On the Feasibility of Identity-based Encryption with Equality Test against Insider Attacks

Keita Emura[1]

[1] National Institute of Information and Communications Technology, Japan.

May 11, 2023

Abstract

As a generalization of public key encryption with keyword search, public key encryption with equality test was proposed, and identity-based encryption with equality test (IBEET) is its identity-based variant. In IBEET, anyone can check whether two ciphertexts of distinct identities are encryptions of the same plaintext or not using trapdoors. Due to its functionality, IBEET cannot provide any indistinguishability-based security for trapdoor holders. As a variant of IBEET, IBEET against insider attacks (IBEETIA) was proposed, where a token is defined for each identity and is used for encryption, and anyone can check whether two ciphertexts of distinct identities are encryptions of the same plaintext or not without using trapdoors, and an indistinguishability security of IBEETIA was defined. Lee et al. (ACISP 2018) and Duong et al. (ProvSec 2019) proposed a paring-based and a lattice-based constructions, respectively. That is, current concrete IBEETIA schemes are constructed by identity-based encryption (IBE) related complexity assumptions. According to the implication result shown by Boneh et al. (FOCS 2008), IBE is recognized as a strong cryptographic primitive because no black-box construction of IBE from trapdoor permutations exist. However, Emura and Takayasu (IEICE Transactions 2023) demonstrated that symmetric key encryption and pseudo-random permutations are sufficient to construct IBEETIA which is secure in the previous security definition. These results suggest us to explore a condition of IBEETIA that requires to employ IBE-related complexity assumptions. In this paper, we demonstrate a sufficient condition that IBEETIA implies IBE. We define one-wayness against chosen-plaintext/ciphertext attacks for the token generator (OW-TG-CPA/CCA) and for token holders (OW-TH-CPA/CCA), which were not considered in the previous security definition. We show that OW-TG-CPA secure IBEETIA with additional conditions implies OW-CPA secure IBE, and show that Lee et al. and Duong et al. schemes provide the OW-TG-CPA security. On the other hand, we propose a generic construction of OW-TH-CCA secure IBEETIA from public key encryption. Our results suggest a design principle to efficiently construct IBEETIA without employing IBE-related complexity assumptions.

Keywords. Identity-based encryption with equality test against insider attacks, Searchable Encryption, Generic Construction

1 Introduction

As a generalization of public key encryption with keyword search [4], public key encryption with equality test (PKEET) was proposed [24], where anyone can check whether two ciphertexts of distinct public keys are encryptions of the same plaintext or not using trapdoors. Identity-based encryption with equality test (IBEET) is its identity-based variant, where anyone can check whether two ciphertexts of distinct identities are encryptions of the same plaintext or not using trapdoors. Lee et al. [17] demonstrated that IBEET can generically be constructed from 3-level hierarchical identity-based encryption (HIBE).[1] Asano et al. [2] further improved the Lee et al. generic construction to capture state-of-the-art lattice-based IBE schemes [16, 23].

IBEET cannot provide any indistinguishability-based security for insiders who have a trapdoor due to the test functionality of IBEET. Concretely, two security notions are defined against two types of adversaries called Type-I and Type-II. The Type-I adversary is allowed to obtain trapdoors for the challenge identity, and thus the one-wayness against the Type-I adversary was defined. The Type-II adversary is not allowed to obtain trapdoors for the challenge identity, and thus the indistinguishability against the Type-II adversary was defined.[2]

IBEET against insider attacks. Since IBEET does not provide any indistinguishability security against insiders who have trapdoors (the Type-I adversary above), Wu et al. [22] introduced IBEET against insider attacks ($\mathsf{IBEETIA}$). We briefly introduce $\mathsf{IBEETIA}$ as follows. The setup algorithm outputs a master token key MTK, in addition to a master public key MPK and a master secret key MSK. As in IBE, MSK is used for generating a secret key $\mathsf{sk}_{\mathsf{ID}}$ for an identity ID. MTK is used for generating a token $\mathsf{tok}_{\mathsf{ID}}$ for an identity ID. The encryption algorithm takes $\mathsf{tok}_{\mathsf{ID}}$ in addition to MPK, ID, a plaintext M, and outputs a ciphertext ct. The decryption algorithm takes MPK, ct, and both $\mathsf{sk}_{\mathsf{ID}}$ and $\mathsf{tok}_{\mathsf{ID}}$, and outputs M. Anyone can run the test algorithm that takes MPK and two ciphertexts without using trapdoors. $\mathsf{IBEETIA}$ can provide an indistinguishability when an adversary is not allowed to obtain a token $\mathsf{tok}_{\mathsf{ID}^*}$ for the challenge identity ID^*, a ciphertext generated by ID^*, and a secret key $\mathsf{sk}_{\mathsf{ID}^*}$.

Lee et al. [18] pointed out a security flaw of the Wu et al. scheme [22], and proposed a pairing-based $\mathsf{IBEETIA}$ scheme. Moreover, Duong et al. [11] proposed a lattice-based $\mathsf{IBEETIA}$ scheme. They employed IBE-related complexity assumptions. According to the implication result shown by Boneh et al. [6], IBE is recognized as a strong cryptographic primitive because no black-box construction of IBE from trapdoor permutations exist. At the first place, these selections are reasonable because HIBE is employed to construct IBEET [2,17]. However, Emura and Takayasu [14] demonstrated that symmetric key encryption and pseudo-random permutations are sufficient to construct $\mathsf{IBEETIA}$ which is secure in the security definition given in [11, 18, 22]. They paid attention to the syntax that a token $\mathsf{tok}_{\mathsf{ID}}$ is used for both encryption and decryption, as in symmetric key encryption, and they set $\mathsf{tok}_{\mathsf{ID}}$ as a key $\mathsf{SKE.sk}$ of a symmetric encryption scheme. Interestingly, $\mathsf{sk}_{\mathsf{ID}} = \perp$ in the Emura-Takayasu construction because $\mathsf{sk}_{\mathsf{ID}}$ is not used for decryption. Emura and Takayasu [14] mentioned that:

> *One may wonder whether our construction provides the same functionality of previous* $\mathsf{IBEETIA}$ *schemes since* $\mathsf{sk}_{\mathsf{ID}} = \perp$. *More precisely, because the* Dec *algorithm takes both* $\mathsf{sk}_{\mathsf{ID}}$ *and* $\mathsf{tok}_{\mathsf{ID}}$ *as input, anyone who has* $\mathsf{tok}_{\mathsf{ID}}$ *can decrypt all ciphertexts generated by* ID *in our construction, whereas one who has* $\mathsf{tok}_{\mathsf{ID}}$ *but does not have* $\mathsf{sk}_{\mathsf{ID}}$ *cannot decrypt such ciphertexts in the previous schemes. We emphasize that this difference does not violate not only the correctness but also the wIND-CCA security.*

Here, wIND-CCA stands for weak indistinguishability against chosen ciphertext attacks. According to this observation, it is natural to consider a security notion against token holders who have $\mathsf{tok}_{\mathsf{ID}}$ but do not have $\mathsf{sk}_{\mathsf{ID}}$, as in the Type-I adversary in IBEET.

Our Contribution. In this paper, we demonstrate a sufficient condition that $\mathsf{IBEETIA}$ implies IBE. We define one-wayness against chosen-plaintext/ciphertext attacks for the token generator

[1] Lee et al. [17] also demonstrated that PKEET can generically be constructed from 2-level HIBE.

[2] Chosen-ciphertext attack (CCA) is considered against both types of adversaries. See [2, 17] for more details.

(OW-TG-CPA/CCA), where an adversary has MTK, and for token holders (OW-TH-CPA/CCA), where an adversary is allowed to issue token queries for any identity. Our main results are explained as follows.

- We show that OW-TG-CPA secure $\mathsf{IBEETIA}$ implies OW-CPA secure IBE. Here, we assume the following additional conditions, where $\mathsf{sk}_{\mathsf{ID}}$ is related to MSK and is independent to MTK, and $\mathsf{tok}_{\mathsf{ID}}$ is related to MTK and is independent to MSK.
  - This structure was employed in the previous constructions [11, 14, 18, 22]. For example, in the Lee et al. scheme [18], $\mathsf{tok}_{\mathsf{ID}}$ is a pseudo-random permutation key and a message authenticated code (MAC) key, and $\mathsf{sk}_{\mathsf{ID}}$ is a secret key of the Boneh-Franklin IBE scheme [5] with the form $H(\mathsf{ID})^\alpha$. In the Duong et al. scheme [11], $\mathsf{tok}_{\mathsf{ID}}$ is a trapdoor for a matrix A' of the Agrawal-Boneh-Boyen (ABB) IBE scheme [1] and $\mathsf{sk}_{\mathsf{ID}}$ is a secret key of the ABB IBE scheme generated by a trapdoor for a matrix A. Here, two matrices A' and A are independently chosen.
  - This structure and the OW-TG-CPA security allow us to construct IBE from $\mathsf{IBEETIA}$. Intuitively, $(\mathsf{MPK}, \mathsf{MTK})$ is set as a master public key of IBE. Revealing MTK allows anyone to generate a ciphertext by computing $\mathsf{tok}_{\mathsf{ID}}$ and supports exponentially many identities.
- On the other hand, we propose a generic construction of OW-TH-CCA secure $\mathsf{IBEETIA}$ from CCA-secure public key encryption (PKE), pseudo-random permutations, and a hash function. Because the encryption algorithm takes a token $\mathsf{tok}_{\mathsf{ID}}$ as input and the number of tokens are bounded by a polynomial of the security parameter, there is room for constructing an OW-TG-CCA secure $\mathsf{IBEETIA}$ scheme from cryptographic primitives which are weaker than IBE.
  - Unlike to the previous constructions, $\mathsf{sk}_{\mathsf{ID}}$ and $\mathsf{tok}_{\mathsf{ID}}$ are related. Let $(\mathsf{PKE.pk}, \mathsf{PKE.dk})$ be a pair of public and decryption key of a PKE scheme. $\mathsf{tok}_{\mathsf{ID}}$ contains a pseudo-random permutation key k and $\mathsf{PKE.pk}$, and $\mathsf{sk}_{\mathsf{ID}} = \mathsf{PKE.dk}$. Since revealing $\mathsf{PKE.pk}$ does not affect the security, this structure provides the OW-TH-CPA security. Moreover, CCA security of the underlying PKE scheme provides OW-TH-CCA security.

Briefly, OW-TG-CPA guarantees that a plaintext is not recovered from a ciphertext even against the token generator that has MTK and issues tokens to users similar to the key generation center of IBE. OW-TG-CPA is a stronger notion because it considers a kind of the key escrow problem of IBE [8, 12, 13]. In other words, $\mathsf{IBEETIA}$ implies IBE when such a stronger security notion is considered. OW-TH-CPA guarantees that a plaintext is not recovered from a ciphertext even against token holders who have tokens, as in the Type-I adversary in IBEET, and seems sufficient in practice. Our results suggest a design principle to efficiently construct $\mathsf{IBEETIA}$ without employing IBE-related complexity assumptions.

