Information Sciences 373 (2016) 419–440

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Semi-generic construction of public key encryption and identity-based encryption with equality test

Hyung Tae Lee[a], San Ling[a], Jae Hong Seo[b,*], Huaxiong Wang[a]

[a] Division of Mathematical Sciences, School of Physical & Mathematical Sciences, Nanyang Technological University, Singapore
[b] Department of Mathematics, Myongji University, Republic of Korea

ARTICLE INFO

Article history:
Received 17 March 2016
Revised 29 August 2016
Accepted 5 September 2016
Available online 6 September 2016

Keywords:
Public key encryption
Identity-based encryption
Equality test
Random oracle model

ABSTRACT

Public key encryption with equality test (PKEET), which was first introduced by Yang et al. (CT-RSA, 2010), has various applications including facilitating keyword search on encrypted data and partitioning encrypted data on the cloud. It can be also applied to manage personal health records on the internet. For these reasons, there have been improvements on earlier PKEET schemes in terms of performance and functionality.

We present a *semi-generic* method for PKEET constructions, assuming only the existence of IND-CCA2 secure traditional public key encryption (PKE) schemes, the hardness of Computational Diffie-Hellman (CDH) problems, and random oracles. Our approach has several advantages; it enables us to understand requirements for the equality test functionality more clearly. Furthermore, our approach is quite general, in that if we change the underlying PKE scheme with the identity-based encryption (IBE) scheme (and we assume the hardness of Bilinear Diffie-Hellman problems instead of CDH), then we obtain the first IBE scheme with equality test (IBEET) satisfying analogous security arguments to those of PKEET. Although an IBEET construction was recently proposed, but we note that it satisfies only weak security requirements.

1. Introduction

Public key encryption with equality test (PKEET), which was first introduced by Yang et al. [19], is a public key encryption (PKE) scheme that supports the capability for testing equality between ciphertexts using different public keys as well as the same public key. This property can be applied to various scenarios in practice. In particular, it is very useful for managing outsourced databases in a secure way.

Let us consider the following scenario to elucidate an advantage of using PKEET in secure outsourced database applications. Suppose that each user stores his/her emails with an email service provider. In this case, we face with two seemingly conflicting requirements, protecting data privacy and managing stored data efficiently. For the former, storing emails in the encrypted form seems necessary. But, it precludes operations over stored data without decrypting it, in particular, keyword search over stored emails. To support it, the email service provider makes senders append encrypted keywords to an en-

* Corresponding author.
E-mail addresses: hyungtaelee@ntu.edu.sg (H.T. Lee), lingsan@ntu.edu.sg (S. Ling), jaehongseo@mju.ac.kr, jhsbhs@gmail.com (J.H. Seo), hxwang@ntu.edu.sg (H. Wang).

http://dx.doi.org/10.1016/j.ins.2016.09.013
0020-0255/

crypted email, and may check encrypted keywords to response to user's keyword search queries or to filter out spam emails. However, traditional PKE schemes do not allow such operations to be performed over encrypted data.

Fully homomorphic encryption (FHE) [8] could be a suitable candidate to resolve the above issue, but the service provider cannot also check the result of operations without decrypting it. Searchable encryption [3] or deterministic encryption [2] could also be utilized. However, these primitives are basically designed to perform tests on ciphertexts generated by the same public key. Hence, the email service provider in the above scenario has to generate a token for each user in the system to monitor stored emails. On the other hand, a goal of PKEET is to enable one who has trapdoors to check equality among ciphertexts generated by different public keys as well as the same public key, so that the email service provider can perform tests on ciphertexts regardless of exploited public keys.

Furthermore, as suggested by Tang [18], PKEET can also be applied to emerging computing scenarios, e.g., internet-based private health record (PHR) applications [15,18]. In a PHR system, each patient may obtain his/her data from various sources: prescription results from a doctor, treatment from a hospital, test results from a laboratory, and so on. The patient receives such data as encrypted using his/her own public key, and stores them with the service provider. When he/she wants to match his/her data with that of others in order to get some help, he/she requests the service provider to search for them over encrypted data using different public keys. Due to its various applications as above, many researchers have developed PKEET schemes [9,11–13,16–18] for the purpose of achieving better performance and providing different levels of authorities for equality testing.

1.1. Our contribution

We provide a semi-generic PKEET construction that exploits traditional PKE schemes having sufficiently large plaintext spaces. Our PKEET system model follows that of Tang's all-or-nothing PKEET scheme [18]. In this scheme, each user issues a trapdoor to a specified tester and the authority to test the equality of all of his/her ciphertexts. Thus, the tester who has knowledge of the two user's trapdoors, is able to check the equality of ciphertexts using their public keys. We note that this model can also be regarded as a PKEET scheme supporting flexible authorization, where we only consider the authorization for equality test on all receiver's ciphertexts (so called Type-1 authorization in [12]).

In Section 1 of [18], Tang initially attempted to construct a generic PKEET scheme by defining an encryption algorithm for a message m by:

$$(\mathsf{C}_1, \mathsf{C}_2) = (\mathsf{PKE1}(pk_1, m), \mathsf{PKE2}(pk_2, \mathcal{H}_1(m)))$$

where PKE1 and PKE2 are traditional PKE schemes and $\mathcal{H}_1$ is a cryptographic hash function. Then, each user issues the secret key sk_2 for PKE2 to the tester and he/she can check their equality by decrypting the C_2's for both ciphertexts and then comparing $\mathcal{H}_1(m)$ values. Immediately, however, Tang demonstrated that the above formulation could not achieve the IND-CCA2 security [18] because an adversary could query

$$(\mathsf{C}_1^*, \mathsf{C}_2) = (\mathsf{PKE1}(pk_1, m_b), \mathsf{PKE2}(pk_2, \mathcal{H}_1(m_\beta)))$$

to the decryption oracle by guessing $b \in \{0, 1\}$, chosen by the challenger as β, and then generate the second component C_2 themselves where $(\mathsf{C}_1^*, \mathsf{C}_2^*)$ is the challenge ciphertext.

We resolve the above problem by providing a way to prevent decryption queries of the obtained ciphertexts by modifying the challenge ciphertext shown above. Our solution is as follows: Let $\mathbb{G}$ be a cyclic group with a generator g of prime order p, and $y = g^x$ is an additional public key for a randomly chosen element $x \in \mathbb{Z}_p^*$. In the encryption algorithm, r is randomly selected from $\mathbb{Z}_p^*$ and is used to compute g^r. Then, the algorithm attaches g^r to the message m and its hash value $\mathcal{H}_1(m)$. Their ciphertexts are generated using traditional PKE schemes. In addition, the algorithm provides the hash value of generated ciphertexts by attaching y^r. That is, our encryption algorithm for a message m is defined by

$$\mathsf{PKEET.Enc}(pk, m) = (\mathsf{PKE1}(pk_1, m\|g^r), \mathsf{PKE2}(pk_2, \mathcal{H}_1(m)\|g^r), \mathcal{H}_2(\mathsf{C}_1, \mathsf{C}_2, y^r))$$

where $\mathsf{C}_1 = \mathsf{PKE1}(pk_1, m\|g^r)$, $\mathsf{C}_2 = \mathsf{PKE2}(pk_2, \mathcal{H}_1(m)\|g^r)$, and $\mathcal{H}_1$ and $\mathcal{H}_2$ are cryptographic hash functions. Informally speaking, an adversary must know y^r to generate a valid ciphertext by modifying the challenge ciphertext, where y^r is the solution to the Computational Diffie-Hellman (CDH) problem for the instance (g, g^r, y). We demonstrate that our semi-generic construction achieves one-wayness under adaptive chosen ciphertext attack (OW-CCA2) security against Type-I adversaries, who have a trapdoor information for the equality test, and the indistinguishability under adaptive chosen ciphertext attack (IND-CCA2) security against Type-II adversaries, who do not have that information. (See Section 2.1 for the details of types of adversaries.) Those are shown assuming that the exploited PKE schemes are IND-CCA2 secure and the CDH assumption holds in the random oracle model. Moreover, we attempt to interpret Tang's all-or-nothing PKEET scheme [18], which coincides with our system model, in the view of our semi-generic construction.

Our construction can be easily extended to the identity-based setting by replacing PKE schemes and the CDH assumption with traditional identity-based encryption (IBE) schemes and the bilinear Diffie-Hellman (BDH) assumption, respectively. Thus, our modified encryption algorithm for IBE with equality test (IBEET) is defined as follows: Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of prime order p. Let g be a generator of $\mathbb{G}$ and set a public parameter $g_1 = g^s$ for a randomly chosen element s from $\mathbb{Z}_p^*$. $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map defined over $\mathbb{G}$ and $\mathbb{G}_T$. We define an encryption algorithm of our semi-generic IBEET construction with an identity ID and a message m by

$$\mathsf{IBEET.Enc}(pp, \mathsf{ID}, m)$$

$$= (\mathsf{IBE1}(pp_1, \mathsf{ID}, m\|g^r), \mathsf{IBE2}(pp_2, \mathsf{ID}, \mathcal{H}_1(m)\|g^r), \mathcal{H}_2(\mathsf{C}_1, \mathsf{C}_2, e(\mathcal{H}_3(\mathsf{ID}), g_1)^r))$$

for a randomly chosen $r \in \mathbb{Z}_p^*$, where IBE1 and IBE2 are traditional IBE schemes, $\mathsf{C}_1 = \mathsf{IBE1}(pp_1, \mathsf{ID}, m\|g^r)$, $\mathsf{C}_2 = \mathsf{IBE2}(pp_2, \mathsf{ID}, \mathcal{H}_1(m)\|g^r)$, and $\mathcal{H}_1$, $\mathcal{H}_2$, and $\mathcal{H}_3$ are cryptographic hash functions.

We demonstrate that our suggestion achieves one-wayness under adaptive chosen identity and adaptive chosen ciphertext attacks (OW-ID-CCA2) security against Type-I adversaries and the indistinguishability under adaptive chosen identity and adaptive chosen ciphertext attacks (IND-ID-CCA2) security against Type-II adversaries, assuming that the exploited IBE schemes are IND-ID-CCA2 secure and the BDH assumption holds in the random oracle model. Similar to our PKEET construction, we remark that an adversary must know the $e(\mathcal{H}_3(\mathsf{ID}), g_1)^r$ value to generate a valid ciphertext by modifying the challenge ciphertext, where $e(\mathcal{H}_3(\mathsf{ID}), g_1)^r = e(g^x, g^s)^r = e(g, g)^{xsr}$ is the solution to the BDH problem for the instance (g, g^x, g^r, g^s), assuming that $\mathcal{H}_3(\mathsf{ID})$ is $g^x \in \mathbb{G}$ for some element x in $\mathbb{Z}_p^*$.

As far as we know, our suggestion is the first IBEET scheme that achieves both the OW-ID-CCA2 security against Type-I adversaries and the IND-ID-CCA2 security against Type-II adversaries in the random oracle model. Furthermore, we demonstrate that it has comparable performance to the previous result [11], which achieves only the OW-ID-CCA2 security against Type-I adversaries in the random oracle model. According to our analysis, our construction, exploiting Boneh and Franklin's IND-ID-CCA2 secure IBE schemes [4], requires $6\mathsf{E}$, $3\mathsf{P} + 2\mathsf{E}$, and $2\mathsf{P} + 2\mathsf{E}$, for encryption, decryption, and testing, respectively, whereas the previous [11] construction requires $6\mathsf{E}$, $2\mathsf{P} + 2\mathsf{E}$, and $4\mathsf{P}$, respectively, where E and P denote the cost of computing exponents and pairing operations, respectively. (See Table 2 in Section 6.2 for the details of performance comparison.)

1.2. Related works

PKE/IBE with equality test. Yang et al. [19] first proposed the concept of PKEET, which allows anyone to check whether two ciphertexts under different public keys contain the same message, and provided an instantiation that is OW-CCA2 secure. Later, Tang [16,17] proposed an enhanced version of PKEET allowing only a tester authorized by two users to perform an equality test on ciphertexts between those users. However, this suggestion has a drawback that an interactive protocol between two users must be performed to initiate the authorization. To improve efficiency (with putting up with the looseness of authorization power), Tang [18] proposed an all-or-nothing PKEET scheme, which specifies the person who is to perform the equality test on all ciphertexts. Subsequently, Ma et al. [13] presented PKE with a delegated equality test that works by hiring a delegated party, who is the only party allowed to perform the test by communicating with a cloud server that stores the ciphertexts. Huang et al. [9] proposed PKE with an authorized equality test. In this formulation, a user separately issues warrants on all of his/her ciphertexts or only a specified ciphertext. Recently, Ma et al. [12] presented a PKEET scheme that simultaneously supports four types of flexible authorization. Following their work, Lin et al. [10] provided an improved construction without the use of bilinear maps.

In the identity-based setting, Ma [11] presented a scheme for IBEET that is OW-ID-CCA2 secure against Type-I adversaries. However, the IND-ID-CCA2 security was not considered, and the scheme does not achieve that level of security.

PKE/IBE with keyword search. PKE with keyword search (PKEKS) [3], which supports the functionality to perform an equality test between keywords embedded in a tag and a ciphertext, is similar to PKEET. The main difference between the two constructions is that PKEKS allows tests on ciphertexts under only a fixed public key, related to the issued tag, whereas PKEET allows equality tests on ciphertexts under different public keys as well as the same public keys. Similarly, IBE with keyword search (IBEKS) [1], which is an extension of PKEKS to the identity-based setting, has similar features to those of IBEET. Similar to the relation between PKEKS and PKEET, IBEKS also allows equality tests on ciphertexts under a fixed identity, related to the issued tag, whereas IBEET allows equality tests on ciphertexts under different identities as well as the same identity.

1.3. Organization of the paper

In Section 2, we introduce basic definitions related to our PKEET and IBEET constructions. Section 3 provides our semi-generic construction for PKEET and an interpretation of Tang's all-or-nothing PKEET scheme in the view of our semi-generic construction. A security analysis of our PKEET construction is given in Section 4. Section 5 extends the construction to the identity-based setting. In Section 6, we provide comparisons of our works with previous results. Some detailed proofs of theorems and lemmas are presented in the Appendices.

2. Basic definitions

In this section, we look at some basic concepts of PKE and IBE with equality test.

Notation. Throughout the paper, $\mathsf{negl}(\cdot)$ denotes a negligible function. For an algorithm A, $A \to a$ denotes that an algorithm A outputs a.

2.1. Public key encryption with equality test

System model of our PKEET. Our PKEET system model consists of users (including a sender and a receiver) and the tester (e.g., the cloud server): A sender encrypts a data using a receiver's public key and passes it to the receiver. Then, the receiver stores it to the cloud server and he/she can decrypt his/her ciphertexts using his/her secret key. One day, once the receiver wants to delegate the test capability for all of his/her ciphertexts, he/she issues a trapdoor to the tester who can access to the cloud server. Since then, the tester can perform equality test on ciphertexts under the public key of the receiver who passed the trapdoor to the tester.

Definition of PKEET. We provide the formal definition of PKEET and its correctness conditions in Definitions 1 and 2, respectively. Since PKEET is an extension of the traditional PKE, Definition 1 already contains the definition of PKE, so we refrain from rewriting the definition of PKE; instead, we note that PKE consists of the three algorithms in Definition 1: the key generation algorithm where a public parameter pp is a security parameter λ, the encryption algorithm, and the decryption algorithm.

Definition 1. A public key encryption scheme with equality test (PKEET) consists of the following six probabilistic polynomial time (PPT) algorithms $\mathsf{PKEET} = (\mathsf{PKEET.Setup}, \mathsf{PKEET.KeyGen}, \mathsf{PKEET.Enc}, \mathsf{PKEET.Dec}, \mathsf{PKEET.Trapdoor}, \mathsf{PKEET.Test})$:

- $\mathsf{PKEET.Setup}(\lambda)$: It takes a security parameter λ as an input and returns a public parameter pp.
- $\mathsf{PKEET.KeyGen}(pp)$: It takes the public parameter pp as an input and returns a pair of public and secret keys (pk, sk).
- $\mathsf{PKEET.Enc}(pk, m)$: It takes the public key pk and a plaintext $m \in \mathcal{M}$ as inputs and returns a ciphertext C. $\mathcal{M}$ denotes the plaintext space of the scheme.
- $\mathsf{PKEET.Dec}(sk, C)$: It takes the secret key sk and a ciphertext C as inputs and returns a plaintext m.

Suppose that each user has his own index i for $1 \le i \le N$. We denote a pair of public and secret keys of user i by (pk_i, sk_i). A ciphertext for user i is denoted by C_i.

- $\mathsf{PKEET.Trapdoor}(sk_i)$: It takes a user i's secret key sk_i as an input and returns a trapdoor td_i for user i's ciphertexts.
- $\mathsf{PKEET.Test}(\mathsf{td}_i, C_i, \mathsf{td}_j, C_j)$: It takes a user i's ciphertext C_i with the trapdoor td_i and a user j's ciphertext C_j with the trapdoor td_j as inputs. Then, it outputs 0 or 1.

Next, we define the correctness of the PKEET scheme.

Definition 2 (Correctness of the PKEET scheme)**.** We say a PKEET scheme is correct if for any security parameter λ, $\mathsf{PKEET.Setup}(\lambda) \to pp$, $\mathsf{PKEET.KeyGen}(pp) \to (pk_i, sk_i)$, and $\mathsf{PKEET.KeyGen}(\lambda) \to (pk_j, sk_j)$, the following conditions are satisfied:

1. For any message $m \in \mathcal{M}$, $\mathsf{PKEET.Dec}(sk_i, \mathsf{PKEET.Enc}(pk_i, m)) = m$ always holds.
2. For any ciphertexts C_i and C_j, if $\mathsf{PKEET.Dec}(sk_i, C_i) = \mathsf{PKEET.Dec}(sk_j, C_j) \neq \perp$,

$$\Pr[\mathsf{PKEET.Test}(\mathsf{td}_i, C_i, \mathsf{td}_j, C_j)] = 1,$$

   where $\mathsf{PKEET.Trapdoor}(sk_i) \to \mathsf{td}_i$ and $\mathsf{PKEET.Trapdoor}(sk_j) \to \mathsf{td}_j$.
3. For any ciphertexts C_i and C_j, if $\mathsf{PKEET.Dec}(sk_i, C_i) \neq \mathsf{PKEET.Dec}(sk_j, C_j)$,

$$\Pr[\mathsf{PKEET.Test}(\mathsf{td}_i, C_i, \mathsf{td}_j, C_j)] \le \mathsf{negl}(\lambda),$$

   where $\mathsf{PKEET.Trapdoor}(sk_i) \to \mathsf{td}_i$ and $\mathsf{PKEET.Trapdoor}(sk_j) \to \mathsf{td}_j$.

Remark 1 (Perfect Correctness vs. Computational Correctness)**.** For the third condition of Definition 2, we admit a negligible probability of test failure. It seems natural not to allow such a failure in the correctness definition, though it is negligible. We note that all existing PKEET schemes except for Yang et al.'s construction[1] [19], satisfy only our relaxed computational correctness definition, instead of the perfect correctness definition; their equality test algorithms exploit the hashed values of some input values, which are expected to be the same, but hash functions inherently lead to negligible errors on the results of equality tests because of the possibility of collision on their outputs.

