IEICE TRANS. FUNDAMENTALS, VOL.E91–A, NO.1 JANUARY 2008

PAPER Special Section on Cryptography and Information Security

Universally Composable Identity-Based Encryption**

Ryo NISHIMAKI[†*a)], *Nonmember*, **Yoshifumi MANABE**[†,††b)], *and* **Tatsuaki OKAMOTO**[†,†††c)], *Members*

SUMMARY Identity-based encryption (IBE) is one of the most important primitives in cryptography, and various security notions of IBE (e.g., IND-ID-CCA2, NM-ID-CCA2, IND-sID-CPA etc.) have been introduced. The relations among them have been clarified recently. This paper, for the first time, investigates the security of IBE in the universally composable (UC) framework. This paper first defines the UC-security of IBE, i.e., we define the ideal functionality of IBE, $\mathcal{F}_{\mathrm{IBE}}$. We then show that UC-secure IBE is equivalent to conventionally-secure (IND-ID-CCA2-secure) IBE.

key words: *identity-based encryption, IND-ID-CCA2, universal composition*

1. Introduction

1.1 Background

The concept of identity-based encryption (IBE), introduced by Shamir [21], is a variant of public-key encryption (PKE), where the identity of a user is employed in place of the user's public-key.

Boneh and Franklin [6] defined IND-ID-CCA2 (indistinguishability against adaptive chosen-ciphertext attacks under chosen identity attacks) as the desirable security of IBE schemes. Canetti, Halevi, and Katz [11], [12] defined a weaker notion of security in which the adversary, ahead of time, commits to the challenge identity it will attack. We refer to this notion as *selective identity* (sID) adaptive chosen-ciphertext secure IBE (IND-sID-CCA2). In addition, they also defined a weaker security notion of IBE, selective-identity *chosen-plaintext* (CPA) secure IBE (IND-sID-CPA). Attrapadung, Cui, Galindo, Hanaoka, Hasuo, Imai, Matsuura, Yang and Zhang [1] introduced *non-malleability* (NM) and *semantic security* (SS) to the set of security notions of IBE. Thus, the security definitions considered up to now in the literature are: G-A1-A2, where G $\in$ {IND, NM, SS}, A1 $\in$ {ID, sID}, ID denotes chosen identity attacks, and A2 $\in$ {CPA, CCA1, CCA2}.

Attrapadung, Cui, Galindo, Hanaoka, Hasuo, Imai, Matsuura, Yang and Zhang [1] clarified the relationship among these notions, and showed that IND-ID-CCA2 is equivalent to the strongest security notion among them, NM-ID-CCA2.

Since Canetti introduced universal composability (UC) as a framework for analyzing the security of cryptographic primitives/protocols [8], investigating the relation between UC-secure primitives/protocols and conventionally-secure primitives/protocols has been a significant topic in cryptography [2], [3], [9], [10], [13], [16], [20]. Since UC represents stronger security requirements, a lot of conventionally-secure protocols fail to meet UC security requirements. For example, we cannot design secure two party protocols in the UC framework with no setup assumption [8], [10], [14], [15], [17], [18], while there are conventionally-secure two party protocols (e.g., commitment and zero-knowledge proofs) with no setup assumption.

We do know, however, that the conventional security notions are equivalent to UC security notions for a few cryptographic primitives. For example, UC-secure PKE is equivalent to conventionally-secure (IND-CCA2-secure) PKE [8], UC-secure signatures are equivalent to conventionally-secure (existentially unforgeable against chosen message attacks: EUF-CMA-secure) signatures [9] and UC-secure Key Encapsulation Mechanism (KEM) is equivalent to conventionally-secure (IND-CCA2-KEM-secure) KEM [19].

IBE is a more complex cryptographic primitive than PKE or signatures, so it is not clear whether conventionally-secure (i.e., IND-ID-CCA2-secure) IBE is equivalent to UC-secure IBE or not. Since IBE is one of the most significant primitives [4]–[7], [22] like PKE and signatures in cryptography, it is important to clarify the relationship between UC security and the conventional security notions of IBE. The UC security of IBE, however, has not been investigated. That is, we have the following problems:

1. What is the security definition of IBE in the UC framework (i.e., how to define an ideal functionality of IBE)?
2. Is UC-secure IBE equivalent to IND-ID-CCA2-secure IBE?

Manuscript received March 23, 2007.
Manuscript revised July 15, 2007.

[†]The authors are with the Graduate School of Informatics, Kyoto University, Kyoto-shi, 606-8501 Japan.
[††]The author is with NTT Communication Science Laboratories, NTT Corporation, Atsugi-shi, 243-0198 Japan.
[†††]The author is with NTT Information Sharing Platform Laboratories, NTT Corporation, Musashino-shi, 180-8585 Japan.
[*]Presently, with NTT Information Sharing Platform Laboratories, NTT Corporation, and with Department of Mathematical and Computing Sciences, Tokyo Institute of Technology.

[**]A preliminary version of this paper was presented at VIETCRYPT, LNCS, vol.4341, pp.337–353, Sept., 2006

a) E-mail: nishimaki.ryo@lab.ntt.co.jp
b) E-mail: manabe.yoshifumi@lab.ntt.co.jp
c) E-mail: okamoto.tatsuaki@lab.ntt.co.jp

DOI: 10.1093/ietfec/e91–a.1.262

1.2 Our Results

This paper answers the above problems:

1. This paper defines the UC-security of IBE, i.e., we define the ideal functionality of IBE, $\mathcal{F}_{\text{IBE}}$.
2. We show that UC-secure IBE is equivalent to conventionally-secure (IND-ID-CCA2-secure) IBE.

2. Preliminaries

2.1 Notations

We describe probabilistic algorithms and experiments using standard notations and conventions. For probabilistic algorithm A, $A(x_1, x_2, \ldots; r)$ denotes the random variable of A's output on inputs $x_1, x_2, \ldots$ and coins r. We let $y \xleftarrow{R} A(x_1, x_2, \ldots)$ denote that y is randomly selected from $A(x_1, x_2, \ldots; r)$ according to its distribution. If S is a finite set, then $x \xleftarrow{U} S$ denotes that x is uniformly selected from S. If α is neither an algorithm nor a set, then $x \leftarrow \alpha$ indicates that we assign α to x.

