More Efficient Adaptively Secure Lattice-based IBE with Equality Test in the Standard Model *

Kyoichi Asano[1, 2], Keita Emura[2], and Atsushi Takayasu[3]

[1]The University of Electro-Communications, Japan.
[2]National Institute of Information and Communications Technology, Japan.
[3]The University of Tokyo, Japan.

November 28, 2022

Abstract

Identity-based encryption with equality test (IBEET) is a variant of identity-based encryption (IBE), where any users who have trapdoors can check whether two ciphertexts are encryption of the same plaintext. Although several lattice-based IBEET schemes have been proposed, they have drawbacks in either security or efficiency. Specifically, most schemes satisfy only selective security, while adaptively secure schemes in the standard model suffer from large master public keys that consist of linear numbers of matrices. In other words, known lattice-based IBEET schemes perform poorly compared to the state-of-the-art lattice-based IBE schemes (without equality test). In this paper, we propose a semi-generic construction of CCA-secure lattice-based IBEET from a certain class of lattice-based IBE schemes. As a result, we obtain the first lattice-based IBEET schemes with adaptive security and CCA security in the standard model. Furthermore, our semi-generic construction can use several state-of-the-art lattice-based IBE schemes as underlying schemes. Then, we have adaptively secure lattice-based IBEET schemes whose public keys have only poly-log matrices.

1 Introduction

Encryption is a fundamental tool for providing data confidentiality. On the other hand, it affects several functions such as searching, comparing, partitioning, and so on. Yang et al. [YTH+10] proposed public key encryption with equality test (PKEET) which allows us to check whether two plaintexts of two ciphertexts are the same or not. This equality check allows us to provide a keyword search on encrypted data, data partitioning on an encrypted database, and so on. Although anyone can run the test algorithm in the Yang's definition, a trapdoor for running the test algorithm is introduced in subsequent works, e.g., [LSQ18, LLS+20]. Identity-based encryption with equality test (IBEET) [Ma16] is an extension of PKEET that can simplify the certificate management of PKEET. As in identity-based encryption (IBE), an identity id is used as a public key for generating a ciphertext $\mathsf{ct}_{\mathsf{id}}$. A secret key $\mathsf{sk}_{\mathsf{id}}$ of an identity id can generate a trapdoor $\mathsf{td}_{\mathsf{id}}$. By using trapdoors $\mathsf{td}_{\mathsf{id}_0}$ and $\mathsf{td}_{\mathsf{id}_1}$, we can check whether $\mathsf{ct}_{\mathsf{id}_0}$ and $\mathsf{ct}_{\mathsf{id}_1}$ are encryptions of the same plaintexts.

Although several CCA-secure IBEET schemes have been proposed by assuming the hardness of Diffie-Hellman-type assumptions, e.g., [LLS+16, Ma16], they are vulnerable against quantum

*An extended abstract appeared at ISC 2022 [AET22]. This is the full version.

attacks. To achieve post-quantum security, several lattice-based IBEET schemes have been proposed. There are two ways for constructing lattice-based IBEET schemes. One is instantiating lattice-based schemes from generic constructions of IBEET and the other is direct constructions by modifying known lattice-based IBE schemes.

At first, we review two known generic constructions of IBEET that can instantiate lattice-based schemes. Lin et al. [LSQ18] proposed a generic construction of CCA-secure IBEET from CCA-secure IBE in the random oracle model. Lee et al. [LLS+20] proposed a generic construction of CCA-secure IBEET from three-level CPA-secure hierarchical identity-based encryption (HIBE) and one-time signatures (OTSs) in the standard model, where OTSs are used for achieving CCA security via the Canetti-Halevi-Katz (CHK) transformation [CHK04]. Lee et al.'s construction provides adaptively secure lattice-based schemes in the quantum random oracle model (QROM) based on adaptively secure lattice-based HIBE schemes in the QROM [ABB10b, CHK+12, Zha12]. Lee et al.'s construction also provides selectively secure lattice-based schemes in the standard model based on selectively secure HIBE schemes in the standard model [ABB10a, CHK+12]. However, their construction does not provide *purely* adaptively secure lattice-based IBEET schemes in the standard model since there are no known adaptively secure lattice-based HIBE schemes in the standard model. Although Singh et al. [SRB12] constructed an adaptively secure lattice-based HIBE scheme in the standard model based on Agrawal et al.'s adaptively secure non-hierarchical IBE scheme [ABB10a], the scheme achieves only bounded security in the sense that the size of a modulus q depends on the number of adversary's key extraction queries. Thus, the instantiation of the Lee et al.'s generic construction from the Singh et al.'s HIBE scheme does not satisfy purely adaptive security. Next, we review four known direct constructions of lattice-based IBEET schemes [DLR+19, NSD+20, SDL20, WWY+21], where all known schemes were studied in the standard model. Duong et al.'s IBEET scheme [DLR+19] and Nguyen et al.'s IBEET scheme [NSD+20] are based on Agrawal et al.'s adaptively secure IBE scheme [ABB10a] achieving adaptive and CPA security. Unfortunately, due to the nature of Agrawal et al.'s IBE scheme, these IBEET schemes achieve only bounded security as the case of Singh et al.'s adaptively secure HIBE scheme [SRB12]. Susilo et al.'s IBEET scheme [SDL20] that is similar to Lee et al.'s generic construction [LLS+20] is based on Agrawal et al.'s selectively secure IBE scheme [ABB10a] achieving selective and CCA security. Wu et al.'s IBEET scheme [WWY+21] is based on Tsabary's IBE scheme [Tsa19] achieving adaptive and CPA security.

Summarizing the situation, almost all known lattice-based IBEET schemes in the standard model achieve only selective and CCA security [LLS+20, SDL20] or adaptive and CPA security [DLR+19, NSD+20, WWY+21] with the only exception that Lee et al.'s generic construction [LLS+20] instantiated by Singh et al.'s HIBE scheme [SRB12]. Moreover, almost all adaptively secure schemes achieve only bounded security with the only exception that Wu et al.'s CPA-secure scheme [WWY+21]. Therefore, constructing purely adaptive and CCA-secure lattice-based IBEET scheme is an interesting open problem. Moreover, known adaptively secure IBEET schemes have a common bottleneck in terms of efficiency. Although there are adaptively secure lattice-based IBE schemes such as Yamada's IBE scheme [Yam17] and Jager-Kurek-Niehues's (JKN) IBE scheme [JKN21][1] whose public keys consist of poly-log matrices, public keys of known adaptively secure lattice-based IBEET schemes [SRB12, DLR+19, NSD+20, WWY+21] consist of matrices whose numbers are (almost) linear in the length of identities or the security parameter. Therefore, it is desirable to construct adaptively secure lattice-based IBEET schemes whose public keys consist of poly-log matrices.

[1] Although Yamada's scheme is purely secure under the LWE assumption, JKN scheme enjoys smaller LWE parameters at the expense of additionally employing near-collision resistance hash functions.

Table 1: Comparison of lattice-based IBEET schemes in the standard model.

Scheme	$\lvert\mathsf{mpk}\rvert$	adaptive?	CCA?	Comment
LLS+20 [LLS+20] +ABB10 [ABB10a]	$O(n^2 \log n)$	selective	CCA	
LLS+20 [LLS+20] +SRB12 [SRB12]	$O(Ln^2 \log Q)$	adaptive	CCA	Q-bounded
DLRS19 [DLR+19]	$O(Ln^2 \log Q)$	adaptive	CPA	Q-bounded
NSD+20 [NSD+20]	$O(Ln^2 \log Q)$	adaptive	CPA	Q-bounded
SDL20 [SDL20]	$O(n^2 \log n)$	selective	CCA	
WWY+21[WWY+21]	$O(\lambda n^2 \log n)$	adaptive	CPA	
Ours [LLS+20] +Yam17 [Yam17]	$O((\log \lambda)^3 n^2 \log n)$	adaptive	CCA	
Ours [LLS+20] +JKN21 [JKN21]	$O((\log \lambda) n^2 \log n)$	adaptive	CCA	Near collision resistant hash function is required

1.1 Our Contribution

In this paper, we construct the first purely adaptive and CCA-secure lattice-based IBEET schemes in the standard model. One promising way for constructing such a desirable scheme is constructing adaptively secure lattice-based HIBE schemes based on known adaptively secure IBE schemes. Specifically, as the Waters pairing-based HIBE scheme [Wat05], we can obtain such a HIBE scheme by sacrificing reduction loss. However, we take another approach to resolve the problem without sacrificing reduction loss very much. In particular, we propose a semi-generic construction of CCA-secure lattice-based IBEET from CPA-secure lattice-based IBE whose structure is similar to Agrawal-Boneh-Boyen (ABB)'s IBE scheme [ABB10a] which we call ABB-type IBE. The resulting IBEET schemes achieve adaptive security if the underlying IBE schemes satisfy adaptive security. Intuitively, a ciphertext and secret key for the same id of ABB-type IBE is associated with the same publicly computable matrix. Thanks to the semi-generic construction, we propose the first purely adaptive and CCA-secure lattice-based IBEET schemes. Moreover, since ABB-type IBE covers Yamada's IBE scheme [Yam17] and JKN IBE scheme [JKN21], we can obtain the first adaptive lattice-based IBEET schemes whose public keys consist of poly-log matrices. See Table 1 for the detailed comparison, where Q denotes the number of adversary's secret key extraction queries and L denotes the length of identities in [DLR+19, NSD+20].

The idea of our semi-generic construction is similar to Lee et al.'s generic construction [LLS+20] from three-level CPA-secure HIBE.[2] Recall that Lee et al. proved that adaptively secure three-level CPA-secure HIBE is sufficient for constructing adaptively and CCA-secure IBEET. Basically, ciphertexts of all IBEET schemes consist of two types of ciphertexts, one is responsible only for decryption and the other is also responsible for equality test. Lee et al. utilized each three hierarchical levels for id, ciphertext type 0 or 1, and verification keys of OTSs for the CHK transforma-

[2] In this paper, we do not follow Lee et al.'s argument [LLS+20] in a security proof but follow Asano et al.'s one [AET+22] which is an attribute-based extension of Lee et al.'s work with a refined proof.

tion [CHK04], respectively. Then, by using the CPA security of the underlying HIBE, Lee et al. hide the challenge plaintext for both types of ciphertexts one by one. Let's take a closer look at this proof strategy. It is widely known that the CHK transformation can convert CPA-secure IBE to CCA-secure public key encryption even when the underlying IBE satisfies only selective security. In other words, Lee et al.'s generic construction does not require HIBE with adaptive security for all hierarchical levels to construct adaptively secure and CCA-secure IBEET. In turn, a special three-level HIBE scheme that satisfies adaptive security only for the first level and selective security for the other levels is sufficient for our purpose. To this end, we construct such special three-level HIBE schemes from ABB-type IBE such as Yamada's scheme [Yam17] and JKN scheme [JKN21]. Briefly speaking, the first level and the other levels of our HIBE scheme are the same as those of the underlying IBE scheme, i.e., Yamada's scheme and the JKN scheme, and Agrawal-Boneh-Boyen's selectively secure HIBE scheme, respectively. Then, a slight modification of Agrawal et al.'s proof technique is applicable to the special three-level HIBE. Moreover, since we employ a semi-generic construction from ABB-type IBE, we do not need to make complex arguments to achieve adaptive security such as [Yam17, JKN21].

2 Preliminaries

Notation. Throughout the paper, λ denotes a security parameter. For a finite set S, $s \leftarrow_{\$} S$ denotes a sampling of an element s from S uniformly at random and let $|S|$ denotes a cardinality of S. We represent vectors by bold-face letters like $\mathbf{u}$, and matrices by bold-face capital letters like $\mathbf{U}$. For an $n \times m$ matrix $\mathbf{A}_1$ and $n \times m'$ matrix $\mathbf{A}_2$, let $[\mathbf{A}_1 \| \mathbf{A}_2]$ denotes the $n \times (m + m')$ matrix formed by concatenating $\mathbf{A}_1$ and $\mathbf{A}_2$. We use a similar notation for vectors. For a matrix $\mathbf{R}$, $\|\mathbf{R}\|$ denotes its norm. Let $\approx$ and $\approx_c$ denote statistical indistinguishability and computational indistinguishability, respectively.

2.1 Lattices

A (full-rank) m-dimensional integer lattice $\Lambda \subseteq \mathbb{Z}^m$ is a set of m-dimensional integer vectors with the form $\left\{\sum_{i\in[m]} x_i \mathbf{b}_i \mid x_i \in \mathbb{Z}\right\}$, where $\{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$ is called the basis of the lattice .

Gaussian Distributions. For an integer $m > 0$, let $D_{\Lambda,\tau}$ be the discrete Gaussian distribution over a m-dimensional integer lattice Λ with parameter $\tau > 0$. Regarding the Gaussian distribution, the following lemmas hold.

Lemma 2.1 ([Reg09], Lemma 2.5)**.** We have $\Pr\left[\|\mathbf{x}\| > \tau\sqrt{m} : \mathbf{x} \leftarrow D_{\mathbb{Z}^m,\tau}\right] \leq 2^{-2m}$.

Lemma 2.2 ([ABB10a], Lemma 15)**.** Let $\mathbf{R}$ be a $n \times m$ matrix chosen at random from $\{-1, 1\}^{n\times m}$. Then there is a universal constant C such that

$$\Pr\left[\|\mathbf{R}\| > C\sqrt{n+m}\right] < e^{-(n+m)}.$$

Gadget Matrix. Let $n, q \in \mathbb{Z}$ and $m \geq n\lceil \log q \rceil$. A gadget matrix $\mathbf{G}$ is defined as $\mathbf{I}_n \otimes (1, 2, \ldots, 2^{\lceil \log q \rceil - 1})$ padded with $m - n\lceil \log q \rceil$ zero columns. For any t, there exists an efficient deterministic algorithm $\mathbf{G}^{-1} : \mathbb{Z}_q^{n\times t} \to \{0,1\}^{m\times t}$ that takes $\mathbf{U} \in \mathbb{Z}_q^{n\times t}$ as input and outputs $\mathbf{V} \in \{0,1\}^{m\times t}$ such that $\mathbf{GV} = \mathbf{U}$.