[1] However, Yang et al.'s proposal allows anyone to perform equality tests and so achieves the weak security notion (i.e., OW-CCA2 security for Type-I adversaries) only.

Security definitions of PKEET. Now, we look into the security model of the PKEET scheme. We consider the following two types of adversaries for the security model of the PKEET scheme under our system model:

- Type-I adversary: We assume an adversary who has the trapdoor about the receiver of the challenge ciphertext and want him/her not to reveal the message contained in the challenge ciphertext.
- Type-II adversary: We assume an adversary who does not have the trapdoor about the receiver of the challenge ciphertext and want him/her not to distinguish whether the challenge ciphertext contains which message between two candidates.

Below we first define the OW-CCA2 security of the PKEET scheme against Type-I adversaries.

Definition 3 (OW-CCA2 of PKEET with a Trapdoor)**.** We say that a PKEET scheme is OW-CCA2 secure against Type-I adversaries if for any PPT adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following game with the challenger $\mathcal{C}$ is negligible in the security parameter λ:

1. **Setup**: $\mathcal{C}$ takes the security parameter λ as an input, runs $\mathsf{PKEET.Setup}(\lambda) \rightarrow pp$, and sends the public parameter pp to $\mathcal{A}$. Then, for $1 \leq i \leq N$, $\mathcal{C}$ runs $\mathsf{PKEET.KeyGen}(pp) \rightarrow (pk_i, sk_i)$ and sends all pk_i's to $\mathcal{A}$.
2. **Phase 1**: $\mathcal{A}$ may query the following oracles polynomially many times adaptively and in any order. The constraint is that an index t does not appear as an input in the $\mathcal{O}^{sk}$ oracle queries.
   - $\mathcal{O}^{sk}$: an oracle that on input an index i, returns sk_i.
   - $\mathcal{O}^{\mathsf{Dec}}$: an oracle that on input an index i and a ciphertext C_i, returns $\mathsf{PKEET.Dec}(sk_i, C_i)$ using the secret key sk_i.
   - $\mathcal{O}^{\mathsf{td}}$: an oracle that on input an index i, returns $\mathsf{PKEET.Trapdoor}(sk_i) \rightarrow \mathsf{td}_i$ using the secret key sk_i.
3. **Challenge**: $\mathcal{C}$ selects a random message $m \in \mathcal{M}$ for the plaintext space $\mathcal{M}$, runs $\mathsf{PKEET.Enc}(pk_t, m) \rightarrow C_t^*$, and sends C_t^* to $\mathcal{A}$.
4. **Phase 2**: $\mathcal{C}$ responds to $\mathcal{A}$'s queries as **Phase 1**. The constraints for $\mathcal{A}$'s queries are that
   (a) the index t does not appear as an input in the $\mathcal{O}^{sk}$ oracle queries;
   (b) a pair of the index t and the ciphertext C_t^* does not appear as an input in the $\mathcal{O}^{\mathsf{Dec}}$ oracle queries.
5. **Guess**: $\mathcal{A}$ outputs m'.

We say that the adversary $\mathcal{A}$ wins if $m' = m$ and the advantage of $\mathcal{A}$ in the above game is defined to

$$\mathbf{Adv}^{\text{OW-CCA2}}_{\text{PKEET},\mathcal{A}}(\lambda) := \Pr[\mathcal{A} \text{ wins}].$$

When the size of the plaintext space is polynomial in the security parameter or the min-entropy of the message distribution is much lower than the security parameter, an adversary who has a trapdoor information for the user t, may perform equality tests with the challenge ciphertext by generating ciphertexts of all messages. To prevent such trivial attacks, we assume that the size of the plaintext space is exponential in the security parameter and the min-entropy of the message distribution is sufficiently higher than the security parameter.

Now, we provide the definition of the IND-CCA2 security of the PKEET scheme against Type-II adversaries below.

Definition 4 (IND-CCA2 of PKEET without a Trapdoor)**.** We say that a PKEET is IND-CCA2 secure against Type-II adversaries if for any PPT adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following game with the challenger $\mathcal{C}$ is negligible in the security parameter λ:

1. **Setup**: This step is the same as that of the OW-CCA2 security game in Definition 3.
2. **Phase 1**: This step is almost the same as that of the OW-CCA2 security game in Definition 3, except that the constraint is that an index t does not appear as an input in the $\mathcal{O}^{sk}$ and $\mathcal{O}^{\mathsf{td}}$ oracle queries.
3. **Challenge**: $\mathcal{A}$ selects messages $m_0, m_1 \in \mathcal{M}$ of the same length and sends $\mathcal{C}$ them. $\mathcal{C}$ selects a random bit $b \in \{0, 1\}$, runs $\mathsf{PKEET.Enc}(pk_t, m_b) \rightarrow C_{t,b}^*$, and sends $C_{t,b}^*$ to $\mathcal{A}$.
4. **Phase 2**: $\mathcal{C}$ responds to $\mathcal{A}$'s queries as **Phase 1**. The constraints for $\mathcal{A}$'s queries are that
   (a) the index t does not appear as an input in the $\mathcal{O}^{sk}$ and $\mathcal{O}^{\mathsf{td}}$ oracle queries;
   (b) a pair of the index t and the ciphertext $C_{t,b}^*$ does not appear as an input in the $\mathcal{O}^{\mathsf{Dec}}$ oracle queries.
5. **Guess**: $\mathcal{A}$ outputs $b' \in \{0, 1\}$.

We say that the adversary $\mathcal{A}$ wins if $b' = b$ in the above game and the advantage of $\mathcal{A}$ is defined to

$$\mathbf{Adv}^{\text{IND-CCA2}}_{\text{PKEET},\mathcal{A}}(\lambda) := \Pr[\mathcal{A} \text{ wins}] - \frac{1}{2}.$$

2.2. Identity-based encryption with equality test

In this subsection, we provide some definitions of IBEET. Basic definitions including the system model, the definition of IBEET scheme, and its OW-ID-CCA2 security against Type-I adversaries, follow Ma's ones [11]. Further, we define the IND-ID-CCA2 security against Type-II adversaries for IBEET scheme by modifying the adaptive IND-ID-CCA2 security of traditional IBE schemes.

System model of our IBEET. Our IBEET system model consists of users (including a sender and a receiver), the key generation center (KGC), and the tester (e.g., the cloud server): As traditional IBE schemes, the KGC issues a user's secret key according to a user's identity to the user. The rest is almost the same with that of our PKEET. A sender encrypts a data using a receiver's identity and passes it to the receiver. Then, the receiver stores it to the cloud server and he/she can decrypt his/her ciphertexts using his/her secret key issued by the KGC. One day, once the receiver wants to delegate the test capability for all of his/her ciphertexts, he/she issues a trapdoor to the tester who can access to the cloud server. Since then, the tester can perform equality test on ciphertexts under the identity of the receiver who passed the trapdoor to the tester.

Definition of IBEET. We provide the formal definition of IBEET under our system model as follows.

Definition 5. An identity-based encryption scheme with equality test (IBEET) consists of the following PPT algorithms $\mathsf{IBEET} = (\mathsf{IBEET.Setup}, \mathsf{IBEET.Extract}, \mathsf{IBEET.Enc}, \mathsf{IBEET.Dec}, \mathsf{IBEET.Trapdoor}, \mathsf{IBEET.Test})$:

- $\mathsf{IBEET.Setup}(\lambda)$: It takes a security parameter λ as an input and returns a public parameter pp and a master secret key msk.
- $\mathsf{IBEET.Extract}(pp, msk, \mathsf{ID})$: It takes the public parameter pp, the master secret key msk, and an identity $\mathsf{ID} \in \{0,1\}^*$ as inputs, and outputs a user ID's secret key d_{ID}.
- $\mathsf{IBEET.Enc}(pp, \mathsf{ID}, m)$: It takes the public parameter pp, an identity $\mathsf{ID} \in \{0,1\}^*$, and a message m as inputs, and outputs a ciphertext C.
- $\mathsf{IBEET.Dec}(pp, d_{\mathsf{ID}}, C)$: It takes the public parameter pp, a user ID's secret key d_{ID}, and a ciphertext C as inputs, and outputs a message m'.
- $\mathsf{IBEET.Trapdoor}(d_{\mathsf{ID}})$: It takes a user ID's secret key d_{ID} as an input and outputs a trapdoor $\mathsf{td}_{\mathsf{ID}}$ for user ID's ciphertexts.
- $\mathsf{IBEET.Test}(\mathsf{td}_{\mathsf{ID}_i}, C_{\mathsf{ID}_i}, \mathsf{td}_{\mathsf{ID}_j}, C_{\mathsf{ID}_j})$: It takes a user ID_i's ciphertext with the trapdoor $\mathsf{td}_{\mathsf{ID}_i}$ and a user ID_j's ciphertext with the trapdoor $\mathsf{td}_{\mathsf{ID}_j}$ as inputs. Then, it outputs 0 or 1.

Remark 2. We note that a traditional IBE scheme consists of the first four algorithms in Definition 5: the setup algorithm, the key extraction algorithm, the encryption algorithm, and the decryption algorithm.

Now, we define the correctness of the IBEET scheme below.

Definition 6 (Correctness of the IBEET scheme)**.** We say an IBEET scheme is correct if for any security parameter λ, all $\mathsf{IBEET.Setup}(\lambda) \to (pp, msk)$, $\mathsf{IBEET.Extract}(pp, msk, \mathsf{ID}_i) \to d_{\mathsf{ID}_i}$, and $\mathsf{IBEET.Extract}(pp, msk, \mathsf{ID}_j) \to d_{\mathsf{ID}_j}$, the following conditions are satisfied:

1. For any message $m \in \mathcal{M}$, $\mathsf{IBEET.Dec}(pp, d_{\mathsf{ID}_i}, \mathsf{IBEET.Enc}(pp, \mathsf{ID}_i, m)) = m$ always holds.
2. For any ciphertexts C_i and C_j, if $\mathsf{IBEET.Dec}(pp, d_{\mathsf{ID}_i}, C_i) = \mathsf{IBEET.Dec}(pp, d_{\mathsf{ID}_j}, C_j) \neq \perp$,

   $$\Pr[\mathsf{IBEET.Test}(\mathsf{td}_{\mathsf{ID}_i}, C_{\mathsf{ID}_i}, \mathsf{td}_{\mathsf{ID}_j}, C_{\mathsf{ID}_j})] = 1,$$

   where $\mathsf{IBEET.Trapdoor}(d_{\mathsf{ID}_i}) \to \mathsf{td}_{\mathsf{ID}_i}$ and $\mathsf{IBEET.Trapdoor}(d_{\mathsf{ID}_j}) \to \mathsf{td}_{\mathsf{ID}_j}$.
3. For any ciphertexts C_i and C_j, if $\mathsf{IBEET.Dec}(pp, d_{\mathsf{ID}_i}, C_i) \neq \mathsf{IBEET.Dec}(pp, d_{\mathsf{ID}_j}, C_j)$,

   $$\Pr[\mathsf{IBEET.Test}(\mathsf{td}_{\mathsf{ID}_i}, C_{\mathsf{ID}_i}, \mathsf{td}_{\mathsf{ID}_j}, C_{\mathsf{ID}_j})] \leq \mathsf{negl}(\lambda),$$

   where $\mathsf{IBEET.Trapdoor}(d_{\mathsf{ID}_i}) \to \mathsf{td}_{\mathsf{ID}_i}$ and $\mathsf{IBEET.Trapdoor}(d_{\mathsf{ID}_j}) \to \mathsf{td}_{\mathsf{ID}_j}$.

Security definitions of IBEET. As the security definitions of PKEET, we consider two types of adversaries, Type-I adversaries who have the trapdoor and Type-II adversaries who do not have that information. We first define the OW-ID-CCA2 security of the IBEET scheme against Type-I adversaries.

Definition 7 (OW-ID-CCA2 of IBEET with a Trapdoor)**.** We say that a IBEET scheme is OW-ID-CCA2 secure against Type-I adversaries if for any PPT adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following game with the challenger $\mathcal{C}$ is negligible in the security parameter λ:

1. **Setup**: $\mathcal{C}$ takes a security parameter λ as an input, runs $\mathsf{IBEET.Setup}(\lambda) \rightarrow (pp, msk)$. $\mathcal{C}$ sends the public parameter pp to $\mathcal{A}$ and keeps the master secret key msk private.
2. **Phase 1**: $\mathcal{A}$ may query the following oracles polynomially many times adaptively and in any order:
   - $\mathcal{O}^{\mathsf{Ext}}$: an oracle that on input an identity ID, returns d_{ID}.
   - $\mathcal{O}^{\mathsf{Dec}}$: an oracle that on input an identity ID and a ciphertext C, runs $\mathsf{IBEET.Dec}(pp, d_{\mathsf{ID}}, C) \rightarrow m'$ and outputs m'.
   - $\mathcal{O}^{\mathsf{td}}$: an oracle that on input an identity ID, runs $\mathsf{IBEET.Trapdoor}(d_{\mathsf{ID}}) \rightarrow \mathsf{td}_{\mathsf{ID}}$ and outputs $\mathsf{td}_{\mathsf{ID}}$.
3. **Challenge**: $\mathcal{A}$ submits an identity ID^* which was not queried to the $\mathcal{O}^{\mathsf{Ext}}$ oracle in Step 2. $\mathcal{C}$ selects a random message m, runs $\mathsf{IBEET.Enc}(pp, \mathsf{ID}^*, m) \rightarrow C^*_{\mathsf{ID}^*}$, and sends $C^*_{\mathsf{ID}^*}$ to $\mathcal{A}$.
4. **Phase 2**: $\mathcal{C}$ responds to $\mathcal{A}$'s queries as **Phase 1**. The constraints for $\mathcal{A}$'s queries are that
   (a) the identity ID^* does not appear as an input in the $\mathcal{O}^{\mathsf{Ext}}$ oracle queries,
   (b) a pair of the identity ID^* and the ciphertext $C^*_{\mathsf{ID}^*}$ does not appear as an input in the $\mathcal{O}^{\mathsf{Dec}}$ oracle queries.
5. **Guess**: $\mathcal{A}$ outputs m'.

We say that the adversary $\mathcal{A}$ wins if $m' = m$ and the advantage of $\mathcal{A}$ in the above game is defined to

$$\mathbf{Adv}^{\text{OW-ID-}CCA2}_{\text{IBEET},\mathcal{A}}(\lambda) := \Pr[\mathcal{A} \text{ wins}].$$

Now, we provide the definition of the IND-ID-CCA2 security of the IBEET scheme against Type-II adversaries by modifying the adaptive IND-ID-CCA2 security of traditional IBE schemes.

Definition 8 (IND-ID-CCA2 of IBEET without a Trapdoor)**.** We say that an IBEET scheme is IND-ID-CCA2 secure against Type-II adversaries if for any PPT adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following game with the challenger $\mathcal{C}$ is negligible in the security parameter λ:

1. **Setup**: This step is the same as that of the OW-ID-CCA2 security game in Definition 7.
2. **Phase 1**: This step is the same as that of the OW-ID-CCA2 security game in Definition 7.
3. **Challenge**: The adversary selects an identity ID^*, which was never queried to the $\mathcal{O}^{\mathsf{Ext}}$ and $\mathcal{O}^{\mathsf{td}}$ oracles in Step 2, and two messages m_0, m_1 of the same length and sends $\mathcal{C}$ them. $\mathcal{C}$ selects a random bit $b \in \{0, 1\}$, runs $\mathsf{IBEET.Enc}(pp, \mathsf{ID}^*, m_b) \rightarrow C^*_{\mathsf{ID}^*,b}$, and sends $C^*_{\mathsf{ID}^*,b}$ to $\mathcal{A}$.
4. **Phase 2**: $\mathcal{C}$ responds to $\mathcal{A}$'s queries as **Phase 1**. The constraints for $\mathcal{A}$'s queries are that
   (a) the identity ID^* does not appear as an input in the $\mathcal{O}^{\mathsf{Ext}}$ and $\mathcal{O}^{\mathsf{td}}$ oracle queries;
   (b) a pair of the identity ID^* and the ciphertext $C^*_{\mathsf{ID}^*,b}$ does not appear as an input in the $\mathcal{O}^{\mathsf{Dec}}$ oracle queries.
5. **Guess**: $\mathcal{A}$ outputs $b' \in \{0, 1\}$.

We say that the advesary wins if $b = b'$ in the above game and the advantage of $\mathcal{A}$ is defined to

$$\mathbf{Adv}^{\text{IND-ID-CCA2}}_{\text{IBEET},\mathcal{A}}(\lambda) := \Pr[\mathcal{A} \text{ wins}] - \frac{1}{2}.$$

2.3. Cryptographic assumptions

Now, we introduce two well-known cryptographic assumptions, the CDH assumption and the BDH assumption. We will use them to prove the security of our PKEET and IBEET schemes, respectively.

Definition 9 (CDH Problem and Assumption)**.** Let $\mathbb{G}$ be a cyclic group of order $p = p(\lambda)$ with a generator g for a security parameter λ. The Computational Diffie-Hellman (CDH) problem is defined as follows: Given (g, g^x, g^y) for randomly chosen $x, y \in \mathbb{Z}^*_p$, a PPT algorithm $\mathcal{A}$ finds the value g^{xy} with the advantage

$$\mathbf{Adv}^{\text{CDH}}_{\mathbb{G},\mathcal{A}}(\lambda) := \Pr[\mathcal{A}(g, g^x, g^y) = g^{xy}].$$

We say that the CDH assumption on $\mathbb{G}$ holds if for any PPT algorithm $\mathcal{A}$, the advantage of $\mathcal{A}$ is negligible in the security parameter λ.

Definition 10 (BDH Problem and Assumption)**.** Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of order $p = p(\lambda)$ for a security parameter λ. Let $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear map and g be a generator of $\mathbb{G}$. The Bilinear Diffie-Hellman (BDH) problem is defined as follows: Given (g, g^x, g^y, g^z) for randomly chosen $x, y, z \in \mathbb{Z}^*_p$, a PPT algorithm $\mathcal{A}$ finds the value $e(g, g)^{xyz}$ with the advantage

$$\mathbf{Adv}^{\text{BDH}}_{\mathbb{G},\mathbb{G}_T,\mathcal{A}}(\lambda) := \Pr[\mathcal{A}(g, g^x, g^y, g^z) = e(g, g)^{xyz}].$$

We say that the BDH assumption on $(\mathbb{G}, \mathbb{G}_T, e)$ holds if for any PPT algorithm $\mathcal{A}$, the advantage of $\mathcal{A}$ is negligible in the security parameter λ.

3. Our semi-generic construction of PKE with equality test

In this section, we provide a semi-generic PKEET construction that exploits traditional PKE schemes. Subsequently, we interpret Tang's all-or-nothing PKEET scheme, which coincides with our system model, in the view of our semi-generic construction.