2 Preliminaries

SKE. An symmetric key encryption scheme SKE consists of the following three algorithms ($\mathsf{SKE.KeyGen}$, $\mathsf{SKE.Enc}$, $\mathsf{SKE.Dec}$). The key generation algorithm $\mathsf{SKE.KeyGen}$ takes a security parameter $\lambda \in \mathbb{N}$, and outputs a secret key $\mathsf{SKE.sk}$. The encryption algorithm $\mathsf{SKE.Enc}$ takes $\mathsf{SKE.sk}$ and a plaintext M as input, and outputs a ciphertext $\mathsf{ct}_{\mathsf{SKE}}$. The decryption algorithm $\mathsf{SKE.Dec}$ takes $\mathsf{SKE.sk}$ and $\mathsf{ct}_{\mathsf{SKE}}$ as input, and output M or $\perp$.

PKE. An public key encryption scheme PKE consists of the following three algorithms ($\mathsf{PKE.KeyGen}$, $\mathsf{PKE.Enc}$, $\mathsf{PKE.Dec}$). The key generation algorithm $\mathsf{PKE.KeyGen}$ takes a security parameter $\lambda \in \mathbb{N}$, and outputs a public key $\mathsf{PKE.pk}$ and a decryption key $\mathsf{PKE.dk}$. The encryption algorithm $\mathsf{PKE.Enc}$ takes $\mathsf{PKE.pk}$ and a plaintext M as input, and outputs a ciphertext $\mathsf{ct_{PKE}}$. The decryption algorithm $\mathsf{PKE.Dec}$ takes $\mathsf{PKE.pk}$, $\mathsf{PKE.dk}$, and $\mathsf{ct_{PKE}}$ as input, and output M or $\perp$. For all $\lambda \in \mathbb{N}$, $(\mathsf{PKE.pk}, \mathsf{PKE.dk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$ and $M \in \mathcal{M}$, it is required that $\Pr[M \leftarrow \mathsf{PKE.Dec}(\mathsf{PKE.pk}, \mathsf{PKE.dk}, \mathsf{PKE.Enc}(\mathsf{PKE.pk}, M))] = 1 - \mathsf{negl}(\lambda)$ holds.

Next, we define the indistinguishability against chosen-ciphertext attacks (IND-CCA) as follows. Let $\mathcal{A}$ be a PPT adversary and $\mathcal{C}$ be the challenger. $\mathcal{C}$ runs $(\mathsf{PKE.pk}, \mathsf{PKE.dk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$ and sends $\mathsf{PKE.pk}$ to $\mathcal{A}$. $\mathcal{A}$ is allowed to issue decryption queries. $\mathcal{A}$ sends $\mathsf{ct_{PKE}}$ to $\mathcal{C}$. $\mathcal{C}$ returns the result of $\mathsf{PKE.Dec}(\mathsf{PKE.pk}, \mathsf{PKE.dk}, \mathsf{ct_{PKE}})$. In the challenge phase, $\mathcal{A}$ declares two equality-length challenge plaintexts (M_0^*, M_1^*). $\mathcal{C}$ flips a coin $b \in \{0,1\}$, runs $\mathsf{ct}^*_{\mathsf{PKE}} \leftarrow \mathsf{PKE.Enc}(\mathsf{PKE.pk}, M_b^*)$, and sends $\mathsf{ct}^*_{\mathsf{PKE}}$ to $\mathcal{A}$. $\mathcal{A}$ is allowed to issue decryption queries. $\mathcal{A}$ sends $\mathsf{ct_{PKE}} \neq \mathsf{ct}^*_{\mathsf{PKE}}$ to $\mathcal{C}$. $\mathcal{C}$ returns the result of $\mathsf{PKE.Dec}(\mathsf{PKE.pk}, \mathsf{PKE.dk}, \mathsf{ct_{PKE}})$. Finally, $\mathcal{A}$ outputs $b' \in \{0,1\}$. The advantage of $\mathcal{A}$ is defined as $\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{PKE},\mathcal{A}}(1^\lambda) := |\Pr[b = b'] - 1/2|$. We say that PKE is IND-CCA secure if $\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{PKE},\mathcal{A}}(1^\lambda)$ is negligible.

IBE. An identity-based encryption scheme IBE consists of the following four algorithms ($\mathsf{IBE.Setup}$, $\mathsf{IBE.Extract}$, $\mathsf{IBE.Enc}$, $\mathsf{IBE.Dec}$). The setup algorithm $\mathsf{IBE.Setup}$ takes a security parameter $\lambda \in \mathbb{N}$, and outputs a master public key MPK and a master secret key MSK. The key extraction algorithm $\mathsf{IBE.Extract}$ takes MPK, MSK, and an identity $\mathsf{ID} \in \mathcal{ID}$ as input, and outputs a secret key $\mathsf{sk_{ID}}$. Here, $\mathcal{ID}$ is the identity space which is implicitly included in MPK. The encryption algorithm $\mathsf{IBE.Enc}$ takes MPK, ID, and a plaintext $M \in \mathcal{M}$ as input, and outputs a ciphertext $\mathsf{ct_{IBE}}$. Here, $\mathcal{M}$ is the message space which is implicitly included in MPK. The decryption algorithm $\mathsf{IBE.Dec}$ takes MPK, $\mathsf{ct_{IBE}}$, and $\mathsf{sk_{ID}}$, and outputs M or $\perp$. For all $\lambda \in \mathbb{N}$, $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$, $\mathsf{ID} \in \mathcal{ID}$, and $M \in \mathcal{M}$, it is required that $\Pr[M \leftarrow \mathsf{IBE.Dec}(\mathsf{MPK}, \mathsf{IBE.Enc}(\mathsf{MPK}, \mathsf{ID}, M), \mathsf{IBE.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{ID}))] = 1 - \mathsf{negl}(\lambda)$ holds.

Next, we define the one-wayness against chosen-plaintext attacks (OW-CPA) as follows. Let $\mathcal{A}$ be a PPT adversary and $\mathcal{C}$ be the challenger. $\mathcal{C}$ runs $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$ and sends MPK to $\mathcal{A}$. $\mathcal{A}$ is allowed to issue key extraction queries. $\mathcal{A}$ sends ID to $\mathcal{C}$. $\mathcal{C}$ runs $\mathsf{sk_{ID}} \leftarrow \mathsf{IBE.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{ID})$ and sends $\mathsf{sk_{ID}}$ to $\mathcal{A}$. In the challenge phase, $\mathcal{A}$ declares the challenge identity ID^* which was not sent as a key extraction query. $\mathcal{C}$ randomly choose $M^* \leftarrow \mathcal{M}$, runs $\mathsf{ct}^*_{\mathsf{IBE}} \leftarrow \mathsf{IBE.Enc}(\mathsf{MPK}, \mathsf{ID}^*, M^*)$, and sends $\mathsf{ct}^*_{\mathsf{IBE}}$ to $\mathcal{A}$. Finally, $\mathcal{A}$ outputs $\hat{M}$. The advantage of $\mathcal{A}$ is defined as $\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{IBE},\mathcal{A}}(1^\lambda) := |\Pr[M^* = \hat{M}] - 1/|\mathcal{M}||$. We say that IBE is OW-CPA secure if $\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{IBE},\mathcal{A}}(1^\lambda)$ is negligible.

IBEETIA [11, 14, 18, 22]. An IBEETIA scheme $\mathsf{IBEETIA}$ consists of the following five algorithms ($\mathsf{IBEETIA.Setup}$, $\mathsf{IBEETIA.Extract}$, $\mathsf{IBEETIA.Enc}$, $\mathsf{IBEETIA.Dec}$, $\mathsf{IBEETIA.Test}$) defined below. A master token key MTK is used for generating a token $\mathsf{tok_{ID}}$ for ID, and $\mathsf{tok_{ID}}$ is required in the encryption algorithm. Anyone can run the test algorithm against two ciphertexts without using trapdoors. By restricting who can run the encryption algorithm, $\mathsf{IBEETIA}$ can provide the indistinguishability security against non-insiders who do not have $\mathsf{tok_{ID}}$.

$\mathsf{IBEETIA.Setup}$**:** The setup algorithm takes a security parameter $\lambda \in \mathbb{N}$, and outputs a master public key MPK, a master secret key MSK, and a master token key MTK.

$\mathsf{IBEETIA.Extract}$**:** The key extraction algorithm $\mathsf{IBE.Extract}$ takes MPK, MSK, and MTK, and an identity $\mathsf{ID} \in \mathcal{ID}$ as input, and outputs a secret key $\mathsf{sk_{ID}}$ and a token $\mathsf{tok_{ID}}$. Here, $\mathcal{ID}$ is the identity space which is implicitly included in MPK.

$\mathsf{IBEETIA.Enc}$: The encryption algorithm takes MPK, $\mathsf{tok_{ID}}$, ID, and a plaintext $M \in \mathcal{M}$ as input, and outputs a ciphertext $\mathsf{ct_{IBEETIA}}$. Here, $\mathcal{M}$ is the message space which is implicitly included in MPK.

$\mathsf{IBEETIA.Dec}$: The decryption algorithm takes MPK, $\mathsf{sk_{ID}}$, $\mathsf{tok_{ID}}$, and $\mathsf{ct_{IBEETIA}}$, and outputs M or $\perp$.

$\mathsf{IBEETIA.Test}$: The test algorithm takes MPK and two ciphertexts $\mathsf{ct_{IBEETIA}}$ and $\mathsf{ct'_{IBEETIA}}$, and outputs 1 or 0.

The correctness of an $\mathsf{IBEETIA}$ scheme is defined as follows. Here, a probability space is random coins to run $\mathsf{IBEETIA.Setup}$, $\mathsf{IBEETIA.Extract}$, and $\mathsf{IBEETIA.Enc}$. We note that the first condition is employed in our security proof.