Trapdoors. Here, we summarize the properties of lattice trapdoors based on the presentation by Brakerski and Vaikuntanathan [BV16]. Let $n, m, q \in \mathbb{N}$ and consider a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$. For all

$\mathbf{V} \in \mathbb{Z}_q^{n \times m'}$, we let $\mathbf{A}_\tau^{-1}(\mathbf{V})$ denote the random variable whose distribution is a Gaussian $(D_{\mathbb{Z}^m,\tau})^{m'}$ conditioned on $\mathbf{A} \cdot \mathbf{A}_\tau^{-1}(\mathbf{V}) = \mathbf{V}$. A τ-trapdoor for $\mathbf{A}$ is a procedure that can sample from the distribution $\mathbf{A}_\tau^{-1}(\mathbf{V})$ in time $\mathsf{poly}(n, m, m', \log q)$, for any $\mathbf{V}$. We slightly overload the notation and denote a τ-trapdoor for $\mathbf{A}$ by $\mathbf{A}_\tau^{-1}$. We have the following:

Lemma 2.3 ([Ajt96, GPV08, ABB10a, ABB10b, CHK+12, BLP+13], Properties of Trapdoors)**.** Lattice trapdoors exhibit the following properties.

1. Given $\mathbf{A}_\tau^{-1}$, one can obtain $\mathbf{A}_{\tau'}^{-1}$, for any $\tau' \geq \tau$.
2. Given $\mathbf{A}_\tau^{-1}$, one can deterministically obtain $[\mathbf{A}\|\mathbf{B}]_\tau^{-1}$ for any $\mathbf{B}$.
3. Given $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{R} \in \mathbb{Z}^{m \times N}$ with $N > n\lceil \log q \rceil$, and a full-rank matrix $\mathbf{H} \in \mathbb{Z}_q^{n \times n}$, one can obtain $[\mathbf{A}\|\mathbf{AR} + \mathbf{HG}]_\tau^{-1}$ for $\tau = m \cdot \|\mathbf{R}\| \cdot \omega(\sqrt{\log m})$.
4. Given $\mathbf{A}_\tau^{-1}$, one can randomize it and obtain $\mathbf{A}_{\tau'}^{-1}$ for $\tau' = \tau \cdot \omega(\sqrt{m})$.
5. There exists an efficient procedure $\mathsf{TrapGen}(1^n, 1^m, q)$ that outputs $(\mathbf{A}, \mathbf{A}_{\tau_0}^{-1})$ where $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ for some $m = O(n \log q)$ and is 2^{-n}-close to uniform, where $\tau_0 = \omega(\sqrt{n \log q \log n})$.
6. For $\mathbf{A}_\tau^{-1}$ and $\mathbf{u} \in \mathbb{Z}_q^n$, it follows $\Pr[\|\mathbf{A}_\tau^{-1}(\mathbf{u})\| > \tau\sqrt{m}] = \mathsf{negl}(n)$.

Useful Facts. We will use the following facts.

Lemma 2.4 ([ABB10a], Leftover Hash Lemma)**.** Let $q > 2$ be a prime, m, n, k be positive integers such that $m > (n+1)\log q + \omega(\log n)$, $k = \mathsf{poly}(n)$. Then, if we sample $\mathbf{A} \leftarrow_\$ \mathbb{Z}_q^{n \times m}$, $\mathbf{R} \leftarrow_\$ \{-1, 1\}^{m \times k}$, and $\mathbf{B} \leftarrow_\$ \mathbb{Z}_q^{n \times k}$, then the distribution of $(\mathbf{A}, \mathbf{AR}, \mathbf{R}^\top \mathbf{w})$ is statistically close to the distribution of $(\mathbf{A}, \mathbf{B}, \mathbf{R}^\top \mathbf{w})$ for all vectors $\mathbf{w} \in \mathbb{Z}_q^m$.

Definition 2.1 ([ABB10a], Full Rank Difference Map)**.** Let q be a prime and n be a positive integer. We say that a function $\mathsf{FRD} : \mathcal{ID} \to \mathbb{Z}_q^{n \times n}$ is an encoding with full-rank differences if: for all distinct $\mathsf{id}, \mathsf{id}' \in \mathcal{ID}$, a matrix $\mathsf{FRD}(\mathsf{id}) - \mathsf{FRD}(\mathsf{id}') \in \mathbb{Z}_q^{n \times n}$ is full rank, and FRD is computable in polynomial time in $n \log q$.

Hardness Assumption. We will use the following LWE assumption to prove security of the proposed scheme.

Definition 2.2 ([Reg09], Learning with Errors)**.** For integers n, m, a prime $q > 2$, a real number $\alpha \in (0, 1)$, and a probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$, an advantage for the learning with errors problem $\mathsf{dLWE}_{n,m,q,\alpha}$ of $\mathcal{A}$ is defined as follows:

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{dLWE}_{n,m,q,\alpha}}(\lambda) := \left|\Pr[\mathcal{A}(\mathbf{A}, \mathbf{s}^\top \mathbf{A} + \mathbf{x}^\top) = 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{w}^\top) = 1]\right|,$$

where $\mathbf{A} \leftarrow_\$ \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \leftarrow_\$ \mathbb{Z}_q^n$, $\mathbf{x} \leftarrow D_{\mathbb{Z}^m,\alpha q}$, $\mathbf{w} \leftarrow_\$ \mathbb{Z}_q^m$. We say that the $\mathsf{dLWE}_{n,m,q,\alpha}$ assumption holds if $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{dLWE}_{n,m,q,\alpha}}(\lambda)$ is negligible in λ for all $\mathcal{A}$.

2.2 Identity-based Encryption

Syntax. An IBE scheme Π consists of the four algorithms ($\mathsf{IBE.Setup}$, $\mathsf{IBE.KeyGen}$, $\mathsf{IBE.Enc}$, $\mathsf{IBE.Dec}$) as follows:

$\mathsf{IBE.Setup}(1^\lambda) \to (\mathsf{IBE.mpk}, \mathsf{IBE.msk})$**:** On input the security parameter 1^λ, it outputs a master public key $\mathsf{IBE.mpk}$ and a master secret key $\mathsf{IBE.msk}$. We assume that $\mathsf{IBE.mpk}$ contains a description of a plaintext space and an identity space $\mathcal{ID}$ that are determined only by the security parameter λ.

$\mathsf{IBE.KeyGen}(\mathsf{IBE.mpk}, \mathsf{IBE.msk}, \mathsf{id}) \to \mathsf{IBE.sk_{id}}$**:** On input a master public key $\mathsf{IBE.mpk}$, a master secret key $\mathsf{IBE.msk}$, and an identity $\mathsf{id} \in \mathcal{ID}$, it outputs a secret key $\mathsf{IBE.sk_{id}}$.

$\mathsf{IBE.Enc}(\mathsf{IBE.mpk}, \mathsf{id}, \mathsf{M}) \to \mathsf{IBE.ct_{id}}$**:** On input a master public key $\mathsf{IBE.mpk}$, an identity $\mathsf{id} \in \mathcal{ID}$, and a plaintext $\mathsf{M} \in \mathcal{M}$, it outputs a ciphertext $\mathsf{IBE.ct_{id}}$.

$\mathsf{IBE.Dec}(\mathsf{IBE.mpk}, \mathsf{IBE.sk_{id}}, \mathsf{IBE.ct_{id}}) \to \mathsf{M}$ or $\perp$**:** On input a master public key $\mathsf{IBE.mpk}$, a secret key $\mathsf{IBE.sk_{id}}$, and a ciphertext $\mathsf{IBE.ct_{id}}$, it outputs the decryption result M or $\perp$.

Correctness. For all $\lambda \in \mathbb{N}$, all $(\mathsf{IBE.mpk}, \mathsf{IBE.msk}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$, all $\mathsf{M} \in \mathcal{M}$, all $\mathsf{id} \in \mathcal{ID}$, it is required that $\mathsf{M}' = \mathsf{M}$ holds with overwhelming probability, where $\mathsf{IBE.ct_{id}} \leftarrow \mathsf{IBE.Enc}(\mathsf{IBE.mpk}, \mathsf{id}, \mathsf{M})$, $\mathsf{IBE.sk_{id}} \leftarrow \mathsf{IBE.KeyGen}(\mathsf{IBE.mpk}, \mathsf{IBE.msk}, \mathsf{id})$, and $\mathsf{M}' \leftarrow \mathsf{IBE.Dec}(\mathsf{IBE.mpk}, \mathsf{IBE.sk_{id}}, \mathsf{IBE.ct_{id}})$.

Security. We consider adaptive IND-CPA security defined below.

Definition 2.3 (Adaptive IND-CPA Security)**.** The adaptive IND-CPA security of an IBE scheme Π is defined by a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ as follows:

Init: $\mathcal{C}$ runs $(\mathsf{IBE.mpk}, \mathsf{IBE.msk}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$ and gives $\mathsf{IBE.mpk}$ to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ is allowed to make the following type of query to $\mathcal{C}$.

> **Key extraction query:** $\mathcal{A}$ is allowed to make the query on $\mathsf{id} \in \mathcal{ID}$. Upon the query, $\mathcal{C}$ runs $\mathsf{IBE.sk_{id}} \leftarrow \mathsf{IBE.KeyGen}(\mathsf{IBE.mpk}, \mathsf{IBE.msk}, \mathsf{id})$ and returns $\mathsf{IBE.sk_{id}}$ to $\mathcal{A}$.

Challenge query: $\mathcal{A}$ is allowed to make the query only once. Upon $\mathcal{A}$'s query on $(\mathsf{id}^*, \mathsf{M}^*) \in \mathcal{ID} \times \mathcal{M}$, where id^* was not be queried on key extraction queries in Phase 1. Then, $\mathcal{C}$ flips a coin $\mathsf{coin} \leftarrow_{\$} \{0,1\}$ and runs $\mathsf{IBE.ct^*_{id^*}} \leftarrow \mathsf{IBE.Enc}(\mathsf{IBE.mpk}, \mathsf{id}^*, \mathsf{M}^*)$ if $\mathsf{coin} = 0$, otherwise samples $\mathsf{IBE.ct^*_{id^*}}$ from a ciphertext space uniformly at random. Finally, $\mathcal{C}$ returns $\mathsf{IBE.ct^*_{id^*}}$ to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ is allowed to make key extraction queries as in Phase 1 except that $\mathcal{A}$ cannot query id^*.

Guess: $\mathcal{A}$ outputs a $\widehat{\mathsf{coin}}$ as a guess of coin.

The adversary $\mathcal{A}$ wins in the above game if $\widehat{\mathsf{coin}} = \mathsf{coin}$ and the advantage is defined to

$$\mathsf{Adv}^{\text{IND-CPA}}_{\Pi,\mathcal{A}}(\lambda) := \left| \Pr\left[\widehat{\mathsf{coin}} = \mathsf{coin}\right] - \frac{1}{2} \right|.$$

If $\mathsf{Adv}^{\text{IND-CPA}}_{\Pi,\mathcal{A}}(\lambda)$ is negligible in the security parameter λ for all PPT adversaries $\mathcal{A}$, an IBE scheme Π is said to satisfy adaptive IND-CPA security.

2.3 One-time Signature

Syntax. An one-time signature (OTS) scheme Γ consists of three algorithms ($\mathsf{Sig.Setup}$, $\mathsf{Sig.Sign}$, $\mathsf{Sig.Vrfy}$) with a message space $\mathcal{M}$ as follows:

$\mathsf{Sig.Setup}(1^\lambda) \to (\mathsf{verk}, \mathsf{sigk})$**:** On input the security parameter 1^λ, it outputs a verification key verk and a signing key sigk.

$\mathsf{Sig.Sign}(\mathsf{sigk}, \mathsf{M}) \to \sigma$**:** On input a signing key sigk and a message $\mathsf{M} \in \mathcal{M}$, it outputs a signature σ.

$\mathsf{Sig.Vrfy}(\mathsf{verk}, \mathsf{M}, \sigma) \to 1$ or 0**:** On input a verification key verk, a message $\mathsf{M} \in \mathcal{M}$, and its signature σ, it outputs 1 if the signature is valid and 0 otherwise.

Correctness. For all $\lambda \in \mathbb{N}$, all $(\mathsf{verk}, \mathsf{sigk}) \leftarrow \mathsf{Sig.Setup}(1^\lambda)$, and all $\mathsf{M} \in \mathcal{M}$, it holds that $\mathsf{Sig.Vrfy}(\mathsf{verk}, \mathsf{M}, \mathsf{Sig.Sign}(\mathsf{sigk}, \mathsf{M})) = 1$ with overwhelming probability.

Security. We define a security notion for OTS. Let Γ be an OTS scheme, and we consider a game between an adversary $\mathcal{A}$ and the challenger $\mathcal{C}$. The game is parameterized by the security parameter λ. The game proceeds as follows: $\mathcal{C}$ first runs $(\mathsf{verk}, \mathsf{sigk}) \leftarrow \mathsf{Sig.Setup}(1^\lambda)$ and gives verk to $\mathcal{A}$. $\mathcal{A}$ is allowed to make the *signature generation query* only once: upon a query $\mathsf{M} \in \{0,1\}^*$ from $\mathcal{A}$, $\mathcal{C}$ returns $\sigma \leftarrow \mathsf{Sig.Sign}(\mathsf{sigk}, \mathsf{M})$ to $\mathcal{A}$. $\mathcal{A}$ outputs $(\widehat{\mathsf{M}}, \hat{\sigma})$ and terminates. In this game, $\mathcal{A}$'s advantage is defined by

$$\mathsf{Adv}^{\mathsf{OTS}}_{\Gamma,\mathcal{A}}(\lambda) := \Pr[\mathsf{Sig.Vrfy}(\mathsf{verk}, \widehat{\mathsf{M}}, \hat{\sigma}) \to 1 \wedge (\widehat{\mathsf{M}}, \hat{\sigma}) \neq (\mathsf{M}, \sigma)].$$

Definition 2.4 (Strong Unforgeability)**.** We say that an OTS scheme Γ satisfies strong unforgeability, if the advantage $\mathsf{Adv}^{\mathsf{OTS}}_{\Gamma,\mathcal{A}}(\lambda)$ is negligible for all PPT adversaries $\mathcal{A}$.

