) \mid j \in [0, \texttt{Classify}_{\log T}(t) - 1]\}$.

Obviously, it is true that $\forall j \in [0, \texttt{Classify}_{\log T} - 1]$, $x_j = \neg x'_j$. By the results of (1) and (2), and the definition of $\texttt{Binarize}_{\log T}$, proof for the case (3) is completed.

D.4 Proof of Theorem 7

It is true that $\forall t \in [0, T - 1]$, $\forall i \in [0, \texttt{Classify}_{\log T}(t)]$, $t \bmod 2^i = 0$. In this proof, we refer to this as *the first statement*.

It is also true that $\forall t \in [0, T-1]$, if $\exists i \in [0, \log T]$ s.t.

$$t \bmod 2^i = 0 \wedge \left[i \neq \log T \Rightarrow t \bmod 2^{i+1} = 2^i \right],$$

then $\texttt{Classify}_{\log T}(t) = i$. We refer to this as *the second statement*.

If we can prove that it is true that $\forall t \in [0, T-1]$, $\forall k \in [0, \texttt{Classify}_{\log T}(t) - 1]$,

$$\begin{aligned} t + 2^k \bmod 2^k &= 0, \\ t + 2^k \bmod 2^{k+1} &= 2^k, \\ t - 2^k \bmod 2^k &= 0, \\ t - 2^k \bmod 2^{k+1} &= 2^k, \end{aligned}$$

then accoding to the second (true) statement, Theorem 6 is proven.

Actually, the statement is true since according to the first (true) statement,

$$\begin{aligned} t + 2^k \bmod 2^k &= t \bmod 2^k + 2^k \bmod 2^k = 0 + 0 = 0, \\ t + 2^k \bmod 2^{k+1} &= t \bmod 2^{k+1} + 2^k \bmod 2^{k+1} = 0 + 2^k = 2^k, \\ t - 2^k \bmod 2^k &= t \bmod 2^k + 2^k \bmod 2^k = 0 + 0 = 0, \\ t - 2^k \bmod 2^{k+1} &= t \bmod 2^{k+1} - 2^k \bmod 2^{k+1} = 0 + 2^k = 2^k. \end{aligned}$$

D.5 Proof of Theorem 8

Let us prove the theorem in each one of the cases (1) $t = 0$, (2) $t = 2^{\log T}$, (3) $t \in [0, T-1] \setminus \{0, 2^{\log T}\}$.

(1) Obviously, $t + 2^{\texttt{Classify}_{\log T}(t)} \bmod T = 0 + 2^{\log T} \bmod T = T \bmod T = 0$. Likewise, $t - 2^{\texttt{Classify}_{\log T}(t)} \bmod T = 0$. The proof is done.

(2) Obviously, $t + 2^{\texttt{Classify}_{\log T}(t)} \bmod T = 2^{\log T - 1} + 2^{\log T - 1} \bmod T = 2^{\log T} \bmod T = T \bmod T = 0$. Likewise, $t - 2^{\texttt{Classify}_{\log T}(t)} \bmod T = 0$. The proof is done.

(3) It is true that $t \bmod 2^{\texttt{Classify}_{\log T}(t)} = 0$ and $t \bmod 2^{\texttt{Classify}_{\log T}(t)+1} = 2^{\texttt{Classify}_{\log T}(t)}$. Hence, $t + 2^{\texttt{Classify}_{\log T}(t)} \bmod 2^{\texttt{Classify}_{\log T}(t)+1} = 2^{\texttt{Classify}_{\log T}(t)+1} \bmod 2^{\texttt{Classify}_{\log T}(t)+1} = 0 \bmod 2^{\texttt{Classify}_{\log T}(t)+1}$. Hence, $\texttt{Classify}_{\log T}(t + 2^{\texttt{Classify}_{\log T}(t)} \bmod T) \geq \texttt{Classify}_{\log T}(t) + 1$. Likewise, since $t - 2^{\texttt{Classify}_{\log T}(t)} \bmod 2^{\texttt{Classify}_{\log T}(t)+1} = 0 \bmod 2^{\texttt{Classify}_{\log T}(t)+1}$, it holds $\texttt{Classify}_{\log T}(t - 2^{\texttt{Classify}_{\log T}(t)} \bmod T) \geq \texttt{Classify}_{\log T}(t) + 1$.