3.1. Our construction

Now, we provide our semi-generic PKEET construction by exploiting two traditional PKE schemes, $\mathsf{PKE1} = (\mathsf{PKE1.KeyGen}, \mathsf{PKE1.Enc}, \mathsf{PKE1.Dec})$ and $\mathsf{PKE2} = (\mathsf{PKE2.KeyGen}, \mathsf{PKE2.Enc}, \mathsf{PKE2.Dec})$. The description is as follows:

- $\mathsf{PKEET.Setup}(\lambda)$: It takes a security parameter λ as an input. Let $\mathbb{G}$ be a cyclic group of prime order $p = p(\lambda)$ and g be a generator of a group $\mathbb{G}$. Let $\mathcal{H}_1 : \{0,1\}^{\ell_1} \to \{0,1\}^{\ell_2}$ and $\mathcal{H}_2 : \{0,1\}^* \to \{0,1\}^{\ell_3}$ be cryptographic hash functions for integers $\ell_1 = \ell_1(\lambda)$, $\ell_2 = \ell_2(\lambda)$, and $\ell_3 = \ell_3(\lambda)$. It outputs a public parameter $pp = (\mathbb{G}, g, \mathcal{H}_1, \mathcal{H}_2)$.
  We remark that the plaintext space of our construction is $\{0,1\}^{\ell_1}$. We assume that the plaintext spaces of PKE1 and PKE2 include $\{0,1\}^{\ell_1+\ell}$ and $\{0,1\}^{\ell_2+\ell}$, respectively, where elements of $\mathbb{G}$ are represented in ℓ bits.
- $\mathsf{PKEET.KeyGen}(pp)$: It takes the public parameter pp as an input. Then, it performs $\mathsf{PKE1.KeyGen}(\lambda) \to (\mathsf{pk}_1, \mathsf{sk}_1)$ and $\mathsf{PKE2.KeyGen}(\lambda) \to (\mathsf{pk}_2, \mathsf{sk}_2)$. It selects a random element x from $\mathbb{Z}_p^*$ and computes $X = g^x$. It outputs a public key $pk = (pp, \mathsf{pk}_1, \mathsf{pk}_2, X)$ and a secret key $sk = (pp, \mathsf{sk}_1, \mathsf{sk}_2, x)$.
- $\mathsf{PKEET.Enc}(pk, m)$: It takes the public key $pk = (pp, \mathsf{pk}_1, \mathsf{pk}_2, X)$ and a message $m \in \{0,1\}^{\ell_1}$ as inputs and performs as follows:
  1. Select a random element r from $\mathbb{Z}_p^*$.
  2. Run $\mathsf{PKE1.Enc}(\mathsf{pk}_1, m \| g^r) \to \mathsf{C}_1$ and $\mathsf{PKE2.Enc}(\mathsf{pk}_2, \mathcal{H}_1(m) \| g^r) \to \mathsf{C}_2$.
  3. Compute $\mathsf{C}_3 = \mathcal{H}_2(\mathsf{C}_1, \mathsf{C}_2, X^r)$.
  4. Output $C = (\mathsf{C}_1, \mathsf{C}_2, \mathsf{C}_3)$.
- $\mathsf{PKEET.Dec}(sk, C)$: It takes the secret key $sk = (pp, \mathsf{sk}_1, \mathsf{sk}_2, x)$ and a ciphertext $C = (\mathsf{C}_1, \mathsf{C}_2, \mathsf{C}_3)$ as inputs and performs as follows:
  1. Run $\mathsf{PKE1.Dec}(\mathsf{sk}_1, \mathsf{C}_1) \to m' \| R'$ and $\mathsf{PKE2.Dec}(\mathsf{sk}_2, \mathsf{C}_2) \to h' \| R''$.
  2. Check whether $h' = \mathcal{H}_1(m')$, $R' = R''$, and $\mathsf{C}_3 = \mathcal{H}_2(\mathsf{C}_1, \mathsf{C}_2, (R')^x)$.
  3. If all of them hold, output m'. Otherwise, output $\perp$.

We denote a public key and a secret key of a user i by $pk_i = (pp, \mathsf{pk}_{i,1}, \mathsf{pk}_{i,2}, X_i)$ and $sk_i = (pp, \mathsf{sk}_{i,1}, \mathsf{sk}_{i,2}, x_i)$, respectively. $C_i = (\mathsf{C}_{i,1}, \mathsf{C}_{i,2}, \mathsf{C}_{i,3})$ denotes a user i's ciphertext.

- $\mathsf{PKEET.Trapdoor}(sk_i)$: It takes the secret key $sk_i = (pp, \mathsf{sk}_{i,1}, \mathsf{sk}_{i,2}, x_i)$ as an input and outputs $\mathsf{td}_i = \mathsf{sk}_{i,2}$.
- $\mathsf{PKEET.Test}(\mathsf{td}_i, C_i, \mathsf{td}_j, C_j)$: It takes a user i's ciphertext $C_i = (\mathsf{C}_{i,1}, \mathsf{C}_{i,2}, \mathsf{C}_{i,3})$ with his/her trapdoor $\mathsf{td}_i = \mathsf{sk}_{i,2}$ and a user j's ciphertext $C_j = (\mathsf{C}_{j,1}, \mathsf{C}_{j,2}, \mathsf{C}_{j,3})$ with his/her trapdoor $\mathsf{td}_j = \mathsf{sk}_{j,2}$ as inputs. Then, it performs as follows:
  1. Run $\mathsf{PKE2.Dec}(\mathsf{td}_i, \mathsf{C}_{i,2}) \to h_i' \| R_i''$ and $\mathsf{PKE2.Dec}(\mathsf{td}_j, \mathsf{C}_{j,2}) \to h_j' \| R_j''$.
  2. Check whether $h_i' = h_j'$. If it holds, output 1. Otherwise, output 0.

Correctness. The following theorem demonstrates the correctness of our construction.

Theorem 1. *Our construction in Section 3.1 is correct according to Definition 2 under assuming that the exploited* PKE1 *and* PKE2 *schemes are correct and* $\mathcal{H}_1$ *is a collision-resistant hash function.*

Proof. Let $C_i = (\mathsf{C}_{i,1}, \mathsf{C}_{i,2}, \mathsf{C}_{i,3})$ be a valid ciphertext generated by running $\mathsf{PKEET.Enc}(pk_i, m_i)$ where pk_i is a user i's public key. Because PKE1 and PKE2 are correct, $\mathsf{PKE1.Dec}(\mathsf{sk}_{i,1}, \mathsf{C}_{i,1}) \to m_i \| g^r$ and $\mathsf{PKE2.Dec}(\mathsf{sk}_{i,2}, \mathsf{C}_{i,2}) \to \mathcal{H}_1(m_i) \| g^r$ for some $r \in \mathbb{Z}_p^*$. Furthermore, since $X^r = (g^r)^x$, $\mathcal{H}_2(\mathsf{C}_{i,1}, \mathsf{C}_{i,2}, (g^r)^x)$ is the same as $\mathsf{C}_{i,3} = \mathcal{H}_2(\mathsf{C}_{i,1}, \mathsf{C}_{i,2}, X^r)$. Hence, the first condition for the correctness holds.

Let pk_j be a user j's public key and $C_j = (\mathsf{C}_{j,1}, \mathsf{C}_{j,2}, \mathsf{C}_{j,3})$ be a valid ciphertext obtained by running $\mathsf{PKEET.Enc}(pk_j, m_j)$. For the PKEET.Test algorithm, when the trapdoors td_i and td_j for users i and j are given, if $m_i = m_j$, then the algorithm outputs 1 because $\mathsf{PKE2.Dec}(\mathsf{td}_i, \mathsf{C}_{i,2}) \to \mathcal{H}_1(m_i) \| R_i$, $\mathsf{PKE2.Dec}(\mathsf{td}_j, \mathsf{C}_{j,2}) \to \mathcal{H}_1(m_j) \| R_j$, and $\mathcal{H}_1(m_i) = \mathcal{H}_1(m_j)$. Otherwise, $\mathcal{H}_1(m_i)$ is different from $\mathcal{H}_1(m_j)$ with overwhelming probability since $\mathcal{H}_1$ is a collision-resistant function. Hence, both the second and third conditions for the correctness hold. □

Remark 3. We have provided a description of our semi-generic construction by exploiting two PKE schemes, PKE1 and PKE2. On the other hand, according to our security analysis in the next section, our proposed construction satisfies the requirements for PKEET if both PKE1 and PKE2 are IND-CCA2 secure. Hence, we may employ the same PKE scheme for PKE1 and PKE2 if the exploited PKE scheme is IND-CCA2 secure. We note that in that case, the size of the system parameter pp

could be slightly reduced up to a small constant factor, but the complexity of all algorithms in our construction will not be reduced.

3.2. Interpretation of Tang's PKEET scheme as our semi-generic construction

Now, we attempt to interpret Tang's all-or-nothing PKEET scheme [18], which coincides with our PKEET system model. The encryption algorithm of his construction is

$$\begin{aligned} C &= \mathsf{PKEET.Enc}(pk, m) = (c_1, c_2, c_3, c_4, c_5) \\ &= (g^u,\ g^v,\ \mathcal{H}_3((g^x)^u) \oplus m \| u,\ g^{\mathcal{H}_4((g^y)^v)+m},\ \mathcal{H}_2(c_1\|c_2\|c_3\|c_4\|m\|u)), \end{aligned}$$

where g is a generator of a cyclic group $\mathbb{G}$ of prime order p, (g^x, g^y) is a public key, u and v are randomly chosen elements from $\mathbb{Z}_p^*$ by the encryption algorithm, and $\mathcal{H}_2$, $\mathcal{H}_3$, and $\mathcal{H}_4$ are cryptographic hash functions. Here, (x, y) is a secret key and y is a trapdoor issued to the tester for equality test. Then, the tester can check the equality by obtaining g^m using y by computing $c_4/g^{\mathcal{H}_4(c_2^y)}$.

We can regard his algorithm as

$$\begin{aligned} \mathsf{C}_1 &= \mathsf{PKE1}(pk_1, m) = (c_1, c_3) = (g^u, \mathcal{H}_3((g^x)^u) \oplus m \| u), \\ \mathsf{C}_2 &= \mathsf{PKE2}(pk_2, g^m) = (c_2, c_4) = (g^v, g^{\mathcal{H}_4((g^y)^v)} \cdot g^m), \\ \mathsf{C}_3 &= c_5 = \mathcal{H}_2(c_1\|c_2\|c_3\|c_4\|m\|u), \end{aligned}$$

where $pk_1 = g^x$ and $pk_2 = g^y$. Once we define a hash function $\mathcal{H}_1(m)$ by g^m, C_2 can be regarded as $\mathsf{PKE2}(pk_2, \mathcal{H}_1(m))$ and it is very similar to ours. The only main difference between them is that his scheme prevents the attack by inserting m and u to the inputs of the hash function $\mathcal{H}_2$, whereas ours relies on the CDH assumption by inserting the CDH instance in the encryption phase.

4. Security analysis of our PKEET construction

In this section, we demonstrate that our PKEET construction is OW-CCA2 secure against PPT Type-I adversaries and IND-CCA2 secure against PPT Type-II adversaries.

4.1. OW-CCA2 security against type-I adversaries

We first look into the OW-CCA2 security of our construction against Type-I adversaries.

Theorem 2. *Our construction in Section 3.1 is OW-CCA2 secure against PPT Type-I adversaries, under assuming that the exploited encryption scheme* PKE1 *is IND-CCA2 secure, the CDH assumption on* $\mathbb{G}$ *holds, and* $\mathcal{H}_1$ *and* $\mathcal{H}_2$ *are modelled as random oracles.*

Proof. Let $\mathcal{A}$ be a PPT Type-I adversary who breaks the OW-CCA2 security of our construction with advantage $\epsilon_{\mathcal{A}}$. Then, we can construct a PPT algorithm $\mathcal{B}$ who breaks the IND-CCA2 security of the PKE1 scheme using $\mathcal{A}$ as follows:

1. On receiving a public key pk^* from the challenger $\mathcal{C}$ of the IND-CCA2 security game of the PKE1 scheme, $\mathcal{B}$ performs as follows:
   (a) Run $\mathsf{PKE1.KeyGen}(\lambda) \to (\mathsf{pk}_{i,1}, \mathsf{sk}_{i,1})$ for $1 \le i \le N$, $i \ne t$, and set $\mathsf{pk}_{t,1} = \mathsf{pk}^*$.
   (b) Run $\mathsf{PKE2.KeyGen}(\lambda) \to (\mathsf{pk}_{i,2}, \mathsf{sk}_{i,2})$ for $1 \le i \le N$.
   (c) Generate a cyclic group $\mathbb{G}$ of prime order p with a generator g. Select a random element x_i from $\mathbb{Z}_p^*$ and compute $X_i = g^{x_i}$ for $1 \le i \le N$.
   (d) Send all $pk_i = (pp, \mathsf{pk}_{i,1}, \mathsf{pk}_{i,2}, X_i)$'s for $1 \le i \le N$ to $\mathcal{A}$.
2. For $\mathcal{A}$'s oracle queries, $\mathcal{B}$ responds as follows:
   - $\mathcal{O}^{\mathcal{H}_1}$ query: On input m, it returns h_1 if (m, h_1) has previously been stored in the hash list $L^{\mathcal{H}_1}$, which was originally initiated as an empty set. Otherwise, it randomly selects h_1 from $\{0,1\}^{\ell_2}$, adds (m, h_1) to the hash list $L^{\mathcal{H}_1}$, and returns h_1.
   - $\mathcal{O}^{\mathcal{H}_2}$ query: On input $(\mathsf{C}_1, \mathsf{C}_2, R)$, it returns h_2 if $(\mathsf{C}_1, \mathsf{C}_2, R, h_2)$ has previously been stored in the hash list $L^{\mathcal{H}_2}$, which was originally initiated as an empty set. Otherwise, it randomly selects h_2 from $\{0,1\}^{\ell_3}$, adds $(\mathsf{C}_1, \mathsf{C}_2, R, h_2)$ to the hash list $L^{\mathcal{H}_2}$, and returns h_2.
   - $\mathcal{O}^{sk}$ query: On input $i \ne t$, it outputs $sk_i = (pp, \mathsf{sk}_{i,1}, \mathsf{sk}_{i,2}, x_i)$.
   - $\mathcal{O}^{\mathsf{td}}$ query: On input i, it runs $\mathsf{PKEET.Trapdoor}(sk_i) \to \mathsf{td}_i$ and returns td_i. We note that because $\mathcal{B}$ knows $\mathsf{sk}_{t,2}$, it can respond to queries on an input t without the whole secret key sk_t.
   - $\mathcal{O}^{\mathsf{Dec}}$ query: On input a pair of an index and a ciphertext (i, C_i) for $C_i = (\mathsf{C}_{i,1}, \mathsf{C}_{i,2}, \mathsf{C}_{i,3})$,
     – if $i \ne t$, it runs $\mathsf{PKEET.Dec}(sk_i, C_i) \to m'$ and returns m';

– otherwise, $\mathcal{B}$ queries $\mathsf{C}_{t,1}$ to the decryption oracle of the IND-CCA2 security game of the PKE1 scheme. Once returning $m'\|R'$, $\mathcal{B}$ runs $\mathsf{PKE2.Dec}(\mathsf{sk}_{t,2}, \mathsf{C}_{t,2}) \rightarrow h'\|R''$ and checks whether $h' = \mathcal{H}_1(m')$, $R' = R''$, and $\mathsf{C}_{t,3} = \mathcal{H}_2(\mathsf{C}_{t,1}, \mathsf{C}_{t,2}, (R')^{x_t})$. If all of them hold, it returns m'. Otherwise, it returns $\perp$.

3. $\mathcal{B}$ selects two random messages m_0, m_1 from $\{0,1\}^{\ell_1}$ and r from $\mathbb{Z}_p^*$, and sends $m_0\|g^r$, $m_1\|g^r$ to $\mathcal{C}$. Then, $\mathcal{C}$ may select a random bit $\beta \in \{0, 1\}$, run

$$\mathsf{PKE1.Enc}(\mathsf{pk}_{t,1}, m_\beta\|g^r) \rightarrow \mathsf{C}^*_{t,1},$$

and send it to $\mathcal{B}$.

4. Once $\mathcal{B}$ receives the challenge ciphertext $\mathsf{C}^*_{t,1}$, he selects a random element $h'_1 \in \{0,1\}^{\ell_2}$, runs

$$\mathsf{PKE2.Enc}(\mathsf{pk}_{t,2}, h'_1\|g^r) \rightarrow \mathsf{C}^*_{t,2},$$

and $\mathsf{C}^*_{t,3} = \mathcal{H}_2(\mathsf{C}^*_{t,1}, \mathsf{C}^*_{t,2}, X^r_t)$. $\mathcal{B}$ sends $C^*_t = (\mathsf{C}^*_{t,1}, \mathsf{C}^*_{t,2}, \mathsf{C}^*_{t,3})$ to $\mathcal{A}$ as a challenge ciphertext.

5. $\mathcal{B}$ responds to $\mathcal{A}$'s oracle queries as the same as in Step 2, except for the decryption oracle queries on ciphertexts of form $(\mathsf{C}^*_{t,1}, \mathsf{C}_{t,2}, \mathsf{C}_{t,3})$. For the decryption oracle queries on ciphertexts of form $(\mathsf{C}^*_{t,1}, \mathsf{C}_{t,2}, \mathsf{C}_{t,3})$, $\mathcal{B}$ outputs $\perp$.
6. $\mathcal{A}$ outputs m'. If $m' = m_b$ for $b = 0, 1$, $\mathcal{B}$ outputs $\beta' = b$. Otherwise, it outputs a random bit $\beta' \in \{0, 1\}$.