2.4 Hash Functions

Let $\mathsf{H} : \mathcal{M} \to \mathcal{M}$ be a hash function. We require the following properties of hash functions for our schemes.

Definition 2.5 (One-wayness)**.** We say that a hash function H is one-way (or preimage resistant) if for all PPT adversaries $\mathcal{A}$,

$$\mathsf{Adv}^{\mathrm{OW}}_{\mathsf{H},\mathcal{A}}(\lambda) := \Pr\left[\mathsf{M}^* \leftarrow_{\$} \mathcal{M}, \widehat{\mathsf{M}} \leftarrow \mathcal{A}(\mathsf{H}(\mathsf{M}^*)) : \mathsf{H}(\widehat{\mathsf{M}}) = \mathsf{H}(\mathsf{M}^*)\right]$$

is negligible in λ.

Definition 2.6 (Collision Resistance)**.** We say that a hash function H is collision resistant if for all PPT adversaries $\mathcal{A}$,

$$\mathsf{Adv}^{\mathrm{CR}}_{\mathsf{H},\mathcal{A}}(\lambda) := \Pr[(\mathsf{M}_0, \mathsf{M}_1) \leftarrow \mathcal{A} : \mathsf{M}_0 \neq \mathsf{M}_1 \wedge \mathsf{H}(\mathsf{M}_0) = \mathsf{H}(\mathsf{M}_1)]$$

is negligible in λ.

2.5 Identity-based Encryption with Equality Test

Syntax. An IBEET scheme Σ consists of the six algorithms (Setup, KeyGen, Enc, Dec, $\mathsf{Trapdoor}$, Test) as follows:

$\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$**:** On input the security parameter 1^λ, it outputs a master public key mpk and a master secret key msk. We assume that mpk contains a description of a plaintext space $\mathcal{M}$ and an identity space $\mathcal{ID}$ that are determined only by the security parameter λ.

$\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id}) \to \mathsf{sk}_{\mathsf{id}}$**:** On input mpk, msk, and an identity $\mathsf{id} \in \mathcal{ID}$, it outputs a secret key $\mathsf{sk}_{\mathsf{id}}$.

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, \mathsf{M}) \to \mathsf{ct}_{\mathsf{id}}$**:** On input mpk, $\mathsf{id} \in \mathcal{ID}$, and a plaintext $\mathsf{M} \in \mathcal{M}$, it outputs a ciphertext $\mathsf{ct}_{\mathsf{id}}$.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}, \mathsf{ct}_{\mathsf{id}}) \to \mathsf{M}$ or $\perp$**:** On input mpk, $\mathsf{sk}_{\mathsf{id}}$, and $\mathsf{ct}_{\mathsf{id}}$, it outputs the decryption result M or $\perp$.

$\mathsf{Trapdoor}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}) \to \mathsf{td}_{\mathsf{id}}$**:** On input mpk and $\mathsf{sk}_{\mathsf{id}}$, it outputs the trapdoor $\mathsf{td}_{\mathsf{id}}$.

$\mathsf{Test}(\mathsf{mpk}, \mathsf{td}_{\mathsf{id}_0}, \mathsf{ct}_{\mathsf{id}_0}, \mathsf{td}_{\mathsf{id}_1}, \mathsf{ct}_{\mathsf{id}_1}) \to 1$ or 0**:** On input mpk, two trapdoors $\mathsf{td}_{\mathsf{id}_0}$ and $\mathsf{td}_{\mathsf{id}_1}$, and two ciphertexts $\mathsf{ct}_{\mathsf{id}_0}$ and $\mathsf{ct}_{\mathsf{id}_1}$, it outputs 1 or 0.

Correctness. In short, IBEET should satisfy three conditions for the correctness, i.e., (1) $\mathsf{ct}_{\mathsf{id}}$ can be correctly decrypted by $\mathsf{sk}_{\mathsf{id}}$, (2) the Test algorithm outputs 1 if $\mathsf{ct}_{\mathsf{id}_0}$ and $\mathsf{ct}_{\mathsf{id}_1}$ are encryptions of the same plaintext, (3) the Test algorithm outputs 0 if $\mathsf{ct}_{\mathsf{id}_0}$ and $\mathsf{ct}_{\mathsf{id}_1}$ are encryptions of distinct plaintexts.[3] Then, the three conditions are formally defined as follows:

(1) For all $\lambda \in \mathbb{N}$, all $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, all $\mathsf{M} \in \mathcal{M}$, and all $\mathsf{id} \in \mathcal{ID}$, it is required that $\mathsf{M}' = \mathsf{M}$ holds with overwhelming probability, where $\mathsf{ct}_{\mathsf{id}} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, \mathsf{M})$, $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$, and $\mathsf{M}' \leftarrow \mathsf{Dec}(\mathsf{mpk}, \mathsf{ct}_{\mathsf{id}}, \mathsf{sk}_{\mathsf{id}})$.

(2) For all $\lambda \in \mathbb{N}$, all $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, all $\mathsf{M} \in \mathcal{M}$, and all $\mathsf{id}_0, \mathsf{id}_1 \in \mathcal{ID}$, it is required that $1 \leftarrow \mathsf{Test}(\mathsf{mpk}, \mathsf{td}_{\mathsf{id}_0}, \mathsf{ct}_{\mathsf{id}_1}, \mathsf{td}_{\mathsf{id}_1}, \mathsf{ct}_{\mathsf{id}_1})$ holds with overwhelming probability, where $\mathsf{sk}_{\mathsf{id}_i} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id}_i)$, $\mathsf{ct}_{\mathsf{id}_i} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}_i, \mathsf{M})$, and $\mathsf{td}_{\mathsf{id}_i} \leftarrow \mathsf{Trapdoor}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}_i})$ for $i = 0, 1$.

(3) For all $\lambda \in \mathbb{N}$, all $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, all $\mathsf{id}_0, \mathsf{id}_1 \in \mathcal{ID}$, and all PPT adversaries $\mathcal{A}$ that take mpk and msk as input and output two plaintexts M_0 and M_1, it is required that $\mathsf{M}_0 \neq \mathsf{M}_1 \wedge 1 \leftarrow \mathsf{Test}(\mathsf{mpk}, \mathsf{td}_{\mathsf{id}_0}, \mathsf{ct}_{\mathsf{id}_0}, \mathsf{td}_{\mathsf{id}_1}, \mathsf{ct}_{\mathsf{id}_1})$ holds with negligible probability, where $(\mathsf{M}_0, \mathsf{M}_1) \leftarrow \mathcal{A}(\mathsf{mpk}, \mathsf{msk})$, and for $i = 0, 1$, $\mathsf{sk}_{\mathsf{id}_i} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id}_i)$, $\mathsf{ct}_{\mathsf{id}_i} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}_i, \mathsf{M}_i)$, and $\mathsf{td}_{\mathsf{id}_i} \leftarrow \mathsf{Trapdoor}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}_i})$.

Security. Let $\mathsf{ct}^*_{\mathsf{id}^*}$ be the challenge ciphertext. For the security of IBEET, we consider two different types of adversaries by whether one has a trapdoor for the target identity id^* or not.

- Type-I adversary: Adversaries of this type have a trapdoor $\mathsf{td}_{\mathsf{id}^*}$, and can perform the equality test against $\mathsf{ct}^*_{\mathsf{id}^*}$. Hence, we consider one-wayness.

[3] To the best of our knowledge, there are no IBEET schemes that satisfy the condition (3) unconditionally. Thus, PPT adversaries $\mathcal{A}$ appears in the formal condition of (3) by following [AET+22].

- Type-II adversary: Adversaries of this type do not have a trapdoor $\mathsf{td}_{\mathsf{id}^*}$, and cannot perform the equality test against $\mathsf{ct}^*_{\mathsf{id}^*}$. Hence, we consider indistinguishability.

Then, security against adversaries of these types are formally defined as follows.

Definition 2.7 (Adaptive OW-CCA2 Security against Type-I Adversaries)**.** The adaptive OW-CCA2 security against Type-I adversaries of an IBEET scheme Σ is defined by a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ as follows:

Init: $\mathcal{C}$ runs $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and gives mpk to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ is allowed to make the following three types of queries to $\mathcal{C}$:

Key extraction query: Upon $\mathcal{A}$'s query on id, $\mathcal{C}$ runs $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$ and returns $\mathsf{sk}_{\mathsf{id}}$ to $\mathcal{A}$.

Decryption query: Upon $\mathcal{A}$'s query on $(\mathsf{id}, \mathsf{ct}_{\mathsf{id}})$, $\mathcal{C}$ runs $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$ and $\mathsf{M} \leftarrow \mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}, \mathsf{ct}_{\mathsf{id}})$, and returns M to $\mathcal{A}$.

Trapdoor query: Upon $\mathcal{A}$'s query on id, $\mathcal{C}$ runs $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$ and $\mathsf{td}_{\mathsf{id}} \leftarrow \mathsf{Trapdoor}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}})$, and returns $\mathsf{td}_{\mathsf{id}}$ to $\mathcal{C}$.

Challenge query: $\mathcal{A}$ is allowed to make the query only once. Upon $\mathcal{A}$'s query on $\mathsf{id}^* \in \mathcal{ID}$, where id^* was not queried on key extraction queries in Phase 1, $\mathcal{C}$ chooses $\mathsf{M}^* \leftarrow_{\$} \mathcal{M}$ and runs $\mathsf{ct}^*_{\mathsf{id}^*} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}^*, \mathsf{M}^*)$. Finally, $\mathcal{C}$ returns $\mathsf{ct}^*_{\mathsf{id}^*}$ to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ is allowed to make the key extraction queries, decryption queries, and trapdoor queries as in Phase 1 with the following exceptions:

Key extraction query: Upon $\mathcal{A}$'s query on $\mathsf{id} \in \mathcal{ID}$, $\mathsf{id} = \mathsf{id}^*$ does not hold.

Decryption query: Upon $\mathcal{A}$'s query on $(\mathsf{id}, \mathsf{ct}_{\mathsf{id}})$, $(\mathsf{id}, \mathsf{ct}_{\mathsf{id}}) = (\mathsf{id}^*, \mathsf{ct}^*_{\mathsf{id}^*})$ does not hold.

Guess: $\mathcal{A}$ outputs $\widehat{\mathsf{M}}$ as a guess of M^*.

The adversary $\mathcal{A}$ wins in the above game if $\widehat{\mathsf{M}} = \mathsf{M}^*$ and the advantage is defined to

$$\mathsf{Adv}^{\mathsf{OW\text{-}CCA2}}_{\Sigma,\mathcal{A}}(\lambda) := \left| \Pr\left[\widehat{\mathsf{M}} = \mathsf{M}^*\right] - \frac{1}{|\mathcal{M}|} \right|.$$

If $\mathsf{Adv}^{\mathsf{OW\text{-}CCA2}}_{\Sigma,\mathcal{A}}(\lambda)$ is negligible in the security parameter λ for all PPT adversaries $\mathcal{A}$, an IBEET scheme Σ is said to satisfy adaptive OW-CCA2 security against Type-I adversaries.

Definition 2.8 (Adaptive IND-CCA2 Security against Type-II Adversaries)**.** The adaptive IND-CCA2 security against Type-II adversaries of an IBEET scheme Σ is defined by a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ as follows:

Init: $\mathcal{C}$ runs $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and gives mpk to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ is allowed to make the following three types of queries to $\mathcal{C}$:

Key extraction query: $\mathcal{A}$ is allowed to make the query on $\mathsf{id} \in \mathcal{ID}$ to $\mathcal{C}$. Upon the query, $\mathcal{C}$ runs $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$ and returns $\mathsf{sk}_{\mathsf{id}}$ to $\mathcal{A}$.

Decryption query: $\mathcal{A}$ is allowed to make the query on $(\mathsf{id}, \mathsf{ct}_{\mathsf{id}})$ to $\mathcal{C}$. Upon the query, $\mathcal{C}$ runs $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$ and $\mathsf{M} \leftarrow \mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}, \mathsf{ct}_{\mathsf{id}})$, and returns M to $\mathcal{A}$.

Trapdoor query: $\mathcal{A}$ is allowed to make the query on $\mathsf{id} \in \mathcal{ID}$ to $\mathcal{C}$. Upon the query, $\mathcal{C}$ runs $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$ and $\mathsf{td}_{\mathsf{id}} \leftarrow \mathsf{Trapdoor}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}})$, and returns $\mathsf{td}_{\mathsf{id}}$ to $\mathcal{C}$.

Challenge query: $\mathcal{A}$ is allowed to make the query only once. Upon $\mathcal{A}$'s query on $(\mathsf{id}^*, \mathsf{M}_0^*, \mathsf{M}_1^*) \in \mathcal{ID} \times \mathcal{M}^2$, where M_0^* and M_1^* have the same length and id^* was not queried on key extraction queries and trapdoor queries in Phase 1. Then, $\mathcal{C}$ flips a coin $\mathsf{coin} \leftarrow_{\$} \{0,1\}$ and runs $\mathsf{ct}_{\mathsf{id}^*}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}^*, \mathsf{M}_{\mathsf{coin}}^*)$. Finally, $\mathcal{C}$ returns $\mathsf{ct}_{\mathsf{id}^*}^*$ to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ is allowed to make the key extraction queries, decryption queries, and trapdoor queries as in Phase 1 with the following exceptions:

Key extraction query: Upon $\mathcal{A}$'s query on $\mathsf{id} \in \mathcal{ID}$, $\mathsf{id} = \mathsf{id}^*$ does not hold.

Decryption query: Upon $\mathcal{A}$'s query on $(\mathsf{id}, \mathsf{ct}_{\mathsf{id}})$, $(\mathsf{id}, \mathsf{ct}_{\mathsf{id}}) = (\mathsf{id}^*, \mathsf{ct}_{\mathsf{id}^*}^*)$ does not hold.

Trapdoor query: Upon $\mathcal{A}$'s query on $\mathsf{id} \in \mathcal{ID}$, $\mathsf{id} = \mathsf{id}^*$ does not hold.

Guess: $\mathcal{A}$ outputs $\widehat{\mathsf{coin}}$ as a guess of coin.