D.6 Proof of Theorem 9

By Theorem 8, it is true that $\forall L, R \in [0, T-1]$ with $D \leftarrow \texttt{Divide}_{\log T}(L, R)$, $D + 2^{\texttt{Classify}_{\log T}(D)} - 1 \geq R$. There are two cases, namely (1) $D + 2^{\texttt{Classify}_{\log T}(D)} - 1 = R$, and (2) $D + 2^{\texttt{Classify}_{\log T}(D)} - 1 > R$. We prove the theorem in each case.

In the 1st case, it is obvious that $\exists k_1 \in [0, \texttt{Classify}_{\log T}(D)]$ s.t. $D + 2^{k_1} - 1 = R$, since $k_1 = \texttt{Classify}_{\log T}(D)$ is the integer. By Theorem 4, we complete to prove Theorem 9.

In the 2nd case, it is true that $\exists k_1 \in [0, \texttt{Classify}_{\log T}(D) - 1]$, $\exists k_2 \in [0, k_1 - 1]$, $\cdots$, $\exists k_n \in [0, k_{n-1} - 1]$ s.t. $D - 1 + \sum_{l=1}^{n} 2^{k_l} = R$. By using Theorem 4 and Theorem 7 with n times and $n-1$ times in total, respectively, we complete to prove Theorem 9.

D.7 Proof of Theorem 10

By Theorem 8, it is true that $\forall L, R \in [0, T-1]$ with $D \leftarrow \texttt{Divide}_{\log T}(L, R)$, $D - 2^{\texttt{Classify}_{\log T}(D)} \bmod T < L$. Hence, it is true that $\exists k_1 \in [0, \texttt{Classify}_{\log T}(D) - 1]$, $\exists k_2 \in [0, k_1 - 1]$, $\cdots$, $\exists k_n \in [0, k_{n-1} - 1]$ s.t. $D - \sum_{l=1}^{n} 2^{k_l} \bmod T = L$. By using Theorem 5 and Theorem 7 with n times and $n-1$ times in total, respectively, we complete to prove Theorem 10.

D.8 Proof of Theorem 11

We only prove the adaptive security, because the selective one can be analogously proven.

Precisely speaking, we prove the following statement: For every integer $\lambda, T \in \mathbb{N}$ and every $\mathsf{A} \in \mathbb{PPT}_\lambda$ which runs in time t, there exists $\mathsf{B} \in \mathbb{PPT}_\lambda$ which runs in time (which is almost the same as) t, such that $\mathtt{Adv}^{\mathtt{IND\text{-}R\text{-}CPA}}_{\Sigma_{\mathtt{IK\text{-}WIBE\text{-}RE}},\mathsf{A},\lambda,T}(\lambda) = \mathtt{Adv}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathtt{WIBE}},\mathsf{B},\lambda,d}(\lambda)$, where $d := \log T$ and $n := 2\log T - 3$. Note that the reason why the integer n is set as $2\log T - 3$ is that total number of wildcarded IDs in $\mathbb{T}_{[L^*,R^*]}$ for the target range $[L^*, R^*]$ is $(\log T - 1) + (\log T - 2)$ at the maximum.