1. For all $\lambda \in \mathbb{N}$, $\mathsf{ID} \in \mathcal{ID}$, and $M \in \mathcal{M}$, $M' = M$ holds with overwhelming probability where $(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}) \leftarrow \mathsf{IBEETIA.Setup}(1^\lambda)$, $(\mathsf{sk_{ID}}, \mathsf{tok_{ID}}) \leftarrow \mathsf{IBEETIA.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}, \mathsf{ID})$, $\mathsf{ct_{IBEETIA}} \leftarrow \mathsf{IBEETIA.Enc}(\mathsf{MPK}, \mathsf{tok_{ID}}, \mathsf{ID}, M)$, and $M' \leftarrow \mathsf{IBEETIA.Dec}(\mathsf{MPK}, \mathsf{sk_{ID}}, \mathsf{tok_{ID}}, \mathsf{ct_{IBEETIA}})$.
2. For all $\lambda \in \mathbb{N}$, $\mathsf{ID}, \mathsf{ID}' \in \mathcal{ID}$, and $M, M' \in \mathcal{M}$, $\mathsf{IBEETIA.Test}(\mathsf{MPK}, \mathsf{ct_{IBEETIA}}, \mathsf{ct'_{IBEETIA}}) = 1$ holds with overwhelming probability, where $(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}) \leftarrow \mathsf{IBEETIA.Setup}(1^\lambda)$, $(\mathsf{sk_{ID}}, \mathsf{tok_{ID}}) \leftarrow \mathsf{IBEETIA.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}, \mathsf{ID})$, $(\mathsf{sk_{ID'}}, \mathsf{tok_{ID'}}) \leftarrow \mathsf{IBEETIA.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}, \mathsf{ID}')$, $\mathsf{ct_{IBEETIA}} \leftarrow \mathsf{IBEETIA.Enc}(\mathsf{MPK}, \mathsf{tok_{ID}}, \mathsf{ID}, M)$, and $\mathsf{ct'_{IBEETIA}} \leftarrow \mathsf{IBEETIA.Enc}(\mathsf{MPK}, \mathsf{tok_{ID'}}, \mathsf{ID}', M')$.
3. For all $\lambda \in \mathbb{N}$, $\mathsf{ID}, \mathsf{ID}' \in \mathcal{ID}$, and $M, M' \in \mathcal{M}$ such that $M \neq M'$, $\mathsf{IBEETIA.Test}(\mathsf{MPK}, \mathsf{ct_{IBEETIA}}, \mathsf{ct'_{IBEETIA}}) = 1$ holds with negligible probability, where $(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}) \leftarrow \mathsf{IBEETIA.Setup}(1^\lambda)$, $(\mathsf{sk_{ID}}, \mathsf{tok_{ID}}) \leftarrow \mathsf{IBEETIA.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}, \mathsf{ID})$, $(\mathsf{sk_{ID'}}, \mathsf{tok_{ID'}}) \leftarrow \mathsf{IBEETIA.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}, \mathsf{ID}')$, $\mathsf{ct_{IBEETIA}} \leftarrow \mathsf{IBEETIA.Enc}(\mathsf{MPK}, \mathsf{tok_{ID}}, \mathsf{ID}, M)$, and $\mathsf{ct'_{IBEETIA}} \leftarrow \mathsf{IBEETIA.Enc}(\mathsf{MPK}, \mathsf{tok_{ID'}}, \mathsf{ID}', M')$.

Next, we define the weak indistinguishability against chosen ciphertext attacks (wIND-CCA). Unlike to IBE, the encryption algorithm takes $\mathsf{tok_{ID}}$ as input. Thus, an adversary $\mathcal{A}$ is allowed to issue an encryption query (ID, M) where $\mathsf{ID} \neq \mathsf{ID}^*$. Since $\mathcal{A}$ is not allowed to obtain a ciphertext generated by ID^*, the security notion is stated as weak. If $\mathcal{A}$ declares the challenge identity ID^* before the setup phase, we call it selectively wIND-CCA secure.

Definition 1 (wIND-CCA [14, 18, 22])**.** *Let $\mathcal{A}$ be a PPT adversary and $\mathcal{C}$ be the challenger. $\mathcal{C}$ runs* $(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}) \leftarrow \mathsf{IBEETIA.Setup}(1^\lambda)$ *and sends* MPK *to $\mathcal{A}$. $\mathcal{A}$ is allowed to issue queries below.*

Key Extraction: *$\mathcal{A}$ sends* ID *to $\mathcal{C}$. $\mathcal{C}$ runs* $(\mathsf{sk_{ID}}, \mathsf{tok_{ID}}) \leftarrow \mathsf{IBEETIA.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}, \mathsf{ID})$ *and sends* $\mathsf{sk_{ID}}$ *to $\mathcal{A}$. Here $\mathcal{C}$ does not send* $\mathsf{tok_{ID}}$ *to $\mathcal{A}$.*

Encryption: *$\mathcal{A}$ sends* (ID, M) *to $\mathcal{C}$. $\mathcal{C}$ runs* $(\mathsf{sk_{ID}}, \mathsf{tok_{ID}}) \leftarrow \mathsf{IBEETIA.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}, \mathsf{ID})$ *and* $\mathsf{ct_{IBEETIA}} \leftarrow \mathsf{IBEETIA.Enc}(\mathsf{MPK}, \mathsf{tok_{ID}}, \mathsf{ID}, M)$*, and sends* $\mathsf{ct_{IBEETIA}}$ *to $\mathcal{A}$.*

Decryption: *$\mathcal{A}$ sends* $(\mathsf{ID}, \mathsf{ct_{IBEETIA}})$ *to $\mathcal{C}$. $\mathcal{C}$ runs* $(\mathsf{sk_{ID}}, \mathsf{tok_{ID}}) \leftarrow \mathsf{IBEETIA.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}, \mathsf{ID})$ *and return the result of* $\mathsf{IBEETIA.Dec}(\mathsf{MPK}, \mathsf{sk_{ID}}, \mathsf{tok_{ID}}, \mathsf{ct_{IBEETIA}})$*.*

In the challenge phase, $\mathcal{A}$ declares two plaintexts (M_0^*, M_1^*) and the challenge identity ID^*. It is required that ID^* was not sent as a key extraction query and an encryption query. $\mathcal{C}$ randomly flips a coin $b \leftarrow \{0,1\}$, runs $(\mathsf{sk}_{\mathsf{ID}^*}, \mathsf{tok}_{\mathsf{ID}^*}) \leftarrow \mathsf{IBEETIA.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}, \mathsf{ID})$, $\mathsf{ct}^*_{\mathsf{IBEETIA}} \leftarrow \mathsf{IBEETIA.Enc}(\mathsf{MPK}, \mathsf{tok}_{\mathsf{ID}^*}, \mathsf{ID}^*, M_b^*)$, and sends $\mathsf{ct}^*_{\mathsf{IBEETIA}}$ to $\mathcal{A}$. $\mathcal{A}$ is not allowed to issue $(\mathsf{ID}^*, \mathsf{ct}^*_{\mathsf{IBEETIA}})$ as a decryption query. Finally, $\mathcal{A}$ outputs $b' \in \{0,1\}$. The advantage of $\mathcal{A}$ is defined as

$$\mathsf{Adv}^{\mathsf{wIND\text{-}CCA}}_{\mathsf{IBEETIA},\mathcal{A}}(1^\lambda) := |\Pr[b = b'] - 1/2|$$

We say that $\mathsf{IBEETIA}$ is wIND-CCA secure if $\mathsf{Adv}^{\mathsf{wIND\text{-}CCA}}_{\mathsf{IBEETIA},\mathcal{A}}(1^\lambda)$ is negligible.

3 OW-CPA Security of IBEETIA

In this section, we define one-wayness against chosen-plaintext attacks for the token generator (OW-TG-CPA) which guarantees that a plaintext is not recovered from a ciphertext even against the token generator. $\mathcal{A}$ is allowed to obtain MTK in addition to MPK. Thus, no encryption oracle is defined. If $\mathcal{A}$ declares the challenge identity ID^* before the setup phase, we call it selectively secure. OW-TG-CCA security can also be defined. We remark that OW-TG-CPA is sufficient to show our implication result.

Definition 2 (OW-TG-CPA)**.** *Let $\mathcal{A}$ be a PPT adversary and $\mathcal{C}$ be the challenger. $\mathcal{C}$ runs $(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}) \leftarrow \mathsf{IBEETIA.Setup}(1^\lambda)$ and sends $(\mathsf{MPK}, \mathsf{MTK})$ to $\mathcal{A}$. $\mathcal{A}$ is allowed to issue queries below.*

Key Extraction: *$\mathcal{A}$ sends ID to $\mathcal{C}$. $\mathcal{C}$ runs $(\mathsf{sk}_{\mathsf{ID}}, \mathsf{tok}_{\mathsf{ID}}) \leftarrow \mathsf{IBEETIA.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}, \mathsf{ID})$ and sends $\mathsf{sk}_{\mathsf{ID}}$ to $\mathcal{A}$. Here $\mathcal{C}$ does not send $\mathsf{tok}_{\mathsf{ID}}$ to $\mathcal{A}$.*

In the challenge phase, $\mathcal{A}$ declares the challenge identity ID^* which was not sent as a key extraction query. $\mathcal{C}$ randomly chooses $M^* \leftarrow \mathcal{M}$, runs $(\mathsf{sk}_{\mathsf{ID}^*}, \mathsf{tok}_{\mathsf{ID}^*}) \leftarrow \mathsf{IBEETIA.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}, \mathsf{ID}^*)$, $\mathsf{ct}^*_{\mathsf{IBEETIA}} \leftarrow \mathsf{IBEETIA.Enc}(\mathsf{MPK}, \mathsf{tok}_{\mathsf{ID}^*}, \mathsf{ID}^*, M^*)$, and sends $\mathsf{ct}^*_{\mathsf{IBEETIA}}$ to $\mathcal{A}$. Finally, $\mathcal{A}$ outputs $\hat{M}$. The advantage of $\mathcal{A}$ is defined as

$$\mathsf{Adv}^{\mathsf{OW\text{-}TG\text{-}CPA}}_{\mathsf{IBEETIA},\mathcal{A}}(1^\lambda) := |\Pr[M^* = \hat{M}] - 1/|\mathcal{M}||$$

We say that $\mathsf{IBEETIA}$ is OW-TG-CPA secure if $\mathsf{Adv}^{\mathsf{OW\text{-}TG\text{-}CPA}}_{\mathsf{IBEETIA},\mathcal{A}}(1^\lambda)$ is negligible.

Next, we define one-wayness against chosen-ciphertext attacks for token holders (OW-TH-CCA), where an adversary is allowed to issue token queries for any identity which guarantees that a plaintext is not recovered from a ciphertext even against token holders who have tokens. OW-TH-CPA is also defined when no decryption oracle is considered. The token extraction oracle can be simulated when MTK is given. Thus, the OW-TG-CPA security implies the OW-TH-CPA security.