The adversary $\mathcal{A}$ wins in the above game if $\widehat{\mathsf{coin}} = \mathsf{coin}$ and the advantage is defined to

$$\mathsf{Adv}_{\Sigma,\mathcal{A}}^{\text{IND-CCA2}}(\lambda) := \left| \Pr\left[\widehat{\mathsf{coin}} = \mathsf{coin}\right] - \frac{1}{2} \right|.$$

If $\mathsf{Adv}_{\Sigma,\mathcal{A}}^{\text{IND-CCA2}}(\lambda)$ is negligible in the security parameter λ for all PPT adversaries $\mathcal{A}$, an IBEET scheme Σ is said to satisfy adaptive IND-CCA2 security against Type-II adversaries.

3 Construction

In this section, we give our semi-generic construction of IBEET from ABB-type IBE. At first, we define the ABB-type IBE in Section 3.1. Then, we show our semi-generic construction in Section 3.2. In Section 3.3, we prove the correctness of our construction.

3.1 ABB-type Identity-based Encryption

At first, we briefly recall a multi-bit variant of the Agrawal-Boneh-Boyen selectively secure IBE scheme [ABB10a], where the plaintext is an ℓ-bit binary string. The IBE scheme has a master public key $(\mathbf{A}, \mathbf{B}, \mathbf{U}) \in (\mathbb{Z}_q^{n \times m})^2 \times \mathbb{Z}_q^{n \times \ell}$ and master secret key $\mathbf{A}_{\tau_0}^{-1}$. A ciphertext for id consists of three vectors $\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2$ such that

$$\begin{aligned} \mathbf{c}_0^\top &= \mathbf{s}^\top \mathbf{U} + \mathbf{e}_0^\top + \mathsf{M} \cdot \lceil q/2 \rceil \in \mathbb{Z}_q^\ell, \\ \mathbf{c}_1^\top &= \mathbf{s}^\top \mathbf{A} + \mathbf{e}_1^\top \in \mathbb{Z}_q^m, \qquad \mathbf{c}_2^\top = \mathbf{s}^\top [\mathbf{B} + \mathsf{FRD}(\mathsf{id})\mathbf{G}] + \mathbf{e}_2^\top \in \mathbb{Z}_q^m, \end{aligned}$$

where $\mathbf{s}$ is a uniformly random vector and $\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2$ are short vectors, e.g., sampled according to discrete Gaussian vectors. A secret key for id is $[\mathbf{A} \| \mathbf{B} + \mathsf{FRD}(\mathsf{id})\mathbf{G}]_\tau^{-1}(\mathbf{U})$ and decryption succeeds by using the relation

$$\mathbf{c}_0^\top - [\mathbf{c}_1^\top \| \mathbf{c}_2^\top] \cdot [\mathbf{A} \| \mathbf{B} + \mathsf{FRD}(\mathsf{id})\mathbf{G}]_\tau^{-1}(\mathbf{U}) = \mathsf{M} \cdot \lceil q/2 \rceil + \mathsf{noise}.$$

Several improved variants which we call ABB-type IBE have been proposed to achieve adaptive security. To capture ABB-type IBE, we use the following auxiliary algorithm:

- $\mathsf{PubEval}(\{\mathbf{B}_i\}_{i\in[u]}, \mathsf{id}) \to \mathbf{B}_{\mathsf{id}}$: On input matrices $\{\mathbf{B}_i\}_{i\in[u]}$ and an identity $\mathsf{id} \in \mathcal{ID}$, it outputs $\mathbf{B}_{\mathsf{id}} \in \mathbb{Z}_q^{n\times m}$.

Intuitively, Agrawal et al.'s selectively secure IBE scheme uses a matrix $\mathbf{B}$ in a master public key to compute a matrix $\mathbf{B}_{\mathsf{id}} = \mathbf{B} + \mathsf{FRD}(\mathsf{id})\mathbf{G}$ that is associated with both ciphertext and secret key. To achieve adaptive security, we use u matrices $\{\mathbf{B}_i\}_{i\in[u]}$ in a master public key and compute a matrix $\mathbf{B}_{\mathsf{id}}$ by using the $\mathsf{PubEval}$ algorithm. Although the first (Q-bounded) adaptively secure IBE scheme of Agrawal et al. [ABB10a] uses $u = O(\lambda)$ matrices, there are a series of works to reduce u. Yamada's adaptively secure scheme [Yam17] that is purely secure under the LWE assumption uses $u = O(\log^3 \lambda)$ matrices, while JKN's scheme [JKN21] utilizes a near collision resistant hash function and further reduces u to be $O(\log \lambda)$.

Then, we formally define ABB-type IBE as follows.

$\mathsf{IBE.Setup}(1^\lambda) \to (\mathsf{IBE.mpk}, \mathsf{IBE.msk})$**:** On input the security parameter 1^λ, it chooses parameters n, m, q, τ_0, τ_1, α, α', ℓ, runs $(\mathbf{A}, \mathbf{A}_{\tau_0}^{-1}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$, and chooses random matrices $\{\mathbf{B}_i\}_{i\in[u]} \leftarrow_{\$} (\mathbb{Z}_q^{n\times m})^u$ and $\mathbf{U} \leftarrow_{\$} \mathbb{Z}_q^{n\times \ell}$. Finally, it outputs $\mathsf{IBE.mpk} := (\mathbf{A}, \{\mathbf{B}_i\}_{i\in[u]}, \mathbf{U})$ and $\mathsf{IBE.msk} := \mathbf{A}_{\tau_0}^{-1}$.

$\mathsf{IBE.Enc}(\mathsf{IBE.mpk}, \mathsf{id}, \mathsf{M}) \to \mathsf{IBE.ct}_{\mathsf{id}}$**:** Parse $\mathsf{IBE.mpk} = (\mathbf{A}, \{\mathbf{B}_i\}_{i\in[u]}, \mathbf{U})$. It samples $\mathbf{s} \leftarrow_{\$} \mathbb{Z}_q^n$, $\mathbf{e}_0 \leftarrow D_{\mathbb{Z}^\ell, \alpha q}$ and $\mathbf{e}_1, \mathbf{e}_2 \leftarrow D_{\mathbb{Z}^m, \alpha' q}$, runs $\mathbf{B}_{\mathsf{id}} \leftarrow \mathsf{PubEval}(\{\mathbf{B}_i\}_{i\in[u]}, \mathsf{id})$, and sets

$$\mathbf{c}_0^\top = \mathbf{s}^\top \mathbf{U} + \mathbf{e}_0^\top + \mathsf{M} \cdot \lceil q/2 \rceil \in \mathbb{Z}_q^\ell,$$
$$\mathbf{c}_1^\top = \mathbf{s}^\top \mathbf{A} + \mathbf{e}_1^\top \in \mathbb{Z}_q^m, \qquad \mathbf{c}_2^\top = \mathbf{s}^\top \mathbf{B}_{\mathsf{id}} + \mathbf{e}_2^\top \in \mathbb{Z}_q^m.$$

Finally, it outputs $\mathsf{IBE.ct}_{\mathsf{id}} = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2)$.

$\mathsf{IBE.KeyGen}(\mathsf{IBE.mpk}, \mathsf{IBE.msk}, \mathsf{id}) \to \mathsf{IBE.sk}_{\mathsf{id}}$**:** Parse $\mathsf{IBE.mpk} = (\mathbf{A}, \{\mathbf{B}_i\}_{i\in[u]}, \mathbf{U})$ and $\mathsf{IBE.msk} = \mathbf{A}_{\tau_0}^{-1}$. For an identity $\mathsf{id} \in \mathcal{ID}$, it runs $\mathbf{B}_{\mathsf{id}} \leftarrow \mathsf{PubEval}(\{\mathbf{B}_i\}_{i\in[u]}, \mathsf{id})$, obtains trapdoor $[\mathbf{A}\|\mathbf{B}_{\mathsf{id}}]_{\tau_1}^{-1}$ by using the trapdoor $\mathbf{A}_{\tau_0}^{-1}$ and Items 1 and 4 of Lemma 2.3, and outputs $\mathsf{sk}_{\mathsf{id}} := [\mathbf{A}\|\mathbf{B}_{\mathsf{id}}]_{\tau_1}^{-1}$.

$\mathsf{IBE.Dec}(\mathsf{IBE.mpk}, \mathsf{IBE.sk}_{\mathsf{id}}, \mathsf{IBE.ct}_{\mathsf{id}}) \to \mathsf{M}$ or $\perp$**:** Parse $\mathsf{IBE.mpk} = (\mathbf{A}, \{\mathbf{B}_i\}_{i\in[u]}, \mathbf{U})$, $\mathsf{IBE.sk}_{\mathsf{id}} = [\mathbf{A}\|\mathbf{B}_{\mathsf{id}}]_{\tau_1}^{-1}$, and $\mathsf{IBE.ct}_{\mathsf{id}} = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2)$. It samples $\mathbf{E} \leftarrow [\mathbf{A}\|\mathbf{B}_{\mathsf{id}}]_{\tau_1}^{-1}(\mathbf{U})$, computes $\mathbf{m}^\top = \mathbf{c}_0^\top - [\mathbf{c}_1^\top \| \mathbf{c}_2^\top]\mathbf{E} \in \mathbb{Z}_q^\ell$, and sets i-th bit of ℓ bit string M as 1 if $|\mathbf{m}_i - \lceil q/2 \rceil| < \lceil q/4 \rceil$ and 0 otherwise. Finally, it outputs M.

Remark 1. The definition of the $\mathsf{PubEval}$ algorithm and the role of $\mathbf{B}_{\mathsf{id}}$ are slightly different from those of Yamada [Yam17] and Jager et al. [JKN21]. In their IBE schemes, there are $u+1$ matrices $(\mathbf{B}_0, \{\mathbf{B}_i\}_{i\in[u]})$ in $\mathsf{IBE.mpk}$. The $\mathsf{PubEval}$ algorithm takes $\{\mathbf{B}_i\}_{i\in[u]}$ as input and outputs $\mathbf{B}_{\mathsf{id}}$. Then, not $\mathbf{B}_{\mathsf{id}}$ itself but $\mathbf{B}_0 + \mathbf{B}_{\mathsf{id}}$ is associated with $\mathbf{c}_2$ and $\mathsf{IBE.sk}_{\mathsf{id}}$. Their definition is effective to prove adaptive security of their schemes. However, we do not use their proof techniques directly in this paper. Thus, our simplified definition makes the notation simpler.

Remark 2. To be precise, $\mathsf{IBE.sk}_{\mathsf{id}}$ of [Yam17, JKN21] is not $\mathsf{sk}_{\mathsf{id}} = [\mathbf{A}\|\mathbf{B}_{\mathsf{id}}]_{\tau_1}^{-1}$ but $\mathbf{E}$ that is created during $\mathsf{IBE.Dec}$ in the above description. In this paper, we use $\mathsf{sk}_{\mathsf{id}} = [\mathbf{A}\|\mathbf{B}_{\mathsf{id}}]_{\tau_1}^{-1}$ since it makes the discussion of our generic construction simpler. We note that the modification does not violate the security proofs of [Yam17, JKN21] since we apply Item 4 of Lemma 2.3 during $\mathsf{IBE.KeyGen}$.

3.2 Constructions of IBEET schemes from ABB-type IBE

We use ABB-type IBE to construct a lattice-based IBEET scheme. In addition to $\mathsf{IBE.mpk} = (\mathbf{A}, \{\mathbf{B}_i\}_{i\in[u]}, \mathbf{U})$, mpk has two random matrices $\mathbf{C}_1, \mathbf{C}_2 \in \mathbb{Z}_q^{n\times m}$. Before presenting our scheme, we introduce two auxiliary algorithms.

- $\widehat{\mathsf{Enc}}(\mathsf{mpk}, (\mathsf{id}, b, \mathsf{verk}), \mathsf{M}) \to \mathsf{ct}_{\mathsf{id},b}$: It runs $\mathsf{IBE.Enc}(\mathsf{IBE.mpk}, \mathsf{id}, \mathsf{M})$ to compute $\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2$, samples $\mathbf{R} \leftarrow_{\$} \{-1, 1\}^{m\times 2m}$ and computes

$$\mathbf{c}_3^\top = \mathbf{s}^\top[\mathbf{C}_1 + b\mathbf{G}\|\mathbf{C}_2 + \mathsf{FRD}(\mathsf{verk})\mathbf{G}] + \mathbf{e}_1^\top\mathbf{R} \in \mathbb{Z}_q^{2m},$$

  where $\mathbf{s}, \mathbf{e}_1$ are sampled during $\mathsf{IBE.Enc}$ and $b \in \{0, 1\}$. Finally, it outputs $\mathsf{ct}_{\mathsf{id},b} := (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$.

- $\widehat{\mathsf{Dec}}(\mathsf{mpk}, \mathbf{E}_b, \mathsf{ct}_{\mathsf{id},b}) \to \mathsf{M}'$: It computes $\mathbf{m}^\top = \mathbf{c}_0^\top - [\mathbf{c}_1^\top\|\mathbf{c}_2^\top\|\mathbf{c}_3^\top]\mathbf{E}_b$ and recovers M' from $\mathbf{m}$ in the same way as $\mathsf{IBE.Dec}$.

Then, we show our IBEET scheme.

$\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$**:** It runs $(\mathsf{IBE.mpk}, \mathsf{IBE.msk}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$, sample $\mathbf{C}_1, \mathbf{C}_2 \leftarrow_{\$} \mathbb{Z}_q^{n\times m}$, selects a OTS scheme $\Gamma = (\mathsf{Sig.Setup}, \mathsf{Sig.Sign}, \mathsf{Sig.Vrfy})$ and a hash function H, and outputs $\mathsf{mpk} := (\mathsf{IBE.mpk}, \mathbf{C}_1, \mathbf{C}_2, \Gamma, \mathsf{H})$ and $\mathsf{msk} := \mathsf{IBE.msk}$.