B behaves as in Fig. 23. B_1 gives mpk to A_1 and gets $([L^*, R^*], m_0, m_1, st'_1)$. Here, obviously, mpk given to A_1 by B_1 properly distributes, i.e., distributes identically to the real mpk in the experiment for $\Sigma_{\mathtt{IK\text{-}WIBE\text{-}RE}}$. From $[L^*, R^*]$, B_1 derives a set of wIDs $\mathbb{T}_{[L^*,R^*]}$ parsed as $\{wID^*_i \mid i \in [1, k^*]\}$. If $k^* = n$, B_1 determines $\mathbb{T}_{[L^*,R^*]}$ as n target wIDs. Otherwise, B_1 determines $\mathbb{T}_{[L^*,R^*]}$ as k^* target wIDs, chooses $n - k^*$ wIDs from $\mathbb{T}_{[L^*,R^*]}$ uniformly at random, then determines them as the remaining $n - k^*$ target wIDs. The reason why we make B_1 do that is that any one among n target wIDs must satisfy a condition that for any numerical value t queried to $O^{\mathsf{KGen}_1}_{msk}$ or $O^{\mathsf{KGen}_2}_{msk}$, the wID is not satisfied by $\mathtt{Binarize}_d(t) \in \{0,1\}^L$. Obviously, every one of the randomly chosen $n - k^*$ wIDs satisfies the condition. After that, B_1 outputs the n wIDs and the two plaintexts m_0 and m_1. B_2 receives n ciphertexts for the n wIDs. Only the first k^* ciphertexts are given to A_2, and the other ones are ignored. Obviously, the ciphertexts given to A_2 properly distribute. Finally, B_2 outputs the bit b' outputted by A_2.

Thus, B perfectly simulates $\mathbf{Expt}^{\mathtt{IND\text{-}R\text{-}CPA}}_{\Sigma_{\mathtt{IK\text{-}WIBE\text{-}RE}},\mathsf{A},0}$ (resp. $\mathbf{Expt}^{\mathtt{IND\text{-}R\text{-}CPA}}_{\Sigma_{\mathtt{IK\text{-}WIBE\text{-}RE}},\mathsf{A},1}$) for A when B (unconsciously) plays $\mathbf{Expt}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathtt{WIBE}},\mathsf{B},0}$ (resp. $\mathbf{Expt}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathtt{WIBE}},\mathsf{B},1}$). Hence, it holds that $\Pr[1 \leftarrow \mathbf{Expt}^{\mathtt{IND\text{-}R\text{-}CPA}}_{\Sigma_{\mathtt{IK\text{-}WIBE\text{-}RE}},\mathsf{A},0}] = \Pr[1 \leftarrow \mathbf{Expt}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathtt{IK\text{-}WIBE\text{-}RE}},\mathsf{B},0}]$ (resp. $\Pr[1 \leftarrow \mathbf{Expt}^{\mathtt{IND\text{-}R\text{-}CPA}}_{\Sigma_{\mathtt{IK\text{-}WIBE\text{-}RE}},\mathsf{A},1}] = \Pr[1 \leftarrow \mathbf{Expt}^{\mathtt{IND\text{-}}n\mathtt{WID\text{-}CPA}}_{\Sigma_{\mathtt{WIBE}},\mathsf{B},1}]$).

The reason why the running time of B is almost the same as that of A is that B does not need to execute any inefficient computation such as paring, exponentiation or multiplication on any (bilinear) groups by himself. He needs to execute some algorithms such as $\mathtt{Binarize}_d$, $\mathtt{Divide}_d$, $\mathtt{LatterWID}_d$, $\mathtt{FormerWID}_d$, and $\mathtt{Merge}_d$, by himself. However, they can be executed much more efficiently than the heavy computation related to (bilinear) groups. Thus, we do not count the time to execute them.

$\mathsf{B}_1^{O^{\mathsf{KGen}_1}_{msk}}(mpk)$:

1. $([L^*, R^*], m_0, m_1, st'_1) \leftarrow \mathsf{A}_1^{O^{\mathsf{KGen}_1}_{msk}}(mpk)$,
   - ▷ $O^{\mathsf{KGen}_1}_{msk}(t \in [0, T-1])$:
     2. B_1 queries $\mathtt{Binarize}_d(t) \in \{0,1\}^d$ to $O^{\mathsf{KGen}_1}_{msk}$ to get sk_t, then returns sk_t.
3. $D^* := \mathtt{Divide}_d(L^*, R^*)$, $\mathbb{T}_{[D^*,R^*]} := \mathtt{LatterWID}_d(D^*, R^*)$.
4. $\mathbb{T}_{[L^*,D^*-1]} := \mathtt{FormerWID}_d(L^*, D^* - 1)$
5. $\mathbb{T}_{[L^*,R^*]} := \mathtt{Merge}_d(\mathbb{T}_{[D^*,R^*]}, \mathbb{T}_{[L^*,D^*-1]})$.
6. Parse $\mathbb{T}_{[L^*,R^*]}$ as $\{wID^*_i \mid i \in [1, k^*]\}$.
7. If $k^* < n$, for every $j \in [k^* + 1, n]$, $wID^*_j \xleftarrow{\mathrm{U}} \{wID^*_i \mid i \in [1, k^*]\}$.
8. **Return** $(\{wID^*_i \mid i \in [1, n]\}, m_0, m_1, st_1)$, where $st_1 := (st'_1, k^*)$.