We say that function $f : \mathbb{N} \to \mathbb{R}$ is negligible in security parameter k if for every constant $c \in \mathbb{N}$, there exists $k_c \in \mathbb{N}$ such that $f(k) < k^{-c}$ for any $k > k_c$. Hereafter, we often use $f < \epsilon(k)$ to mean that f is negligible in k. On the other hand, we use $f > \nu(k)$ to mean that f is non-negligible in k. i.e., function $f : \mathbb{N} \to \mathbb{R}$ is non-negligible in k, if there exists a constant $c \in \mathbb{N}$ such that for every $k_c \in \mathbb{N}$, there exists $k > k_c$ such that $f(k) > k^{-c}$. A distribution ensemble $X = \{X(k, z)\}_{k \in \mathbb{N}, z \in \{0,1\}^*}$ is an infinite set of probability distributions, where a distribution $X(k, z)$ is associated with each $k \in \mathbb{N}$ and $z \in \{0, 1\}^*$. The ensembles considered in this paper describe outputs of computations where parameter z represents input, and parameter k is taken to be the security parameter. Two *binary* distribution ensembles X and Y are statistically indistinguishable (written $X \approx Y$) if for any $c, d \in \mathbb{N}$ there exists $k_c \in \mathbb{N}$ such that for any $k > k_c$ and any $z \in \cup_{k \le k^d} \{0, 1\}^k$ we have:

$$|\Pr[X(k, z) = 1] - \Pr[Y(k, z) = 1]| < k^{-c}$$

2.2 Definition of Identity-Based Encryption

Identity-Based Encryption Schemes.

In this section, we define identity-based encryption schemes and its security notions.

Definition 2.1 (Identity-Based Encryption Schemes): Identity-based encryption scheme Σ consists of four algorithms $(\mathsf{Set}, \mathsf{Xtr}, \mathsf{Enc}, \mathsf{Dec})$:

Setup: Set takes security parameter k and returns PK (system parameters) and MK (master-key). The system parameters include a description of a finite message space M, and a description of a finite ciphertext space C. Intuitively, the system parameters should be publicly known, while MK is known only by the setup party.

Extract: Xtr takes as input PK, MK, and an arbitrary $ID \in \{0, 1\}^*$, and returns private key dk. Here ID is an arbitrary string that will be used as a public key, and dk is the corresponding private decryption key. The extract algorithm extracts a private key from the given public key.

Encrypt: Enc takes as input PK, ID, and $m \in \mathsf{M}$. It returns ciphertext $c \in \mathsf{C}$.

Decrypt: Dec takes as input PK, $c \in \mathsf{C}$, and private key dk. It returns $m \in \mathsf{M}$.

These algorithms must satisfy the standard consistency constraint, namely, $\forall m \in \mathsf{M} : \mathsf{Dec}(PK, c, dk) = m$ where $c = \mathsf{Enc}(PK, ID, m)$, $dk = \mathsf{Xtr}(PK, MK, ID)$

Security of Identity-Based Encryption Schemes.

Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary; we say $\mathcal{A}$ is polynomial time if both probabilistic algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ are polynomial time. In the first stage, given the system parameters, the adversary computes and outputs challenge template τ. $\mathcal{A}_1$ can output some information, s, which will be transferred to $\mathcal{A}_2$. In the second stage, the adversary is challenged with target ciphertext c^* generated from τ by a probabilistic function, in a manner depending on the goal. We say adversary $\mathcal{A}$ successfully breaks the scheme if she achieves her goal. We consider a security goal, IND [1], and three attack models, ID-CPA, ID-CCA, ID-CCA2, listed in order of increasing strength. The difference among the models is whether or not $\mathcal{A}_1$ or $\mathcal{A}_2$ is granted access to decryption oracles.

We describe in Table 1 and Table 2 the ability with which the adversary can, in the different attack models, access the Extraction Oracle $\mathsf{Xtr}(PK, MK, \cdot)$, the Encryption Oracle $\mathsf{Enc}(PK, ID, \cdot)$ and the Decryption Oracle $\mathsf{Dec}(PK, \cdot, dk)$. When we say $\mathcal{O}_i = \{\mathcal{XO}_i, \mathcal{EO}_i, \mathcal{DO}_i\} = \{\mathsf{Xtr}(PK, MK, \cdot), \mathsf{Enc}(PK, ID, \cdot), \perp\}$, where $i \in \{1, 2\}$, we mean that no decryption oracle can be used.

Let $\Sigma = (\mathsf{Set}, \mathsf{Xtr}, \mathsf{Enc}, \mathsf{Dec})$ be an identity based encryption scheme and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary. For atk $\in$ {id-cpa, id-cca, id-cca2} and $k \in \mathbb{N}$ let,

$$\begin{aligned}\mathbf{Adv}^{\text{ind-atk}}_{\Sigma,\mathcal{A}}(k) = {} & \Pr[\mathbf{Exp}^{\text{ind-atk-1}}_{\Sigma,\mathcal{A}}(k) = 1] \\ & - \Pr[\mathbf{Exp}^{\text{ind-atk-0}}_{\Sigma,\mathcal{A}}(k) = 1]\end{aligned}$$

Table 1 Oracle Set $\mathcal{O}_1$ in the Definitions of the Notions for IBE.

	$\mathcal{O}_1 = \{\mathcal{XO}_1, \mathcal{EO}_1, \mathcal{DO}_1\}$
ID-CPA	$\{\mathsf{Xtr}(PK, MK, \cdot), \mathsf{Enc}(PK, ID, \cdot), \perp\}$
ID-CCA	$\{\mathsf{Xtr}(PK, MK, \cdot), \mathsf{Enc}(PK, ID, \cdot), \mathsf{Dec}(PK, \cdot, dk)\}$
ID-CCA2	$\{\mathsf{Xtr}(PK, MK, \cdot), \mathsf{Enc}(PK, ID, \cdot), \mathsf{Dec}(PK, \cdot, dk)\}$

Table 2 Oracle Set $\mathcal{O}_2$ in the Definitions of the Notions for IBE.