Definition 3 (OW-TH-CCA)**.** *Let $\mathcal{A}$ be a PPT adversary and $\mathcal{C}$ be the challenger. $\mathcal{C}$ runs $(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}) \leftarrow \mathsf{IBEETIA.Setup}(1^\lambda)$ and sends MPK to $\mathcal{A}$. $\mathcal{A}$ is allowed to issue queries below.*

Key Extraction: *$\mathcal{A}$ sends ID to $\mathcal{C}$. $\mathcal{C}$ runs $(\mathsf{sk}_{\mathsf{ID}}, \mathsf{tok}_{\mathsf{ID}}) \leftarrow \mathsf{IBEETIA.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}, \mathsf{ID})$ and sends $\mathsf{sk}_{\mathsf{ID}}$ to $\mathcal{A}$. Here $\mathcal{C}$ does not send $\mathsf{tok}_{\mathsf{ID}}$ to $\mathcal{A}$.*

Token Extraction: *$\mathcal{A}$ sends ID to $\mathcal{C}$. $\mathcal{C}$ runs $(\mathsf{sk}_{\mathsf{ID}}, \mathsf{tok}_{\mathsf{ID}}) \leftarrow \mathsf{IBEETIA.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}, \mathsf{ID})$ and sends $\mathsf{tok}_{\mathsf{ID}}$ to $\mathcal{A}$. Here $\mathcal{C}$ does not send $\mathsf{sk}_{\mathsf{ID}}$ to $\mathcal{A}$.*

Decryption: *$\mathcal{A}$ sends* $(\mathsf{ID}, \mathsf{ct}_{\mathsf{IBEETIA}})$ *to* $\mathcal{C}$*.* $\mathcal{C}$ *runs* $(\mathsf{sk}_{\mathsf{ID}}, \mathsf{tok}_{\mathsf{ID}}) \leftarrow \mathsf{IBEETIA.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}, \mathsf{ID})$ *and return the result of* $\mathsf{IBEETIA.Dec}(\mathsf{MPK}, \mathsf{sk}_{\mathsf{ID}}, \mathsf{tok}_{\mathsf{ID}}, \mathsf{ct}_{\mathsf{IBEETIA}})$*.*

In the challenge phase, $\mathcal{A}$ *declares the challenge identity* ID^* *which was not sent as a key extraction query. Note that* $\mathcal{A}$ *is allowed to obtain* $\mathsf{tok}_{\mathsf{ID}^*}$*.* $\mathcal{C}$ *randomly chooses* $M^* \leftarrow \mathcal{M}$*, runs* $(\mathsf{sk}_{\mathsf{ID}^*}, \mathsf{tok}_{\mathsf{ID}^*}) \leftarrow \mathsf{IBEETIA.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}, \mathsf{ID}^*)$*,* $\mathsf{ct}^*_{\mathsf{IBEETIA}} \leftarrow \mathsf{IBEETIA.Enc}(\mathsf{MPK}, \mathsf{tok}_{\mathsf{ID}^*}, \mathsf{ID}^*, M^*)$*, and sends* $\mathsf{ct}^*_{\mathsf{IBEETIA}}$ *to* $\mathcal{A}$*.* $\mathcal{A}$ *is not allowed to issue* $(\mathsf{ID}^*, \mathsf{ct}^*_{\mathsf{IBEETIA}})$ *as a decryption query. Finally,* $\mathcal{A}$ *outputs* $\hat{M}$*. The advantage of* $\mathcal{A}$ *is defined as*

$$\mathsf{Adv}^{\mathsf{OW\text{-}TH\text{-}CCA}}_{\mathsf{IBEETIA},\mathcal{A}}(1^\lambda) := |\Pr[M^* = \hat{M}] - 1/|\mathcal{M}||$$

We say that $\mathsf{IBEETIA}$ *is OW-TH-CCA secure if* $\mathsf{Adv}^{\mathsf{OW\text{-}TH\text{-}CCA}}_{\mathsf{IBEETIA},\mathcal{A}}(1^\lambda)$ *is negligible.*

4 Our IBE Construction from IBEETIA

In this section, we construct an OW-CPA secure IBE scheme from an OW-TG-CPA secure $\mathsf{IBEETIA}$ scheme. In addition to the OW-TG-CPA security, we assume the following additional conditions, where $\mathsf{sk}_{\mathsf{ID}}$ is related to MSK and is independent to MTK, and $\mathsf{tok}_{\mathsf{ID}}$ is related to MTK and is independent to MSK. This is a sufficient condition that $\mathsf{IBEETIA}$ implies IBE. Concretely, the $\mathsf{IBEETIA.Extract}$ algorithm is divided to two sub-algorithms as follows.

$\mathsf{IBEETIA.ExtractSK}$**:** The secret key extraction algorithm takes as MPK, MSK, and $\mathsf{ID} \in \mathcal{ID}$ as input, and outputs a secret key $\mathsf{sk}_{\mathsf{ID}}$.

$\mathsf{IBEETIA.ExtractTK}$**:** The token extraction algorithm takes as MPK, MTK, and $\mathsf{ID} \in \mathcal{ID}$ as input, and outputs a token $\mathsf{tok}_{\mathsf{ID}}$.

Then, the $\mathsf{IBEETIA.Extract}$ algorithm is defined as follows.

$\mathsf{IBEETIA.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}, \mathsf{ID})$**:** Run $\mathsf{sk}_{\mathsf{ID}} \leftarrow \mathsf{IBEETIA.ExtractSK}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{ID})$ and $\mathsf{tok}_{\mathsf{ID}} \leftarrow \mathsf{IBEETIA.ExtractTK}(\mathsf{MPK}, \mathsf{MTK}, \mathsf{ID})$, and output $(\mathsf{sk}_{\mathsf{ID}}, \mathsf{tok}_{\mathsf{ID}})$.

We give our IBE construction as follows. Here, the test functionality of $\mathsf{IBEETIA}$ is not used. Importantly, IBE must support exponentially many identities, i.e., there are exponentially many public keys. If IBE supports only polynomially many identities, IBE can be constructed from PKE by assigning a public key of PKE to each identity. Due to the OW-TG-CPA security, an adversary is allowed to obtain MTK. This allows us to contain MTK to the master public key of IBE. Then, unlike to $\mathsf{IBEETIA}$, anyone can generate a ciphertext by internally computing $\mathsf{tok}_{\mathsf{ID}}$ using MTK and by running the $\mathsf{IBEETIA.Enc}$ algorithm. Thus, as in IBE, exponentially many identities are supported. This publicly-executable encryption algorithm is mandatory to construct IBE.

$\mathsf{IBE.Setup}(1^\lambda)$**:** Run $(\mathsf{MPK}', \mathsf{MSK}, \mathsf{MTK}) \leftarrow \mathsf{IBEETIA.Setup}(1^\lambda)$ and output $\mathsf{MPK} = (\mathsf{MPK}', \mathsf{MTK})$ and MSK.

$\mathsf{IBE.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{ID})$**:** Parse $\mathsf{MPK} = (\mathsf{MPK}', \mathsf{MTK})$. Run $\mathsf{sk}_{\mathsf{ID}} \leftarrow \mathsf{IBEETIA.ExtractSK}(\mathsf{MPK}', \mathsf{MSK}, \mathsf{ID})$ and output $\mathsf{sk}_{\mathsf{ID}}$.

$\mathsf{IBE.Enc}(\mathsf{MPK}, \mathsf{ID}, M)$**:** Parse $\mathsf{MPK} = (\mathsf{MPK}', \mathsf{MTK})$. Run $\mathsf{tok}_{\mathsf{ID}} \leftarrow \mathsf{IBEETIA.ExtractTK}(\mathsf{MPK}, \mathsf{MTK}, \mathsf{ID})$ and $\mathsf{ct}_{\mathsf{IBEETIA}} \leftarrow \mathsf{IBEETIA.Enc}(\mathsf{MPK}', \mathsf{tok}_{\mathsf{ID}}, \mathsf{ID}, M)$. Output $\mathsf{ct}_{\mathsf{IBE}} = \mathsf{ct}_{\mathsf{IBEETIA}}$.

$\mathsf{IBE.Dec}(\mathsf{MPK}, \mathsf{ct}_{\mathsf{IBE}}, \mathsf{sk}_{\mathsf{ID}})$**:** Parse $\mathsf{MPK} = (\mathsf{MPK}', \mathsf{MTK})$ and $\mathsf{ct}_{\mathsf{IBE}} = \mathsf{ct}_{\mathsf{IBEETIA}}$. Run $\mathsf{tok}_{\mathsf{ID}} \leftarrow \mathsf{IBEETIA.ExtractTK}(\mathsf{MPK}, \mathsf{MTK}, \mathsf{ID})$. Output the result of $\mathsf{IBEETIA.Dec}(\mathsf{MPK}', \mathsf{sk}_{\mathsf{ID}}, \mathsf{tok}_{\mathsf{ID}}, \mathsf{ct}_{\mathsf{IBEETIA}})$.

Correctness of the IBE scheme is directly followed by the first condition of the correctness of the underlying IBEETIA scheme.

Theorem 1. *The proposed* IBE *scheme is OW-CPA secure if the underlying* IBEETIA *is OW-TG-CPA secure.*

Proof. Let $\mathcal{A}$ be the adversary of the OW-CPA security of IBE and $\mathcal{C}$ be the challenger of the OW-TG-CPA security of IBEETIA. We construct an algorithm $\mathcal{B}$ that breaks the OW-TG-CPA security using $\mathcal{A}$ as follows.

First, $\mathcal{C}$ runs $(\mathsf{MPK}', \mathsf{MSK}, \mathsf{MTK}) \leftarrow \mathsf{IBEETIA.Setup}(1^\lambda)$ and sends $(\mathsf{MPK}', \mathsf{MTK})$ to $\mathcal{B}$. $\mathcal{B}$ sets $\mathsf{MPK} = (\mathsf{MPK}', \mathsf{MTK})$ and sends MPK to $\mathcal{A}$. When $\mathcal{A}$ issues a key extraction query ID, $\mathcal{B}$ forwards ID to $\mathcal{C}$. $\mathcal{C}$ runs $(\mathsf{sk}_{\mathsf{ID}}, \mathsf{tok}_{\mathsf{ID}}) \leftarrow \mathsf{IBEETIA.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}, \mathsf{ID})$ and sends $\mathsf{sk}_{\mathsf{ID}}$ to $\mathcal{B}$. In the challenge phase, $\mathcal{A}$ declares ID^*. $\mathcal{B}$ sends ID^* to $\mathcal{C}$. $\mathcal{C}$ randomly chooses $M^* \leftarrow \mathcal{M}$, runs $(\mathsf{sk}_{\mathsf{ID}^*}, \mathsf{tok}_{\mathsf{ID}^*}) \leftarrow \mathsf{IBEETIA.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}, \mathsf{ID}^*)$, $\mathsf{ct}^*_{\mathsf{IBEETIA}} \leftarrow \mathsf{IBEETIA.Enc}(\mathsf{MPK}, \mathsf{tok}_{\mathsf{ID}^*}, \mathsf{ID}^*, M^*)$, and sends $\mathsf{ct}^*_{\mathsf{IBEETIA}}$ to $\mathcal{B}$. $\mathcal{B}$ sets $\mathsf{ct}^*_{\mathsf{IBE}} = \mathsf{ct}^*_{\mathsf{IBEETIA}}$ and sends $\mathsf{ct}^*_{\mathsf{IBE}}$ to $\mathcal{A}$. Finally, $\mathcal{A}$ outputs $\hat{M}$ and $\mathcal{B}$ outputs the same $\hat{M}$. Then, $\mathcal{B}$ breaks the OW-TG-CPA security with the advantage at least $\mathsf{Adv}^{\text{OW-CPA}}_{\mathsf{IBE},\mathcal{A}}(1^\lambda)$. □

5 Proposed Generic Construction of IBEETIA

In this section, we propose a generic construction of IBEETIA that provides the OW-TH-CCA security, in addition to the wIND-CCA security and correctness. The proposed construction basically follows the Emura-Takayasu construction, except that a PKE scheme is employed. Unlike to the OW-TG-CCA security, an adversary is not allowed to obtain MTK, and is allowed to issue token queries. Because the number of tokens is bounded by a polynomial of the security parameter and the encryption algorithm takes a token $\mathsf{tok}_{\mathsf{ID}}$ as input, the number of identities are also bounded by a polynomial of the security parameter. Thus, there is room for constructing an OW-TG-CCA secure IBEETIA scheme from cryptographic primitives which are weaker than IBE.