$\mathsf{B}_2^{O^{\mathsf{KGen}_2}_{msk}}(st_1, \{C^*_i \mid i \in [1, n]\})$:

9. Parse st_1 as (st'_1, k^*).
10. **Return** $b' \leftarrow \mathsf{A}_2^{O^{\mathsf{KGen}_2}_{msk}}(st'_1, \{C^*_i \mid i \in [1, k^*]\})$,
    - ▷ $O^{\mathsf{KGen}_1}_{msk}(t \in [0, T-1])$:
      B_2 replies as B_1 replied to a query to $O^{\mathsf{KGen}_1}_{msk}$ from A_1.

Fig. 23. Algorithm B in the proof of Theorem 11

D.9 Proof of Theorem 12

Precisely speaking, we prove the following statement: For every $\lambda, L \in \mathbb{N}$ and every $\mathsf{A} \in \mathbb{PPT}_\lambda$ running in time t, there exists $\mathsf{B} \in \mathbb{PPT}_\lambda$ running in time $t + ((2+q_k)t_{exp} + q_k t_{mul})|wID^*|_*$, where $|\mathbb{W}(wID^*)|$ denotes number of wildcard symbol $*$ in the target wildcarded ID wID^* and t_{exp} (resp. t_{mul}) denotes computational time per one exponentiation (resp. multiplication) on the group $\mathbb{G}$, such that $\mathsf{Adv}^{\mathtt{IND\text{-}WID\text{-}CPA}}_{\Pi_{\mathrm{WIBE}},\mathsf{A},\lambda,L}(\lambda) = 2^L \cdot \mathsf{Adv}^{\mathtt{IND\text{-}ID\text{-}CPA}}_{\Pi_{\text{Wat-IBE}},\mathsf{B},\lambda,L'}(\lambda)$, where $L' := L - |wID^*|_*$.

We consider B which behaves as in Fig. 24. As we explained earlier, B plays $\mathbf{Expt}^{\mathtt{IND\text{-}ID\text{-}CPA}}_{\Pi_{\text{Wat-IBE}},\mathsf{B}}$ with $L' = L - |wID^*|_*$ which depends on the target wildcarded ID $wID^* \in \{0,1\}^L$ which will be chosen by A. We make B (or B_0) randomly guess the position of all wildcard symbols in wID^*, i.e., $\{i \ s.t. \ wID^*[i] = *\}$, before the experiment starts. Note that $\{i \ s.t. \ wID^*[i] \in \{0,1\}\}$ can be computed from $\{i \ s.t. \ wID^*[i] = *\}$ (and L) by $[1,L] \setminus \{i \ s.t. \ wID^*[i] = *\}$. Hereafter, we consider a situation where B_0 correctly guesses it. Note that the situation occurs with probability $1/2^L$. We assume that information about $\{i \ s.t. \ wID^*[i] = *\}$ is transmitted to B_1 at the same time as mpk, which means that B_1 knows $\{i \ s.t. \ wID^*[i] = *\}$ before wID^* is chosen by A_1.