	$\mathcal{O}_2 = \{\mathcal{XO}_2, \mathcal{EO}_2, \mathcal{DO}_2\}$
ID-CPA	$\{\mathsf{Xtr}(PK, MK, \cdot), \mathsf{Enc}(PK, ID, \cdot), \perp\}$
ID-CCA	$\{\mathsf{Xtr}(PK, MK, \cdot), \mathsf{Enc}(PK, ID, \cdot), \perp\}$
ID-CCA2	$\{\mathsf{Xtr}(PK, MK, \cdot), \mathsf{Enc}(PK, ID, \cdot), \mathsf{Dec}(PK, \cdot, dk)\}$

where for $b, d \in \{0, 1\}$ and $|m_0| = |m_1|$,

$$\begin{gathered}\text{Experiment } \mathbf{Exp}^{\text{ind-atk-b}}_{\Sigma,\mathcal{A}}(k)\\ (PK, MK) \xleftarrow{\mathsf{R}} \mathsf{Set}(1^k);\\ (m_0, m_1, s, ID) \xleftarrow{\mathsf{R}} \mathcal{A}_1^{\mathcal{O}_1}(PK);\\ c^* \xleftarrow{\mathsf{R}} \mathsf{Enc}(PK, ID, m_b);\\ d \xleftarrow{\mathsf{R}} \mathcal{A}_2^{\mathcal{O}_2}(m_0, m_1, s, c^*, ID);\\ \text{return } d\end{gathered}$$

Definition 2.2 (IND-ID-CCA2-secure): We say that Σ is IND-ID-CCA2-secure, if $\mathbf{Adv}^{\text{ind-id-cca2}}_{\Sigma,\mathcal{A}}(k)$ is negligible for any $\mathcal{A}$.

2.3 Universal Composability

The universally composable security framework allows the security properties of cryptographic tasks to be defined such that security is maintained under a general composition with an unbounded number of instances of arbitrary protocols running concurrently. Security in this framework is called universally composable (UC) security. Informally, we describe this framework as follows: (See [8] for more details.)

We consider the real life world, the ideal process world, and environment $\mathcal{Z}$ that tries to distinguish these two worlds.

The real life world.

In this world, there are adversary $\mathcal{A}$ and protocol π which realizes a functionality among some parties. Let $\mathtt{REAL}_{\pi,\mathcal{A},\mathcal{Z}}(k, z, \mathbf{r})$ denote the output of environment $\mathcal{Z}$ when interacting with adversary $\mathcal{A}$ and parties $P_1, \ldots, P_n$ running protocol π (hereafter denoted as $(\mathcal{A}, \pi)$) on security parameter k, auxiliary input z and random input $\mathbf{r} = (r_{\mathcal{Z}}, r_{\mathcal{A}}, r_1, \ldots, r_n)$ (z and $r_{\mathcal{Z}}$ for $\mathcal{Z}$, $r_{\mathcal{A}}$ for $\mathcal{A}$, r_i for party P_i). Let $\mathtt{REAL}_{\pi,\mathcal{A},\mathcal{Z}}(k, z)$ denote the random variable describing $\mathtt{REAL}_{\pi,\mathcal{A},\mathcal{Z}}(k, z, \mathbf{r})$ when $\mathbf{r}$ is uniformly chosen.

The ideal process world.

In this world, there are simulator $\mathcal{S}$ that simulates the real life world, ideal functionality $\mathcal{F}$, and dummy parties. Let $\mathtt{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}(k, z, \mathbf{r})$ denote the output of environment $\mathcal{Z}$ when interacting with adversary $\mathcal{S}$ and ideal functionality $\mathcal{F}$ (hereafter denoted as $(\mathcal{S}, \mathcal{F})$) on security parameter k, auxiliary input z and random input $\mathbf{r} = (r_{\mathcal{Z}}, r_{\mathcal{S}}, r_{\mathcal{F}})$ (z and $r_{\mathcal{Z}}$ for $\mathcal{Z}$, $r_{\mathcal{S}}$ for $\mathcal{S}$, $r_{\mathcal{F}}$ for party $\mathcal{F}$). Let $\mathtt{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}(k, z)$ denote the random variable describing $\mathtt{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}(k, z, \mathbf{r})$ when $\mathbf{r}$ is uniformly chosen. Let $\mathcal{F}$ be an ideal functionality and let π be a protocol. We say that π UC-realizes $\mathcal{F}$, if for any adversary $\mathcal{A}$, there exists simulator $\mathcal{S}$, such that for any environment $\mathcal{Z}$ we have:

$$\mathtt{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}(k, z) \approx \mathtt{REAL}_{\pi,\mathcal{A},\mathcal{Z}}(k, z)$$

where $\mathcal{A}$, $\mathcal{S}$ and $\mathcal{Z}$ are probabilistic polynomial-time (PPT) interactive Turing machines (ITMs).

3. Universally Composable Identity-Based Encryption

3.1 The IBE Functionality $\mathcal{F}_{\text{IBE}}$

Our definition of $\mathcal{F}_{\text{IBE}}$ follows the one for $\mathcal{F}_{\text{PKE}}$ of regular public-key encryption schemes given by Canetti [8]. We present IBE functionality $\mathcal{F}_{\text{IBE}}$ in Fig 1.

The idea of $\mathcal{F}_{\text{IBE}}$ is to allow parties to obtain idealized (information theoretically secure) ciphertexts for messages by using their IDs, such that private keys do not appear in the

Functionality $\mathcal{F}_{\text{IBE}}$

$\mathcal{F}_{\text{IBE}}$ proceeds as follows, given domain M of plaintexts and domain N of ID. Let $\mu \in \mathsf{M}$ be a fixed message.

Setup

In the first activation, expect to receive a value $(\mathtt{Setup}, sid, T)$ from some party T. Then do:

1. Hand $(\mathtt{Setup}, sid, T)$ to the adversary. Upon receiving value $(\mathtt{Algorithms}, sid, x, e, d)$ from the adversary, where x, e, d are descriptions of PPT ITMs, output $(\mathtt{Encryption\ Algorithm}, sid, e)$ to T. (If T is corrupted, the adversary may not send x and/or d.)
2. Record (T, x, e, d). (If T is corrupted, x and/or d may not be included.)

Extract

Upon receiving value $(\mathtt{Extract}, sid, ID, D)$ from party D, proceed as follows:

1. If $ID \notin \mathsf{N}$ or (ID, P) is recorded in ID-Reg for some $P(\neq D)$, then output an error message to D. Else if it has not received $\mathtt{Setup}$ yet, ignore the request.
2. Else if T is not corrupted, record (ID, D) in ID-Reg and output $(\mathtt{Extracted}, sid, ID, D)$ to D and T.
3. Otherwise, if T is corrupted, hand $(\mathtt{Corrupted\ Extract}, sid, ID, D)$ to the adversary. Upon receiving $(\mathtt{Corrupted\ Decrypt}, sid, ID, \tilde{d}_{ID})$ from the adversary, record $(ID, D, \tilde{d}_{ID}, \mathsf{corrupted})$ and output $(\mathtt{Extracted}, sid, ID, D)$ to D and T. If it does not receive it then output an error message to D and T.