5.1 Emura-Takayasu IBEETIA Construction

Before giving the proposed construction, we revisit the Emura-Taykayasu construction [14]. Let π be a pseudo-random permutation with a key space $\mathcal{K}$ and $\mathsf{SKE} = (\mathsf{SKE.KeyGen}, \mathsf{SKE.Enc}, \mathsf{SKE.Dec})$ be a CCA-secure SKE scheme. We denote $\mathcal{R}$ as a randomness space for key generation. That is, we denote $\mathsf{SKE.sk} \leftarrow \mathsf{SKE.KeyGen}(1^\lambda; r)$ for $r \xleftarrow{\$} \mathcal{R}$.

$\mathsf{IBEETIA.Setup}(1^\lambda)$**:** Choose $k \xleftarrow{\$} \mathcal{K}$. Output $\mathsf{MPK} = \perp$, $\mathsf{MSK} = \perp$, and $\mathsf{MTK} = k$.

$\mathsf{IBEETIA.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}, \mathsf{ID})$**:** Parse $\mathsf{MTK} = k$. Choose $r_{\mathsf{ID}} \xleftarrow{\$} \mathcal{R}$ and $\mathsf{SKE.sk}_{\mathsf{ID}} \leftarrow \mathsf{SKE.KeyGen}(1^\lambda; r_{\mathsf{ID}})$ and output $\mathsf{sk}_{\mathsf{ID}} = \perp$ and $\mathsf{tok}_{\mathsf{ID}} = (\mathsf{SKE.sk}_{\mathsf{ID}}, k)$.

$\mathsf{IBEETIA.Enc}(\mathsf{MPK}, \mathsf{tok}_{\mathsf{ID}}, \mathsf{ID}, M)$**:** Parse $\mathsf{tok}_{\mathsf{ID}} = (\mathsf{SKE.sk}_{\mathsf{ID}}, k)$. Run $x_M \leftarrow \pi(k, M)$ and $\mathsf{ct}_{\mathsf{SKE}} \leftarrow \mathsf{SKE.Enc}(\mathsf{SKE.sk}_{\mathsf{ID}}, M)$ and outputs $\mathsf{ct}_{\mathsf{IBEETIA}} = (x_M, \mathsf{ct}_{\mathsf{SKE}})$.

$\mathsf{IBEETIA.Dec}(\mathsf{MPK}, \mathsf{sk}_{\mathsf{ID}}, \mathsf{tok}_{\mathsf{ID}}, \mathsf{ct}_{\mathsf{IBEETIA}})$**:** Parse $\mathsf{tok}_{\mathsf{ID}} = (\mathsf{SKE.sk}_{\mathsf{ID}}, k)$ and $\mathsf{ct}_{\mathsf{IBEETIA}} = (x_M, \mathsf{ct}_{\mathsf{SKE}})$. Run $M' \leftarrow \mathsf{SKE.Dec}(\mathsf{SKE.sk}_{\mathsf{ID}}, \mathsf{ct}_{\mathsf{SKE}})$ and $x_{M'} \leftarrow \pi(k, M')$. If $x_M = x_{M'}$, then output M', and $\perp$ otherwise.

$\mathsf{IBEETIA.Test}(\mathsf{MPK}, \mathsf{ct}_{\mathsf{IBEETIA}}, \mathsf{ct}'_{\mathsf{IBEETIA}})$**:** Parse $\mathsf{ct}_{\mathsf{IBEETIA}} = (x_M, \mathsf{ct}_{\mathsf{SKE}})$ and $\mathsf{ct}'_{\mathsf{IBEETIA}} = (x_{M'}, \mathsf{ct}'_{\mathsf{SKE}})$. If $x_M = x_{M'}$, then output 1, and 0 otherwise.

Since π is permutation, if $M = M'$, then $x_M = x_{M'}$ holds and if $M \neq M'$, then $x_M \neq x_{M'}$ holds. Moreover, the IBEETIA construction is wIND-CCA secure if π is a pseudo-random permutation and SKE is CCA secure. Intuitively, if $\mathsf{MTK} = k$ is hidden, then x_M is indistinguishable from a value generated by a random function due to the pseudo-randomness. We remark that k is not revealed in the definition of the wIND-CCA security. Moreover, no information of M is revealed from $\mathsf{ct}_{\mathsf{SKE}}$ due to the IND-CCA security of the SKE scheme. We briefly explain how to prove that the construction is wIND-CCA secure as follows. In the security proof, a simulator $\mathcal{B}$ obtains the challenge ciphertext $\mathsf{ct}^*_{\mathsf{SKE}}$ from the challenger of the SKE scheme. Then $\mathcal{B}$ randomly chooses $x_{M_b^*}$ from $\mathcal{M}$ and sets $(x_{M_b^*}, \mathsf{ct}^*_{\mathsf{SKE}})$ as the challenge ciphertext of IBEETIA. One may wonder how to respond a decryption query $(x_M, \mathsf{ct}^*_{\mathsf{SKE}}) \wedge x_M \neq x_{M_b^*}$ since $\mathcal{B}$ is not allowed to issue a decryption query $\mathsf{ct}^*_{\mathsf{SKE}}$ to $\mathcal{C}$. Because π is permutation, there is only one valid $x_{M_b^*}$. Thus, $\mathcal{B}$ simply returns $\perp$ if $(x_M, \mathsf{ct}^*_{\mathsf{SKE}}) \wedge x_M \neq x_{M_b^*}$.

Obviously, the Emura-Takayasu construction does not provide the OW-TH-CPA security because $\mathsf{tok}_{\mathsf{ID}} = (\mathsf{SKE.sk}, k)$ and revealing $\mathsf{tok}_{\mathsf{ID}}$ immediately breaks the security of the underlying SKE scheme. In other words, symmetric key primitives are sufficient if no security against token holders is required.

5.2 Proposed Construction

We tweak the Emura-Takayasu construction to provide the OW-TH-CCA security. Basically, we employ a PKE scheme, and set $\mathsf{sk}_{\mathsf{ID}} = \mathsf{PKE.dk}$ and $\mathsf{tok}_{\mathsf{ID}} = (\mathsf{PKE.pk}, k)$. Then, revealing PKE.pk does not affect the security of PKE. We need to care the fact that no security of π can be assumed if k is revealed. Especially, M may be recovered from $\pi(k, M)$ without contradicting the security of π. For example, when a block cipher is employed as π, revealing the key k immediately recovers M. As the first attempt, we employ a hash function H and set $x_M \leftarrow \pi(k, H(M))$. Due to the collision resistance of H, if $M \neq M'$, then $H(M) \neq H(M')$. Thus, employing H does not affect the correctness. Moreover, due to the one-wayness of H, M is not recovered from $H(M)$. Informally, for $M^* \xleftarrow{\$} \mathcal{M}$, the challenge ciphertext is $(x^*, \mathsf{ct}^*_{\mathsf{PKE}})$ where $x^* = \pi(k, H(M^*))$ and $\mathsf{ct}^*_{\mathsf{PKE}} \leftarrow \mathsf{PKE.Enc}(\mathsf{PKE.pk}, M^*)$. First, we replace $\mathsf{ct}^*_{\mathsf{PKE}}$ such that $\mathsf{ct}^*_{\mathsf{PKE}} \leftarrow \mathsf{PKE.Enc}(\mathsf{PKE.pk}, 0^{|M^*|})$ due to the IND-CCA security of PKE, and second we break the one-wayness of H using the output of the OW-TH-CCA adversary. Here, we need to consider that usually PKE does not hide the plaintext size. Thus, if we set $x^* = \pi(k, H(M))$ where $H(M)$ is given from the challenger of the one-wayness of H, the simulator has no way to generate $\mathsf{ct}^*_{\mathsf{PKE}} \leftarrow \mathsf{PKE.Enc}(\mathsf{PKE.pk}, 0^{|M|})$. Thus, we assume that the size of all plaintexts are the same, e.g., by adding a padding to each plaintext. We further need to consider the case that an adversary sends $(x_M, \mathsf{ct}^*_{\mathsf{PKE}})$ where $x^* \neq x_M$. Since H is deterministic, $H(M)$ is uniquely determined when M is fixed. Thus, we can reject a decryption query $(x_M, \mathsf{ct}^*_{\mathsf{PKE}})$ if $x^* \neq x_M$.

Let $H : \mathcal{M} \to \{0,1\}^\lambda$ be a hash function and $\mathsf{PKE} = (\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ be a CCA-secure PKE scheme. We denote $\mathcal{R}$ as a randomness space for key generation. That is, we denote $(\mathsf{PKE.pk}, \mathsf{PKE.dk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda; r)$ for $r \xleftarrow{\$} \mathcal{R}$.

$\mathsf{IBEETIA.Setup}(1^\lambda)$: Specify a hash function H and choose $k \xleftarrow{\$} \mathcal{K}$. Output $\mathsf{MPK} = H$, $\mathsf{MSK} = \perp$, and $\mathsf{MTK} = k$.

$\mathsf{IBEETIA.Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{MTK}, \mathsf{ID})$: Parse $\mathsf{MTK} = k$. Choose $r_{\mathsf{ID}} \xleftarrow{\$} \mathcal{R}$ and $(\mathsf{PKE.pk}_{\mathsf{ID}}, \mathsf{PKE.dk}_{\mathsf{ID}}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda; r_{\mathsf{ID}})$ and output $\mathsf{sk}_{\mathsf{ID}} = \mathsf{PKE.dk}_{\mathsf{ID}}$ and $\mathsf{tok}_{\mathsf{ID}} = (\mathsf{PKE.pk}_{\mathsf{ID}}, k)$.