Let us evaluate the advantage of $\mathcal{B}$. First, we note that the simulation of $\mathcal{B}$ may fail in the following two cases:

- Let $\mathcal{F}_1$ be the event that a ciphertext $(\mathsf{C}^*_{t,1}, \mathsf{C}_{t,2}, \mathsf{C}_{t,3})$ queried to the decryption oracle in Step 5, is valid. In this case, we assume that the decryption oracle returns $\perp$ in the above simulation. However, in case when $\mathsf{PKE2.Enc}(\mathsf{pk}_{t,2}, \bar{h}_1\|g^r) \rightarrow \mathsf{C}_{t,2}$ and $\mathsf{C}_{t,3} = \mathcal{H}_2(\mathsf{C}^*_{t,1}, \mathsf{C}_{t,2}, X^r_t)$ for some $\bar{h}_1 \in \{0,1\}^{\ell_2}$, this ciphertext becomes valid. To generate such a ciphertext, the adversary $\mathcal{A}$ must query to the oracle $\mathcal{O}^{\mathcal{H}_2}$ on an input $(\mathsf{C}^*_{t,1}, \mathsf{C}_{t,2}, X^r_t)$. Here, we can assume that if the adversary detects such a wrong simulation, then he can solve the CDH problem on the instance (g, g^r, g^{x_t}), because he generates $X^r_t = (g^{x_t})^r$ with only (g, g^r, g^{x_t}). However, because we assume that the CDH assumption on $\mathbb{G}$ holds, $\Pr[\mathcal{F}_1] \leq \mathbf{Adv}^{\mathrm{CDH}}_{\mathcal{A},\mathbb{G}}$ is negligible in the security parameter.
- Let $\mathcal{F}_2$ be the event that m_0 or m_1 are queried to the oracle $\mathcal{O}^{\mathcal{H}_1}$. If m_0 or m_1 are queried and the outputs of the oracle queries on m_0 or m_1 are not h'_1, then the challenge ciphertext for the adversary $\mathcal{A}$ may be invalid. Since both m_0 and m_1 are completely hidden from the viewpoint of $\mathcal{A}$, the probability that $\mathcal{F}_2$ occurs is

$$\Pr[\mathcal{F}_2] = 1 - \binom{2^{\ell_1} - 2}{q_{\mathcal{H}_1}} \Big/ \binom{2^{\ell_1}}{q_{\mathcal{H}_1}} = 1 - \left(1 - \frac{q_{\mathcal{H}_1}}{2^{\ell_1}}\right)\left(1 - \frac{q_{\mathcal{H}_1}}{2^{\ell_1} - 1}\right) \leq \frac{2q_{\mathcal{H}_1}}{2^{\ell_1} - 1},$$

where $q_{\mathcal{H}_1}$ is the number of different inputs to be queried to the $\mathcal{O}^{\mathcal{H}_1}$ oracle. Whereas the size of the plaintext space is exponential in the security parameter, $q_{\mathcal{H}_1}$ is polynomial in the security parameter, and hence $\Pr[\mathcal{F}_2]$ is negligible in the security parameter.

We denote the event that $\mathcal{F}_1$ or $\mathcal{F}_2$ occur by $\mathcal{F}$, i.e., $\mathcal{F} = \mathcal{F}_1 \vee \mathcal{F}_2$. In the above simulation, $\mathcal{B}$ outputs $\beta' = \beta$ in the following two cases:

- $\mathsf{E}_{\mathcal{A},1}$: the event that $\mathcal{A}$ correctly outputs m_β. In this case, $\mathcal{B}$ correctly answers with probability 1. If the event $\mathcal{F}$ does not occur, then our simulation is correct and the event $\mathsf{E}_{\mathcal{A},1}$ directly implies the success of $\mathcal{A}$, so that we can utilize $\mathcal{A}$ for our purpose. However, if the event $\mathcal{F}$ occurs, then we cannot expect the adversarial behaviour from our simulation, but in that case the adversarial success probability is bounded by 1 at most. Therefore, we have the following bound for the adversarial success probability $\epsilon_{\mathcal{A}}$.

$$\Pr[\mathsf{E}_{\mathcal{A},1} \wedge \neg\mathcal{F}] \leq \epsilon_{\mathcal{A}} \leq \Pr[\mathsf{E}_{\mathcal{A},1} \wedge \neg\mathcal{F}] + \Pr[\mathcal{F}] \tag{1}$$

- $\mathsf{E}_{\mathcal{A},2}$: the event that $\mathcal{A}$ outputs neither m_β nor $m_{1-\beta}$. In this case, $\mathcal{B}$ correctly answers with probability $\frac{1}{2}$. With the similar argument to the above, we have the following bound for probability $1 - \epsilon_{\mathcal{A}} - \frac{1}{2^{\ell_1}}$, which is that $\mathcal{A}$ outputs neither m_β nor $m_{1-\beta}$, not in the simulation, but in the real game.

$$\Pr[\mathsf{E}_{\mathcal{A},2} \wedge \neg\mathcal{F}] \leq 1 - \epsilon_{\mathcal{A}} - \frac{1}{2^{\ell_1}} \leq \Pr[\mathsf{E}_{\mathcal{A},2} \wedge \neg\mathcal{F}] + \Pr[\mathcal{F}] \tag{2}$$

Hence,

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{IND\text{-}CCA2}}_{\mathcal{B}} &= \Pr[\beta = \beta'] - \frac{1}{2} \\
&\geq \Pr[\beta = \beta' \wedge \neg\mathcal{F}] - \frac{1}{2} \\
&= 1 \cdot \Pr[\mathsf{E}_{\mathcal{A},1} \wedge \neg\mathcal{F}] + \frac{1}{2} \cdot \Pr[\mathsf{E}_{\mathcal{A},2} \wedge \neg\mathcal{F}] - \frac{1}{2} \\
&\geq \epsilon_{\mathcal{A}} - \Pr[\mathcal{F}] + \frac{1}{2}\left(1 - \epsilon_{\mathcal{A}} - \frac{1}{2^{\ell_1}} - \Pr[\mathcal{F}]\right) - \frac{1}{2}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2} + \frac{\epsilon_{\mathcal{A}}}{2} - \frac{1}{2^{\ell_1+1}} - \frac{3}{2}\Pr[\mathcal{F}] - \frac{1}{2} \\
&\geq \frac{1}{2} + \frac{\epsilon_{\mathcal{A}}}{2} - \frac{1}{2^{\ell_1+1}} - \frac{3}{2}(\mathbf{Adv}^{\mathsf{CDH}}_{\mathcal{A},\mathbb{G}} + \frac{2q_{\mathcal{H}_1}}{2^{\ell_1}-1}) - \frac{1}{2} \\
&= \frac{\epsilon_{\mathcal{A}}}{2} - \frac{3}{2}\mathbf{Adv}^{\mathsf{CDH}}_{\mathcal{A},\mathbb{G}} - \mathsf{negl}(\lambda) \qquad (3)
\end{aligned}
$$

and therefore

$$
\begin{aligned}
\epsilon_{\mathcal{A}} &= \mathbf{Adv}^{\mathsf{OW\text{-}CCA2}}_{\mathsf{OURS}} \\
&\leq 2\mathbf{Adv}^{\mathsf{IND\text{-}CCA2}}_{\mathsf{PKE1}} + 3\mathbf{Adv}^{\mathsf{CDH}}_{\mathbb{G}} + \mathsf{negl}(\lambda).
\end{aligned}
$$

Since we assume that the PKE1 scheme is IND-CCA2 secure and the CDH assumption on $\mathbb{G}$ holds, our construction is OW-CCA2 secure. □

4.2. IND-CCA2 security against type-II adversaries

Now, we look into the IND-CCA2 security of our construction against Type-II adversaries.

Theorem 3. *Our construction in Section 3.1 is IND-CCA2 secure against PPT Type-II adversaries under assuming that the exploited encryption schemes* PKE1 *and* PKE2 *are IND-CCA2 secure, the CDH assumption on $\mathbb{G}$ holds, and $\mathcal{H}_1$ and $\mathcal{H}_2$ are modelled as random oracles.*

Proof. We first define an original IND-CCA2 security game of our construction and its modification. Then, we will demonstrate that the advantages of any PPT adversaries of those games are negligible in the security parameter by showing that the sum and difference of the advantages of adversaries of those games are negligible in the security parameter.

Game 0. This game is the same as the original IND-CCA2 security game in Definition 4.

1. The challenger $\mathcal{C}$ takes a security parameter λ as an input, runs $\mathsf{PKEET.Setup}(\lambda) \rightarrow pp$, and sends the public parameter pp to the adversary $\mathcal{A}$. Then, he runs $\mathsf{PKEET.KeyGen}(pp) \rightarrow (pk_i, sk_i)$ for $1 \leq i \leq N$ and sends all pk_i's to the adversary $\mathcal{A}$.
2. For $\mathcal{A}$'s queries to the oracles $\mathcal{O}^{\mathcal{H}_1}$, $\mathcal{O}^{\mathcal{H}_2}$, $\mathcal{O}^{sk}$, and $\mathcal{O}^{\mathsf{td}}$, $\mathcal{C}$ responds as $\mathcal{B}$ in Step 2 of the security game in the proof of Theorem 2. We note that the trapdoor query to the oracle $\mathcal{O}^{\mathsf{td}}$ on input t is not allowed in this game. For queries to the oracle $\mathcal{O}^{\mathsf{Dec}}$, $\mathcal{C}$ responds as follows:
   - $\mathcal{O}^{\mathsf{Dec}}$ query: On input a ciphertext $C_i = (\mathsf{C}_{i,1}, \mathsf{C}_{i,2}, \mathsf{C}_{i,3})$ under the public key pk_i, it runs $\mathsf{PKEET.Dec}(sk_i, C_i) \rightarrow m'$ and returns m'.
3. $\mathcal{A}$ selects two messages $m_0, m_1 \in \{0,1\}^{\ell_1}$ and sends $\mathcal{C}$ them. $\mathcal{C}$ selects a random bit $b \in \{0, 1\}$, runs $\mathsf{PKEET.Enc}(pk_t, m_b) \rightarrow C^*_{t,b} = (\mathsf{C}^*_{t,b,1}, \mathsf{C}^*_{t,b,2}, \mathsf{C}^*_{t,b,3})$ and sends $C^*_{t,b}$ to $\mathcal{A}$.
4. For $\mathcal{A}$'s oracle queries, $\mathcal{C}$ responds as in Step 2. The constraints are that
   (a) the index t does not appear as an input in the $\mathcal{O}^{sk}$ and $\mathcal{O}^{\mathsf{td}}$ oracle queries;
   (b) a pair of the index t and the ciphertext $C^*_{t,b}$ does not appear as an input in the $\mathcal{O}^{\mathsf{Dec}}$ oracle queries.
5. $\mathcal{A}$ outputs $b' \in \{0, 1\}$.

Game 1. This game is almost the same as **Game** 0, except for the challenge ciphertext. In the challenge phase, once $\mathcal{C}$ receives two messages m_0, m_1 from $\mathcal{A}$, $\mathcal{C}$ selects a random bit $b \in \{0, 1\}$, and generates a challenge ciphertext as the followings:

1. Select a random element r from $\mathbb{Z}^*_p$.
2. Run $\mathsf{PKE1.Enc}(\mathsf{pk}_{t,1}, m_b \| g^r) \rightarrow \mathsf{C}^*_{t,b,1}$.
3. Run $\mathsf{PKE2.Enc}(\mathsf{pk}_{t,2}, \mathcal{H}_1(m_{1-b}) \| g^r) \rightarrow \mathsf{C}^*_{t,1-b,2}$.
4. Compute $\mathsf{C}^*_{t,b,3} = \mathcal{H}_2(\mathsf{C}^*_{t,b,1}, \mathsf{C}^*_{t,1-b,2}, X^r_t)$.

and send $C^*_{t,b} = (\mathsf{C}^*_{t,b,1}, \mathsf{C}^*_{t,1-b,2}, \mathsf{C}^*_{t,b,3})$ to $\mathcal{A}$.

Let ϵ_i be the advantage of $\mathcal{A}$ in **Game** i, i.e., $\epsilon_i = \Pr[\mathcal{A} \text{ outputs } b] - \frac{1}{2}$. We remark that the probability ϵ_1 is related to the event that the adversary $\mathcal{A}$ in **Game** 1 outputs the index b of the first component $\mathsf{C}^*_{t,b,1}$ of the challenge ciphertext $C^*_{t,b}$, not the second component $\mathsf{C}^*_{t,1-b,2}$.

The following lemmas will demonstrate that $\epsilon_0 + \epsilon_1$ and $\epsilon_0 - \epsilon_1$ are negligible in the security parameter under assuming that the exploited PKE1 and PKE2 schemes are IND-CCA2 secure, respectively, the CDH assumption on $\mathbb{G}$ holds, and $\mathcal{H}_1$ and $\mathcal{H}_2$ are modelled as random oracles. As a result, we conclude that the advantage ϵ_0 of any adversaries in the original security game **Game** 0, is negligible in the security parameter.

Lemma 1. *Assuming that the employed* PKE1 *is IND-CCA2 secure, the CDH assumption on $\mathbb{G}$ holds, and $\mathcal{H}_1$ and $\mathcal{H}_2$ are modelled as random oracles, $\epsilon_0 + \epsilon_1$ is negligible in the security parameter.*

Proof. Let $\mathcal{A}_1$ be a PPT adversary who breaks the IND-CCA2 security of our construction. Then, we can construct a PPT algorithm $\mathcal{B}_1$ who breaks the IND-CCA2 security of the PKE1 scheme using $\mathcal{A}_1$ as follows:

1. On receiving a public key pk^* from the challenger $\mathcal{C}_1$ of the IND-CCA2 security game of the PKE1 scheme, $\mathcal{B}_1$ runs as $\mathcal{B}$ in the proof of Theorem 2.
2. For $\mathcal{A}_1$'s queries to the oracles, $\mathcal{B}_1$ responds as $\mathcal{B}$ in the proof of Theorem 2 with decryption oracle queries of $\mathcal{C}_1$ if needed. The only difference between this step and that of the proof of Theorem 2 is that t does not appear as an input in the $\mathcal{O}^{\mathsf{td}}$ oracle queries in this game.
3. On receiving two messages $m_0, m_1 \in \{0,1\}^{\ell_1}$ from $\mathcal{A}_1$, $\mathcal{B}_1$ randomly selects r from $\mathbb{Z}_p^*$ and forwards $m_0 \| g^r$ and $m_1 \| g^r$ to $\mathcal{C}_1$. Then, $\mathcal{C}_1$ may select a random bit $\beta \in \{0,1\}$, run $\mathsf{PKE1.Enc}(\mathsf{pk}^*, m_\beta \| g^r) \rightarrow \mathsf{C}^*_\beta$, and send C^*_β to $\mathcal{B}_1$.
4. On receiving C^*_β from $\mathcal{C}_1$, $\mathcal{B}_1$ selects a random bit $b \in \{0,1\}$, sets $\mathsf{C}^*_{t,b,1} = \mathsf{C}^*_\beta$, runs $\mathsf{PKE2.Enc}(\mathsf{pk}_{t,2}, \mathcal{H}_1(m_b) \| g^r) \rightarrow \mathsf{C}^*_{t,b,2}$, and computes $\mathsf{C}^*_{t,b,3} = \mathcal{H}_2(\mathsf{C}^*_{t,b,1}, \mathsf{C}^*_{t,b,2}, X_t^r)$. Then, $\mathcal{B}_1$ returns $C^*_{t,b} = (\mathsf{C}^*_{t,b,1}, \mathsf{C}^*_{t,b,2}, \mathsf{C}^*_{t,b,3})$ to $\mathcal{A}_1$.
5. $\mathcal{B}_1$ responds to $\mathcal{A}_1$'s oracle queries as in Step 2 of this game, except for the decryption oracle queries on ciphertexts of the form $(\mathsf{C}^*_{t,1}, \mathsf{C}_{t,2}, \mathsf{C}_{t,3})$. For the decryption oracle queries on ciphertexts of form $(\mathsf{C}^*_{t,1}, \mathsf{C}_{t,2}, \mathsf{C}_{t,3})$, it outputs $\perp$.
6. On receiving $\mathcal{A}_1$'s output b', $\mathcal{B}_1$ outputs $\beta' = b'$.

We note that the above simulation may fail when the event $\mathcal{F}_1$ defined in the proof of Theorem 2 occurs. As the analysis in the proof of Theorem 2, $\Pr[\mathcal{F}_1] \leq \mathbf{Adv}^{\mathsf{CDH}}_{\mathcal{A}_1,\mathbb{G}}$ and it is negligible in the security parameter because we assume that the CDH assumption on $\mathbb{G}$ holds.

Now, let us evaluate the advantage of $\mathcal{B}_1$. Because b is randomly chosen by $\mathcal{B}_1$, $\Pr[\beta = b] = \Pr[\beta \neq b] = \frac{1}{2}$. Further, if $\mathcal{B}_1$ correctly guesses β, i.e., $\beta = b$, then the challenge ciphertext C^*_t for $\mathcal{A}_1$ has the form of

$$\begin{aligned} C^*_{t,b} &= (\mathsf{C}^*_{t,b,1}, \mathsf{C}^*_{t,b,2}, \mathsf{C}^*_{t,b,3}) \\ &= (\mathsf{PKE1.Enc}(\mathsf{pk}^*, m_\beta \| g^r), \mathsf{PKE2.Enc}(\mathsf{pk}_{t,2}, \mathcal{H}_1(m_\beta) \| g^r), \mathsf{C}^*_{t,b,3}), \end{aligned}$$

which has the same form of the challenge ciphertext in **Game** 0. Hence, when $\mathcal{A}_1$ correctly answers so that $b' = b$, $\mathcal{B}'$ outputs the correct answer $\beta' = b' = b = \beta$.

If $\mathcal{B}_1$ does not correctly guess β, i.e., $\beta \neq b$, then the challenge ciphertext C^*_t has the form of

$$\begin{aligned} C^*_{t,b} &= (\mathsf{C}^*_{t,b,1}, \mathsf{C}^*_{t,b,2}, \mathsf{C}^*_{t,b,3}) \\ &= (\mathsf{PKE1.Enc}(\mathsf{pk}^*, m_\beta \| g^r), \mathsf{PKE2.Enc}(\mathsf{pk}_{t,2}, \mathcal{H}_1(m_{1-\beta}) \| g^r), \mathsf{C}^*_{t,b,3}), \end{aligned}$$

which has the same form of the challenge ciphertext in **Game** 1. Hence, when $\mathcal{A}_1$ outputs the index related to the first component of the challenge ciphertext in **Game** 1, β' is the same as β. Hence,

$$\begin{aligned} &\mathbf{Adv}^{\mathsf{IND\text{-}CCA2}}_{\mathsf{PKE1},\mathcal{B}_1} \\ &= \Pr[\beta = \beta'] - \frac{1}{2} \\ &\geq \Pr[\beta = \beta' | \neg\mathcal{F}_1] \Pr[\neg\mathcal{F}_1] - \frac{1}{2} \\ &= \Big(\Pr[\mathcal{A} \text{ outputs } \beta \text{ in } \mathbf{Game\ 0} | \neg\mathcal{F}_1 \wedge \beta = b] \Pr[\beta = b] \\ &\quad + \Pr[\mathcal{A} \text{ outputs } \beta \text{ in } \mathbf{Game\ 1} | \neg\mathcal{F}_1 \wedge \beta \neq b] \Pr[\beta \neq b] \Big) \Pr[\neg\mathcal{F}_1] - \frac{1}{2} \\ &\geq \left(\left(\frac{1}{2} + \epsilon_0 - \Pr[\mathcal{F}_1] \right) \frac{1}{2} + \left(\frac{1}{2} + \epsilon_1 - \Pr[\mathcal{F}_1] \right) \frac{1}{2} \right) (1 - \Pr[\mathcal{F}_1]) - \frac{1}{2} \\ &\geq \left(1 - \mathbf{Adv}^{\mathsf{CDH}}_{\mathcal{A}_1,\mathbb{G}} \right) \frac{\epsilon_0 + \epsilon_1}{2} - \frac{3\mathbf{Adv}^{\mathsf{CDH}}_{\mathcal{A}_1,\mathbb{G}}}{2} + \mathsf{negl}(\lambda). \end{aligned} \tag{4}$$

Since the PKE1 scheme is IND-CCA2 secure, the advantage of $\mathcal{B}_1$ is negligible in the security parameter. Therefore, $\epsilon_0 + \epsilon_1$ is also negligible in the security parameter. □

Lemma 2. *Assuming that the exploited* PKE2 *is IND-CCA2 secure, the CDH assumption on $\mathbb{G}$ holds, and $\mathcal{H}_1$ and $\mathcal{H}_2$ are modelled as random oracles, $\epsilon_0 - \epsilon_1$ is negligible in the security parameter.*

Proof. The proof of this lemma is very similar to that of Lemma 1. Let $\mathcal{A}_2$ be a PPT adversary who breaks the IND-CCA2 security of our construction. Then, we can construct a PPT algorithm $\mathcal{B}_2$ who breaks the IND-CCA2 security of the PKE2 scheme using $\mathcal{A}_2$ as follows:

1. On receiving a public key pk^* from the challenger $\mathcal{C}_2$ of the IND-CCA2 security game of the PKE2 scheme, $\mathcal{B}_2$ performs as follows:
   (a) Run $\mathsf{PKE1.KeyGen}(\lambda) \rightarrow (\mathsf{pk}_{i,1}, \mathsf{sk}_{i,1})$ for $1 \leq i \leq N$.