Encrypt

Upon receiving value $(\mathtt{Encrypt}, sid, m, ID, e')$ from some party E, proceed as follows:

1. If $m \notin \mathsf{M}$ or $ID \notin \mathsf{N}$ then output an error message to E. Else, if $e' = e$, the setup party is uncorrupted, (ID, P) is recorded in ID-Reg for some P and decryptor P is uncorrupted, then let $c = e'(ID, \mu)$ and record (m, c, ID) in Plain-Cipher. Else, let $c = e'(ID, m)$.
2. Output $(\mathtt{Ciphertext}, sid, c)$ to E

Decrypt

Upon receiving value $(\mathtt{Decrypt}, sid, c, ID)$ from D, proceed as follows:

1. If the following two conditions are satisfied then hand $(\mathtt{Plaintext}, sid, m)$ to D.
   a. (ID, D) is recorded in ID-Reg.
   b. (m, c, ID) is stored in Plain-Cipher.
2. Else if (ID, D) is not recorded in ID-Reg or $(ID, D, \tilde{d}_{ID}, \mathsf{corrupted})$ is not recorded then hand $\mathtt{not\text{-}recorded}$ to D.
3. Else if T is corrupted and $(ID, D, \tilde{d}_{ID}, \mathsf{corrupted})$ is recorded then return $(\mathtt{Plaintext}, sid, \tilde{d}_{ID}(c))$ to D.
4. Otherwise, return $(\mathtt{Plaintext}, sid, d(c, x(ID)))$ to D.

Fig. 1 The ideal identity-based encryption functionality, $\mathcal{F}_{\text{IBE}}$.

interface, but at the same time the designated decryptor can retrieve the plaintexts. There may be multiple designated decryptors who let the setup party extract their private keys from their IDs.

$\mathcal{F}_{\text{IBE}}$ should be defined as follows: If no party is corrupted, setup, extract, encrypt and decrypt are information theoretically securely executed. $\mathcal{F}_{\text{IBE}}$ realizes such idealized setup, extract, encrypt and decrypt by recording IDs, ciphertexts and plaintexts. (i.e., $\mathcal{F}_{\text{IBE}}$ plays the role of the centralized database of encrypted messages and the corresponding ciphertexts and IDs used to encrypt.) $\mathcal{F}_{\text{IBE}}$ is written in a way that can be realized by protocols that have only local operations (setup, extract, encrypt, decrypt). All communication is left to the protocols that call $\mathcal{F}_{\text{IBE}}$. The important difference between PKE and IBE is that IBE schemes have the extract algorithm. Users need to extract private keys corresponding to their IDs to decrypt ciphertexts. They cannot locally generate private keys. The setup party generates user's private keys. $\mathcal{F}_{\text{IBE}}$ takes four types of input: setup, extract, encrypt, and decrypt.

Upon receiving a setup request from party T (the setup party), $\mathcal{F}_{\text{IBE}}$ asks the adversary to provide three descriptions of PPT algorithms: Extract algorithm x, encryption algorithm e, and decryption algorithm d. (Note that x, e, and d can be probabilistic.) It then outputs to T the description of encryption algorithm e. While the encryption algorithm is public and given to the environment (via T), the extract algorithm does not appear in the interface between $\mathcal{F}_{\text{IBE}}$ and T, because we need not such secret information in the ideal world. In the ideal world, we can realize IBE functionality by only using record. If the master key is output (and thus sent to $\mathcal{Z}$) then $\mathcal{Z}$ can extract private keys by itself and easily distinguish the two worlds. Note that decryption algorithm d does not include any secret information (See Remark 2.). However, it does not appear in the interface, because we do not need it. Encryption algorithm e also plays the role of system parameters. Note that if T is corrupted, $\mathcal{F}_{\text{IBE}}$ may not record extraction algorithm x, because corrupted T may not output the master-key.

Upon receiving a request from some party D to extract a private key with an ID and D (party ID), $\mathcal{F}_{\text{IBE}}$ proceeds as follows. If ID is not in domain N or (ID, P) is recorded in ID-Reg (ID-Reg is a kind of file which records the correspondence of an arbitrary identity to a party name.) for some P, then $\mathcal{F}_{\text{IBE}}$ outputs an error message to D. If $\mathcal{F}_{\text{IBE}}$ has not received `Setup` yet, then $\mathcal{F}_{\text{IBE}}$ ignores the request. Else if T is not corrupted, $\mathcal{F}_{\text{IBE}}$ records pair (ID, D) in ID-Reg and outputs message "extracted" to T and D. (Notice that one party may extract multiple private keys.) $\mathcal{F}_{\text{IBE}}$ only records the correspondence between parties and IDs. If T is corrupted, private keys may not be extracted appropriately, so $\mathcal{F}_{\text{IBE}}$ records the decryption algorithm which is sent by the adversary. The algorithm may include incorrect private keys. $\mathcal{Z}$ may obtain private keys of some parties only when $\mathcal{Z}$ corrupts them. Thus $\mathcal{F}_{\text{IBE}}$ need not output private keys in the interface of extract, the adversary may not send them. Note that the adversary may not send decryption algorithm. In this case, decryption cannot be executed at all.

Upon receiving a request from some arbitrary party E to encrypt message m with ID and encryption algorithm e', $\mathcal{F}_{\text{IBE}}$ proceeds as follows. If m is not in domain M or ID is not in domain N, $\mathcal{F}_{\text{IBE}}$ outputs an error message to E. Else, $\mathcal{F}_{\text{IBE}}$ outputs formal ciphertext c to E, where c is computed as follows. If $e' = e$, the setup party is uncorrupted, (ID, P) is recorded in ID-Reg for some P and decryptor P is uncorrupted, then $c = e(ID, \mu)$, where $\mu \in \mathsf{M}$ is some fixed message. In this case, (m, c, ID) is recorded for future decryption. Else, $c = e'(ID, m)$. In this case, no secrecy is guaranteed, since c may depend on m in arbitrary ways. Notice that if $\mathcal{F}_{\text{IBE}}$ receives $(\texttt{Encrypt}, sid, m, ID, e)$ before receiving $(\texttt{Extract}, sid, ID, P_h, e)$ for any party P_h, then c is not information theoretically secure, $(c = e(ID, m))$ even if $(\texttt{Extract}, sid, ID, P_h, e)$ arrives later for uncorrupted party P_h. However, this does not influence the UC security of IBE, because $c = e(ID, m)$ in the ideal world is the same as $c = e(ID, m)$ in the real world. (i.e., $\mathcal{Z}$ cannot distinguish two worlds.)