$\mathsf{IBEETIA.Enc}(\mathsf{MPK}, \mathsf{tok}_{\mathsf{ID}}, \mathsf{ID}, M)$: Parse $\mathsf{MPK} = H$ and $\mathsf{tok}_{\mathsf{ID}} = (\mathsf{PKE.pk}_{\mathsf{ID}}, k)$. Run $x_M \leftarrow \pi(k, H(M))$ and $\mathsf{ct}_{\mathsf{PKE}} \leftarrow \mathsf{PKE.Enc}(\mathsf{PKE.pk}_{\mathsf{ID}}, M)$ and outputs $\mathsf{ct}_{\mathsf{IBEETIA}} = (x_M, \mathsf{ct}_{\mathsf{PKE}})$.

$\mathsf{IBEETIA.Dec}(\mathsf{MPK}, \mathsf{sk}_{\mathsf{ID}}, \mathsf{tok}_{\mathsf{ID}}, \mathsf{ct}_{\mathsf{IBEETIA}})$: Parse $\mathsf{MPK} = H$, $\mathsf{sk}_{\mathsf{ID}} = \mathsf{PKE.dk}_{\mathsf{ID}}$, $\mathsf{tok}_{\mathsf{ID}} = (\mathsf{PKE.pk}_{\mathsf{ID}}, k)$, and $\mathsf{ct}_{\mathsf{IBEETIA}} = (x_M, \mathsf{ct}_{\mathsf{PKE}})$. Run $M' \leftarrow \mathsf{PKE.Dec}(\mathsf{PKE.dk}_{\mathsf{ID}}, \mathsf{ct}_{\mathsf{PKE}})$ and $x_{M'} \leftarrow \pi(k, H(M'))$. If $x_M = x_{M'}$, then output M', and $\perp$ otherwise.

$\mathsf{IBEETIA.Test}(\mathsf{MPK}, \mathsf{ct}_{\mathsf{IBEETIA}}, \mathsf{ct}'_{\mathsf{IBEETIA}})$: Parse $\mathsf{ct}_{\mathsf{IBEETIA}} = (x_M, \mathsf{ct}_{\mathsf{PKE}})$ and $\mathsf{ct}'_{\mathsf{IBEETIA}} = (x_{M'}, \mathsf{ct}'_{\mathsf{PKE}})$. If $x_M = x_{M'}$, then output 1, and 0 otherwise.

Due to the correctness of the underlying PKE scheme, π is a permutation, and H is a deterministic hash function, the first and second conditions of the correctness hold. Since π is a permutation, and H is collision resistant, $x_M \neq x_{M'}$ holds if $M \neq M'$. Thus, the third condition of the correctness holds.

Theorem 2. *The proposed* $\mathsf{IBEETIA}$ *construction is wIND-CCA secure if the underlying* PKE *is IND-CCA secure and* π *is a pseudo-random permutation.*

Proof. The proof proceeds with the following sequence of games.

Game_0: This game is a real wIND-CCA security game. Queries issued by $\mathcal{A}$ is answered as follows.

Key Extraction: $\mathcal{A}$ sends ID to $\mathcal{C}$. If ID has been issued as a query, $\mathcal{C}$ retrieves r_{ID}. Otherwise, $\mathcal{C}$ chooses $r_{\mathsf{ID}} \xleftarrow{\$} \mathcal{R}$ and preserves $(\mathsf{ID}, r_{\mathsf{ID}})$ locally. $\mathcal{C}$ runs $(\mathsf{PKE.pk}_{\mathsf{ID}}, \mathsf{PKE.dk}_{\mathsf{ID}}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda; r_{\mathsf{ID}})$ and returns $\mathsf{sk}_{\mathsf{ID}} = \mathsf{PKE.dk}_{\mathsf{ID}}$ to $\mathcal{A}$.

Encryption: $\mathcal{A}$ sends (ID, M) to $\mathcal{C}$. If ID has been issued as a query, $\mathcal{C}$ retrieves r_{ID} and preserves $(\mathsf{ID}, r_{\mathsf{ID}})$ locally. Otherwise, $\mathcal{C}$ chooses $r_{\mathsf{ID}} \xleftarrow{\$} \mathcal{R}$. $\mathcal{C}$ runs $(\mathsf{PKE.pk}_{\mathsf{ID}}, \mathsf{PKE.dk}_{\mathsf{ID}}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda; r_{\mathsf{ID}})$, runs $x_M \leftarrow \pi(k, H(M))$ and $\mathsf{ct}_{\mathsf{PKE}} \leftarrow \mathsf{PKE.Enc}(\mathsf{PKE.pk}_{\mathsf{ID}}, M)$, and returns $\mathsf{ct}_{\mathsf{IBEETIA}} = (x_M, \mathsf{ct}_{\mathsf{PKE}})$ to $\mathcal{A}$.

Decryption: $\mathcal{A}$ sends $(\mathsf{ID}, \mathsf{ct}_{\mathsf{IBEETIA}})$ to $\mathcal{C}$. If ID has been issued as a query, $\mathcal{C}$ retrieves r_{ID} and preserves $(\mathsf{ID}, r_{\mathsf{ID}})$ locally. Otherwise, $\mathcal{C}$ chooses $r_{\mathsf{ID}} \xleftarrow{\$} \mathcal{R}$. $\mathcal{C}$ runs $(\mathsf{PKE.pk}_{\mathsf{ID}}, \mathsf{PKE.dk}_{\mathsf{ID}}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda; r_{\mathsf{ID}})$. $\mathcal{C}$ runs $M' \leftarrow \mathsf{PKE.Dec}(\mathsf{PKE.dk}_{\mathsf{ID}}, \mathsf{ct}_{\mathsf{PKE}})$ and $x_{M'} \leftarrow \pi(k, H(M'))$. If $x_M = x_{M'}$, then return M', and $\perp$ otherwise.

Game_1: This game is the same as Game_0 except that $\mathcal{C}$ replaces the permutation π with a random function.

Game_2: This game is the same as Game_1 except that $\mathcal{C}$ randomly chooses $x_M \xleftarrow{\$} \mathcal{M}$ instead of using the random function and stores (M, x_M) locally. If the same M is queried, then $\mathcal{C}$ retrieves x_M.

Game_3: Let q_{ID} denote the number of distinct identities that are chosen when the $\mathcal{A}$ queries the oracles, and $r_1, r_2, \ldots, r_{q_{\mathsf{ID}}}$ are random coins for running the $\mathsf{PKE.KeyGen}$ algorithm. This game is the same as Game_2 except that $\mathcal{C}$ randomly chooses $q^* \xleftarrow{\$} \{1, 2, \ldots, q_{\mathsf{ID}}\}$ at the beginning of the game, and aborts the game if there exists $i \in \{1, 2, \ldots, q_{\mathsf{ID}}\} \setminus \{q^*\}$ such that $r_{q^*} = r_i$ holds.

By Lemma 1 in [14], Game_0 and Game_1 are computationally indistinguishable if π is a pseudo-random. By Lemma 2 in [14], Game_1 and Game_2 are statistically indistinguishable. By Lemma 3 in [14], Game_2 and Game_3 are statistically indistinguishable.

Lemma 1 ($\mathcal{A}$'s advantage in Game_3)**.** *For any PPT adversary $\mathcal{A}$, there exists a reduction algorithm $\mathcal{B}$ for breaking the IND-CCA security of* PKE*.*

Proof. We show that $\mathcal{A}$'s advantage in Game_3 is negligible. Concretely, we construct an algorithm $\mathcal{B}$ that breaks the IND-CCA security of the PKE scheme. Let $\mathcal{C}$ be the IND-CCA challenger. First, $\mathcal{C}$ runs $(\mathsf{PKE.pk}, \mathsf{PKE.dk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$ and sends $\mathsf{PKE.pk}$ to $\mathcal{B}$. $\mathcal{B}$ specifies a has function H and chooses $k \xleftarrow{\$} \mathcal{K}$. $\mathcal{B}$ sends $\mathsf{MPK} = H$ to $\mathcal{A}$. During the game, $\mathcal{B}$ aborts if $\mathsf{ID}_{q^*} \neq \mathsf{ID}^*$. From now on, we assume that $\mathcal{B}$'s guess is correct.

$\mathcal{B}$ responds $\mathcal{A}$'s queries as follows.

Key Extraction: $\mathcal{A}$ sends $\mathsf{ID} \neq \mathsf{ID}^*$ to $\mathcal{B}$. If ID has been issued as a query, $\mathcal{B}$ retrieves r_{ID}. Otherwise, $\mathcal{B}$ chooses $r_{\mathsf{ID}} \xleftarrow{\$} \mathcal{R}$ and preserves $(\mathsf{ID}, r_{\mathsf{ID}})$ locally. $\mathcal{B}$ runs $(\mathsf{PKE.pk}_{\mathsf{ID}}, \mathsf{PKE.dk}_{\mathsf{ID}}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda; r_{\mathsf{ID}})$ and returns $\mathsf{sk}_{\mathsf{ID}} = \mathsf{PKE.dk}_{\mathsf{ID}}$ to $\mathcal{A}$.

Encryption: $\mathcal{A}$ sends (ID, M) to $\mathcal{B}$. $\mathcal{B}$ randomly chooses $x_M \xleftarrow{\$} \mathcal{M}$ if (M, x_M) is not stored locally. Otherwise, $\mathcal{B}$ retrieves x_M and preserves (M, x_M) locally.

- If $\mathsf{ID} \neq \mathsf{ID}^*$ (i.e., this is the i-th query where $i \neq q^*$), if ID has been issued as a query, $\mathcal{B}$ retrieves r_{ID}. Otherwise, $\mathcal{B}$ chooses $r_{\mathsf{ID}} \xleftarrow{\$} \mathcal{R}$ and preserves $(\mathsf{ID}, r_{\mathsf{ID}})$ locally. $\mathcal{B}$ runs $(\mathsf{PKE.pk}_{\mathsf{ID}}, \mathsf{PKE.dk}_{\mathsf{ID}}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda; r_{\mathsf{ID}})$, runs $\mathsf{ct}_{\mathsf{PKE}} \leftarrow \mathsf{PKE.Enc}(\mathsf{PKE.pk}_{\mathsf{ID}}, M)$, and returns $\mathsf{ct}_{\mathsf{IBEETIA}} = (x_M, \mathsf{ct}_{\mathsf{PKE}})$ to $\mathcal{A}$.
- If $\mathsf{ID} = \mathsf{ID}^*$ (i.e., this is the q^*-th query), then $\mathcal{B}$ runs $\mathsf{ct}_{\mathsf{PKE}} \leftarrow \mathsf{PKE.Enc}(\mathsf{PKE.pk}, M)$, and returns $\mathsf{ct}_{\mathsf{IBEETIA}} = (x_M, \mathsf{ct}_{\mathsf{PKE}})$ to $\mathcal{A}$.