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, \mathsf{M}) \to \mathsf{ct}_{\mathsf{id}}$**:** Parse $\mathsf{mpk} = (\mathsf{IBE.mpk}, \mathbf{C}_1, \mathbf{C}_2, \Gamma, \mathsf{H})$. It runs

- $(\mathsf{verk}, \mathsf{sigk}) \leftarrow \mathsf{Sig.Setup}(1^\lambda)$,
- $\mathsf{ct}_{\mathsf{id},0} \leftarrow \widehat{\mathsf{Enc}}(\mathsf{mpk}, (\mathsf{id}, 0, \mathsf{verk}), \mathsf{M})$,
- $\mathsf{ct}_{\mathsf{id},1} \leftarrow \widehat{\mathsf{Enc}}(\mathsf{mpk}, (\mathsf{id}, 1, \mathsf{verk}), \mathsf{H}(\mathsf{M}))$,
- $\sigma \leftarrow \mathsf{Sig.Sign}(\mathsf{sigk}, [\mathsf{ct}_{\mathsf{id},0}\|\mathsf{ct}_{\mathsf{id},1}])$.

Output $\mathsf{ct}_{\mathsf{id}} := (\mathsf{verk}, \mathsf{ct}_{\mathsf{id},0}, \mathsf{ct}_{\mathsf{id},1}, \sigma)$.

$\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id}) \to \mathsf{sk}_{\mathsf{id}}$**:** Parse $\mathsf{mpk} = (\mathsf{IBE.mpk}, \mathbf{C}_1, \mathbf{C}_2, \Gamma, \mathsf{H})$ and $\mathsf{msk} = \mathsf{IBE.msk}$. It runs $\mathsf{IBE.sk}_{\mathsf{id}} = [\mathbf{A}\|\mathbf{B}_{\mathsf{id}}]_{\tau_1}^{-1} \leftarrow \mathsf{IBE.KeyGen}(\mathsf{IBE.mpk}, \mathsf{IBE.msk}, \mathsf{id})$ and outputs $\mathsf{sk}_{\mathsf{id}} := \mathsf{IBE.sk}_{\mathsf{id}}$.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}, \mathsf{ct}_{\mathsf{id}}) \to \mathsf{M}$ or $\perp$**:** Parse $\mathsf{mpk} = (\mathsf{IBE.mpk}, \mathbf{C}_1, \mathbf{C}_2, \Gamma, \mathsf{H})$, $\mathsf{ct}_{\mathsf{id}} = (\mathsf{verk}, \mathbf{c}_{\mathsf{id},0}, \mathbf{c}_{\mathsf{id},1}, \sigma)$, and $\mathsf{sk}_{\mathsf{id}} = [\mathbf{A}\|\mathbf{B}_{\mathsf{id}}]_{\tau_1}^{-1}$. If $0 \leftarrow \mathsf{Sig.Vrfy}(\mathsf{verk}, [\mathbf{c}_{\mathsf{id},0}\|\mathbf{c}_{\mathsf{id},1}], \sigma)$, it outputs $\perp$. Otherwise, it computes $\mathbf{E}_b \leftarrow [\mathbf{A}\|\mathbf{B}_{\mathsf{id}}\|\mathbf{C}_1 + b\mathbf{G}\|\mathbf{C}_2 + \mathsf{FRD}(\mathsf{verk})\mathbf{G}]_{\tau_1}^{-1}(\mathbf{U})$ for $b \in \{0, 1\}$ from $[\mathbf{A}\|\mathbf{B}_{\mathsf{id}}]_{\tau_1}^{-1}$ by using Item 2 of Lemma 2.3. It runs $\mathsf{M} \leftarrow \widehat{\mathsf{Dec}}(\mathsf{mpk}, \mathbf{E}_0, \mathsf{ct}_{\mathsf{id},0})$ and $h \leftarrow \widehat{\mathsf{Dec}}(\mathsf{mpk}, \mathbf{E}_1, \mathsf{ct}_{\mathsf{id},1})$. It outputs M if $\mathsf{H}(\mathsf{M}) = h$ and $\perp$ otherwise.

$\mathsf{Trapdoor}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}) \to \mathsf{td}_{\mathsf{id}}$**:** Parse $\mathsf{mpk} = (\mathsf{IBE.mpk}, \mathbf{C}_1, \mathbf{C}_2, \Gamma, \mathsf{H})$ and $\mathsf{sk}_{\mathsf{id}} = [\mathbf{A}\|\mathbf{B}_{\mathsf{id}}]_{\tau_1}^{-1}$. It computes $\mathsf{td}_{\mathsf{id}} := \mathbf{E} \leftarrow [\mathbf{A}\|\mathbf{B}_{\mathsf{id}}\|\mathbf{C}_1 + \mathbf{G}]_{\tau_1}^{-1}(\mathbf{U})$ from $[\mathbf{A}\|\mathbf{B}_{\mathsf{id}}]_{\tau_1}^{-1}$ by using Item 2 of Lemma 2.3. It outputs $\mathsf{td}_{\mathsf{id}}$.

$\mathsf{Test}(\mathsf{mpk}, \mathsf{td}_{\mathsf{id}}, \mathsf{ct}_{\mathsf{id}}, \mathsf{td}_{\mathsf{id}'}, \mathsf{ct}_{\mathsf{id}'}) \to 1$ or 0**:** Parse $\mathsf{td}_{\mathsf{id}} = \mathbf{E}_{\mathsf{id}} \in \mathbb{Z}_q^{3m\times\ell}$, $\mathsf{td}_{\mathsf{id}'} = \mathbf{E}_{\mathsf{id}'} \in \mathbb{Z}_q^{3m\times\ell}$, $\mathsf{ct}_{\mathsf{id}} = (\mathsf{verk}, \mathsf{ct}_{\mathsf{id},0}, \mathsf{ct}_{\mathsf{id},1}, \sigma)$, and $\mathsf{ct}_{\mathsf{id}'} = (\mathsf{verk}', \mathsf{ct}_{\mathsf{id}',0}, \mathsf{ct}_{\mathsf{id}',1}, \sigma')$. If $0 \leftarrow \mathsf{Sig.Vrfy}(\mathsf{verk}, [\mathsf{ct}_{\mathsf{id},0}\|\mathsf{ct}_{\mathsf{id},1}], \sigma) \vee 0 \leftarrow \mathsf{Sig.Vrfy}(\mathsf{verk}', [\mathsf{ct}_{\mathsf{id}',0}\|\mathsf{ct}_{\mathsf{id}',1}], \sigma')$, it outputs 0. Otherwise, it runs $h \leftarrow \widehat{\mathsf{Dec}}(\mathsf{mpk}, [\mathbf{E}_{\mathsf{id}}^\top\|\mathbf{O}_{\ell,m}]^\top, \mathsf{ct}_{\mathsf{id},1})$ and $h' \leftarrow \widehat{\mathsf{Dec}}(\mathsf{mpk}, [\mathbf{E}_{\mathsf{id}'}^\top\|\mathbf{O}_{\ell,m}]^\top, \mathsf{ct}_{\mathsf{id}',1})$, where $\mathbf{O}_{\ell,m}$ is an $\ell\times m$ zero matrix. It outputs 1 if $h = h'$ and 0 otherwise.

3.3 Correctness

First, we show that a ciphertext encrypted by $\widehat{\mathsf{Enc}}$ is correctly decrypted by $\widehat{\mathsf{Dec}}$ with overwhelming probability. When we run $\mathsf{ct}_{\mathsf{id},b} = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3) \leftarrow \widehat{\mathsf{Enc}}(\mathsf{mpk}, (\mathsf{id}, b, \mathsf{verk}), \mathsf{M})$ and $\widehat{\mathsf{Dec}}(\mathsf{mpk}, \mathbf{E}_b, \mathsf{ct}_{\mathsf{id},b})$, we have

$$\mathbf{m}^\top = \mathbf{c}_0^\top - [\mathbf{c}_1^\top \| \mathbf{c}_2^\top \| \mathbf{c}_3^\top]\mathbf{E}_b = \mathsf{M} \cdot \lceil q/2 \rceil + \underbrace{\mathbf{e}_0^\top - [\mathbf{e}_1^\top \| \mathbf{e}_2^\top \| \mathbf{e}_1^\top \mathbf{R}]\mathbf{E}_b}_{\text{error term}} \in \mathbb{Z}_q^\ell.$$

Lemma 3.1. Assuming $\alpha' > \alpha$, the error term is bounded by $\alpha q\sqrt{\ell} + O(\alpha' q \tau_1 m^{3/2})$ with overwhelming probability.

Proof. Let $\mathbf{e}_{b,i} = [\mathbf{e}_{b,i,1} \| \mathbf{e}_{b,i,2}]$ and $e_{0,i}$ denote the i-th column of $\mathbf{E}_b$ and the i-th element of $\mathbf{e}_0$, respectively, where $\mathbf{e}_{b,i,1} \in \mathbb{Z}_q^{2m}, \mathbf{e}_{b,i,2} \in \mathbb{Z}_q^{2m}$. Then, the i-th element of the error term is bounded as follows with overwhelming probability.

$$\begin{aligned}
|e_{0,i} - [\mathbf{e}_1^\top \| \mathbf{e}_2^\top \| \mathbf{e}_1^\top \mathbf{R}]\mathbf{e}_{b,i}| &\le |e_{0,i}| + \|[\mathbf{e}_1 \| \mathbf{e}_2]\| \cdot \|\mathbf{e}_{b,i,1}\| + \|\mathbf{R}^\top \mathbf{e}_1\| \cdot \|\mathbf{e}_{b,i,2}\| \\
&\le \alpha q\sqrt{\ell} + 2\alpha' q \tau_1 m + O(\alpha' q \tau_1 m^{3/2}) \\
&\le \alpha q\sqrt{\ell} + O(\alpha' q \tau_1 m^{3/2})
\end{aligned}$$

The first inequality above follows from Cauchy-Schwartz and second inequality follows from Lemma 2.1 and 2.2, and Lemma 2.3. □

Parameter selection. To satisfy the correctness conditions (1)–(3) and the security proof works for our IBEET scheme, it is required that

- each element of the error term is less than $q/5$ with overwhelming probability (i.e., $q > \Omega(\alpha q\sqrt{\ell} + \alpha' q \tau_1 m^{3/2})$),
- that $\mathsf{TrapGen}$ can operate (i.e., $m \ge 6n\lceil \log q \rceil$),
- that the leftover hash lemma (Lemma 2.4) can be applied in the security proof (i.e., $m > (n+1)\log q + \omega(\log n)$),
- that τ_1 is sufficiently large compared with $\tau_0 = \omega(\sqrt{n \log q \log n})$ so that we can apply Items 4 and 3 of Lemma 2.3 in the real scheme and security proof, respectively (i.e., $\tau_1 > \tau_0 \cdot \omega(\sqrt{m})$, $\tau_1 > m \cdot \|\mathbf{R}\| \cdot \omega(\sqrt{\log m})$),
- that ℓ is sufficiently large so that a hash function H satisfies one-wayness and collision resistance (i.e., $\ell = \omega(n)$).

For ABB-type IBE to be correct, there are other restrictions. For example, although we omit the detail, Yamada's IBE scheme [Yam17] additionally requires that

- $\tau_1 > m \cdot (1+\delta) \cdot \omega(\sqrt{m \log m})$,[4]
- $\alpha'/2\alpha > \sqrt{2} \cdot m(1+\delta)$ and $\alpha q > \omega(\sqrt{\log m})$,
- $\alpha q > 2\sqrt{2n}$,

[4]The condition is slightly worse than Yamada's original IBE scheme since we have to apply Item 4 of Lemma 2.3.

where $\delta = m^3 \cdot O(\log^2 \lambda) \cdot (O(\lambda) + 1)$. Therefore, we can set the following parameter if we use Yamada's scheme as the underlying ABB-type IBE:

$$m = O(n \log q), \qquad q = n^{9/2} \cdot \delta^2 \cdot \omega(\log^{9/2} n), \qquad \tau_1 = m^{3/2} \cdot \delta \cdot \omega(\sqrt{\log m})$$
$$\alpha q = 3\sqrt{n}, \qquad \alpha' q = 5\sqrt{n} \cdot m \cdot \delta, \qquad \ell = \Theta(n).$$

Theorem 3.1. Our IBEET scheme Σ satisfies correctness if the underlying IBE scheme Π and the OTS scheme Γ satisfy correctness, and the hash function H satisfies collision resistance.

Proof. We prove the three conditions (1)–(3) one by one.

We can prove the condition (1) by using Lemma 3.1 and the underlying OTS scheme Γ. For all $\lambda \in \mathbb{N}$, all $(\mathsf{IBE.mpk}, \mathsf{IBE.msk}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$ and Γ, all $\mathsf{M} \in \mathcal{M}$, and all $\mathsf{id} \in \mathcal{ID}$, it is required that

$$\mathsf{Sig.Vrfy}(\mathsf{verk}, [\mathsf{ct}_{\mathsf{id},0} \| \mathsf{ct}_{\mathsf{id},1}], \sigma) \to 1 \wedge \mathsf{M}' = \mathsf{M} \wedge h = \mathsf{H}(\mathsf{M})$$

holds with overwhelming probability, where

- $(\mathsf{verk}, \mathsf{sigk}) \leftarrow \mathsf{Sig.Setup}(1^\lambda)$,
- $\mathsf{ct}_{\mathsf{id},0} \leftarrow \widehat{\mathsf{Enc}}(\mathsf{mpk}, (\mathsf{id}, 0, \mathsf{verk}), \mathsf{M})$,
- $\mathsf{ct}_{\mathsf{id},1} \leftarrow \widehat{\mathsf{Enc}}(\mathsf{mpk}, (\mathsf{id}, 1, \mathsf{verk}), \mathsf{H}(\mathsf{M}))$,
- $\sigma \leftarrow \mathsf{Sig.Sign}(\mathsf{sigk}, [\mathsf{ct}_{\mathsf{id},0} \| \mathsf{ct}_{\mathsf{id},1}])$,
- $[\mathbf{A} \| \mathbf{B}_{\mathsf{id}}]_{\tau_1}^{-1} \leftarrow \mathsf{IBE.KeyGen}(\mathsf{IBE.mpk}, \mathsf{IBE.msk}, \mathsf{id})$,
- obtain $\mathbf{E}_0 \leftarrow [\mathbf{A} \| \mathbf{B}_{\mathsf{id}} \| \mathbf{C}_1 \| \mathbf{C}_2 + \mathsf{FRD}(\mathsf{verk})\mathbf{G}]_{\tau_1}^{-1}(\mathbf{U})$ from $[\mathbf{A} \| \mathbf{B}_{\mathsf{id}}]_{\tau_1}^{-1}$,
- obtain $\mathbf{E}_1 \leftarrow [\mathbf{A} \| \mathbf{B}_{\mathsf{id}} \| \mathbf{C}_1 + \mathbf{G} \| \mathbf{C}_2 + \mathsf{FRD}(\mathsf{verk})\mathbf{G}]_{\tau_1}^{-1}(\mathbf{U})$ from $[\mathbf{A} \| \mathbf{B}_{\mathsf{id}}]_{\tau_1}^{-1}$,
- $\mathsf{M}' \leftarrow \widehat{\mathsf{Dec}}(\mathsf{mpk}, \mathbf{E}_0, \mathsf{ct}_{\mathsf{id},0})$,
- $h \leftarrow \widehat{\mathsf{Dec}}(\mathsf{mpk}, \mathbf{E}_1, \mathsf{ct}_{\mathsf{id},1})$.