It is easy to verify that mpk' given to A_1 distributes identically to the real one in $\mathbf{Expt}^{\mathtt{IND\text{-}ID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},0}$ or $\mathbf{Expt}^{\mathtt{IND\text{-}ID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},1}$. It is also easy to verify that $C^\star$ given to A_2 distributes identically to the real one in $\mathbf{Expt}^{\mathtt{IND\text{-}ID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},0}$ (resp. $\mathbf{Expt}^{\mathtt{IND\text{-}ID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},1}$) when $b = 0$ (resp. $b = 1$). It is also obvious that each secret-key $sk_{ID'}$ generated by B_1 or B_2 on $\mathtt{KGen}$ oracle distributes identically to the real one since it can be simply written as $(g_2^\alpha \cdot (u' \prod_{i \ s.t. \ ID'[i]=1} u_i)^r, g^r)$, where $r \xleftarrow{\mathrm{U}} \mathbb{Z}_p$. Thus, B perfectly simulates $\mathbf{Expt}^{\mathtt{IND\text{-}WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},0}$ (resp. $\mathbf{Expt}^{\mathtt{IND\text{-}WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},1}$) for A when B correctly guesses $\{i \ s.t. \ wID^*[i] = *\}$ (with probability $1/2^L$) and (unconsciously) plays $\mathbf{Expt}^{\mathtt{IND\text{-}ID\text{-}CPA}}_{\Sigma_{\text{Wat-IBE}},\mathsf{B},0}$ (resp. $\mathbf{Expt}^{\mathtt{IND\text{-}ID\text{-}CPA}}_{\Sigma_{\text{Wat-IBE}},\mathsf{B},1}$). Hence, it holds that $\Pr[1 \leftarrow \mathbf{Expt}^{\mathtt{IND\text{-}WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},0}]/2^L = \Pr[1 \leftarrow \mathbf{Expt}^{\mathtt{IND\text{-}ID\text{-}CPA}}_{\Sigma_{\text{Wat-IBE}},\mathsf{B},0}]$ (resp. $\Pr[1 \leftarrow \mathbf{Expt}^{\mathtt{IND\text{-}WID\text{-}CPA}}_{\Sigma_{\mathrm{WIBE}},\mathsf{A},1}]/2^L = \Pr[1 \leftarrow \mathbf{Expt}^{\mathtt{IND\text{-}ID\text{-}CPA}}_{\Sigma_{\text{Wat-IBE}},\mathsf{B},1}]$).

The reason why the the running time of B is $t + ((2+q_k)t_{exp} + q_k t_{mul})|wID^*|_*$ is that B_1 (resp. B_2) in step 3 (resp. step 14) needs to calculate $|wID^*|_*$ exponentiations, and B_1 and B_2 collectively need to calculate $q_k \cdot |wID^*|_*$ exponentiations and multiplications to generate q_k secret-keys on $\mathtt{KGen}_1$ and $\mathtt{KGen}_2$ in total.

E Detailed Comparison of Ciphertext-Sizes between IK/PQ-WIBE-RE Instantiated by Π_{WIBE}

Let $\mathbb{T}^{\mathtt{PQ}}_{[L,R]}$ denote the set of wildcarded IDs (deterministically) chosen from a range $[L,R]$ in PQ-WIBE-RE w. Π_{WIBE}. Let $\mathbb{T}^{\mathtt{IK}}_{[L,D-1]}$ (resp. $\mathbb{T}^{\mathtt{IK}}_{[D,R]}$) denote the set of wIDs determined from $[L,D-1]$ (resp. $[D,R]$) in IK-WIBE-RE w. Π_{WIBE}. Let $\mathbb{T}^{\mathtt{IK}}_{[L,R]}$ denote the finally determined set of wIDs for $[L,R]$ in IK-WIBE-RE w. Π_{WIBE}.