Decryption: $\mathcal{A}$ sends $(\mathsf{ID}, \mathsf{ct}_{\mathsf{IBEETIA}})$ to $\mathcal{B}$. Parse $\mathsf{ct}_{\mathsf{IBEETIA}} = (x_M, \mathsf{ct}_{\mathsf{PKE}})$.

- If $\mathsf{ID} \neq \mathsf{ID}^*$ (i.e., this is the i-th query where $i \neq q^*$), if ID has been issued as a query, $\mathcal{B}$ retrieves r_{ID}. Otherwise, $\mathcal{B}$ chooses $r_{\mathsf{ID}} \xleftarrow{\$} \mathcal{R}$ and preserves $(\mathsf{ID}, r_{\mathsf{ID}})$ locally. $\mathcal{B}$ runs $M' \leftarrow \mathsf{PKE.Dec}(\mathsf{PKE.dk}_{\mathsf{ID}}, \mathsf{ct}_{\mathsf{PKE}})$. $\mathcal{B}$ randomly chooses $x_{M'} \xleftarrow{\$} \mathcal{M}$ and preserves $(M', x_{M'})$ locally if (M', x_M) is not stored locally. Otherwise, $\mathcal{B}$ retrieves x_M, and returns M' if $x_M = x_{M'}$, and $\perp$ otherwise.
- If $\mathsf{ID} = \mathsf{ID}^*$ (i.e., this is the q^*-th query), $\mathcal{B}$ sends $\mathsf{ct}_{\mathsf{PKE}}$ to $\mathcal{C}$ as a decryption query, and obtains M'. $\mathcal{B}$ randomly chooses $x_{M'} \xleftarrow{\$} \mathcal{M}$ and preserves $(M', x_{M'})$ locally if (M', x_M) is not stored locally. Otherwise, $\mathcal{B}$ retrieves x_M and returns M' if $x_M = x_{M'}$, and $\perp$ otherwise.

When $\mathcal{A}$ declares $(\mathsf{ID}^*, M_0^*, M_1^*)$, then $\mathcal{B}$ sends (M_0^*, M_1^*) to $\mathcal{C}$. $\mathcal{C}$ randomly chooses $b \xleftarrow{\$} \{0,1\}$, computes $\mathsf{ct}_{\mathsf{PKE}}^* \leftarrow \mathsf{PKE.Enc}(\mathsf{PKE.pk}, M_b^*)$, and sends $\mathsf{ct}_{\mathsf{PKE}}^*$ to $\mathcal{B}$. $\mathcal{B}$ randomly chooses $x^* \xleftarrow{\$} \mathcal{M}$ and sends $\mathsf{ct}_{\mathsf{IBEETIA}}^* = (x^*, \mathsf{ct}_{\mathsf{PKE}}^*)$ to $\mathcal{A}$.

$\mathcal{B}$ responds $\mathcal{A}$'s queries as follows.

Key Extraction: $\mathcal{B}$ responds the query as in the pre-challenge phase.

Encryption: $\mathcal{B}$ responds the query as in the pre-challenge phase.

Decryption: $\mathcal{A}$ sends $(\mathsf{ID}, \mathsf{ct}_{\mathsf{IBEETIA}})$ to $\mathcal{B}$. Parse $\mathsf{ct}_{\mathsf{IBEETIA}} = (x_M, \mathsf{ct}_{\mathsf{PKE}})$.

- If $\mathsf{ID} \neq \mathsf{ID}^*$ (i.e., this is the i-th query where $i \neq q^*$), $\mathcal{B}$ responds the query as in the pre-challenge phase.
- If $\mathsf{ID} = \mathsf{ID}^*$ (i.e., this is the q^*-th query), if $\mathsf{ct}_{\mathsf{PKE}} \neq \mathsf{ct}^*_{\mathsf{PKE}}$, then $\mathcal{B}$ sends $\mathsf{ct}_{\mathsf{PKE}}$ to $\mathcal{C}$ as a decryption query, and obtains M'. $\mathcal{B}$ randomly chooses $x_{M'} \xleftarrow{\$} \mathcal{M}$ and preserves $(M', x_{M'})$ locally if (M', x_M) is not stored locally. Otherwise, $\mathcal{B}$ retrieves x_M and returns M' if $x_M = x_{M'}$, and $\perp$ otherwise. If $\mathsf{ct}_{\mathsf{PKE}} = \mathsf{ct}^*_{\mathsf{PKE}}$ (and then $x_M \neq x^*$), then $\mathcal{B}$ returns $\perp$ because x^* is deterministically fixed when the challenge plaintext is fixed and thus $(x_M, \mathsf{ct}^*_{\mathsf{PKE}})$ is an invalid ciphertext.

Finally, $\mathcal{A}$ outputs a bit b'. $\mathcal{B}$ outputs the same b'. If the guess q^* is correct (with the probability at least $1/(q^{\mathsf{ext}} + q^{\mathsf{enc}} + q^{\mathsf{dec}})$ where q^{ext}, q^{enc}, and q^{dec} are the number of key extraction, encryption, and decryption queries, respectively), then $\mathcal{B}$'s simulation is perfect and $\mathcal{B}$ can break the IND-CCA security. □

This concludes the proof of Theorem 2. □

Theorem 3. *The proposed* IBEETIA *construction is OW-TH-CCA secure if the underlying* PKE *is IND-CCA secure and* H *is a one-way hash function.*

Proof.

Game_0**:** This game is a real OW-TH-CCA security game. Queries issued by $\mathcal{A}$ is answered as follows.

Key Extraction: $\mathcal{A}$ sends ID to $\mathcal{C}$. If ID has been issued as a query, $\mathcal{C}$ retrieves r_{ID}. Otherwise, $\mathcal{C}$ chooses $r_{\mathsf{ID}} \xleftarrow{\$} \mathcal{R}$ and preserves $(\mathsf{ID}, r_{\mathsf{ID}})$ locally. $\mathcal{C}$ runs $(\mathsf{PKE.pk}_{\mathsf{ID}}, \mathsf{PKE.dk}_{\mathsf{ID}}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda; r_{\mathsf{ID}})$ and returns $\mathsf{sk}_{\mathsf{ID}} = \mathsf{PKE.dk}_{\mathsf{ID}}$ to $\mathcal{A}$.

Token Extraction: $\mathcal{A}$ sends ID to $\mathcal{C}$. If ID has been issued as a query, $\mathcal{C}$ retrieves r_{ID}. Otherwise, $\mathcal{C}$ chooses $r_{\mathsf{ID}} \xleftarrow{\$} \mathcal{R}$ and preserves $(\mathsf{ID}, r_{\mathsf{ID}})$ locally. $\mathcal{C}$ runs $(\mathsf{PKE.pk}_{\mathsf{ID}}, \mathsf{PKE.dk}_{\mathsf{ID}}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda; r_{\mathsf{ID}})$ and returns $\mathsf{tok}_{\mathsf{ID}} = (\mathsf{PKE.pk}_{\mathsf{ID}}, k)$ to $\mathcal{A}$.

Decryption: $\mathcal{A}$ sends $(\mathsf{ID}, \mathsf{ct}_{\mathsf{IBEETIA}})$ to $\mathcal{C}$. If ID has been issued as a query, $\mathcal{C}$ retrieves r_{ID} and preserves $(\mathsf{ID}, r_{\mathsf{ID}})$ locally. Otherwise, $\mathcal{C}$ chooses $r_{\mathsf{ID}} \xleftarrow{\$} \mathcal{R}$. $\mathcal{C}$ runs $(\mathsf{PKE.pk}_{\mathsf{ID}}, \mathsf{PKE.dk}_{\mathsf{ID}}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda; r_{\mathsf{ID}})$. $\mathcal{C}$ runs $M' \leftarrow \mathsf{PKE.Dec}(\mathsf{PKE.dk}_{\mathsf{ID}}, \mathsf{ct}_{\mathsf{PKE}})$ and $x_{M'} \leftarrow \pi(k, H(M'))$. If $x_M = x_{M'}$, then return M', and $\perp$ otherwise.

Game_1**:** Let q_{ID} denote the number of distinct identities that are chosen when the $\mathcal{A}$ queries the oracles, and $r_1, r_2, \ldots, r_{q_{\mathsf{ID}}}$ are random coins for running the $\mathsf{PKE.KeyGen}$ algorithm. This game is the same as Game_0 except that $\mathcal{C}$ randomly chooses $q^* \xleftarrow{\$} \{1, 2, \ldots, q_{\mathsf{ID}}\}$ at the beginning of the game, and aborts the game if there exists $i \in \{1, 2, \ldots, q_{\mathsf{ID}}\} \setminus \{q^*\}$ such that $r_{q^*} = r_i$ holds.

Game_2**:** This game is the same as Game_1 except that $\mathcal{C}$ prepares the challenge ciphertext as follows. In the challenge phase, $\mathcal{A}$ declares ID^*. Let $\mathsf{PKE.pk}$ be the public key used in the q^*-th query. $\mathcal{C}$ randomly chooses $M^* \xleftarrow{\$} \mathcal{M}$, runs $\mathsf{ct}^*_{\mathsf{PKE}} \leftarrow \mathsf{PKE.Enc}(\mathsf{PKE.pk}_{\mathsf{ID}}, 0^{|M^*|})$, computes $x^* \leftarrow \pi(k, H(M^*))$, and sends $\mathsf{ct}^*_{\mathsf{IBEETIA}} = (x^*, \mathsf{ct}^*_{\mathsf{PKE}})$ to $\mathcal{A}$.

By Lemma 3 in [14], Game_0 and Game_1 are statistically indistinguishable.