The correctness of the OTS scheme Γ ensures that $\mathsf{Sig.Vrfy}(\mathsf{verk}, [\mathsf{ct}_{\mathsf{id},0} \| \mathsf{ct}_{\mathsf{id},1}], \sigma) \to 1$ holds with overwhelming probability. Moreover, Lemma 3.1 ensures that $\mathsf{M} = \mathsf{M}' \wedge h = \mathsf{H}(\mathsf{M})$ holds with overwhelming probability. Therefore, the condition (1) holds.

We can prove the condition (2) by using the Lemma 3.1 and the correctness of the underlying OTS scheme Γ. For all $\lambda \in \mathbb{N}$, all $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and Γ, all $\mathsf{M} \in \mathcal{M}$, and all $\mathsf{id}_0, \mathsf{id}_1 \in \mathcal{ID}$, it is required that

$$\left(\wedge_{i \in \{0,1\}} \mathsf{Sig.Vrfy}(\mathsf{verk}_i, [\mathsf{ct}_{\mathsf{id}_i,0} \| \mathsf{ct}_{\mathsf{id}_i,1}], \sigma_i) \to 1\right) \wedge h_0 = h_1$$

holds with overwhelming probability, where for $i \in \{0, 1\}$,

- $(\mathsf{verk}_i, \mathsf{sigk}_i) \leftarrow \mathsf{Sig.Setup}(1^\lambda)$,
- $\mathsf{ct}_{\mathsf{id}_i,0} \leftarrow \widehat{\mathsf{Enc}}(\mathsf{mpk}, (\mathsf{id}_i, 0, \mathsf{verk}_i), \mathsf{M})$,
- $\mathsf{ct}_{\mathsf{id}_i,1} \leftarrow \widehat{\mathsf{Enc}}(\mathsf{mpk}, (\mathsf{id}_i, 1, \mathsf{verk}_i), \mathsf{H}(\mathsf{M}))$,

- $\sigma_i \leftarrow \mathsf{Sig.Sign}(\mathsf{sigk}_i, \left[\mathsf{ct}_{\mathsf{id}_i,0} \| \mathsf{ct}_{\mathsf{id}_i,1}\right])$,
- $\left[\mathbf{A} \| \mathbf{B}_{\mathsf{id}_i}\right]_{\tau_1}^{-1} \leftarrow \mathsf{IBE.KeyGen}(\mathsf{IBE.mpk}, \mathsf{IBE.msk}, \mathsf{id}_i)$,
- obtain $\mathbf{E} \leftarrow \left[\mathbf{A} \| \mathbf{B}_{\mathsf{id}_i} \| \mathbf{C}_1 + \mathbf{G} \| \mathbf{C}_2 + \mathsf{FRD}(\mathsf{verk}_i)\mathbf{G}\right]_{\tau_1}^{-1}(\mathbf{U})$ from $\left[\mathbf{A} \| \mathbf{B}_{\mathsf{id}_i}\right]_{\tau_1}^{-1}$,
- $h_i \leftarrow \widehat{\mathsf{Dec}}(\mathsf{mpk}, \mathbf{E}, \mathsf{ct}_{\mathsf{id}_i,1})$.

The correctness of the OTS scheme Γ ensures that $\mathsf{Sig.Vrfy}(\mathsf{verk}_i, \left[\mathsf{ct}_{\mathsf{id}_i,0} \| \mathsf{ct}_{\mathsf{id}_i,1}\right], \sigma_i) \rightarrow 1$ holds for $i \in \{0,1\}$ with overwhelming probability. Moreover, the Lemma 3.1 ensures that $h_i = \mathsf{H}(\mathsf{M})$ for $i \in \{0,1\}$, i.e., $h_0 = h_1$, holds with overwhelming probability. Therefore, the condition (2) holds.

We can prove the condition (3) by using the Lemma 3.1 and collision resistance of the underlying hash function H. For this purpose, we use an adversary $\mathcal{A}$ for breaking the condition (3) to construct a PPT adversary $\mathcal{B}$ that breaks the collision resistance of H. Here, we say that $\mathcal{A}$ breaks the condition (3) if it holds that $\mathsf{M}_0 \neq \mathsf{M}_1 \wedge \mathsf{Test}(\mathsf{mpk}, \mathsf{td}_{\mathsf{id}_0}, \mathsf{ct}_{\mathsf{id}_0}, \mathsf{td}_{\mathsf{id}_1}, \mathsf{ct}_{\mathsf{id}_1}) \rightarrow 1$, where $(\mathsf{M}_0, \mathsf{M}_1) \leftarrow \mathcal{A}(\mathsf{mpk}, \mathsf{msk})$, and for $i = 0, 1$, $\mathsf{ct}_{\mathsf{id}_i} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}_i, \mathsf{M}_i)$, $\mathsf{sk}_{\mathsf{id}_i} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id}_i)$ and $\mathsf{td}_{\mathsf{id}_i} \leftarrow \mathsf{Trapdoor}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}_i})$. For all $\lambda \in \mathbb{N}$, all $(\mathsf{IBE.mpk}, \mathsf{IBE.msk}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$ and (Γ, H), all PPT adversaries $\mathcal{A}$, all $(\mathsf{id}_0, \mathsf{id}_1) \in \mathcal{ID}^2$, after $\mathcal{A}$ outputs $(\mathsf{M}_0, \mathsf{M}_1)$, $\mathcal{B}$ also outputs the same $(\mathsf{M}_0, \mathsf{M}_1)$. If $\mathcal{A}$ breaks the condition (3), it holds that $\mathsf{M}_0 \neq \mathsf{M}_1 \wedge h_0 = h_1$, where for $i \in \{0,1\}$,

- $(\mathsf{verk}_i, \mathsf{sigk}_i) \leftarrow \mathsf{Sig.Setup}(1^\lambda)$,
- $\mathsf{ct}_{\mathsf{id}_i,1} \leftarrow \widehat{\mathsf{Enc}}(\mathsf{mpk}, (\mathsf{id}_i, 1, \mathsf{verk}_i), \mathsf{H}(\mathsf{M}_i))$,
- $\left[\mathbf{A} \| \mathbf{B}_{\mathsf{id}_i}\right]_{\tau_1}^{-1} \leftarrow \mathsf{IBE.KeyGen}(\mathsf{IBE.mpk}, \mathsf{IBE.msk}, \mathsf{id}_i)$,
- obtain $\mathbf{E} = \left[\mathbf{A} \| \mathbf{B}_{\mathsf{id}_i} \| \mathbf{C}_1 + \mathbf{G} \| \mathbf{C}_2 + \mathsf{FRD}(\mathsf{verk}_i)\mathbf{G}\right]_{\tau_1}^{-1}(\mathbf{U})$ from $\left[\mathbf{A} \| \mathbf{B}_{\mathsf{id}_i}\right]_{\tau_1}^{-1}$,
- $h_i \leftarrow \widehat{\mathsf{Dec}}(\mathsf{mpk}, \mathbf{E}, \mathsf{ct}_{\mathsf{id}_i,1})$.

The Lemma 3.1 ensures that $h_i = \mathsf{H}(\mathsf{M}_i)$ holds for $i \in \{0,1\}$ with overwhelming probability. Therefore, if $\mathcal{A}$ breaks condition (3), it holds that $\mathsf{H}(\mathsf{M}_0) = \mathsf{H}(\mathsf{M}_1)$ and $\mathcal{B}$ breaks the collision resistance of H with overwhelming probability since it holds that $\mathsf{M}_0 \neq \mathsf{M}_1 \wedge \mathsf{H}(\mathsf{M}_0) = \mathsf{H}(\mathsf{M}_1)$. Therefore, the condition (3) holds.

From the above, it is proved that our proposed construction is correct.

□

4 Security

In this section, we discuss the security of our proposed IBEET scheme.

4.1 OW-CCA2 Security against Type-I Adversaries

In this subsection, we prove the following theorem.

Theorem 4.1 (OW-CCA2 Security against Type-I Adversaries)**.** If the underlying IBE scheme Π satisfies adaptive IND-CPA security, OTS scheme Γ satisfies strong unforgeability, and H satisfies one-wayness, then our proposed IBEET scheme Σ satisfies adaptive OW-CCA2 security against Type-I adversaries. In particular, there are PPT algorithms $\mathcal{F}, \mathcal{B}_1$, and $\mathcal{D}$ such that

$$\mathsf{Adv}^{\mathsf{OW\text{-}CCA2}}_{\Sigma,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{OTS}}_{\Gamma,\mathcal{F}}(\lambda) + 2\mathsf{Adv}^{\mathsf{IBE}}_{\Pi,\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{OW}}_{\mathsf{H},\mathcal{D}}(\lambda) + \mathsf{negl}(\lambda).$$

Proof. Let $\mathsf{ct}^*_{\mathsf{id}^*} = (\mathsf{verk}^*, \mathsf{ct}^*_{\mathsf{id}^*,0}, \mathsf{ct}^*_{\mathsf{id}^*,1}, \sigma^*)$ be the challenge ciphertext for the target identity id^*. We prove the theorem via game sequence from $\mathbf{Game}_0$ to $\mathbf{Game}_4$. Let W_i denote an event that $\mathcal{A}$ wins in $\mathbf{Game}_i$ for $i \in \{0, \dots, 4\}$.

$\mathbf{Game}_0$: This game is the same as the original adaptive OW-CCA2 security game in Definition 2.7 between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$.

$\mathbf{Game}_1$: This game is the same as $\mathbf{Game}_0$ except that $\mathcal{C}$ runs $(\mathsf{verk}^*, \mathsf{sigk}^*) \leftarrow \mathsf{Sig.Setup}(1^\lambda)$ that will be used for creating the challenge ciphertext immediately after Init phase instead of running in challenge query and if $\mathcal{A}$ makes the decryption queries on $(\mathsf{id}, \mathsf{ct}_{\mathsf{id}}) = (\mathsf{id}, (\mathsf{verk}, \mathsf{ct}_{\mathsf{id},0}, \mathsf{ct}_{\mathsf{id},1}, \sigma))$ such that

$$\mathsf{verk} = \mathsf{verk}^* \wedge \mathsf{Sig.Vrfy}(\mathsf{verk}, [\mathsf{ct}_{\mathsf{id},0} \| \mathsf{ct}_{\mathsf{id},1}], \sigma) \to 1 \wedge (\mathsf{ct}_{\mathsf{id},0}, \mathsf{ct}_{\mathsf{id},1}, \sigma) \neq (\mathsf{ct}^*_{\mathsf{id}^*,0}, \mathsf{ct}_{\mathsf{id}^*,1}, \sigma^*)$$

then $\mathcal{C}$ aborts the game and returns $\mathsf{M} \leftarrow_{\$} \mathcal{M}$.

$\mathbf{Game}_2$: This game is the same as $\mathbf{Game}_1$ except the way $\mathcal{C}$ generates $(\mathbf{C}_1, \mathbf{C}_2) \in \mathsf{mpk}$ and $\mathsf{ct}^*_{\mathsf{id}^*,0} \in \mathsf{ct}^*_{\mathsf{id}^*}$, where the creation of $\mathsf{ct}^*_{\mathsf{id}^*,1} \in \mathsf{ct}^*_{\mathsf{id}^*}$ is unchanged. In Init phase of $\mathbf{Game}_2$, $\mathcal{C}$ samples $\mathbf{R}^*_1, \mathbf{R}^*_2 \leftarrow_{\$} \mathbb{Z}_q^{m \times m}$ and sets $\mathbf{C}_1 = \mathbf{A}\mathbf{R}^*_1$, $\mathbf{C}_2 = \mathbf{A}\mathbf{R}^*_2 - \mathsf{FRD}(\mathsf{verk}^*)\mathbf{G}$. Upon $\mathcal{A}$'s challenge query on $\mathsf{id}^\star$, $\mathcal{C}$ creates $(\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2)$ in the same way as the real scheme by running $\mathsf{IBE.Enc}(\mathsf{IBE.mpk}, \mathsf{id}^*, \mathsf{M}^*)$ and creates $\mathbf{c}_3$ by computing

$$\begin{aligned}\mathbf{c}_3^\top &= \mathbf{c}_1^\top [\mathbf{R}^*_1 \| \mathbf{R}^*_2] = (\mathbf{s}^\top \mathbf{A} + \mathbf{e}_1^\top)[\mathbf{R}^*_1 \| \mathbf{R}^*_2] \\ &= \mathbf{s}^\top [\mathbf{C}_1 \| \mathbf{C}_2 + \mathsf{FRD}(\mathsf{verk}^*)\mathbf{G}] + \mathbf{e}_1^\top [\mathbf{R}^*_1 \| \mathbf{R}^*_2].\end{aligned}$$

Then, $\mathcal{C}$ sets $\mathsf{ct}_{\mathsf{id}^*,0} = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$.