(b) Run $\mathsf{PKE2.KeyGen}(\lambda) \to (\mathsf{pk}_{i,2}, \mathsf{sk}_{i,2})$ for $1 \le i \le N$, $i \ne t$, and set $\mathsf{pk}_{t,2} = \mathsf{pk}^*$.
(c) Generate a cyclic group $\mathbb{G}$ of prime order p with a generator g. Select a random element x_i from $\mathbb{Z}_p^*$ and compute $X_i = g^{x_i}$ for $1 \le i \le N$.
(d) Send all $pk_i = (pp, \mathsf{pk}_{i,1}, \mathsf{pk}_{i,2}, X_i)$'s for $1 \le i \le N$ to $\mathcal{A}$.

2. For $\mathcal{A}_2$'s queries to the oracles $\mathcal{O}^{\mathcal{H}_1}$, $\mathcal{O}^{\mathcal{H}_2}$, $\mathcal{O}^{sk}$, and $\mathcal{O}^{\mathsf{td}}$, $\mathcal{B}_2$ responds as $\mathcal{B}_1$ in Step 2 of Lemma 1. For queries to $\mathcal{O}^{\mathsf{Dec}}$, $\mathcal{B}_2$ responds as follows:
   - $\mathcal{O}^{\mathsf{Dec}}$ query: On input a pair of an index and a ciphertext (i, C_i) for $C_i = (\mathsf{C}_{i,1}, \mathsf{C}_{i,2}, \mathsf{C}_{i,3})$,
     - if $i \ne t$, it runs $\mathsf{PKEET.Dec}(sk_i, C_i) \to m'$ and returns m';
     - otherwise, $\mathcal{B}_2$ queries $\mathsf{C}_{t,2}$ to the decryption oracle of the IND-CCA2 security game of the PKE2 scheme. Once returning $h' \| R''$, $\mathcal{B}_2$ runs $\mathsf{PKE1.Dec}(\mathsf{sk}_{t,1}, \mathsf{C}_{t,1}) \to m' \| R'$ and checks whether $h' = \mathcal{H}_1(m')$, $R' = R''$, and $\mathsf{C}_{t,3} = \mathcal{H}_2(\mathsf{C}_{t,1}, \mathsf{C}_{t,2}, (R')^{x_t})$. If all of them hold, it returns m' to $\mathcal{A}_2$. Otherwise, it returns $\perp$.
3. On receiving two messages $m_0, m_1 \in \{0,1\}^{\ell_1}$ from $\mathcal{A}_2$, $\mathcal{B}_2$ randomly selects r from $\mathbb{Z}_p^*$ and forwards $\mathcal{H}_1(m_0)\|g^r$, $\mathcal{H}_1(m_1)\|g^r$ to $\mathcal{C}_2$. Then, $\mathcal{C}_2$ may select a random bit $\beta \in \{0, 1\}$, run $\mathsf{PKE2.Enc}(\mathsf{pk}^*, \mathcal{H}_1(m_\beta)\|g^r) \to \mathsf{C}^*_\beta$, and send C^*_β to $\mathcal{B}_2$.
4. On receiving C^*_β from $\mathcal{C}_2$, $\mathcal{B}_2$ selects a random bit $b \in \{0, 1\}$, sets $\mathsf{C}^*_{t,b,2} = \mathsf{C}^*_\beta$, runs $\mathsf{PKE1.Enc}(\mathsf{pk}_{t,1}, m_b\|g^r) \to \mathsf{C}^*_{t,b,1}$, and computes $\mathsf{C}^*_{t,b,3} = \mathcal{H}_2(\mathsf{C}^*_{t,b,1}, \mathsf{C}^*_{t,b,2}, X_t^r)$. Then, $\mathcal{B}_2$ returns $C^*_{t,b} = (\mathsf{C}^*_{t,b,1}, \mathsf{C}^*_{t,b,2}, \mathsf{C}^*_{t,b,3})$ to $\mathcal{A}_2$.
5. $\mathcal{B}_2$ responds to $\mathcal{A}_2$'s oracle queries as in Step 2 of this game, except for the decryption oracle queries on ciphertexts of the form $(\mathsf{C}_{t,1}, \mathsf{C}^*_{t,2}, \mathsf{C}_{t,3})$. For the decryption oracle queries on ciphertexts of form $(\mathsf{C}_{t,1}, \mathsf{C}^*_{t,2}, \mathsf{C}_{t,3})$, it outputs $\perp$.
6. On receiving $\mathcal{A}_2$'s output b', $\mathcal{B}_2$ outputs $\beta' = b'$.

As the similar reason in the proof of Lemma 1, the above simulation may fail when the event $\mathcal{F}_1$ defined in the proof of Theorem 2 occurs and $\Pr[\mathcal{F}_1] \le \mathbf{Adv}^{\mathsf{CDH}}_{\mathcal{A}_2,\mathbb{G}}$.

Because b is randomly chosen by $\mathcal{B}_2$, $\Pr[\beta = b] = \Pr[\beta \ne b] = \frac{1}{2}$. Further, if $\mathcal{B}_2$ correctly guesses β, i.e., $\beta = b$, then the challenge ciphertext C^*_t for $\mathcal{A}_2$ has the form of

$$\begin{aligned} C^*_{t,b} &= (\mathsf{C}^*_{t,b,1}, \mathsf{C}^*_{t,b,2}, \mathsf{C}^*_{t,b,3}) \\ &= (\mathsf{PKE1.Enc}(\mathsf{pk}^*, m_\beta\|g^r), \mathsf{PKE2.Enc}(\mathsf{pk}_{t,2}, \mathcal{H}_1(m_\beta)\|g^r), \mathsf{C}^*_{t,b,3}), \end{aligned}$$

which has the same form of the challenge ciphertext in **Game** 0. Hence, when $\mathcal{A}_1$ correctly answers so that $b' = b$, $\mathcal{B}'$ outputs the correct answer $\beta' = b' = b = \beta$.

If $\mathcal{B}_2$ does not correctly guess β, i.e., $\beta \ne b$, then the challenge ciphertext C^*_t has the form of

$$\begin{aligned} C^*_{t,b} &= (\mathsf{C}^*_{t,b,1}, \mathsf{C}^*_{t,b,2}, \mathsf{C}^*_{t,b,3}) \\ &= (\mathsf{PKE1.Enc}(\mathsf{pk}^*, m_{1-\beta}\|g^r), \mathsf{PKE2.Enc}(\mathsf{pk}_{t,2}, \mathcal{H}_1(m_\beta)\|g^r), \mathsf{C}^*_{t,b,3}), \end{aligned}$$

which has the same form of the challenge ciphertext in **Game** 1. Hence, when $\mathcal{A}_1$ outputs the index related to the second component of the challenge ciphertext in **Game** 1, β' is the same as β. The probability of such the event is $\frac{1}{2} - \epsilon_1$ because we define ϵ_1 by $\Pr[\mathcal{A} \text{ outputs } 1-\beta] - \frac{1}{2}$. Hence,

$$\begin{aligned} &\mathbf{Adv}^{\mathsf{IND\text{-}CCA2}}_{\mathsf{PKE2},\mathcal{B}_2} \\ &= \Pr[\beta = \beta'] - \frac{1}{2} \\ &\ge \Pr[\beta = \beta' | \neg\mathcal{F}_1]\Pr[\neg\mathcal{F}_1] - \frac{1}{2} \\ &= \Big(\Pr[\mathcal{A} \text{ outputs } \beta \text{ in } \mathbf{Game\ 0} | \neg F_1 \wedge \beta = b]\Pr[\beta = b] \\ &\quad + \Pr[\mathcal{A} \text{ outputs } \beta \text{ in } \mathbf{Game\ 1} | \neg\mathcal{F}_1 \wedge \beta \ne b]\Pr[\beta \ne b]\Big)\Pr[\neg\mathcal{F}_1] - \frac{1}{2} \\ &\ge \left(\left(\frac{1}{2} + \epsilon_0 - \Pr[\mathcal{F}_1]\right)\frac{1}{2} + \left(\frac{1}{2} - \epsilon_1 - \Pr[\mathcal{F}_1]\right)\frac{1}{2}\right)(1 - \Pr[\mathcal{F}_1]) - \frac{1}{2} \\ &\ge \left(1 - \mathbf{Adv}^{\mathsf{CDH}}_{\mathcal{A}_2,\mathbb{G}}\right)\frac{\epsilon_0 - \epsilon_1}{2} - \frac{3\mathbf{Adv}^{\mathsf{CDH}}_{\mathcal{A}_2,\mathbb{G}}}{2} + \mathsf{negl}(\lambda). \end{aligned} \tag{5}$$

Since we assume that the PKE2 scheme is IND-CCA2 secure, the advantage of $\mathcal{B}_2$ is negligible in the security parameter. Therefore, $\epsilon_0 - \epsilon_1$ is also negligible in the security parameter. □

From Lemma 1 and 2, both $\epsilon_0+\epsilon_1$ and $\epsilon_0-\epsilon_1$ are negligible in the security parameter. Hence, ϵ_0 should be negligible in the security parameter. More precisely, from the relations (4) and (5),

$$\begin{aligned}\epsilon_0 &= \mathbf{Adv}^{\text{IND-CCA2}}_{\text{OURS}}\\ &\le \left(\frac{1}{1-\mathbf{Adv}^{\text{CDH}}_{\mathbb{G}}}\right)\left(\mathbf{Adv}^{\text{IND-CCA2}}_{\text{PKE1}}+\mathbf{Adv}^{\text{IND-CCA2}}_{\text{PKE2}}+3\mathbf{Adv}^{\text{CDH}}_{\mathbb{G}}\right)+\text{negl}(\lambda).\end{aligned}$$

Since we assume that the exploited PKE1 and PKE2 schemes are IND-CCA2 secure and the CDH assumption on $\mathbb{G}$ holds, our construction is IND-CCA2 secure. □

5. Our semi-generic construction of IBE with equality test

In this section, we present our semi-generic IBEET construction that exploits traditional IBE schemes. Then, we show that our construction is OW-ID-CCA2 secure against Type-I adversaries and IND-ID-CCA2 secure against Type-II adversaries in the random oracle model.

5.1. Our semi-generic IBEET construction

Now, we provide our semi-generic IBEET construction by exploiting two traditional IBE schemes, $\text{IBE1}=(\text{IBE1.Setup}, \text{IBE1.Extract}, \text{IBE1.Enc}, \text{IBE1.Dec})$ and $\text{IBE2}=(\text{IBE2.Setup}, \text{IBE2.Extract}, \text{IBE2.Enc}, \text{IBE2.Dec})$. We note that one may employ the same IBE scheme for IBE1 and IBE2 as in our PKEET constructions. The description is as follows:

- $\text{IBEET.Setup}(\lambda)$: It takes a security parameter λ as an input and runs $\text{IBE1.Setup}(\lambda)\to(pp_1, msk_1)$ and $\text{IBE2.Setup}(\lambda)\to(pp_2, msk_2)$. Let $\mathbb{G}$ and $\mathbb{G}_T$ be cyclic groups of prime order $p=p(\lambda)$. Let $e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$ be a bilinear map and g be a generator of the group $\mathbb{G}$. It randomly selects a random element s from $\mathbb{Z}_p^*$ and set $g_1=g^s$. Let $\mathcal{H}_1:\{0,1\}^{\ell_1}\to\{0,1\}^{\ell_2}$, $\mathcal{H}_2:\{0,1\}^*\to\{0,1\}^{\ell_3}$, and $\mathcal{H}_3:\{0,1\}^*\to\mathbb{G}$ be cryptographic hash functions for integers $\ell_1=\ell_1(\lambda)$, $\ell_2=\ell_2(\lambda)$, and $\ell_3=\ell_3(\lambda)$. It outputs a public parameter pp and a master secret key msk,

$$\begin{aligned}pp &= (pp_1, pp_2, \mathbb{G}, \mathbb{G}_T, e, g, g_1, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3),\\ msk &= (msk_1, msk_2, s).\end{aligned}$$

- $\text{IBEET.Extract}(pp, msk, \text{ID})$: It takes the public parameter pp, the master secret key msk, and an identity $\text{ID}\in\{0,1\}^*$ as inputs, and performs as follows:
  1. Run $\text{IBE1.Extract}(pp_1, msk_1, \text{ID})\to d_{\text{ID},1}$.
  2. Run $\text{IBE2.Extract}(pp_2, msk_2, \text{ID})\to d_{\text{ID},2}$.
  3. Compute $d_{\text{ID},3}=h_{\text{ID}}^s$ where $h_{\text{ID}}=\mathcal{H}_3(\text{ID})$.
  4. Output $d_{\text{ID}}=(d_{\text{ID},1}, d_{\text{ID},2}, d_{\text{ID},3})$.
- $\text{IBEET.Enc}(pp, \text{ID}, m)$: It takes the public parameter pp, an identity ID, and a message $m\in\{0,1\}^{\ell_1}$ as inputs, and performs as follows:
  1. Select a random element r from $\mathbb{Z}_p^*$.
  2. Run $\text{IBE1.Enc}(pp_1, \text{ID}, m\|g^r)\to \text{C}_1$.
  3. Run $\text{IBE2.Enc}(pp_2, \text{ID}, \mathcal{H}_1(m)\|g^r)\to \text{C}_2$.
  4. Compute $\text{C}_3=\mathcal{H}_2(\text{C}_1, \text{C}_2, e(h_{\text{ID}}, g_1)^r)$ and output $C=(\text{C}_1, \text{C}_2, \text{C}_3)$.
- $\text{IBEET.Dec}(pp, d_{\text{ID}}, C)$: It takes the public parameter pp, the secret key $d_{\text{ID}}=(d_{\text{ID},1}, d_{\text{ID},2}, d_{\text{ID},3})$ of the identity ID, and a ciphertext $C=(\text{C}_1, \text{C}_2, \text{C}_3)$ as inputs, and performs as follows:
  1. Run $\text{IBE1.Dec}(pp_1, d_{\text{ID},1}, \text{C}_1)\to m'\|R'$.
  2. Run $\text{IBE2.Dec}(pp_2, d_{\text{ID},2}, \text{C}_2)\to h'\|R''$.
  3. Check whether $h'=\mathcal{H}_1(m')$, $R'=R''$, and $\text{C}_3=\mathcal{H}_2(\text{C}_1, \text{C}_2, e(d_{\text{ID},3}, R'))$.
  4. If all of them hold, it outputs m'. Otherwise, output $\perp$.
- $\text{IBEET.Trapdoor}(d_{\text{ID}})$: It takes the secret key $d_{\text{ID}}=(d_{\text{ID},1}, d_{\text{ID},2}, d_{\text{ID},3})$ of the identity ID as an input and outputs a trapdoor $\text{td}_{\text{ID}}=d_{\text{ID},2}$ for the identity ID.
- $\text{IBEET.Test}(\text{td}_{\text{ID}_i}, C_{\text{ID}_i}, \text{td}_{\text{ID}_j}, C_{\text{ID}_j})$: It takes a ciphertext $C_{\text{ID}_i}=(\text{C}_{\text{ID}_i,1}, \text{C}_{\text{ID}_i,2}, \text{C}_{\text{ID}_i,3})$ of a user ID_i with his/her trapdoor td_{ID_i} and a ciphertext $C_{\text{ID}_j}=(\text{C}_{\text{ID}_j,1}, \text{C}_{\text{ID}_j,2}, \text{C}_{\text{ID}_j,3})$ of a user ID_j with his/her trapdoor td_{ID_j} as inputs. It performs as follows:
  1. Run $\text{IBE2.Dec}(pp_2, \text{td}_{\text{ID}_i}, \text{C}_{\text{ID}_i,2})\to h'\|R'$.
  2. Run $\text{IBE2.Dec}(pp_2, \text{td}_{\text{ID}_j}, \text{C}_{\text{ID}_j,2})\to h''\|R''$.
  3. If $h'=h''$, then output 1. Otherwise, output 0.

Correctness. The following theorem demonstrates the correctness of our semi-generic IBEET construction.

Theorem 4. *Our construction in Section 5.1 is correct according to Definition 6 under assuming the exploited* IBE1 *and* IBE2 *schemes are correct and $\mathcal{H}_1$ is a collision-resistant hash function.*

Proof. Let $C_{\mathsf{ID}_i} = (C_{\mathsf{ID}_i,1}, C_{\mathsf{ID}_i,2}, C_{\mathsf{ID}_i,3})$ be a valid ciphertext generated by running $\mathsf{IBEET.Enc}(pp, \mathsf{ID}_i, m_i)$. Then, $\mathsf{IBE1.Dec}(pp_1, d_{\mathsf{ID}_i,1}, C_{\mathsf{ID}_i,1}) \rightarrow m_i \| g^r$ and $\mathsf{IBE2.Dec}(pp_2, d_{\mathsf{ID}_i,2}, C_{\mathsf{ID}_i,2}) \rightarrow \mathcal{H}_1(m_i) \| g^r$ for some $r \in \mathbb{Z}_p^*$, because IBE1 and IBE2 are correct. Moreover, since $e(h_{\mathsf{ID}_i}, g_1)^r = e(h_{\mathsf{ID}_i}^s, g^r) = e(d_{\mathsf{ID}_i,3}, g^r)$, $\mathcal{H}_2(C_{\mathsf{ID}_i,1}, C_{\mathsf{ID}_i,2}, e(d_{\mathsf{ID}_i,3}, g^r))$ is the same as $C_{\mathsf{ID}_i,3} = \mathcal{H}_2(C_{\mathsf{ID}_i,1}, C_{\mathsf{ID}_i,2}, e(h_{\mathsf{ID}_i}, g_1)^r)$. Hence, the first condition for the correctness holds.