It holds that for every $\mathtt{A} \in \{\mathtt{PQ}, \mathtt{IK}\}$, $|C^{\mathtt{A}}_{[L,R]}| = \sum_{wID \in \mathbb{T}^{\mathtt{A}}_{[L,R]}} \{|g_T| + (2 + |wID|_*)|g|\}$. We say that two sets of wIDs $\mathbb{T}_1$ and $\mathbb{T}_2$ are *structurally identical* if there exists a wID in one of the sets which covers a range, then there also exists another wID in the other one of the sets which covers the same range. Obviously, if $\mathbb{T}^{\mathtt{PQ}}_{[L,R]}$ and $\mathbb{T}^{\mathtt{IK}}_{[L,R]}$ are structurally identical, then $|C^{\mathtt{PQ}}_{[L,R]}| = |C^{\mathtt{IK}}_{[L,R]}|$. We can easily prove that for every $[L,R]$, $\mathbb{T}^{\mathtt{PQ}}_{[L,R]}$ and $\mathbb{T}^{\mathtt{IK}}_{[L,D-1]} \cup \mathbb{T}^{\mathtt{IK}}_{[D,R]}$ are structurally identical.

For some ranges $[L,R]$, $\mathbb{T}^{\mathtt{IK}}_{[L,R]} = \mathbb{T}^{\mathtt{IK}}_{[L,D-1]} \cup \mathbb{T}^{\mathtt{IK}}_{[D,R]}$, e.g., the first three examples in Table 3. In this case, since $\mathbb{T}^{\mathtt{PQ}}_{[L,R]}$ and $\mathbb{T}^{\mathtt{IK}}_{[L,R]}$ are structurally identical, $|C^{\mathtt{PQ}}_{[L,R]}| = |C^{\mathtt{IK}}_{[L,R]}|$.

For the other ranges $[L,R]$, $\mathbb{T}^{\mathtt{IK}}_{[L,R]} \neq \mathbb{T}^{\mathtt{IK}}_{[L,D-1]} \cup \mathbb{T}^{\mathtt{IK}}_{[D,R]}$, e.g., the last three examples in Table 3. In this case, $|C^{\mathtt{PQ}}_{[L,R]}| > |C^{\mathtt{IK}}_{[L,R]}|$. Consider a case that a wID in $\mathbb{T}^{\mathtt{IK}}_{[D,R]}$ and a wID' in $\mathbb{T}^{\mathtt{IK}}_{[L,D-1]}$, where $|wID|_* = |wID'|_* = k \in [0, \log T - 1]$, are merged into a $wID^\star$ with $|wID^\star|_* = k+1$. In this case, by the merging, ciphertext size is reduced by $|g_T| + (k+1)|g|$. For instance, when $[L,R] = [1, T-2]$, $|C^{\mathtt{PQ}}_{[L,R]}| = (2\log T - 2)|g_T| + (\log^2 T + \log T - 2)|g|$ and $|C^{\mathtt{IK}}_{[L,R]}| = (\log T - 1)|g_T| + \frac{1}{2}(\log^2 T + 3\log T - 4)|g| = \frac{1}{2}|C^{\mathtt{PQ}}_{[L,R]}| + (\log T - 1)|g|$ which is almost the half of $|C^{\mathtt{PQ}}_{[L,R]}|$.

$\mathsf{B}_1^{\mathcal{O}_{msk}^{\mathsf{KGen}_1}}(mpk, (\{i \text{ s.t. } wID^*[i] = *\}))$:

1. Parse mpk as $(p, \mathbb{G}, \mathbb{G}_T, e, g, g_1, g_2, u', \{u'_i \mid i \text{ s.t. } wID^*[i] \in \{0,1\}\})$.
2. For every i s.t. $wID^*[i] \in \{0,1\}$, $u_i := u'_i$.
3. For every i s.t. $wID^*[i] = *$, $\beta_i \xleftarrow{\mathrm{U}} \mathbb{Z}_p$, $u_i := g^{\beta_i}$.
4. $mpk' := (p, \mathbb{G}, \mathbb{G}_T, e, g, g_1, g_2, u', u_0, \cdots, u_{L-1}, v_1)$.
5. $(wID^*, m_0, m_1, st'_1) \leftarrow \mathsf{A}_1^{\mathcal{O}_{msk}^{\mathsf{KGen}_1}}(mpk')$.