Upon receiving a request from party D to decrypt ciphertext c encrypted for ID ID, $\mathcal{F}_{\text{IBE}}$ first checks if there are records (ID, D) in ID-Reg (i.e., D is the decryptor) and (m, c, ID) in Plain-Cipher for some m. If so, then it returns m as the decrypted value. This guarantees perfectly correct decryption for messages that were encrypted via this instance of $\mathcal{F}_{\text{IBE}}$. If (ID, D) is not recorded in ID-Reg or $(ID, D, \tilde{d}_{ID}, \mathsf{corrupted})$ is not recorded, this means that D has not extracted a private key for ID. Accordingly, $\mathcal{F}_{\text{IBE}}$ returns an error message. If T is corrupted and $(ID, D, \tilde{d}_{ID}, \mathsf{corrupted})$ is recorded then this means that corrupted T may have sent the decryption algorithm which includes a (might be inappropriate) private key, so return $(\texttt{Plaintext}, sid, \tilde{d}_{ID}(c))$ to D. If no (m, c, ID) record exists for any m, this means that c was not generated legitimately via this instance of $\mathcal{F}_{\text{IBE}}$, so no correctness guarantee is provided, and $\mathcal{F}_{\text{IBE}}$ returns the value $d(c, x(ID))$. In IBE, multiple users may extract their private keys from a single master key, so single instance of $\mathcal{F}_{\text{IBE}}$ should deal with multiple decryptors.

3.2 UC-Secure IBE is Equivalent to IND-ID-CCA2-Secure IBE

Next, we present a protocol that UC-realizes $\mathcal{F}_{\text{IBE}}$. Let $\Sigma = (\mathsf{Set}, \mathsf{Xtr}, \mathsf{Enc}, \mathsf{Dec})$ be an identity based encryption scheme. We define protocol π_Σ that is constructed from Σ and has the same interface with the environment as $\mathcal{F}_{\text{IBE}}$.

protocol π_Σ.

Setup: Upon input $(\texttt{Setup}, sid, T)$ within some setup party T, T obtains the system parameters PK and master-key MK by running algorithm $\mathsf{Set}(\cdot)$ and sets $x = \mathsf{Xtr}(PK, MK, \cdot)$, $e = \mathsf{Enc}(PK, \cdot, \cdot)$, $d = \mathsf{Dec}(PK, \cdot, \cdot)$. It then outputs $(\texttt{Encryption Algorithm}, sid, e)$.

Extract: Upon input $(\texttt{Extract}, sid, ID, D)$ within some party D, D sends an extraction request for ID to T. If

$ID \notin \mathsf{N}$ or T has extracted the private key for the same ID, $dk_{ID} = x(ID)$ before, then T sends an error message to D and D outputs an error message. Else if T has not executed setup yet, T ignores the request. Else, T obtains private key $dk_{ID} = x(ID)$. T sends the private key securely to D. If D receives the private key then D and T output (Extracted, sid, ID, D) (See Remark 1 below for how to securely transfer.) If D does not receive the private key then it outputs an error message.

Encrypt: Upon input (Encrypt, sid, m, ID, e') within some party E, if $m \notin \mathsf{M}$ or $ID \notin \mathsf{N}$, E outputs an error message. Else, E obtains ciphertext $c = e'(ID, m)$ and outputs (Ciphertext, sid, c). (Note that it does not necessarily hold that ID is E's)

Decrypt: Upon input (Decrypt, sid, c, ID') within D, if $ID' \neq ID$ or D does not have private key dk_{ID} yet, outputs not-recorded. Else, D obtains $m = d(c, dk_{ID})$ and outputs (Plaintext, sid, m).

Remark 1**.** (On the communication between the setup party and the decryptor)

IBE is specified by four algorithms which are locally executed. The procedure to send and receive keys is outside of the definition of the IBE scheme. In order to realize a secure communication mechanism based on IBE, some transmission protocol must be used with IBE. The security of the communication mechanism depends also on the security of the transmission protocol. Our $\mathcal{F}_{\mathrm{IBE}}$ and Σ definitions do not describe procedures to transfer keys securely, because the aim of our paper is investigating the security of IBE, not the communication mechanism. We consider a model that when uncorrupted T writes the private key to D's incoming communication tape, **the adversary can not see it**. (The adversary can see the private key when T is corrupted or D is corrupted.)

Remark 2**.** (On the decryption algorithm d)

Note that decryption algorithm d does **not** include any secret information ($d = \mathsf{Dec}(PK, \cdot, \cdot)$ does not include master key MK and private key dk_{ID}). It includes only system parameters like encryption algorithm e. When setup party T is corrupted, T might not output d at Setup. However, corrupted T might output (possibly inappropriate) private key dk at Extract. This means, D can decrypt ciphertexts by the (possibly inappropriate) private key. For this case, in the ideal world, $\mathcal{F}_{\mathrm{IBE}}$ receives decryption algorithm $\tilde{d}_{ID}$ which may include incorrect private keys from the adversary. ($\tilde{d}_{ID} = \mathsf{Dec}(PK, \cdot, \tilde{dk}_{ID})$, where $\tilde{dk}_{ID}$ might be an incorrect private key)

Remark 3**.** (On the extraction by corrupted decryptor)

Note that even when decryptor D is corrupted, uncorrupted setup party T executes extraction appropriately.

Security against adaptive adversaries.

Recall that, even in the case of $\mathcal{F}_{\mathrm{PKE}}$, when the adversary is allowed to corrupt parties during the course of the computation, and obtain their internal state, realizing $\mathcal{F}_{\mathrm{PKE}}$ is a very hard problem [8]. The reason is as follows: If $\mathcal{Z}$ is allowed to corrupt adaptively, $\mathcal{Z}$ makes uncorrupted party E generate ciphertext c of message m for ID ID whose decryptor D is uncorrupted. $\mathcal{Z}$ then corrupts D and can distinguish whether $c = e(ID, \mu)$ or $c = e(ID, m)$ (($\mathcal{S}, \mathcal{F}_{\mathrm{IBE}}$) or ($\mathcal{A}, \pi_{\Sigma}$)) by obtaining corrupted D's internal states.