Let $C_{\mathsf{ID}_j} = (C_{\mathsf{ID}_j,1}, C_{\mathsf{ID}_j,2}, C_{\mathsf{ID}_j,3})$ be a valid ciphertext generated by running $\mathsf{IBEET.Enc}(pp, \mathsf{ID}_j, m_j)$. For the IBEET.Test algorithm, when the trapdoors $\mathsf{td}_{\mathsf{ID}_i}$ and $\mathsf{td}_{\mathsf{ID}_j}$ for users ID_i and ID_j are given, if $m_i = m_j$, then it outputs 1 since $\mathsf{IBE2.Dec}(pp_2, \mathsf{td}_{\mathsf{ID}_i}, C_{\mathsf{ID}_i,2}) \rightarrow \mathcal{H}_1(m_i) \| R_i$, $\mathsf{IBE2.Dec}(pp_2, \mathsf{td}_{\mathsf{ID}_j}, C_{\mathsf{ID}_j,2}) \rightarrow \mathcal{H}_1(m_j) \| R_j$, and $\mathcal{H}_1(m_i) = \mathcal{H}_1(m_j)$. Otherwise, $\mathcal{H}_1(m_i)$ is different from $\mathcal{H}_1(m_j)$ with overwhelming probability since $\mathcal{H}_1$ is a collision-resistant hash function. Therefore, both the second and third conditions for the correctness hold. □

5.2. Security analysis of our IBEET construction

In this subsection, we explore the security of our IBEET construction. The following theorem demonstrates that our IBEET construction is OW-ID-CCA2 secure against Type-I adversaries.

Theorem 5. *Our construction in Section 5.1 is OW-ID-CCA2 secure against PPT Type-I adversaries, under assuming that the exploited* IBE1 *scheme is IND-ID-CCA2 secure, the BDH assumption holds, and $\mathcal{H}_1$, $\mathcal{H}_2$, and $\mathcal{H}_3$ are modelled as random oracles.*

Sketch of Proof. The proof of this theorem is very similar to that of Theorem 2. As in the proof of Theorem 2, we suppose that there exists a PPT Type-I adversary $\mathcal{A}$ who breaks the OW-ID-CCA2 security of our IBEET construction. Then, we construct a PPT algorithm $\mathcal{B}$ who breaks the IND-ID-CCA2 security of the IBE1 scheme using $\mathcal{A}$. The main differences are as follows:

- While $\mathcal{O}^{sk}$ oracle queries are controlled by $\mathcal{B}$ itself in the proof of Theorem 2, the key extraction queries $\mathcal{O}^{\mathsf{Ext}}$ are served with the aid of the key extraction oracle of the IND-ID-CCA2 security game of the IBE1 scheme, to extract the secret key of the IBE1 scheme in this proof.
- For $\mathcal{O}^{\mathsf{Dec}}$ queries, while $\mathcal{B}$ in the proof of Theorem 2 queries to the decryption oracle on ciphertexts only under the public key pk_t, $\mathcal{B}$ in this proof queries to it on all ciphertexts.
- Additionally, $\mathcal{O}^{\mathcal{H}_3}$ oracle queries are offered as follows: At the initial step, it selects a random element x from $\mathbb{Z}_p^*$ and sets $\bar{g} = g^x$. On input an identity ID, it returns h_{ID} if $(\mathsf{ID}, r_{\mathsf{ID}}, h_{\mathsf{ID}})$ has previously been stored in the hash list $L^{\mathcal{H}_3}$, which was originally initiated as an empty set. Otherwise, it randomly selects r_{ID} from $\mathbb{Z}_p^*$, computes $h_{\mathsf{ID}} = \bar{g}^{r_{\mathsf{ID}}}$, adds $(\mathsf{ID}, r_{\mathsf{ID}}, h_{\mathsf{ID}})$ to the hash list $L^{\mathcal{H}_3}$, and returns h_{ID}.
- Similarly to the event denoted by $\mathcal{F}_1$ in the proof of Theorem 2, we denote by $\mathcal{G}_1$ the event that a ciphertext of the form $(C_{\mathsf{ID}^*,1}^*, C_{\mathsf{ID}^*,2}, C_{\mathsf{ID}^*,3})$ queried to the decryption oracle, is valid where the challenge ciphertext is $(C_{\mathsf{ID}^*,1}^*, C_{\mathsf{ID}^*,2}^*, C_{\mathsf{ID}^*,3}^*)$. Then, we can assume that if $\mathcal{A}$ detects a wrong simulation, then $\mathcal{A}$ can solve the BDH problem on the instance (g, g^x, g^r, g^s) by computing $(e(h_{\mathsf{ID}^*}, g_1)^r)^{r_{\mathsf{ID}^*}^{-1}}$ where $h_{\mathsf{ID}^*} = \bar{g}^{r_{\mathsf{ID}^*}} = (g^x)^{r_{\mathsf{ID}^*}}$. Hence, we can assume that $\Pr[\mathcal{G}_1] \leq \mathbf{Adv}_{\mathcal{A}}^{\mathrm{BDH}}$ is negligible in the security parameter.

Then, we complete the proof of this theorem by following the similar step to that of the proof of Theorem 2. The full proof of Theorem 5 is given in Appendix A. □

Next, we demonstrate that our IBEET construction also achieves the IND-ID-CCA2 security against Type-II adversaries.

Theorem 6. *Our construction in Section 5.1 is IND-ID-CCA2 secure against PPT Type-II adversaries, under assuming that the exploited* IBE1 *and* IBE2 *schemes are IND-ID-CCA2 secure, the BDH assumption holds, and $\mathcal{H}_1$, $\mathcal{H}_2$, and $\mathcal{H}_3$ are modelled as random oracles.*

Sketch of Proof. The proof of this theorem is very similar with that of Theorem 3, which demonstrates the IND-CCA2 security of our PKEET scheme. Similarly to the proof of Theorem 3, we can prove by defining an original IND-ID-CCA2 security game of our IBEET construction and its slight modification, and then demonstrating that the advantages of any PPT adversaries of those games are negligible in the security parameter. We can easily obtain the full proof of this theorem from properly combining the proofs of Theorem 3 and Theorem 5 by considering the identity-based setting. We provide the full proof of this theorem in Appendix B. □

Table 1
Comparison of Our PKEET with existing schemes.

		[19]	[18]	[12]†	Ours
Comp of	Enc	3Exp	5Exp	6Exp	6Exp + 2SE
	Dec	3Exp	2Exp	5Exp	3Exp + 2SE
	Test	2Pairing	4Exp	2Pairing + 2Exp	2Exp + 2SE
Size of	PK	$\|\mathbb{G}\|$	$2\|\mathbb{G}\|$	$3\|\mathbb{G}\|$	$3\|\mathbb{G}\|$
	CT	$3\|\mathbb{G}\| + \|\mathbb{Z}_p\|$	$4\|\mathbb{G}\| + \|\mathbb{Z}_p\| + 2\lambda$	$5\|\mathbb{G}\| + \|\mathbb{Z}_p\|$	$2\|\mathbb{G}\| + 10\lambda$
	TD	-	$\|\mathbb{Z}_p\|$	$\|\mathbb{Z}_p\|$	$\|\mathbb{Z}_p\|$
Security	Type-I	OW-CCA2	OW-CCA2	OW-CCA2	OW-CCA2
	Type-II	-	IND-CCA2	IND-CCA2	IND-CCA2
Assumptions		CDH	CDH	CDH	CDH

†The scheme in [12] originally supports four types of authorization policies for equality tests, but we consider Type-1 authorization only for the above comparison, which has the same functionality with ours. Legends: Comp: computational complexity, Enc: encryption algorithm, Dec: decryption algorithm, Test: test algorithm, PK: public key, CT: ciphertext, TD: trapdoor, Exp: cost for an exponentiation, Pairing: cost for a pairing computation, SE: cost for symmetric encryption (or decryption), λ: security parameter, CDH: computational Diffie-Hellman assumption.

6. Comparisons

In this section, we provide comparisons of our PKEET and IBEET constructions with existing other related works.

6.1. Comparison with previous PKEET schemes

We first present a comparison of our PKEET construction with previous PKEET schemes which support the same functionality with ours. For PKE schemes in our PKEET construction, we exploit the outcome of the Fujisaki and Okamoto (FO) conversion [6] by employing the hashed ElGamal encryption scheme [7] as an asymmetric encryption scheme and AES [5] as a symmetric encryption scheme. The hashed ElGamal encryption scheme requires two exponentiations for encryption and one exponentiation for decryption[2], and its ciphertext consists of one group element and one string whose size is the same as a hash output. In addition to those of the hashed ElGamal encryption, the outcome of the FO conversion using the hashed ElGamal encryption and AES requires costs of the AES encryption and decryption algorithms for encryption and decryption, respectively. A ciphertext of the FO conversion just adds a ciphertext of AES to a ciphertext of the hashed ElGamal encryption. We note that the hashed ElGamal encryption scheme is indistinguishable against chosen plaintext attacks (IND-CPA) under CDH assumptions in the random oracle model and so the outcome of the FO conversion is IND-CCA2 secure if CDH assumptions hold and AES is one-way.

Table 1 provides a comparison between our construction and other related ones. The second, third, fourth, and last columns describe the features of Yang et al.'s PKEET [19], Tang's all-or-nothing PKEET [18], Ma et al.'s PKEET [12], and ours, respectively. We set the size of hash values and the ciphertext size of AES to 2λ, respectively, for the security parameter λ. The table shows that the performance of our construction is slightly worse than the previous best result [18]. This seems to be because our construction is semi-generic and so its performance relies heavily on those of the underlying PKE schemes. Hence, we believe that it could be further improved if a more efficient IND-CCA2 secure PKE scheme is provided.

6.2. Comparison with previous IBEET schemes

Now, we provide a comparison of our IBEET construction with an IBEKS scheme and the existing IBEET scheme [11]. For IBEKS, we consider Abdalla et al.'s generic transformation [1] by exploiting Park and Lee's anonymous hierarchical IBE scheme [14] as in [11]. For our IBEET construction, we exploit Boneh and Franklin's IND-ID-CCA2 secure IBE scheme [4], which requires two exponentiations for encryption, and one pairing computation and one exponentiation for decryption.[3]

Table 2 gives a comparison between our and other related constructions. The second, third, and fourth columns describe the features of an IBEKS scheme exploiting Park and Lee's anonymous hierarchical IBE, Ma's IBEET scheme, and our scheme exploiting Boneh-Franklin's IBE, respectively. In terms of computational complexity, the table shows that our decryption algorithm is slightly slower than the one given by Ma. However, the test algorithm is made faster by considering that in general a pairing computation is more expensive than an exponentiation. We also confirm that the size of our public key, ciphertext, and trapdoor are comparable to those used in Ma's scheme. Finally, we remark that our construction is the only IBEET scheme that achieves the IND-ID-CCA2 security against Type-II adversaries.

[2] We ignore the cost of the hash computation, because in general it is much cheaper than exponentiation.
[3] We ignore the cost of the hash computation, because in general it is much cheaper than exponentiation and pairing operations.

Table 2
Comparison of IBEKS and IBEETs.

		IBEKS ([1]+[14])	IBEET ([11])	Ours (with BF-IBE [4])
Comp of	Enc	7Exp	6Exp	6Exp
	Dec	-	2Pairing+2Exp	3Pairing+2Exp
	Test	4Pairing	4Pairing	2Pairing+2Exp
Size of	PK	$15\mathbb{G}$	$2\mathbb{G}$	$4\mathbb{G}$
	CT	$4\mathbb{G} + \mathbb{G}_T$	$4\mathbb{G} + \mathbb{Z}_p$	$2\mathbb{G}$+5$\|\mathcal{H}\|$
	TD	$12\mathbb{G}$	$\mathbb{G}$	$\mathbb{G}$
Fun	KS	Yes	Yes	Yes
	ET	No	Yes	Yes
Security		IND-ID-CPA	OW-ID-CCA2	IND-ID-CCA2
ROM		No	Yes	Yes
Assumption		ℓ-DBDHE & aug ℓ-DL	BDH	BDH

Legends: IBEKS: identity-based encryption with keyword search, IBEET: identity-based encryption with equality test, BF-IBE: Boneh and Franklin's identity-based encryption, Comp: computational complexity, Fun: Functionality, ROM: random oracle model, Enc: encryption algorithm, Dec: decryption algorithm, Test: test algorithm, PK: public key, CT: ciphertext, TD: trapdoor, KS: keyword search, ET: equality test, Exp: Exponentiation, Pairing: Pairing, $\mathbb{G}, \mathbb{G}_T$: two cyclic group of order p for a pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, $\mathcal{H}$: hash function, ℓ-DBDHE: decisional ℓ-bilinear Diffie-Hellman exponent assumption, aug ℓ-DL: augmented Decisional linear assumption, BDH: bilinear Diffie-Hellman assumption.

7. Conclusion

In this paper, we have provided a semi-generic construction of public key encryption with equality test by exploiting traditional public key encryption schemes. Our construction is OW-CCA2 secure against Type-I adversaries who have the trapdoor for equality test, and IND-CCA2 secure against Type-II adversaries who do not have this information. This is shown assuming that the exploited public key encryption schemes are IND-CCA2 secure and the CDH assumption holds in the random oracle model.

We have also provided an identity-based version of our construction by replacing public key encryption schemes and the CDH assumption with identity-based schemes and the BDH assumption, respectively. As a result, we have obtained the first identity-based encryption scheme with equality test that achieves both OW-ID-CCA2 security against Type-I adversaries and IND-ID-CCA2 security against Type-II adversaries.

In future research, we may consider developing more generic constructions which do not require the CDH or BDH assumptions. It may also be interesting to design generic constructions of public key encryption and identity-based encryption with equality test that support flexible authorization, as done for the scheme in [12].

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments. Hyung Tae Lee, San Ling, and Huaxiong Wang were supported by Research Grant TL-9014101684-01 and the Singapore Ministry of Education under Research Grant MOE2013-T2-1-041. Huaxiong Wang was also supported by NTU under Tier 1 grant RG143/14. Jae Hong Seo was supported by ETRI R&D program (15ZS1500).

Appendix A. Proof of Theorem 5

The proof of Theorem 5 is very similar as that of Theorem 2. Let $\mathcal{A}$ be a PPT Type-I adversary who breaks the OW-ID-CCA2 security of our IBEET construction with advantage $\epsilon_{\mathcal{A}}$. Then, we can construct a PPT algorithm $\mathcal{B}$ who breaks the IND-ID-CCA2 security of the IBE1 scheme using $\mathcal{A}$ as follows:

1. On receiving a public parameter pp_1 from the challenger $\mathcal{C}$ of the IND-ID-CCA2 security game of the IBE1 scheme, $\mathcal{B}$ performs as follows:
   (a) Run IBE2.Setup$(\lambda) \to (pp_2, msk_2)$.
   (b) Generate cyclic groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order p. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map. Select a generator g of the group $\mathbb{G}$. Choose a random element s from $\mathbb{Z}_p^*$ and set $g_1 = g^s$.
   (c) Send $pp = (pp_1, pp_2, \mathbb{G}, \mathbb{G}_T, e, g, g_1)$ to $\mathcal{A}$.
2. For $\mathcal{A}$'s oracle queries, $\mathcal{B}$ responds as follows:
   - $\mathcal{O}^{\mathcal{H}_1}, \mathcal{O}^{\mathcal{H}_2}$ queries: These are the same with those in the proof of Theorem 2, which proves the OW-CCA2 security of our PKEET construction.

- $\mathcal{O}^{\mathcal{H}_3}$ query: At the initial step, it selects a random element x from $\mathbb{Z}_p^*$ and sets $\bar{g} = g^x$. On input an identity ID, it returns h_{ID} if $(\mathsf{ID}, r_{\mathsf{ID}}, h_{\mathsf{ID}})$ has previously been stored in the hash list $L^{\mathcal{H}_3}$, which was originally initiated as an empty set. Otherwise, it randomly selects r_{ID} from $\mathbb{Z}_p^*$, computes $h_{\mathsf{ID}} = \bar{g}^{r_{\mathsf{ID}}}$, adds $(\mathsf{ID}, r_{\mathsf{ID}}, h_{\mathsf{ID}})$ to the hash list $L^{\mathcal{H}_3}$, and returns h_{ID}.
- $\mathcal{O}^{\mathsf{Ext}}$ query: On input an identity ID,
  (a) $\mathcal{B}$ queries ID to the key extraction oracle of the IND-ID-CCA2 security game of the IBE1 scheme and then receives $d_{\mathsf{ID},1}$.
  (b) $\mathcal{B}$ runs $\mathsf{IBE2.Extract}(pp_2, msk_2, \mathsf{ID}) \to d_{\mathsf{ID},2}$.
  (c) $\mathcal{B}$ computes $d_{\mathsf{ID},3} = h_{\mathsf{ID}}^s$ where $h_{\mathsf{ID}} = \mathcal{H}_3(\mathsf{ID})$ (obtained by $\mathcal{O}^{\mathcal{H}_3}$ queries).
  (d) $\mathcal{B}$ sends $d_{\mathsf{ID}} = (d_{\mathsf{ID},1}, d_{\mathsf{ID},2}, d_{\mathsf{ID},3})$ to $\mathcal{A}$.
- $\mathcal{O}^{\mathsf{td}}$ query: On input an identity ID, it runs $\mathsf{IBEET.Trapdoor}(d_{\mathsf{ID}}) \to \mathsf{td}_{\mathsf{ID}}$ and returns $\mathsf{td}_{\mathsf{ID}}$.
- $\mathcal{O}^{\mathsf{Dec}}$ query: On input a pair of an identity and a ciphertext $(\mathsf{ID}, \mathsf{C}_{\mathsf{ID}})$ for $\mathsf{C}_{\mathsf{ID}} = (\mathsf{C}_{\mathsf{ID},1}, \mathsf{C}_{\mathsf{ID},2}, \mathsf{C}_{\mathsf{ID},3})$,
  (a) $\mathcal{B}$ queries $\mathsf{C}_{\mathsf{ID},1}$ to the decryption oracle of the IND-ID-CCA2 security game of the IBE1 scheme and then receives $m' \| R'$.
  (b) $\mathcal{B}$ runs $\mathsf{IBE2.Dec}(pp_2, d_{\mathsf{ID},2}, \mathsf{C}_{\mathsf{ID},2}) \to h_1' \| R''$ and check whether $h_1' = \mathcal{H}_1(m')$, $R' = R''$, and $\mathsf{C}_{\mathsf{ID},3} = \mathcal{H}_2(\mathsf{C}_{\mathsf{ID},1}, \mathsf{C}_{\mathsf{ID},2}, e(d_{\mathsf{ID},3}, R'))$. If all of them hold, it returns m'. Otherwise, it returns $\perp$.