▷ $\mathcal{O}_{msk}^{\mathsf{KGen}_1}(ID' \in \{0,1\}^L)$:

6. B_1 parses ID' as $\|_{i\in[0,L-1]} h'_i$, where $h'_i \in \{0,1\}$.
7. B_1 queries $ID := \|_{i \text{ s.t. } wID^*[i]\in\{0,1\}} h'_i$ to his own $\mathcal{O}_{msk}^{\mathsf{KGen}_1}$ to get $sk_{ID} = (g_2^{\alpha} \cdot (u' \prod_{i \text{ s.t. } wID^*[i]\in\{0,1\} \wedge ID'[i]=1} u_i)^r, g^r) =: (d_1, d_2)$, where $\alpha, r \in \mathbb{Z}_p$ cannot be seen by B_1.
8. B_1 returns $sk_{ID'} := (d_1 \cdot \prod_{i \text{ s.t. } wID^*[i]=* \wedge ID'[i]=1} d_2^{\beta_i}, d_2)$.

9. $st_1 := (st'_1, \{\beta_i \mid i \in \mathbb{W}(wID^*)\})$.
10. Parse wID^* as $\|_{i\in[0,L-1]} h^*_i$ where $h^*_i \in \{0,1,*\}^L$.
11. **Return** (ID^*, m_0, m_1, st_1), where $ID^* := \|_{i \text{ s.t. } wID^*[i]\in\{0,1\}} h^*_i$.

$\mathsf{B}_2^{\mathcal{O}_{msk}^{\mathsf{KGen}_2}}(st_1, C^*)$:

12. Parse st_1 as $(st'_1, \{\beta_i \mid i \text{ s.t. } wID^*[i] = *\})$.
13. Parse C^* as $(e(g_1,g_2)^s \cdot m_b, g^s, (u' \prod_{i\in\mathbb{U}(ID^*)} u_i)^s)$, where s and b are unknown to B_2.
14. $C^\star := ((e(g_1,g_2)^s \cdot m_b, g^s, (u' \prod_{i \text{ s.t. } ID^*[i]=1} u_i)^s, \{(g^s)^{\beta_i} \mid i \text{ s.t. } wID^*[i] = *\})$.
15. **Return** $b' \leftarrow \mathsf{A}_2^{\mathcal{O}_{msk}^{\mathsf{KGen}_2}}(st_1, C^\star)$.

▷ $\mathcal{O}_{msk}^{\mathsf{KGen}_2}(ID' \in \{0,1\}^L)$:

B_2 replies as B_1 replied to a query to $\mathcal{O}_{msk}^{\mathsf{KGen}_1}$ from A_1.

Fig. 24. Algorithm B in the proof of Theorem 12

Table 3. Ciphertext sizes of IK/PQ-WIBE-RE instantiated by Π_{WIBE} for some ranges $[L, R]$

$[L,R]$	$\lvert C^{\mathrm{PQ}}_{[L,R]}\rvert$	$\lvert C^{\mathrm{IK}}_{[L,R]}\rvert$
$[0,0]$	$\lvert g_T\rvert + 2\lvert g\rvert$	
$[0, T-1]$	$\lvert g_T\rvert + (2+\log T)\lvert g\rvert$	
$[1, 2^{\log T-1} - 1 + \sum_{i\in[0,\log T-3]} 2^i]$	$(2\log T - 3)\lvert g_T\rvert + (\log^2 T - 2)\lvert g\rvert$	
$[T-1, 0]$	$2\lvert g_T\rvert + 4\lvert g\rvert$	$\lvert g_T\rvert + 3\lvert g\rvert$
$[1, 2^{\log T-1} - 2]$	$(2\log T - 4)\lvert g_T\rvert$ $+(\log^2 T - \log T - 2)\lvert g\rvert$	$(\log T - 2)\lvert g_T\rvert$ $+\frac{1}{2}(\log^2 T + \log T - 6)\lvert g\rvert$
$[1, T-2]$ or $[2^{\log T-1}+1, 2^{\log T-2} - 2]$	$(2\log T - 2)\lvert g_T\rvert$ $+(\log^2 T + \log T - 2)\lvert g\rvert$	$(\log T - 1)\lvert g_T\rvert$ $+\frac{1}{2}(\log^2 T + 3\log T - 4)\lvert g\rvert$