Lemma 2 (Indistinguishability between Game_1 and Game_2)**.** *For any PPT adversary $\mathcal{A}$, there exists a reduction algorithm $\mathcal{B}_1$ for breaking the IND-CCA security of* PKE*.*

Proof Sketch. The proof is almost the same as that of Lemma 1, except that $\mathcal{A}$ declares ID^* and $\mathcal{B}_1$ sends $(\mathsf{ID}^*, M^*, 0^{|M^*|})$ to the IND-CCA chalelnger. If M^* is encrypted, then $\mathcal{B}_1$ simulates Game_1 and if 0^{M^*} is encrypted, then $\mathcal{B}_1$ simulates Game_2. □

Lemma 3 ($\mathcal{A}$'s advantage in Game_2)**.** *For any PPT adversary $\mathcal{A}$, there exists a reduction algorithm $\mathcal{B}_2$ for breaking the one-wayness of H.*

Proof. Let $\mathcal{C}$ be the challenger of the one-wayness of H. We construct the algorithm $\mathcal{B}_2$ as follows. First, $\mathcal{C}$ sends the description of H to $\mathcal{B}_2$. $\mathcal{B}_2$ sets $\mathsf{MPK} = H$ and sends MPK to $\mathcal{A}$. $\mathcal{B}_2$ prepares all public keys and decryption keys of PKE and thus $\mathcal{B}_2$ can respond to all queries. In the challenge phase, $\mathcal{B}_2$ randomly chooses $M^* \xleftarrow{\$} \mathcal{M}$ and runs $\mathsf{ct}^*_{\mathsf{PKE}} \leftarrow \mathsf{PKE.Enc}(\mathsf{PKE.pk}_{\mathsf{ID}}, 0^{|M^*|})$. $\mathcal{C}$ sends $H(M)$ to $\mathcal{B}_2$. if $H(M^*) = H(M)$, then $\mathcal{B}_2$ outputs M^*. Otherwise, $\mathcal{B}_2$ computes $x^* \leftarrow \pi(k, H(M))$ and sends $\mathsf{ct}^*_{\mathsf{IBEETIA}} = (x^*, \mathsf{ct}^*_{\mathsf{PKE}})$ to $\mathcal{A}$. We remark that, for M, which is not known by $\mathcal{B}_2$, $|M| = |M^*|$ holds because we assume that the size of all plaintexts are the same. Thus, $\mathcal{B}_2$ properly simulates the challenge ciphertext in Game_2. Finally, $\mathcal{A}$ outputs $\hat{M}$. $\mathcal{B}_2$ outputs the same $\hat{M}$ and breaks the one-wayness of H with the advantage of $\mathcal{A}$. □

This concludes the proof of Theorem 3. □

6 Discussion

In this section, we discuss the efficiency and security level of the proposed construction. We give the comparisons in Table 1. We can employ any CCA-secure PKE scheme, e.g., the Cramer-Shoup scheme [9] (which is IND-CCA secure under the decisional Diffie-Hellman assumption) or the CRYSTALS-Kyber key encapsulation mechanism [7] (which is IND-CCA secure under the learning with errors (LWE) assumption over module lattices) with an appropriate data decapsulation mechanism, and so on.

Of course, the Emura-Takayasu construction [14] is the most efficient since it employs only symmetric key primitives. However, it does not provide the OW-TH-CPA security.

The Lee et al. scheme [18] is wIND-CCA secure under the bilinear Diffie-Hellman (BDH) assumption in the random oracle model. The form of ciphertext is:

- $C_1 = F(K_1, H_1(M))$, $C_2 = g^r$, $C_3 = (M||r) \oplus H_2(T||C_2||e(P_{\mathsf{pub}}, H(\mathsf{ID}))^r)$

where F is a permutation, T is a MAC (message authentication code) on C_1 such that $T = \mathsf{MAC}(K_2, C_1)$, H, H_1, and H_2 are hash functions modeled as random oracles, e is a bilinear paring, and $\mathsf{tok}_{\mathsf{ID}} = (K_1, K_2)$. Because the MAC part has the role of rejecting an invalid decryption query, it is not clear whether the Lee et al. scheme provides the OW-TH-CCA security after K_2 is given to the adversary. However, the Lee et al. scheme is at least OW-TH-CPA secure due to the one-wayness of H_1 and the fact that $e(P_{\mathsf{pub}}, H(\mathsf{ID}))^r$ can be regarded as a solution of the BDH problem. In the Lee et al. scheme, $\mathsf{tok}_{\mathsf{ID}} = \mathsf{MTK}$. Thus, the Lee et al. scheme is OW-TG-CPA secure when it is OW-TH-CPA secure. Thus, we state that the Lee et al. scheme is OW-TG-CPA secure in Table 1. Due to our implication result, employing pairings in the Lee et al. scheme is reasonable. On the other hand, proposed construction does not employ parings, e.g., when the Cramer-Shoup scheme is employed. In this perspective, our constriction is more efficient than the Lee et al. scheme.

Table 1: Comparison: STD stands for standard model and $*$ stands for selective security.

Scheme	Tools/Assumptions	IND	OW	STD
Emura-Takayasu [14]	SKE	wIND-CCA	No	Yes
Lee et al. [18]	Pairing	wIND-CCA	OW-TG-CPA	No
Duong et al. [11]	LWE	wIND-CPA*	OW-TG-CPA*	Yes
Proposed Construction	PKE	wIND-CCA	OW-TH-CCA	Yes

The Duong et al. scheme [11] is wIND-CPA secure under the LWE assumption over integer lattices. The form of ciphertext for $M \in \{0,1\}^t$ for some $t \in \mathbb{N}$ is:

- $C_1 = T_{\mathbf{A}'}\mathbf{s}^\top + H(M||T_{\mathbf{A}'})$, C_2, and C_3.

where H is a one-way and collision-resistant hash function, $\mathsf{A}' \in \mathbb{Z}^{n\times m}$ is a matrix (we omit the explanations of n and m here), $T_{\mathbf{A}'}$ is the trapdoor, and $\mathsf{tok}_{\mathsf{ID}} = T_{\mathbf{A}'}$. (C_2, C_3) can be regarded as a ciphertext of the ABB IBE scheme [1], and is independent to A'. In the OW-TH-CPA security definition, the trapdoor $T_{\mathbf{A}'}$ is leaked. Thus, C_1 may leak $H(M||T_{\mathbf{A}'})$. Since the ABB IBE is selectively IND-CPA secure under the LWE assumption, the Duong et al. scheme provides the selective OW-TH-CPA security after $\mathsf{tok}_{\mathsf{ID}} = T_{\mathbf{A}'}$ is given to the adversary. In the Duong et al. scheme, $\mathsf{tok}_{\mathsf{ID}} = \mathsf{MTK}$. Thus, we state that the Duong et al. scheme is selectively OW-TG-CPA secure in Table 1. The proposed construction provides the wIND-CCA security which is stronger that the wIND-CPA security. That is, a post-quantum instantiation of the proposed construction, e.g., from the CRYSTALS-Kyber scheme, provides the first IBEETIA scheme which is wIND-CCA secure under a post-quantum complexity assumption.

7 Conclusion

In this paper, we introduced OW security notions, OW-TG-CPA and OW-TH-CCA/CPA, and demonstrated OW-TG-CPA secure IBEETIA implies IBE, and proposed a generic construction of OW-TH-CCA secure IBEETIA from PKE.

As an extension of IBEET, attribute-based encryption with equality test (ABEET) has been proposed [3,10,19–21,25]. However, ABEET against insider attacks (ABEETIA) has not been considered so far, to the best of our knowledge. Because ABE implies IBE [15], it would be interesting to explore whether a similar separation holds or not, i.e., whether OW-TH-CCA/CPA secure ABEETIA can be constructed from PKE/IBE or not.

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

References

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

[2] Kyoichi Asano, Keita Emura, and Atsushi Takayasu. More efficient adaptively secure lattice-based IBE with equality test in the standard model. In *ISC*, pages 75–83, 2022.

[3] Kyoichi Asano, Keita Emura, Atsushi Takayasu, and Yohei Watanabe. A generic construction of CCA-secure attribute-based encryption with equality test. In *ProvSec*, pages 3–19, 2022.

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

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

[6] Dan Boneh, Periklis A. Papakonstantinou, Charles Rackoff, Yevgeniy Vahlis, and Brent Waters. On the impossibility of basing identity based encryption on trapdoor permutations. In *IEEE FOCS*, pages 283–292, 2008.

[7] Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - kyber: A CCA-secure module-lattice-based KEM. In *EuroS&P*, pages 353–367. IEEE, 2018.

[8] Sherman S. M. Chow. Removing escrow from identity-based encryption. In *Public Key Cryptography*, pages 256–276, 2009.

[9] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO*, pages 13–25, 1998.

[10] Yuzhao Cui, Qiong Huang, Jianye Huang, Hongbo Li, and Guomin Yang. Outsourced ciphertext-policy attribute-based encryption with equality test. In *Inscrypt*, pages 448–467, 2018.

[11] Dung Hoang Duong, Huy Quoc Le, Partha Sarathi Roy, and Willy Susilo. Lattice-based IBE with equality test in standard model. In *ProvSec*, pages 19–40, 2019.

[12] Keita Emura, Shuichi Katsumata, and Yohei Watanabe. Identity-based encryption with security against the KGC: A formal model and its instantiation from lattices. In *ESORICS*, pages 113–133, 2019.

[13] Keita Emura, Shuichi Katsumata, and Yohei Watanabe. Identity-based encryption with security against the KGC: A formal model and its instantiations. *Theoretical Computer Science*, 900:97–119, 2022.

[14] Keita Emura and Atsushi Takayasu. A generic construction of CCA-secure identity-based encryption with equality test against insider attacks. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 106-A(3):193–202, 2023.

[15] Javier Herranz. Attribute-based encryption implies identity-based encryption. *IET Information Security*, 11(6):332–337, 2017.

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

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

[18] Hyung Tae Lee, Huaxiong Wang, and Kai Zhang. Security analysis and modification of ID-based encryption with equality test from ACISP 2017. In *ACISP*, pages 780–786, 2018.

[19] Cong Li, Qingni Shen, Zhikang Xie, Xinyu Feng, Yuejian Fang, and Zhonghai Wu. Large universe CCA2 CP-ABE with equality and validity test in the standard model. *The Computer Journal*, 64(4):509–533, 2021.

[20] Qiang Wang, Li Peng, Hu Xiong, Jianfei Sun, and Zhiguang Qin. Ciphertext-policy attribute-based encryption with delegated equality test in cloud computing. *IEEE Access*, 6:760–771, 2018.

[21] Yuanhao Wang, Yuzhao Cui, Qiong Huang, Hongbo Li, Jianye Huang, and Guomin Yang. Attribute-based equality test over encrypted data without random oracles. *IEEE Access*, 8:32891–32903, 2020.

[22] Tong Wu, Sha Ma, Yi Mu, and Shengke Zeng. ID-based encryption with equality test against insider attack. In *ACISP*, pages 168–183, 2017.

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

[24] Guomin Yang, Chik How Tan, Qiong Huang, and Duncan S. Wong. Probabilistic public key encryption with equality test. In *CT-RSA*, pages 119–131, 2010.

[25] Huijun Zhu, Licheng Wang, Haseeb Ahmad, and Xinxin Niu. Key-policy attribute-based encryption with equality test in cloud computing. *IEEE Access*, 5:20428–20439, 2017.