3. Once receiving the challenge identity ID^* from $\mathcal{A}$, $\mathcal{B}$ selects two random messages m_0, m_1 from $\{0,1\}^{\ell_1}$ and r from $\mathbb{Z}_p^*$, and sends ID^*, $m_0 \| g^r$, and $m_1 \| g^r$ to $\mathcal{C}$. Then, $\mathcal{C}$ may select a random bit $\beta \in \{0,1\}$, run
$$\mathsf{IBE1.Enc}(pp_1, \mathsf{ID}^*, m_\beta \| g^r) \to \mathsf{C}^*_{\mathsf{ID}^*,1},$$
and send it to $\mathcal{B}$.
4. Once $\mathcal{B}$ receives the challenge ciphertext $\mathsf{C}^*_{\mathsf{ID}^*,1}$, he selects a random element $h_1^* \in \{0,1\}^{\ell_2}$, runs
$$\mathsf{IBE2.Enc}(pp_2, \mathsf{ID}^*, h_1^* \| g^r) \to \mathsf{C}^*_{\mathsf{ID}^*,2},$$
and computes $\mathsf{C}^*_{\mathsf{ID}^*,3} = \mathcal{H}_2(\mathsf{C}^*_{\mathsf{ID}^*,1}, \mathsf{C}^*_{\mathsf{ID}^*,2}, e(h_{\mathsf{ID}^*}, g_1)^r)$. $\mathcal{B}$ sends $\mathsf{C}^*_{\mathsf{ID}^*} = (\mathsf{C}^*_{\mathsf{ID}^*,1}, \mathsf{C}^*_{\mathsf{ID}^*,2}, \mathsf{C}^*_{\mathsf{ID}^*,3})$ to $\mathcal{A}$ as the challenge ciphertext.
5. $\mathcal{B}$ responds to $\mathcal{A}$'s oracle queries as the same as in Step 2, except for the decryption oracle queries on ciphertexts of form $(\mathsf{C}^*_{\mathsf{ID}^*,1}, \mathsf{C}_{\mathsf{ID}^*,2}, \mathsf{C}_{\mathsf{ID}^*,3})$. For the decryption oracle queries on ciphertexts of form $(\mathsf{C}^*_{\mathsf{ID}^*,1}, \mathsf{C}_{\mathsf{ID}^*,2}, \mathsf{C}_{\mathsf{ID}^*,3})$, $\mathcal{B}$ outputs $\perp$.
We note that the key extraction query on the challenge identity ID^* and the decryption query on the pair of the challenge identity ID^* and the challenge ciphertext $C^*_{\mathsf{ID}^*}$, are not allowed by the definition of the security game.
6. $\mathcal{A}$ outputs m'. If $m' = m_b$ for $b = 0, 1$, $\mathcal{B}$ outputs $\beta' = b$. Otherwise, it outputs a random bit $\beta' \in \{0, 1\}$.

Let us evaluate $\mathcal{B}$'s advantage. We first note that $\mathcal{B}$'s simulation may fail in the following two cases:

- Let $\mathcal{G}_1$ be the event that a ciphertext of the form $(\mathsf{C}^*_{\mathsf{ID}^*,1}, \mathsf{C}_{\mathsf{ID}^*,2}, \mathsf{C}_{\mathsf{ID}^*,3})$ queried to the decryption oracle in Step 5, is valid. In this case, we assume that the decryption oracle returns $\perp$ in the above simulation. However, in case when $\mathsf{IBE2.Enc}(pp_2, \mathsf{ID}^*, \bar{h}_1 \| g^r) \to \mathsf{C}_{\mathsf{ID}^*,2}$ and $\mathsf{C}_{\mathsf{ID}^*,3} = \mathcal{H}_2(\mathsf{C}^*_{\mathsf{ID}^*,1}, \mathsf{C}_{\mathsf{ID}^*,2}, e(h_{\mathsf{ID}^*}, g_1)^r))$ for some $\bar{h}_1 \in \{0,1\}^{\ell_2}$, this ciphertext becomes valid. To generate such a ciphertext, $\mathcal{A}$ must query to the oracle $\mathcal{O}^{\mathcal{H}_2}$ on an input $(\mathsf{C}^*_{\mathsf{ID}^*,1}, \mathsf{C}_{\mathsf{ID}^*,2}, e(h_{\mathsf{ID}^*}, g_1)^r)$. Here, we can assume that if $\mathcal{A}$ detects such a wrong simulation, then he can solve a BDH problem on an instance (g, g^x, g^r, g^s) by computing $(e(h_{\mathsf{ID}^*}, g_1)^r)^{r_{\mathsf{ID}^*}^{-1}}$, because
$$(e(h_{\mathsf{ID}^*}, g_1)^r)^{r_{\mathsf{ID}^*}^{-1}} = (e(g^{xr_{\mathsf{ID}^*}}, g^s)^r)^{r_{\mathsf{ID}^*}^{-1}} = e(g,g)^{xsr}.$$
Since we assume that the BDH assumption holds, $\Pr[\mathcal{G}_1] \le \mathbf{Adv}_{\mathcal{A}}^{\mathrm{BDH}}$ is negligible in the security parameter.
- Let $\mathcal{G}_2$ be the event that m_0 or m_1 are queried to the oracle $\mathcal{O}^{\mathcal{H}_1}$. If m_0 or m_1 are queried and the outputs of the oracle queries on m_0 or m_1 are not $\bar{h}_1$, then the challenge ciphertext for $\mathcal{A}$ may not be valid. We note that the probability that $\mathcal{G}_2$ occurs is
$$\Pr[\mathcal{G}_2] = 1 - \binom{2^{\ell_1} - 2}{q_{\mathcal{H}_1}} \Big/ \binom{2^{\ell_1}}{q_{\mathcal{H}_1}} = 1 - \left(1 - \frac{q_{\mathcal{H}_1}}{2^{\ell_1}}\right)\left(1 - \frac{q_{\mathcal{H}_1}}{2^{\ell_1} - 1}\right) \le \frac{2q_{\mathcal{H}_1}}{2^{\ell_1} - 1},$$
where $q_{\mathcal{H}_1}$ is the number of different inputs to be queried to the $\mathcal{O}^{\mathcal{H}_1}$ oracle. Whereas the size of the plaintext space is exponential in the security parameter, $q_{\mathcal{H}_1}$ is polynomial in the security parameter, and hence $\Pr[\mathcal{G}_2]$ is negligible in the security parameter.

We denote the event that $\mathcal{G}_1$ or $\mathcal{G}_2$ occur by $\mathcal{G}$, i.e., $\mathcal{G} = \mathcal{G}_1 \vee \mathcal{G}_2$. In the above simulation, $\mathcal{B}$ outputs $\beta' = \beta$ in the following two cases:

- $\mathsf{E}_{\mathcal{A},1}$: the event that $\mathcal{A}$ correctly outputs m_β. In this case, $\mathcal{B}$ correctly answers with probability 1 and it holds that
$$\Pr[\mathsf{E}_{\mathcal{A},1} \wedge \neg\mathcal{G}] \ge \epsilon_{\mathcal{A}} - \Pr[\mathcal{G}].$$

- $\mathsf{E}_{\mathcal{A},2}$: the event that $\mathcal{A}$ outputs neither m_β nor $m_{1-\beta}$. In this case, $\mathcal{B}$ correctly answers with probability $\frac{1}{2}$ and it holds that

$$\Pr[\mathsf{E}_{\mathcal{A},2} \wedge \neg\mathcal{G}] \leq 1 - \epsilon_{\mathcal{A}} - \frac{1}{2^{\ell_2}} \leq \Pr[\mathsf{E}_{\mathcal{A},2} \wedge \neg\mathcal{G}] + \Pr[\mathcal{G}].$$

Here, both the above two bounds are obtained by the same argument in Eqs. (1) and (2). Then, the remaining analysis is exactly the same as in the proof of Theorem 2. That is, we have

$$\mathbf{Adv}^{\text{IND-ID-CCA2}}_{\text{IBE1},\mathcal{B}} \geq \frac{\epsilon_{\mathcal{A}}}{2} - \frac{3}{2}\mathbf{Adv}^{\text{BDH}}_{\mathcal{A}} - \text{negl}(\lambda)$$

and therefore

$$\begin{aligned}\epsilon_{\mathcal{A}} &= \mathbf{Adv}^{\text{OW-ID-CCA2}}_{\text{OURS},\mathcal{A}} \\ &\leq 2\mathbf{Adv}^{\text{IND-ID-CCA2}}_{\text{IBE1}} + 3\mathbf{Adv}^{\text{BDH}}_{\mathcal{A}} + \text{negl}(\lambda).\end{aligned}$$

Since we assume that the IBE1 scheme is IND-ID-CCA2 secure and the BDH assumption holds, our construction is OW-ID-CCA2 secure. □

Appendix B. Proof of Theorem 6

The proof of this theorem is very similar with that of Theorem 3. Similarly to the proof of Theorem 3, we will first define an original IND-ID-CCA2 security game of our construction and its slight modification. Then, we will demonstrate that the advantages of any PPT adversaries of those games are negligible in the security parameters. The original game and its modification are defined as follows:

Game 0. This game is the same as the original IND-ID-CCA2 security game in Definition 8.

1. The challenger $\mathcal{C}$ takes a security parameter λ as an input, runs $\text{IBEET.Setup}(\lambda) \rightarrow (pp, msk)$, sends the public parameter pp to the adversary $\mathcal{A}$, and keeps the master secret key msk private.
2. For $\mathcal{A}$'s queries to the oracles $\mathcal{O}^{\mathcal{H}_1}$, $\mathcal{O}^{\mathcal{H}_2}$, $\mathcal{O}^{\mathcal{H}_3}$, and $\mathcal{O}^{\text{td}}$, the challenger $\mathcal{C}$ responds as $\mathcal{B}$ in Step 2 of the security game in the proof of Theorem 5. For queries to the oracles $\mathcal{O}^{\text{Ext}}$ and $\mathcal{O}^{\text{Dec}}$, $\mathcal{C}$ responds as follows:
   - $\mathcal{O}^{\text{Ext}}$ query: On input an identity ID, it runs $\text{IBEET.Extract}(pp, msk, \text{ID}) \rightarrow d_{\text{ID}}$ and returns d_{ID}.
   - $\mathcal{O}^{\text{Dec}}$ query: On input a ciphertext $C_{\text{ID}} = (C_{\text{ID},1}, C_{\text{ID},2}, C_{\text{ID},3})$ of a user ID, it runs $\text{IBEET.Dec}(pp, d_{\text{ID}}, C_{\text{ID}}) \rightarrow m'$ and returns m'.
3. $\mathcal{A}$ selects an identity ID^*, which was never queried to the $\mathcal{O}^{\text{Ext}}$ oracle and $\mathcal{O}^{\text{td}}$ oracle in Step 2, and two messages m_0 and m_1, and sends them to $\mathcal{C}$. $\mathcal{C}$ selects a random bit $b \in \{0, 1\}$, runs $\text{IBEET.Enc}(pp, \text{ID}^*, m_b) \rightarrow C^*_{\text{ID}^*,b}$, and sends $C^*_{\text{ID}^*,b}$ to $\mathcal{A}$.
4. For $\mathcal{A}$'s oracle queries, $\mathcal{C}$ responds as in Step 2. The constraints are that
   (a) the identity ID^* does not appear as an input in the $\mathcal{O}^{\text{Ext}}$ and $\mathcal{O}^{\text{td}}$ oracle queries;
   (b) a pair of the identity ID^* and the ciphertext $C^*_{\text{ID}^*,b}$ does not appear as an input in the $\mathcal{O}^{\text{Dec}}$ oracle queries.
5. $\mathcal{A}$ outputs $b \in \{0, 1\}$.

Game 1. This game is almost the same as **Game** 0, except for the challenge ciphertext. In the challenge phase, once $\mathcal{C}$ receives the identity ID^* and two messages m_0, m_1 from $\mathcal{A}$, he selects a random bit $b \in \{0, 1\}$, and generates a challenge ciphertext as follows:

1. Select a random element r from $\mathbb{Z}^*_p$.
2. Run $\text{IBE1.Enc}(pp_1, \text{ID}^*, m_b \| g^r) \rightarrow C^*_{\text{ID}^*,b,1}$.
3. Run $\text{IBE2.Enc}(pp_2, \text{ID}^*, \mathcal{H}_1(m_{1-b}) \| g^r) \rightarrow C^*_{\text{ID}^*,1-b,2}$.
4. Compute $C^*_{\text{ID}^*,b,3} = \mathcal{H}_2(C^*_{\text{ID}^*,b,1}, C^*_{\text{ID}^*,1-b,2}, e(h_{\text{ID}^*}, g_1)^r)$,

and sends $C^*_{\text{ID}^*,b} = (C^*_{\text{ID}^*,b,1}, C^*_{\text{ID}^*,1-b,2}, C^*_{\text{ID}^*,b,3})$ to $\mathcal{A}$.

Let ϵ_i be the advantage of $\mathcal{A}$ in **Game** i, i.e., $\epsilon_i = \Pr[\mathcal{A} \text{ outputs } b] - \frac{1}{2}$. We remark that the probability ϵ_1 is related to the event that $\mathcal{A}$ in **Game** 1 outputs the index b of the first component $C^*_{\text{ID}^*,b,1}$ of the challenge ciphertext $C^*_{\text{ID}^*,b}$, not that of the second component $C^*_{\text{ID}^*,1-b,2}$.

The following lemmas will demonstrate that $\epsilon_0 + \epsilon_1$ and $\epsilon_0 - \epsilon_1$ are negligible in the security parameter under assuming that the exploited IBE1 and IBE2 schemes are IND-ID-CCA2 secure, respectively, the BDH assumption holds, and $\mathcal{H}_1$, $\mathcal{H}_2$, and $\mathcal{H}_3$ are modelled as random oracles. As a result, we conclude that the advantage ϵ_0 of any adversaries in the original security game **Game** 0, is negligible in the security parameter.

Lemma 3. *Assuming that the exploited* IBE1 *is IND-ID-CCA2 secure, the BDH assumption holds, and $\mathcal{H}_1$, $\mathcal{H}_2$, and $\mathcal{H}_3$ are modelled as random oracles, $\epsilon_0 + \epsilon_1$ is negligible in the security parameter.*

Proof. Let $\mathcal{A}_1$ be a PPT adversary who breaks the IND-ID-CCA2 security of our construction. Then, we can construct a PPT algorithm $\mathcal{B}_1$ who breaks the IND-ID-CCA2 security of the IBE1 scheme using $\mathcal{A}_1$ as follows:

1. On receiving the public parameter pp_1 from the challenger $\mathcal{C}_1$ of the IND-ID-CCA2 security game of the IBE1 scheme, $\mathcal{B}_1$ runs as $\mathcal{B}$ in the proof of Theorem 5.
2. For $\mathcal{A}_1$'s queries to the oracles, $\mathcal{B}_1$ responds as $\mathcal{B}$ in the proof of Theorem 5 with the key extraction and the decryption oracle queries of $\mathcal{C}_1$ if needed.
3. On receiving the challenge identity ID^* and two messages $m_0, m_1 \in \{0,1\}^{\ell_1}$ from $\mathcal{A}_1$, $\mathcal{B}_1$ randomly selects r from $\mathbb{Z}_p^*$ and forwards ID^*, $m_0\|g^r$ and $m_1\|g^r$ to $\mathcal{C}_1$. Subsequently, $\mathcal{C}_1$ may select a random bit $\beta \in \{0, 1\}$, run $\mathsf{IBE1.Enc}(pp_1, \mathsf{ID}^*, m_\beta\|g^r) \to \mathsf{C}^*_\beta$, and send C^*_β to $\mathcal{B}_1$.
4. On receiving C^*_β from $\mathcal{C}_1$, $\mathcal{B}_1$ selects a random bit $b \in \{0, 1\}$, sets $\mathsf{C}^*_{\mathsf{ID}^*,b,1} = \mathsf{C}^*_\beta$, runs $\mathsf{IBE2.Enc}(pp_2, \mathsf{ID}^*, \mathcal{H}_1(m_b)\|g^r) \to \mathsf{C}^*_{\mathsf{ID}^*,b,2}$, and computes $\mathsf{C}^*_{\mathsf{ID}^*,b,3} = \mathcal{H}_2(\mathsf{C}^*_{\mathsf{ID}^*,b,1}, \mathsf{C}^*_{\mathsf{ID}^*,b,2}, e(h_{\mathsf{ID}^*}, g_1)^r)$. $\mathcal{B}_1$ returns $\mathsf{C}^*_{\mathsf{ID}^*,b} = (\mathsf{C}^*_{\mathsf{ID},b,1}, \mathsf{C}^*_{\mathsf{ID},b,2}, \mathsf{C}^*_{\mathsf{ID},b,3})$ to $\mathcal{A}_1$.
5. $\mathcal{B}_1$ responds to $\mathcal{A}_1$'s oracle queries as in Step 2 of this game, except for the decryption oracle queries on ciphertexts of the form $(\mathsf{C}^*_{\mathsf{ID}^*,b,1}, \mathsf{C}_{\mathsf{ID}^*,b,2}, \mathsf{C}_{\mathsf{ID}^*,b,3})$. For the decryption oracle queries on ciphertexts of form $(\mathsf{C}^*_{\mathsf{ID}^*,b,1}, \mathsf{C}_{\mathsf{ID}^*,b,2}, \mathsf{C}_{\mathsf{ID}^*,b,3})$, it outputs $\perp$.
6. On receiving $\mathcal{A}_1$'s output b', $\mathcal{B}_1$ outputs $\beta' = b'$.

We note that the above simulation may fail when the event $\mathcal{G}_1$ defined in the proof of Theorem 5 occurs. As the analysis in the proof of Theorem 5, $\Pr[\mathcal{G}_1] = \mathbf{Adv}^{\mathrm{BDH}}_{\mathcal{A}_1}$ and it is negligible in the security parameter because we assume that the BDH assumption holds.

Now, let us evaluate the advantage of $\mathcal{B}_1$. Because b is randomly chosen by $\mathcal{B}_1$, $\Pr[\beta = b] = \Pr[\beta \neq b] = \frac{1}{2}$. Further, if $\mathcal{B}_1$ correctly guesses β, i.e., $\beta = b$, then the challenge ciphertext $C^*_{\mathsf{ID}^*,b}$ for $\mathcal{A}_1$ has the form of

$$\begin{aligned} C^*_{\mathsf{ID}^*,b} &= (\mathsf{C}^*_{\mathsf{ID}^*,b,1}, \mathsf{C}^*_{\mathsf{ID}^*,b,2}, \mathsf{C}^*_{\mathsf{ID}^*,b,3}) \\ &= (\mathsf{IBE1.Enc}(pp_1, \mathsf{ID}^*, m_\beta\|g^r), \mathsf{IBE2.Enc}(pp_2, \mathsf{ID}^*, \mathcal{H}_1(m_\beta)\|g^r), \mathsf{C}^*_{\mathsf{ID}^*,b,3}), \end{aligned}$$

which has the same form of the challenge ciphertext in **Game** 0. Hence, when $\mathcal{A}_1$ correctly answers so that $b' = b$, $\mathcal{B}'$ outputs the correct answer $\beta' = b' = b = \beta$.