Theorem 3.1: π_{Σ} UC-realizes $\mathcal{F}_{\mathrm{IBE}}$ with respect to non-adaptive adversaries if and only if IBE scheme Σ is IND-ID-CCA2-secure.

Proof. ("only if" part)

We prove that if Σ is not IND-ID-CCA2-secure, then π_{Σ} does not UC-realize $\mathcal{F}_{\mathrm{IBE}}$. In more detail, assuming that there exists adversary G that can break Σ in the sense of IND-ID-CCA2 with non-negligible probability (i.e., $\mathbf{Adv}^{\text{ind-id-cca2}}_{\Sigma,G} > \nu(k)$), we prove that we can construct environment $\mathcal{Z}$ and real life adversary $\mathcal{A}$ such that for any ideal process adversary (simulator) $\mathcal{S}$, $\mathcal{Z}$ can tell with non-negligible probability whether ($\mathcal{S}, \mathcal{F}_{\mathrm{IBE}}$) or ($\mathcal{A}, \pi_{\Sigma}$) by using adversary G. $\mathcal{Z}$ proceeds as follows:

1. Activates party T with (Setup, sid, T) and obtains encryption algorithm e (and system parameters).
2. Hands e to G and plays the role of $\mathcal{XO}_1$ (the extraction oracle as in Preliminaries), $\mathcal{EO}_1$ (the encryption oracle), and $\mathcal{DO}_1$ (the decryption oracle) for adversary G in the IND-ID-CCA2 game.
3. Obtains (ID^*, m_0, m_1) from G. ID^* is the ID G attacks.
4. If $\mathcal{Z}$ has not activated D with (Extract, sid, ID^*, D) yet, it does so and obtains (Extracted, sid, ID^*, D).
5. Chooses random bit $b \xleftarrow{\cup} \{0, 1\}$, selects an arbitrary party $E(\neq D)$, activates E with (Encrypt, sid, m_b, ID^*, e) and obtains c^*.
6. Hands c^* to G as the target ciphertext.
7. Plays the role of $\mathcal{XO}_2$, $\mathcal{EO}_2$, and $\mathcal{DO}_2$ for adversary G in the IND-ID-CCA2 game, and obtains guess $b' \in \{0, 1\}$.
8. Outputs 1 if $b = b'$, otherwise outputs 0 and halts.

Notice that we consider non-adaptive adversary case. The corrupted parties are denoted $\tilde{P}_1, \ldots, \tilde{P}_t$. In step 2, the adversary issues queries $q_1, \ldots, q_m$ where query q_l is one of:

1. Extraction query $\langle ID_l \rangle$. If this is the x-th extraction, $\mathcal{Z}$ activates $\tilde{P}_x$ with (Extract, $sid, ID_l, \tilde{P}_x$) to obtain private key dk_{ID_l} corresponding to public key ID_l from corrupted $\tilde{P}_x$. When (Extracted, $sid, ID_l, \tilde{P}_x$) is output, private key dk_{ID_l} is transferred to $\tilde{P}_x$ in the real world. So $\mathcal{Z}$ can obtain dk_{ID_l} from corrupted $\tilde{P}_x$. In the ideal world, $\mathcal{Z}$ can do so as in the real world, because simulator $\mathcal{S}$ generates master-key and extraction algorithm x when T is activated with (Setup, sid, T) and uses simulated copy of real life adversary $\mathcal{A}$. In both cases, $\mathcal{Z}$ can hand correct private key dk_{ID_l} to G. If $\mathcal{Z}$ cannot obtain the private key (i.e., $\mathcal{S}$ did not simulate the private key), $\mathcal{Z}$ can easily decide that the world

it is interacting with is the ideal world.
2. Decryption query $\langle ID_l, c_l\rangle$. If this is the first decryption query for ID_l, $\mathcal{Z}$ selects a new uncorrupted party, P_y, and activates P_y with $(\texttt{Extract}, sid, ID_l, P_y)$ and then activates P_y with $(\texttt{Decrypt}, sid, c_l, ID_l)$. Otherwise $\mathcal{Z}$ activates P'_y with $(\texttt{Decrypt}, sid, c_l, ID_l)$, where P'_y is the process $\mathcal{Z}$ activated T with $(\texttt{Extract}, sid, ID_l, P'_y)$. When $\mathcal{Z}$ receives $(\texttt{Plaintext}, sid, v_l)$, it hands v_l to G.

These queries may be asked adaptively, that is, each query q_l may depend on the replies to $q_1, \ldots, q_{l-1}$. In step 7, the adversary issues more queries $q_{m+1}, \ldots, q_n$ where query q_l is one of:

1. Extraction query $\langle ID_l\rangle$ where $ID_l \neq ID^*$. $\mathcal{Z}$ responds as in step 2.
2. Decryption query $\langle ID_l, c_l\rangle \neq \langle ID^*, c^*\rangle$. $\mathcal{Z}$ responds as in step 2.

These queries may be asked adaptively as in step 2.

When $\mathcal{Z}$ interacts with $\mathcal{A}$ and π_Σ, $\mathcal{Z}$ obtains $c^* = \mathsf{Enc}(PK, ID^*, m_b)$ in Step 6. G can break IND-ID-CCA2 security with non-negligible advantage $\mathbf{Adv}^{\text{ind-id-cca2}}_{\Sigma,G} > \nu(k)$. $\Pr[\mathcal{Z} \to 1 | \mathcal{Z} \leftrightarrow \texttt{REAL}]$ denotes the probability that $\mathcal{Z}$ outputs 1 when $\mathcal{Z}$ interacts with $\mathcal{A}$ and π_Σ.

$$
\begin{aligned}
&\Pr[\mathcal{Z} \to 1 | \mathcal{Z} \leftrightarrow \texttt{REAL}] \\
&\quad = \Pr[m_b = m_0] \Pr[b' = 0 | c^* = \mathsf{Enc}(PK, ID^*, m_0)] \\
&\qquad + \Pr[m_b = m_1] \Pr[b' = 1 | c^* = \mathsf{Enc}(PK, ID^*, m_1)] \\
&\quad = \frac{1}{2}(1 - \Pr[b' = 1 | c^* = \mathsf{Enc}(PK, ID^*, m_0)]) \\
&\qquad + \frac{1}{2} \Pr[b' = 1 | c^* = \mathsf{Enc}(PK, ID^*, m_1)] \\
&\quad = \frac{1}{2} + \frac{1}{2}(\Pr[\mathbf{Exp}^{\text{ind-id-cca2-1}}_{\Sigma,G}(k) = 1] \\
&\qquad - \Pr[\mathbf{Exp}^{\text{ind-id-cca2-0}}_{\Sigma,G}(k) = 1]) \\
&\quad > \frac{1}{2} + \frac{1}{2}\nu(k)
\end{aligned}
$$