$\mathbf{Game}_3$: This game is the same as $\mathbf{Game}_2$ except that the way $\mathcal{C}$ generates $\mathbf{E}_0, \mathbf{E}_1$ and $\mathsf{td}_{\mathsf{id}} = \mathbf{E}$ without using msk for answering decryption queries and trapdoor queries, respectively. In $\mathbf{Game}_2$, $\mathcal{C}$ uses $\mathsf{sk}_{\mathsf{id}}$ to generate $\mathbf{E}_0, \mathbf{E}_1$ and $\mathsf{td}_{\mathsf{id}} = \mathbf{E}$, but in $\mathbf{Game}_3$, $\mathcal{C}$ uses $\mathbf{R}^*_1, \mathbf{R}^*_2$ and the property of the gadget matrix (Lemma 2.3, Item 3) to generate $\mathbf{E}_0$, $\mathbf{E}_1$, and $\mathbf{E}$. In particular, $\mathcal{C}$ generates $\mathbf{E}_0, \mathbf{E}_1$ and $\mathsf{td}_{\mathsf{id}} = \mathbf{E}$ in $\mathbf{Game}_3$ as follows.

- Upon $\mathcal{A}$'s decryption query on $(\mathsf{id}, \mathsf{ct}_{\mathsf{id}}) = (\mathsf{id}, (\mathsf{verk}, \mathsf{ct}_{\mathsf{id},0}, \mathsf{ct}_{\mathsf{id},1}, \sigma))$, $\mathcal{C}$ first computes $[\mathbf{A} \| \mathbf{C}_2 + \mathsf{FRD}(\mathsf{verk})\mathbf{G}]^{-1}_{\tau_1} = [\mathbf{A} \| \mathbf{A}\mathbf{R}^*_2 + (\mathsf{FRD}(\mathsf{verk}) - \mathsf{FRD}(\mathsf{verk}^*))\mathbf{G}]^{-1}_{\tau_1}$ by using Item 3 of Lemma 2.3, where $\mathsf{FRD}(\mathsf{verk}) - \mathsf{FRD}(\mathsf{verk}^*)$ is full-rank since $\mathsf{verk} \neq \mathsf{verk}^*$ holds due to the modification in $\mathbf{Game}_1$. Next, $\mathcal{C}$ creates $[\mathbf{A} \| \mathbf{B}_{\mathsf{id}} \| \mathbf{C}_1 \| \mathbf{C}_2 + \mathsf{FRD}(\mathsf{verk})\mathbf{G}]^{-1}_{\tau}$ and $[\mathbf{A} \| \mathbf{B}_{\mathsf{id}} \| \mathbf{C}_1 + \mathbf{G} \| \mathbf{C}_2 + \mathsf{FRD}(\mathsf{verk})\mathbf{G}]^{-1}_{\tau}$ by using Item 2 of Lemma 2.3. Then, $\mathcal{C}$ creates $\mathbf{E}_0, \mathbf{E}_1$ in the same way as the real scheme.
- Upon $\mathcal{A}$'s trapdoor query on id, $\mathcal{C}$ first computes $[\mathbf{A} \| \mathbf{C}_1 + \mathbf{G}]^{-1}_{\tau}$ by using Item 3 of Lemma 2.3. Next, $\mathcal{C}$ creates $[\mathbf{A} \| \mathbf{B}_{\mathsf{id}} \| \mathbf{C}_1 + \mathbf{G}]^{-1}_{\tau}$ by using Item 2 of Lemma 2.3. Then, $\mathcal{C}$ creates $\mathbf{E}$ in the same way as the real scheme.

$\mathbf{Game}_4$: This game is the same as $\mathbf{Game}_3$ except the way $\mathcal{C}$ creates the challenge ciphertext $\mathsf{ct}^*_{\mathsf{id}^*} = (\mathsf{verk}^*, \mathsf{ct}^*_{\mathsf{id}^*,0}, \mathsf{ct}^*_{\mathsf{id}^*,1}, \sigma^*)$. In short, $\mathsf{ct}_{\mathsf{id}^*,0}$ is an encryption of the challenge plaintext M^* in $\mathbf{Game}_3$. In contrast, $\mathsf{ct}_{\mathsf{id}^*,0}$ is a uniformly random element over $\mathbb{Z}_q^{\ell+4m}$ in $\mathbf{Game}_4$. We note that $\mathsf{ct}_{\mathsf{id}^*,1}$ is an encryption $\mathsf{H}(\mathsf{M}^*)$ in both $\mathbf{Game}_3$ and $\mathbf{Game}_4$.

It holds that $\mathbf{Game}_0 \approx_c \mathbf{Game}_1$ from $\mathcal{A}$'s view based on the unforgeability of the OTS scheme Γ, where the standard argument of the CHK transformation [CHK04] is sufficient for the proof. It holds that $\mathbf{Game}_1 \approx \mathbf{Game}_2$ from $\mathcal{A}$'s view due to Lemma 2.4. It holds that $\mathbf{Game}_2 \approx \mathbf{Game}_3$ from $\mathcal{A}$'s view due to the Item 3 of Lemma 2.3. It is computationally infeasible for $\mathcal{A}$ to win in $\mathbf{Game}_4$ based on the one-wayness of the hash function.

All we have to show is $\mathbf{Game}_3 \approx_c \mathbf{Game}_4$ based on the IND-CPA security of the IBE scheme Π. For this purpose, we use $\mathcal{A}$ to construct a PPT adversary $\mathcal{B}_1$ that breaks IND-CPA security of Π. Let $\mathsf{IBE}.\mathcal{C}$ denote a challenger of the IND-CPA security game of Π. After $\mathcal{B}_1$ receives $\mathsf{IBE.mpk}$ and begins the IND-CPA security game with $\mathsf{IBE}.\mathcal{C}$, $\mathcal{B}_1$ runs $(\mathsf{verk}^*, \mathsf{sigk}^*) \leftarrow \mathsf{Sig.Setup}(1^\lambda)$ as we modified in $\mathbf{Game}_1$, samples $\mathbf{R}^*_1, \mathbf{R}^*_2 \leftarrow_\$ \mathbb{Z}_q^{m\times m}$ and computes $\mathbf{C}_1$, $\mathbf{C}_2$ as we modified in $\mathbf{Game}_2$. Then, $\mathcal{B}_1$ select a OTS scheme Γ and a hash function H, and begins the OW-CCA2 security game with $\mathcal{A}$ by giving $\mathsf{mpk} = (\mathsf{IBE.mpk}, \mathbf{C}_1, \mathbf{C}_2, \Gamma, \mathsf{H})$ to $\mathcal{A}$.

In the Phase 1, $\mathcal{B}_1$ answers decryption queries and trapdoor queries as we modified in $\mathbf{Game}_1$ and $\mathbf{Game}_3$. Upon $\mathcal{A}$'s key extraction query on id, $\mathcal{B}_1$ makes a key extraction query on id to $\mathsf{IBE}.\mathcal{C}$ and receives $\mathsf{IBE.sk}_{\mathsf{id}}$. Then, $\mathcal{B}_1$ sends $\mathsf{sk}_{\mathsf{id}} := \mathsf{IBE.sk}_{\mathsf{id}}$ to $\mathcal{A}$

Upon $\mathcal{A}$'s challenge query on id^*, $\mathcal{B}_1$ chooses $\mathsf{M}^* \leftarrow_\$ \mathcal{M}$ as the challenge plaintext of OW-CCA security game. Then, $\mathcal{B}_1$ makes the challenge query on $(\mathsf{id}^*, \mathsf{M}^*)$ to $\mathsf{IBE}.\mathcal{C}$ and receives $\mathsf{IBE.ct}^*_{\mathsf{id}^*} = (\mathbf{c}^*_0, \mathbf{c}^*_1, \mathbf{c}^*_2)$. $\mathcal{B}_1$ retrieves $\mathbf{R}^*_1, \mathbf{R}^*_2$, computes $(\mathbf{c}^*_3)^\top = (\mathbf{c}^*_1)^\top [\mathbf{R}^*_1 \| \mathbf{R}^*_2]$ as we modified in $\mathbf{Game}_2$, and sets $\mathsf{ct}^*_{\mathsf{id}^*,0} = (\mathbf{c}^*_0, \mathbf{c}^*_1, \mathbf{c}^*_2, \mathbf{c}^*_3)$. $\mathcal{B}_1$ runs $\mathsf{ct}^*_{\mathsf{id}^*,1} \leftarrow \widehat{\mathsf{Enc}}(\mathsf{mpk}, (\mathsf{id}^*, 1, \mathsf{verk}^*), \mathsf{H}(\mathsf{M}^*))$, $\sigma^* \leftarrow \mathsf{Sig.Sign}(\mathsf{sigk}^*, \left[\mathsf{ct}^*_{\mathsf{id}^*,0} \| \mathsf{ct}^*_{\mathsf{id}^*,1}\right])$, and gives $\mathsf{ct}^*_{\mathsf{id}^*} = (\mathsf{verk}^*, \mathsf{ct}_{\mathsf{id}^*,0}, \mathsf{ct}_{\mathsf{id}^*,1}, \sigma^*)$ to $\mathcal{A}$. In the Phase 2, $\mathcal{B}_1$ can answer all three types of queries essentially in the same way as in Phase 1. After $\mathcal{A}$ outputs $\widehat{\mathsf{M}}$ as a guess of M^*, $\mathcal{B}_1$ outputs $\widehat{\mathsf{coin}} = 0$ if $\widehat{\mathsf{M}} = \mathsf{M}^*$ and $\widehat{\mathsf{coin}} = 1$ otherwise as a guess of coin flipped by $\mathsf{IBE}.\mathcal{C}$.

We check that $\mathcal{B}_1$'s behavior in the security game of IBE is correct. $\mathcal{B}_1$ makes key extraction queries on id to $\mathsf{IBE}.\mathcal{C}$ only for answering $\mathcal{A}$'s key extraction queries. Since the security game of IBEET ensures that all id on which $\mathcal{A}$ makes key extraction queries satisfy $\mathsf{id} \neq \mathsf{id}^*$, $\mathcal{B}_1$ does not make key extraction queries on id^* to $\mathsf{IBE}.\mathcal{C}$. Next, we check $\mathcal{B}_1$'s behavior in the security game of IBEET is correct. Due to the modification in $\mathbf{Game}_2$, mpk distributes in the same way as the real scheme. Due to the modifications in $\mathbf{Game}_1$, $\mathbf{Game}_2$, and $\mathbf{Game}_3$, $\mathcal{B}_1$ can answer all $\mathcal{A}$'s queries so that the distributions are statistically close to those of the real scheme. Due to the modification in $\mathbf{Game}_2$, $\mathsf{ct}^*_{\mathsf{id}^*,0}$ is a valid encryption of M^* and a uniformly random element over $\mathbf{Z}_q^{\ell+4m}$ if $\beta = 0$ and 1, respectively, based on the standard leftover hash lemma, e.g., Theorem 8.38 of [Sho05]. In other words, the challenge ciphertext $\mathsf{ct}^*_{\mathsf{id}^*}$ distributes according to $\mathbf{Game}_3$ and $\mathbf{Game}_4$ if $\beta = 0$ and 1, respectively. Therefore, it holds that $\mathbf{Game}_3 \approx_c \mathbf{Game}_4$ as required.

□

4.2 IND-CCA2 Security against Type-II Adversaries

In this subsection, we prove the following theorem.

Theorem 4.2 (IND-CCA2 Security against Type-II Adversaries)**.** If the underlying IBE scheme Π satisfies adaptive IND-CPA security, OTS scheme Γ satisfies strong unforgeability, then our

proposed IBEET scheme Σ satisfies adaptive IND-CCA2 security against Type-II adversaries. In particular, there are PPT algorithms $\mathcal{F}, \mathcal{B}_1$, and $\mathcal{B}_2$ such that

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA2}}_{\Sigma,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{OTS}}_{\Gamma,\mathcal{F}}(\lambda) + 2\mathsf{Adv}^{\mathsf{IBE}}_{\Pi,\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{IBE}}_{\Pi,\mathcal{B}_2}(\lambda) + \mathsf{negl}(\lambda).$$

Proof. Let $\mathsf{ct}^*_{\mathsf{id}^*} = (\mathsf{verk}^*, \mathsf{ct}^*_{\mathsf{id}^*,0}, \mathsf{ct}^*_{\mathsf{id}^*,1}, \sigma^*)$ be the challenge ciphertext for the target identity id^*. We prove the theorem via game sequence from $\mathbf{Game}_0$ to $\mathbf{Game}_7$, where $\mathbf{Game}_0$ to $\mathbf{Game}_4$ are almost the same[5] as in the proof of Theorem 4.1.

$\mathbf{Game}_5$: This game is the same as $\mathbf{Game}_4$ except that $\mathcal{C}$ generates $\mathsf{td}_{\mathsf{id}} = \mathbf{E}$ in the same way as $\mathbf{Game}_3$ by using msk.

$\mathbf{Game}_6$: This game is the same as $\mathbf{Game}_5$ except the way $\mathcal{C}$ generates $\mathbf{C}_1 \in \mathsf{mpk}$ and $\mathsf{ct}^*_{\mathsf{id}^*,1} \in \mathsf{ct}^*_{\mathsf{id}^*}$, where the creations of $\mathbf{C}_2 \in \mathsf{mpk}$ and $\mathsf{ct}^*_{\mathsf{id}^*,0} \in \mathsf{ct}^*_{\mathsf{id}^*}$ are unchanged. In Init phase of $\mathbf{Game}_5$, $\mathcal{C}$ samples $\mathbf{R}^*_1 \leftarrow_{\$} \mathbb{Z}_q^{m\times m}$ and sets $\mathbf{C}_1 = \mathbf{A}\mathbf{R}^*_1$. In contrast, in Init phase of $\mathbf{Game}_6$, $\mathcal{C}$ samples $\mathbf{R}^*_1 \leftarrow_{\$} \mathbb{Z}_q^{m\times m}$ and sets $\mathbf{C}_1 = \mathbf{A}\mathbf{R}^*_1 - \mathbf{G}$. Upon $\mathcal{A}$'s challenge query on id^*, $\mathcal{C}$ runs $(\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2) \leftarrow \mathsf{IBE.Enc}(\mathsf{IBE.mpk}, \mathsf{id}^*, \mathsf{M}^*_{\mathsf{coin}})$ and creates $\mathbf{c}_3$ by computing

$$\begin{aligned}\mathbf{c}_3^\top &= \mathbf{c}_1^\top[\mathbf{R}^*_1\|\mathbf{R}^*_2] = (\mathbf{s}^\top\mathbf{A} + \mathbf{e}_1^\top)[\mathbf{R}^*_1\|\mathbf{R}^*_2] \\ &= \mathbf{s}^\top[\mathbf{C}_1 + \mathbf{G}\|\mathbf{C}_2 + \mathsf{FRD}(\mathsf{verk}^*)\mathbf{G}] + \mathbf{e}_1^\top[\mathbf{R}^*_1\|\mathbf{R}^*_2].\end{aligned}$$

Then, $\mathcal{C}$ sets $\mathsf{ct}_{\mathsf{id}^*,1} = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$.