If $\mathcal{B}_1$ does not correctly guess β, i.e., $\beta \neq b$, then the challenge ciphertext $C^*_{\mathsf{ID}^*,b}$ has the form of

$$\begin{aligned} C^*_{\mathsf{ID}^*,b} &= (\mathsf{C}^*_{\mathsf{ID}^*,b,1}, \mathsf{C}^*_{\mathsf{ID}^*,1-b,2}, \mathsf{C}^*_{\mathsf{ID}^*,b,3}) \\ &= (\mathsf{IBE1.Enc}(pp_1, \mathsf{ID}^*, m_\beta\|g^r), \mathsf{IBE2.Enc}(pp_2, \mathsf{ID}^*, \mathcal{H}_1(m_{1-\beta})\|g^r), \mathsf{C}^*_{\mathsf{ID}^*,b,3}), \end{aligned}$$

which has the same form of the challenge ciphertext in **Game** 1. Hence, when $\mathcal{A}_1$ outputs the index related to the first component of the challenge ciphertext in **Game** 1, β' is the same as β. Hence,

$$\begin{aligned} &\mathbf{Adv}^{\mathsf{IND\text{-}ID\text{-}CCA2}}_{\mathcal{B}_1} \\ &= \Pr[\beta = \beta'] - \frac{1}{2} \\ &\geq \Pr[\beta = \beta' | \neg\mathcal{G}_1]\Pr[\neg\mathcal{G}_1] - \frac{1}{2} \\ &= \Big(\Pr[\mathcal{A} \text{ outputs } \beta \text{ in} \mathbf{Game\ 0} | \neg\mathcal{G}_1 \wedge \beta = b]\Pr[\beta = b] \\ &\quad + \Pr[\mathcal{A} \text{ outputs } \beta \text{ in } \mathbf{Game\ 1} | \neg\mathcal{G}_1 \wedge \beta \neq b]\Pr[\beta \neq b]\Big)\Pr[\neg\mathcal{G}_1] - \frac{1}{2} \\ &\geq \left(\left(\frac{1}{2} + \epsilon_0 - \Pr[\mathcal{G}_1]\right)\frac{1}{2} + \left(\frac{1}{2} + \epsilon_1 - \Pr[\mathcal{G}_1]\right)\frac{1}{2}\right)(1 - \Pr[\mathcal{G}_1]) - \frac{1}{2} \\ &\geq \left(1 - \mathbf{Adv}^{\mathrm{BDH}}_{\mathcal{A}_1}\right)\frac{\epsilon_0 + \epsilon_1}{2} - \frac{3\mathbf{Adv}^{\mathrm{BDH}}_{\mathcal{A}_1}}{2} + \mathsf{negl}(\lambda). \end{aligned} \tag{B.1}$$

Since the IBE1 scheme is IND-ID-CCA2 secure, the advantage of $\mathcal{B}_1$ is negligible in the security parameter. Therefore, $\epsilon_0 + \epsilon_1$ is also negligible in the security parameter. □

Lemma 4. *Assuming that the exploited* IBE2 *is IND-ID-CCA2 secure, the BDH assumption holds, and $\mathcal{H}_1$, $\mathcal{H}_2$, and $\mathcal{H}_3$ are modelled as random oracles, $\epsilon_0 - \epsilon_1$ is negligible in the security parameter.*

Proof. The proof of this lemma is very similar to that of Lemma 3. Let $\mathcal{A}_2$ be a PPT adversary who breaks the IND-ID-CCA2 security of our construction. Then, we can construct a PPT algorithm $\mathcal{B}_2$ who breaks the IND-ID-CCA2 security of the IBE2 scheme using $\mathcal{A}_2$ as follows:

1. On receiving a public parameter pp_2 from the challenger $\mathcal{C}_2$ of the IND-ID-CCA2 security game of the IBE2 scheme, $\mathcal{B}_2$ performs as follows:

(a) Run $\mathsf{IBE1.Setup}(\lambda) \rightarrow (pp_1, msk_1)$.
(b) Generate cyclic groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order p. Let $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear map. Select a generator g of the group $\mathbb{G}$. Choose a random element s from $\mathbb{Z}_p^*$ and set $g_1 = g^s$.
(c) Send $pp = (pp_1, pp_2, \mathbb{G}, \mathbb{G}_T, e, g, g_1)$ to $\mathcal{A}_2$.

2. For $\mathcal{A}_2$'s queries to the oracles $\mathcal{O}^{\mathcal{H}_1}$, $\mathcal{O}^{\mathcal{H}_2}$, and $\mathcal{O}^{\mathcal{H}_3}$, $\mathcal{B}_2$ responds as $\mathcal{B}_1$ in Step 2 $\mathcal{B}_1$'s behaviors in the proof of Lemma 3. For queries to $\mathcal{O}^{\mathsf{Ext}}$, $\mathcal{O}^{\mathsf{td}}$, and $\mathcal{O}^{\mathsf{Dec}}$, $\mathcal{B}_2$ responds as follows:
   - $\mathcal{O}^{\mathsf{Ext}}$ query: On input an identity ID,
     (a) $\mathcal{B}_2$ queries ID to the key extraction oracle of the IND-ID-CCA2 security game of the IBE2 scheme and then receives $d_{\mathsf{ID},2}$.
     (b) $\mathcal{B}_2$ runs $\mathsf{IBE1.Extract}(pp_1, msk_1, \mathsf{ID}) \rightarrow d_{\mathsf{ID},1}$.
     (c) $\mathcal{B}_2$ computes $d_{\mathsf{ID},3} = h_{\mathsf{ID}}^s$ where $h_{\mathsf{ID}} = \mathcal{H}_3(\mathsf{ID})$ (obtained by $\mathcal{O}^{\mathcal{H}_3}$ queries).
     (d) $\mathcal{B}_2$ sends $d_{\mathsf{ID}} = (d_{\mathsf{ID},1}, d_{\mathsf{ID},2}, d_{\mathsf{ID},3})$ to $\mathcal{A}_2$.
   - $\mathcal{O}^{\mathsf{td}}$ query: On input an identity ID, it queries ID to the decryption oracle of the IBE2 scheme. Once receiving $d_{\mathsf{ID},2}$, it returns $\mathsf{td}_{\mathsf{ID}} = d_{\mathsf{ID},2}$ to $\mathcal{A}_2$.
   - $\mathcal{O}^{\mathsf{Dec}}$ query: On input a pair of an identity and a ciphertext $(\mathsf{ID}, C_{\mathsf{ID}})$ for $C_{\mathsf{ID}} = (C_{\mathsf{ID},1}, C_{\mathsf{ID},2}, C_{\mathsf{ID},3})$,
     (a) $\mathcal{B}_2$ queries $C_{\mathsf{ID},2}$ to the decryption oracle of the IND-ID-CCA2 security game of the IBE2 scheme and then receives $h_1' \| R''$.
     (b) $\mathcal{B}_2$ runs $\mathsf{IBE1.Dec}(pp_1, d_{\mathsf{ID},1}, C_{\mathsf{ID},1}) \rightarrow m' \| R'$ and check whether $h_1' = \mathcal{H}_1(m')$, $R' = R''$, and $C_{\mathsf{ID},3} = \mathcal{H}_2(C_{\mathsf{ID},1}, C_{\mathsf{ID},2}, e(d_{\mathsf{ID},3}, R'))$. If all of them hold, it returns m'. Otherwise, it returns $\perp$.
3. On receiving the challenge identity ID^* and two messages $m_0, m_1 \in \{0,1\}^{\ell_1}$ from $\mathcal{A}_2$, $\mathcal{B}_2$ randomly selects r from $\mathbb{Z}_p^*$ and forwards ID^*, $\mathcal{H}_1(m_0)\|g^r$, and $\mathcal{H}_1(m_1)\|g^r$ to $\mathcal{C}_2$. Then, $\mathcal{C}_2$ may select a random bit $\beta \in \{0, 1\}$, run $\mathsf{IBE2.Enc}(pp_2, \mathsf{ID}^*, \mathcal{H}_1(m_\beta)\|g^r) \rightarrow C^*_{\mathsf{ID}^*,\beta}$, and send $C^*_{\mathsf{ID}^*,\beta}$ to $\mathcal{B}_2$.
4. On receiving $C^*_{\mathsf{ID}^*,\beta}$ from $\mathcal{C}_2$, $\mathcal{B}_2$ selects a random bit $b \in \{0, 1\}$, sets $C^*_{\mathsf{ID}^*,b,2} = C^*_\beta$, runs $\mathsf{IBE1.Enc}(pp_1, \mathsf{ID}^*, m_b\|g^r) \rightarrow C^*_{\mathsf{ID}^*,b,1}$, and computes $C^*_{\mathsf{ID}^*,b,3} = \mathcal{H}_2(C^*_{\mathsf{ID}^*,b,1}, C^*_{\mathsf{ID}^*,b,2}, e(h_{\mathsf{ID}^*}, g_1)^r)$. Then, $\mathcal{B}_2$ returns $C^*_{\mathsf{ID}^*,b} = (C^*_{\mathsf{ID}^*,b,1}, C^*_{\mathsf{ID}^*,b,2}, C^*_{\mathsf{ID}^*,b,3})$ to $\mathcal{A}_2$.
5. $\mathcal{B}_2$ responds to $\mathcal{A}_2$'s oracle queries as in Step 2 of this game, except for the decryption oracle queries on ciphertexts of form $(C_{\mathsf{ID}^*,b,1}, C^*_{\mathsf{ID}^*,b,2}, C_{\mathsf{ID}^*,b,3})$. For the decryption oracle queries on ciphertexts of form $(C_{\mathsf{ID}^*,b,1}, C^*_{\mathsf{ID}^*,b,2}, C_{\mathsf{ID}^*,b,3})$, it outputs $\perp$.
6. On receiving $\mathcal{A}_2$'s output b', $\mathcal{B}_2$ outputs $\beta' = b'$.

As the similar reason in the proof of Lemma 3, the above simulation may fail when the event $\mathcal{G}_1$ defined in the proof of Theorem 5 occurs and $\Pr[\mathcal{G}_1] = \mathbf{Adv}^{\mathrm{BDH}}_{\mathcal{A}_2}$.

Because b is randomly chosen by $\mathcal{B}_2$, $\Pr[\beta = b] = \Pr[\beta \neq b] = \frac{1}{2}$. Further, if $\mathcal{B}_2$ correctly guesses β, i.e., $\beta = b$, then the challenge ciphertext $C^*_{\mathsf{ID}^*,b}$ for the adversary $\mathcal{A}_2$ has the form of

$$\begin{aligned} C^*_{\mathsf{ID}^*,b} &= (C^*_{\mathsf{ID}^*,b,1}, C^*_{\mathsf{ID}^*,b,2}, C^*_{\mathsf{ID}^*,b,3}) \\ &= (\mathsf{IBE1.Enc}(pp_1, \mathsf{ID}^*, m_\beta\|g^r), \mathsf{IBE2.Enc}(pp_2, \mathsf{ID}^*, \mathcal{H}_1(m_\beta)\|g^r), C^*_{\mathsf{ID}^*,b,3}), \end{aligned}$$

which has the same form of the challenge ciphertext in **Game** 0. Hence, when $\mathcal{A}_1$ correctly answers so that $b' = b$, $\mathcal{B}'$ outputs the correct answer $\beta' = b' = b = \beta$.

If $\mathcal{B}_2$ does not correctly guess β, i.e., $\beta \neq b$, then the challenge ciphertext $C^*_{\mathsf{ID}^*,b}$ has the form of

$$\begin{aligned} C^*_{\mathsf{ID}^*,b} &= (C^*_{\mathsf{ID}^*,b,1}, C^*_{\mathsf{ID}^*,b,2}, C^*_{\mathsf{ID}^*,b,3}) \\ &= (\mathsf{IBE1.Enc}(pp_1, \mathsf{ID}^*, m_{1-\beta}\|g^r), \mathsf{IBE2.Enc}(pp_2, \mathsf{ID}^*, \mathcal{H}_1(m_\beta)\|g^r), C^*_{\mathsf{ID}^*,b,3}), \end{aligned}$$

which has the same form of the challenge ciphertext in **Game** 1. Hence, when $\mathcal{A}_1$ outputs the index related to the second component of the challenge ciphertext in **Game** 1, β' is the same as β. The probability of such the event is $\frac{1}{2} - \epsilon_1$ because we define ϵ_1 by $\Pr[\mathcal{A} \text{ outputs } 1-\beta] - \frac{1}{2}$. Hence,

$$\begin{aligned} &\mathbf{Adv}^{\mathsf{IND\text{-}ID\text{-}CCA2}}_{\mathsf{IBE2},\mathcal{B}_2} \\ &= \Pr[\beta = \beta'] - \frac{1}{2} \\ &\geq \Pr[\beta = \beta' | \neg\mathcal{G}_1]\Pr[\neg\mathcal{G}_1] - \frac{1}{2} \\ &= \Big(\Pr[\mathcal{A} \text{ outputs } \beta \text{ in } \mathbf{Game\ 0} | \neg\mathcal{G}_1 \wedge \beta = b]\Pr[\beta = b] \\ &\quad + \Pr[\mathcal{A} \text{ outputs } \beta \text{ in } \mathbf{Game\ 1} | \neg\mathcal{G}_1 \wedge \beta \neq b]\Pr[\beta \neq b]\Big)\Pr[\neg\mathcal{G}_1] - \frac{1}{2} \\ &\geq \left(\left(\frac{1}{2} + \epsilon_0 - \Pr[\mathcal{G}_1]\right)\frac{1}{2} + \left(\frac{1}{2} - \epsilon_1 - \Pr[\mathcal{G}_1]\right)\frac{1}{2}\right)(1 - \Pr[\mathcal{G}_1]) - \frac{1}{2} \end{aligned}$$

$$\geq \left(1 - \mathbf{Adv}^{\mathrm{BDH}}_{\mathcal{A}_2}\right)\frac{\epsilon_0 - \epsilon_1}{2} - \frac{3\mathbf{Adv}^{\mathrm{BDH}}_{\mathcal{A}_2}}{2} + \mathsf{negl}(\lambda). \tag{B.2}$$

Since we assume that the IBE2 scheme is IND-ID-CCA2 secure, the advantage of $\mathcal{B}_2$ is negligible in the security parameter. Therefore, $\epsilon_0 - \epsilon_1$ is also negligible in the security parameter. □

From Lemma 3 and 4, both $\epsilon_0 + \epsilon_1$ and $\epsilon_0 - \epsilon_1$ are negligible in the security parameter. Hence, ϵ_0 should be negligible in the security parameter. More precisely, from the relations (B.1) and (B.2),

$$\begin{aligned}\epsilon_0 &= \mathbf{Adv}^{\mathrm{IND\text{-}ID\text{-}CCA2}}_{\mathrm{OURS}} \\ &\leq \left(\frac{1}{1 - \mathbf{Adv}^{\mathrm{BDH}}}\right)\left(\mathbf{Adv}^{\mathrm{IND\text{-}ID\text{-}CCA2}}_{\mathrm{IBE1}} + \mathbf{Adv}^{\mathrm{IND\text{-}ID\text{-}CCA2}}_{\mathrm{IBE2}} + 3\mathbf{Adv}^{\mathrm{BDH}}\right) + \mathsf{negl}(\lambda).\end{aligned}$$

Since we assume that the exploited IBE1 and IBE2 schemes are IND-ID-CCA2 secure and the BDH assumption holds, our construction is IND-ID-CCA2 secure. □

References

[1] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, H. Shi, Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions, J. Cryptology 21 (3) (2008) 350–391.
[2] M. Bellare, A. Boldyreva, A. O'Neill, Deterministic and efficiently searchable encryption, in: A. Menezes (Ed.), Advances in Cryptology - CRYPTO 2007, LNCS, 4622, Springer, 2007, pp. 535–552.
[3] D. Boneh, G.D. Crescenzo, R. Ostrovsky, G. Persiano, Public key encryption with keyword search, in: C. Cachin, J. Camenisch (Eds.), Advances in Cryptology - EUROCRYPT 2004, LNCS, 3027, Springer, 2004, pp. 506–522.
[4] D. Boneh, M.K. Franklin, Identity-based encryption from the weil pairing, in: J. Kilian (Ed.), Advances in Cryptology - CRYPTO 2001, LNCS, 2139, Springer, 2001, pp. 213–229.
[5] J. Daemen, V. Rijmen, The Design of Rijndael: AES - the Advanced Encryption Standard, Information Security and Cryptography, Springer, 2002, doi:10.1007/978-3-662-04722-4.
[6] E. Fujisaki, T. Okamoto, Secure integration of asymmetric and symmetric encryption schemes, J. Cryptol. 26 (1) (2013) 80–101.
[7] T.E. Gamal, A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Trans. Inf. Theor. 31 (4) (1985) 469–472.
[8] C. Gentry, Fully homomorphic encryption using ideal lattices, in: M. Mitzenmacher (Ed.), ACM Symposium on Theory of Computing (STOC) 2009, ACM, 2009, pp. 169–178.
[9] K. Huang, R. Tso, Y. Chen, S.M.M. Rahman, A. Almogren, A. Alamri, PKE-AET: public key encryption with authorized equality test, Comput. J. 58 (10) (2015) 2686–2697.
[10] X.-J. Lin, H. Qu, X. Zhang, Public key encryption supporting equality test and flexible authorization without bilinear pairings, IACR Cryptol. ePrint Archive 2016 (2016) 277.
[11] S. Ma, Identity-based encryption with outsourced equality test in cloud computing, Inf. Sci. 328 (2016) 389–402.
[12] S. Ma, Q. Huang, M. Zhang, B. Yang, Efficient public key encryption with equality test supporting flexible authorization, IEEE Trans. Inf. Forensics Secur. 10 (3) (2015) 458–470.
[13] S. Ma, M. Zhang, Q. Huang, B. Yang, Public key encryption with delegated equality test in a multi-user setting, Comput. J. 58 (4) (2015) 986–1002.
[14] J.H. Park, D.H. Lee, Anonymous HIBE: compact construction over prime-order groups, IEEE Trans. Inf. Theor. 59 (4) (2013) 2531–2541.
[15] D.F. Sittig, Personal health records on the internet: a snapshot of the pioneers at the end of the 20th century, I. J. Med. Inf. 65 (1) (2002) 1–6.
[16] Q. Tang, Towards public key encryption scheme supporting equality test with fine-grained authorization, in: U. Parampalli, P. Hawkes (Eds.), ACISP 2011, LNCS, 6812, Springer, 2011, pp. 389–406.
[17] Q. Tang, Public key encryption schemes supporting equality test with authorisation of different granularity, IJACT 2 (4) (2012a) 304–321.
[18] Q. Tang, Public key encryption supporting plaintext equality test and user-specified authorization, Security Commun. Netw. 5 (12) (2012b) 1351–1362.
[19] G. Yang, C.H. Tan, Q. Huang, D.S. Wong, Probabilistic public key encryption with equality test, in: J. Pieprzyk (Ed.), Topics in Cryptology - CT-RSA 2010, LNCS, 5985, Springer, 2010, pp. 119–131.