In contrast, when $\mathcal{Z}$ interacts with the ideal process for $\mathcal{F}_{\text{IBE}}$ and any adversary, the view of the instance of G within $\mathcal{Z}$ is statistically independent of b, thus in this case $b = b'$ with probability exactly one half. To see why G's view is independent of b, recall that the view of G consists of the target ciphertext c^* and the decryptions of all ciphertexts generated by G (except for the decryption of c^*). However, $c^* = e(ID^*, \mu)$ for fixed message μ is independent of b. Furthermore, all ciphertexts c_l generated by G are independent of b, thus decryption $d(c_l, dk_{ID_l})$ is independent of b.

$\Pr[\mathcal{Z} \to 1 | \mathcal{Z} \leftrightarrow \texttt{IDEAL}]$ denotes the probability that $\mathcal{Z}$ outputs 1 when $\mathcal{Z}$ interacts with $\mathcal{S}$ in the ideal process for $\mathcal{F}_{\text{IBE}}$.

$$
\begin{aligned}
&\Pr[\mathcal{Z} \to 1 | \mathcal{Z} \leftrightarrow \texttt{IDEAL}] \\
&\quad = \Pr[m_b = m_0] \Pr[b' = 0 | c^* = e(ID^*, \mu)] \\
&\qquad + \Pr[m_b = m_1] \Pr[b' = 1 | c^* = e(ID^*, \mu)] \\
&\quad = \frac{1}{2}(1 - \Pr[b' = 1 | c^* = e(ID^*, \mu)] \\
&\qquad + \Pr[b' = 1 | c^* = e(ID^*, \mu)]) \\
&\quad = \frac{1}{2}
\end{aligned}
$$

Thus, $\Pr[\mathcal{Z} \to 1 | \mathcal{Z} \leftrightarrow \texttt{REAL}] - \Pr[\mathcal{Z} \to 1 | \mathcal{Z} \leftrightarrow \texttt{IDEAL}] > \frac{1}{2}\nu(k)$. Therefore, $\mathcal{Z}$ can tell whether $(\mathcal{S}, \mathcal{F}_{\text{IBE}})$ or $(\mathcal{A}, \pi_\Sigma)$ with non-negligible probability.

("if" part)

We show that if π_Σ does not UC-realize $\mathcal{F}_{\text{IBE}}$, then Σ is not IND-ID-CCA2-secure. In more detail, we assume for contradiction that there is real life adversary $\mathcal{A}$ such that for any ideal process adversary $\mathcal{S}$ there exists environment $\mathcal{Z}$ that can tell whether $(\mathcal{S}, \mathcal{F}_{\text{IBE}})$ or $(\mathcal{A}, \pi_\Sigma)$. We then show that there exists an IND-ID-CCA2 attacker G against Σ using $\mathcal{Z}$.

First, we show that $\mathcal{Z}$ can distinguish whether $(\mathcal{S}, \mathcal{F}_{\text{IBE}})$ or $(\mathcal{A}, \pi_\Sigma)$ only when setup party T, some encryptor E and some decryptor D are not corrupted. Since we are dealing with non-adaptive adversaries, there are following cases;

- Case A: For every triple (T, E, D), some of the elements are corrupted. There are 6 cases; Case 1: setup party T is corrupted (throughout the protocol), Case 2: Encryptor E is corrupted (throughout the protocol), Case 3: Decryptor D is corrupted (throughout the protocol), Case 4: T and E are corrupted (throughout the protocol), Case 5: T and D are corrupted (throughout the protocol), Case 6: D and E are corrupted (throughout the protocol),
- Case B: There is an uncorrupted triple (T, E, D). That is, T, some E and some D are uncorrupted.

In Case 1, we can construct simulator $\mathcal{S}$ such that no $\mathcal{Z}$ can distinguish whether$(\mathcal{S}, \mathcal{F}_{\text{IBE}})$ or $(\mathcal{A}, \pi_\Sigma)$ as follows:

1. When $\mathcal{Z}$ sends $(\texttt{Setup}, sid, T)$ to corrupted party T (i.e., $\mathcal{S}$), $\mathcal{S}$ receives the message and sends it to $\mathcal{F}_{\text{IBE}}$ on behalf of T and the simulated copy of $\mathcal{A}$, which returns a reply message (which may be $\perp$) to $\mathcal{S}$. When $\mathcal{S}$ receives $(\texttt{Setup}, sid, T)$ from $\mathcal{F}_{\text{IBE}}$, $\mathcal{S}$ sends $\mathcal{A}$'s reply to $\mathcal{F}_{\text{IBE}}$. $\mathcal{S}$ sends $\mathcal{A}$'s reply to $\mathcal{Z}$.
2. When $\mathcal{Z}$ sends $(\texttt{Extract}, sid, ID, D)$ to D, D forwards it to $\mathcal{F}_{\text{IBE}}$. When $\mathcal{S}$ receives $(\texttt{Corrupted Extract}, sid, ID, D)$ from $\mathcal{F}_{\text{IBE}}$, $\mathcal{S}$ sends it to the simulated copy of $\mathcal{A}$, which returns a reply message (which may be $\perp$) to $\mathcal{S}$. If $\mathcal{A}$'s reply is decryption algorithm $\tilde{d}_{ID}$ (which may be inappropriate), $\mathcal{S}$ sends $(\texttt{Corrupted Decrypt}, sid, ID, \tilde{d}_{ID})$ to $\mathcal{F}_{\text{IBE}}$. $\mathcal{F}_{\text{IBE}}$ records $(ID, D, \tilde{d}_{ID}, \texttt{corrputed})$ and returns $(\texttt{Extracted}, sid, ID, D)$. If $\mathcal{A}$'s reply is $\perp$, $\mathcal{S}$ sends it to $\mathcal{F}_{\text{IBE}}$ and $\mathcal{F}_{\text{IBE}}$ returns an error message.
3. When $\mathcal{Z}$ sends $(\texttt{Encrypt}, sid, m, ID, e')$ to E, E forwards it to $\mathcal{F}_{\text{IBE}}$. $\mathcal{F}_{\text{IBE}}$ generates $c = e'(ID, m)$ and returns $(\texttt{Ciphertext}, sid, c)$ to E, since T is corrupted.
4. When $\mathcal{Z}$ sends $(\texttt{Decrypt}, sid, c, ID)$ to D, D forwards it to $\mathcal{F}_{\text{IBE}}$. If decryption algorithm in step 2, $\mathcal{F}_{\text{IBE}}$ returns $(\texttt{Plaintext}, sid, \tilde{d}_{ID}(c))$. Otherwise, $\mathcal{F}_{\text{IBE}}$ outputs $\texttt{not-recorded}$, because $(ID, D, \tilde{d}_{ID}, \texttt{corrputed})$

is not recorded.