$\mathbf{Game}_7$: This game is the same as $\mathbf{Game}_6$ except the way $\mathcal{C}$ creates $\mathsf{ct}^*_{\mathsf{id}^*,1} \in \mathsf{ct}^*_{\mathsf{id}^*}$. In short, $\mathsf{ct}^*_{\mathsf{id}^*,1}$ is an encryption of the challenge plaintext $\mathsf{M}^*_{\mathsf{coin}}$ in $\mathbf{Game}_6$. In contrast, $\mathsf{ct}^*_{\mathsf{id}^*,1}$ is a uniformly random element over $\mathbb{Z}_q^{\ell+4m}$ in $\mathbf{Game}_7$.

It holds that $\mathbf{Game}_0 \approx_c \mathbf{Game}_4$ as we proved in the proof of Theorem 4.1. It holds that $\mathbf{Game}_4 \approx \mathbf{Game}_5$ by following the same argument as in $\mathbf{Game}_2 \approx_c \mathbf{Game}_3$. It holds that $\mathbf{Game}_5 \approx \mathbf{Game}_6$ by following the same argument as in $\mathbf{Game}_1 \approx \mathbf{Game}_2$. In $\mathbf{Game}_7$, $\mathcal{A}$'s advantage is exactly zero since both $\mathsf{ct}^*_{\mathsf{id}^*,0}$ and $\mathsf{ct}^*_{\mathsf{id}^*,1}$ are independent of $\mathsf{M}^*_{\mathsf{coin}}$.

All we have to show is $\mathbf{Game}_6 \approx_c \mathbf{Game}_7$ based on the IND-CPA security of the IBE scheme Π. For this purpose, we use $\mathcal{A}$ to construct a PPT adversary $\mathcal{B}_2$ that breaks IND-CPA security of Π, where $\mathcal{B}_2$'s behavior is similar to $\mathcal{B}_1$ in the proof of $\mathbf{Game}_3 \approx_c \mathbf{Game}_4$. After $\mathcal{B}_2$ receives $\mathsf{IBE.mpk}$ and begins the IND-CPA security game with $\mathsf{IBE}.\mathcal{C}$, $\mathcal{B}$ gives $\mathsf{mpk} = (\mathsf{IBE.mpk}, \mathbf{C}_1, \mathbf{C}_2, \Gamma, \mathsf{H})$ to $\mathcal{A}$. $\mathcal{B}_2$ creates mpk in the same way as $\mathcal{B}_1$ except that $\mathbf{C}_1 = \mathbf{A}\mathbf{R}^*_1 - \mathbf{G}$ as we modified in $\mathbf{Game}_6$.

In the Phase 1, $\mathcal{B}_2$ answers $\mathcal{A}$'s key extraction queries and decryption queries in the same way as $\mathcal{B}_1$. Upon $\mathcal{A}$'s trapdoor query on id, $\mathcal{B}_2$ makes a key extraction query on id to $\mathsf{IBE}.\mathcal{C}$ and receives $\mathsf{IBE.sk}_{\mathsf{id}}$. Then, $\mathcal{B}_2$ creates $\mathsf{td}_{\mathsf{id}} = \mathbf{E}$ in the same way as the real scheme and gives it to $\mathcal{A}$.

Upon $\mathcal{A}$'s challenge query on $(\mathsf{id}^*, \mathsf{M}^*_0, \mathsf{M}^*_1)$, $\mathcal{B}_2$ flips a coin $\mathsf{coin} \leftarrow_{\$} \{0,1\}$ and makes the challenge query on $(\mathsf{id}^*, \mathsf{M}^*_{\mathsf{coin}})$ to $\mathsf{IBE}.\mathcal{C}$ and receives $\mathsf{IBE.ct}^*_{\mathsf{id}^*} = (\mathbf{c}^*_0, \mathbf{c}^*_1, \mathbf{c}^*_2)$. $\mathcal{B}_2$ retrieves $\mathbf{R}^*_1, \mathbf{R}^*_2$, computes $(\mathbf{c}^*_3)^\top = (\mathbf{c}^*_1)^\top[\mathbf{R}^*_1\|\mathbf{R}^*_2]$ as we modified in $\mathbf{Game}_2$ and $\mathbf{Game}_6$, and sets $\mathsf{ct}^*_{\mathsf{id}^*,1} = (\mathbf{c}^*_0, \mathbf{c}^*_1, \mathbf{c}^*_2, \mathbf{c}^*_3)$. $\mathcal{B}_2$ creaetes $\mathsf{ct}^*_{\mathsf{id}^*,0} \leftarrow_{\$} \mathbb{Z}_q^{\ell+4m}$, $\sigma^* \leftarrow \mathsf{Sig.Sign}(\mathsf{sigk}^*, \left[\mathsf{ct}^*_{\mathsf{id}^*,0}\|\mathsf{ct}^*_{\mathsf{id}^*,1}\right])$, and gives $\mathsf{ct}^*_{\mathsf{id}^*} = (\mathsf{verk}^*, \mathsf{ct}^*_{\mathsf{id}^*,0}, \mathsf{ct}^*_{\mathsf{id}^*,1}, \sigma^*)$ to $\mathcal{A}$. In the Phase 2, $\mathcal{B}_2$ can answer all three types of queries essentially in the same way as in Phase 1. After $\mathcal{A}$ outputs $\widehat{\mathsf{coin}}$ as a guess of coin, $\mathcal{B}_2$ outputs $\hat{\beta} = 0$ if $\widehat{\mathsf{coin}} = \mathsf{coin}$ and $\hat{\beta} = 1$ otherwise as a guess of β flipped by $\mathsf{IBE}.\mathcal{C}$.

We check that $\mathcal{B}_2$'s behavior in the security game of IBE is correct. $\mathcal{B}$ makes key extraction queries on id to $\mathsf{IBE}.\mathcal{C}$ only for answering $\mathcal{A}$'s key extraction queries and trapdoor queries. Since

[5] In $\mathbf{Game}_1$, if $\mathcal{C}$ aborts the game, it outputs $\widehat{\mathsf{coin}} \leftarrow_{\$} \{0,1\}$.

the security game of IBEET against Type-II adversary ensures that all id on which $\mathcal{A}$ makes key extraction queries or trapdoor queries satisfy $\mathsf{id} \neq \mathsf{id}^*$, $\mathcal{B}_2$ does not make key extraction queries on id^* to $\mathsf{IBE}.\mathcal{C}$. Next, we check $\mathcal{B}_2$'s behavior in the security game of IBEET is correct. Due to the modification in $\mathbf{Game}_2$ and $\mathbf{Game}_6$, mpk distributes in the same way as the real scheme. Due to the modifications in $\mathbf{Game}_1$, $\mathbf{Game}_2$, $\mathbf{Game}_3$, and $\mathbf{Game}_6$, $\mathcal{B}_2$ can answer all $\mathcal{A}$'s queries so that the distributions are statistically close to those of the real scheme. Due to the modification in $\mathbf{Game}_6$, $\mathsf{ct}^*_{\mathsf{id}^*,0}$ is a valid encryption of $\mathsf{M}^*_{\mathsf{coin}}$ and a uniformly random element over $\mathbf{Z}_q^{\ell+4m}$ if $\beta = 0$ and 1, respectively, based on the standard leftover hash lemma, e.g., Theorem 8.38 of [Sho05]. In other words, the challenge ciphertext $\mathsf{ct}^*_{\mathsf{id}^*}$ distributes according to $\mathbf{Game}_6$ and $\mathbf{Game}_7$ if $\beta = 0$ and 1, respectively. Therefore, it holds that $\mathbf{Game}_6 \approx_c \mathbf{Game}_7$ as required. □

5 Conclusion

In this paper, we construct the first purely adaptive and CCA-secure lattice-based IBEET schemes in the standard model. We pointed out that a special three-level schemes HIBE satisfying adaptive security only for the first level and selective security for the other levels is sufficient for constructing adaptively and CCA-secure IBEET, and we construct such HIBE schemes from ABB-type IBE. How to employ our technique to construct an attribute-based encryption with equality test (ABEET) having the same properties of semi-adaptively secure lattice-based ABE scheme for circuits [BV16] and adaptively secure lattice-based inner-product encryption [KNY+20] is left as a future work of this paper.

References

[ABB10a] Shweta Agrawal, Dan Boneh, and Xavier Boyen. "Efficient Lattice (H)IBE in the Standard Model." In: *EUROCRYPT*. 2010, pp. 553–572.

[ABB10b] Shweta Agrawal, Dan Boneh, and Xavier Boyen. "Lattice Basis Delegation in Fixed Dimension and Shorter-Ciphertext Hierarchical IBE." In: *CRYPTO*. 2010, pp. 98–115.

[AET+22] Kyoichi Asano, Keita Emura, Atsushi Takayasu, and Yohei Watanabe. "A Generic Construction of CCA-secure Attribute-based Encryption with Equality Test." In: *ProvSec*. 2022, pp. 3–19.

[AET22] Kyoichi Asano, Keita Emura, and Atsushi Takayasu. "More Efficient Adaptively Secure Lattice-based IBE with Equality Test in the Standard Model." ISC 2022, to appear.

[Ajt96] Miklós Ajtai. "Generating Hard Instances of Lattice Problems (Extended Abstract)." In: *28th ACM STOC*. ACM, 1996, pp. 99–108.

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. "Classical hardness of learning with errors." In: *45th ACM STOC*. ACM, 2013, pp. 575–584.

[BV16] Zvika Brakerski and Vinod Vaikuntanathan. "Circuit-ABE from LWE: Unbounded Attributes and Semi-adaptive Security." In: *CRYPTO*. 2016, pp. 363–384.

[CHK+12] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. "Bonsai Trees, or How to Delegate a Lattice Basis." In: *J. Cryptol.* 25.4 (2012), pp. 601–639.

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. "Chosen-Ciphertext Security from Identity-Based Encryption." In: *EUROCRYPT*. 2004, pp. 207–222.

[DLR+19] Dung Hoang Duong, Huy Quoc Le, Partha Sarathi Roy, and Willy Susilo. "Lattice-Based IBE with Equality Test in Standard Model." In: *Provable Security*. 2019, pp. 19–40.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. "Trapdoors for hard lattices and new cryptographic constructions." In: *40th ACM STOC*. ACM, 2008, pp. 197–206.

[JKN21] Tibor Jager, Rafael Kurek, and David Niehues. "Efficient Adaptively-Secure IB-KEMs and VRFs via Near-Collision Resistance." In: *Public-Key Cryptography*. 2021, pp. 596–626.

[KNY+20] Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. "Adaptively Secure Inner Product Encryption from LWE." In: *ASIACRYPT*. 2020, pp. 375–404.

[LLS+16] Hyung Tae Lee, San Ling, Jae Hong Seo, and Huaxiong Wang. "Semi-generic construction of public key encryption and identity-based encryption with equality test." In: *Information Sciences* 373 (2016), pp. 419–440.

[LLS+20] Hyung Tae Lee, San Ling, Jae Hong Seo, Huaxiong Wang, and Taek-Young Youn. "Public key encryption with equality test in the standard model." In: *Information Sciences* 516 (2020), pp. 89–108.

[LSQ18] Xi Jun Lin, Lin Sun, and Haipeng Qu. "Generic construction of public key encryption, identity-based encryption and signcryption with equality test." In: *Information Sciences* 453 (2018), pp. 111–126.

[Ma16] Sha Ma. "Identity-based encryption with outsourced equality test in cloud computing." In: *Information Sciences* 328 (2016), pp. 389–402.

[NSD+20] Giang Linh Duc Nguyen, Willy Susilo, Dung Hoang Duong, Huy Quoc Le, and Fuchun Guo. "Lattice-Based IBE with Equality Test Supporting Flexible Authorization in the Standard Model." In: *INDOCRYPT*. 2020, pp. 624–643.

[Reg09] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography." In: *J. ACM* 56.6 (2009), 34:1–34:40.

[SDL20] Willy Susilo, Dung Hoang Duong, and Huy Quoc Le. "Efficient Post-quantum Identity-based Encryption with Equality Test." In: *IEEE ICPADS*. 2020, pp. 633–640.

[Sho05] Victor Shoup. "A Computational Introduction to Number Theory and Algebra." Cambridge University Press, 2005.

[SRB12] Kunwar Singh, C. Pandu Rangan, and A. K. Banerjee. "Adaptively Secure Efficient Lattice (H)IBE in Standard Model with Short Public Parameters." In: *SPACE*. 2012, pp. 153–172.

[Tsa19] Rotem Tsabary. "Fully Secure Attribute-Based Encryption for t-CNF from LWE." In: *CRYPTO*. 2019, pp. 62–85.

[Wat05] Brent Waters. "Efficient Identity-Based Encryption Without Random Oracles." In: *EUROCRYPT*. 2005, pp. 114–127.

[WWY+21] Zhenghao Wu, Jian Weng, Anjia Yang, Lisha Yao, Xiaojian Liang, Zike Jiang, and Jinghang Wen. "Efficient and Fully Secure Lattice-Based IBE with Equality Test." In: *ICICS*. 2021, pp. 301–318.

[Yam17] Shota Yamada. "Asymptotically Compact Adaptively Secure Lattice IBEs and Verifiable Random Functions via Generalized Partitioning Techniques." In: *CRYPTO*. 2017, pp. 161–193.

[YTH+10] Guomin Yang, Chik How Tan, Qiong Huang, and Duncan S. Wong. "Probabilistic Public Key Encryption with Equality Test." In: *CT-RSA*. 2010, pp. 119–131.

[Zha12] Mark Zhandry. "Secure Identity-Based Encryption in the Quantum Random Oracle Model." In: *CRYPTO*. Ed. by Reihaneh Safavi-Naini and Ran Canetti. 2012, pp. 758–775.