In this case, $\mathcal{Z}$ cannot distinguish whether $(\mathcal{S}, \mathcal{F}_{\mathrm{IBE}})$ or $(\mathcal{A}, \pi_\Sigma)$, because the message returned by $\mathcal{S}$ (using $\mathcal{A}$) as T in the ideal world is the same as that returned by $\mathcal{A}$ as T in the real world, and (`Ciphertext`, sid, c) returned by $\mathcal{F}_{\mathrm{IBE}}$ is exactly the same as that returned by E in the real world, and `not-recorded` or (`Plaintext`, $sid, \tilde{d}_{ID}(c)$) returned by $\mathcal{F}_{\mathrm{IBE}}$ is exactly the same as that returned by D in the real world.

In Case 2, we can construct simulator $\mathcal{S}$ such that no $\mathcal{Z}$ can distinguish whether $(\mathcal{S}, \mathcal{F}_{\mathrm{IBE}})$ or $(\mathcal{A}, \pi_\Sigma)$ as follows:

1. When $\mathcal{Z}$ sends (`Setup`, sid, T) to T, T forwards it to $\mathcal{F}_{\mathrm{IBE}}$. $\mathcal{F}_{\mathrm{IBE}}$ sends (`Setup`, sid, T) to $\mathcal{S}$, $\mathcal{S}$ computes (PK, MK) by running algorithm $\mathsf{Set}()$, and generates x, e and d, where $x = \mathsf{Xtr}(PK, MK, \cdot)$, $e = \mathsf{Enc}(PK, \cdot, \cdot)$ and $d = \mathsf{Dec}(PK, \cdot, \cdot)$. $\mathcal{S}$ returns (`Algorithms`, sid, x, e, d) to $\mathcal{F}_{\mathrm{IBE}}$.
2. When $\mathcal{Z}$ sends (`Extract`, sid, ID, D) to D, D forwards it to $\mathcal{F}_{\mathrm{IBE}}$. $\mathcal{F}_{\mathrm{IBE}}$ records (ID, D) and returns (`Extracted`, sid, ID, D).
3. When $\mathcal{Z}$ sends (`Encrypt`, sid, m, ID, e') to corrupted party E (i.e., $\mathcal{S}$), $\mathcal{S}$ receives the message and sends it to the simulated copy of $\mathcal{A}$, which replies to $\mathcal{S}$. $\mathcal{S}$ then returns $\mathcal{A}$'s reply (which may be $\perp$) to $\mathcal{Z}$.
4. When $\mathcal{Z}$ sends (`Decrypt`, sid, c, ID) to D, D forwards it to $\mathcal{F}_{\mathrm{IBE}}$. $\mathcal{F}_{\mathrm{IBE}}$ then returns (`Plaintext`, $sid, d(c, x(ID))$), since E (i.e., $\mathcal{S}$) sent no (`Encrypt`, sid, m, ID, e) to $\mathcal{F}_{\mathrm{IBE}}$, which records nothing as (m, c, ID).

In this case, $\mathcal{Z}$ cannot distinguish whether $(\mathcal{S}, \mathcal{F}_{\mathrm{IBE}})$ or $(\mathcal{A}, \pi_\Sigma)$, because the message returned by $\mathcal{S}$ (using $\mathcal{A}$) as E in the ideal world is the same as that returned by $\mathcal{A}$ as E in the real world, and (`Encryption Algorithm`, sid, e) returned by $\mathcal{F}_{\mathrm{IBE}}$ is exactly the same as that returned by T in the real world, (`Extracted`, sid, ID, D) returned by $\mathcal{F}_{\mathrm{IBE}}$ is exactly the same as that returned by T in the real world, and (`Plaintext`, $sid, d(c, x(ID))$) returned by $\mathcal{F}_{\mathrm{IBE}}$ is exactly the same as that returned by D in the real world.

In Case 3, we can construct simulator $\mathcal{S}$ such that no $\mathcal{Z}$ can distinguish whether $(\mathcal{S}, \mathcal{F}_{\mathrm{IBE}})$ or $(\mathcal{A}, \pi_\Sigma)$ as follows:

1. When $\mathcal{Z}$ sends (`Setup`, sid, T) to T, T forwards it to $\mathcal{F}_{\mathrm{IBE}}$. $\mathcal{F}_{\mathrm{IBE}}$ sends (`Setup`, sid, T) to $\mathcal{S}$, $\mathcal{S}$ computes (PK, MK) by running algorithm $\mathsf{Set}()$, and generates x, e and d, where $x = \mathsf{Xtr}(PK, MK, \cdot)$, $e = \mathsf{Enc}(PK, \cdot, \cdot)$ and $d = \mathsf{Dec}(PK, \cdot, \cdot)$. $\mathcal{S}$ returns (`Algorithms`, sid, x, e, d) to $\mathcal{F}_{\mathrm{IBE}}$.
2. When $\mathcal{Z}$ sends (`Extract`, sid, ID, D) to corrupted D (i.e., $\mathcal{S}$), $\mathcal{S}$ sends it to the simulated copy of $\mathcal{A}$. If $\mathcal{A}$ requests the private key, forwards it to $\mathcal{F}_{\mathrm{IBE}}$ and returns $dk_{ID} = x(ID)$. $\mathcal{F}_{\mathrm{IBE}}$ records (ID, D) and returns (`Extracted`, sid, ID, D).
3. When $\mathcal{Z}$ sends (`Encrypt`, sid, m, ID, e') to E, E forwards it to $\mathcal{F}_{\mathrm{IBE}}$. $\mathcal{F}_{\mathrm{IBE}}$ generates $c = e'(ID, m)$ and returns (`Ciphertext`, sid, c) to E, since D is corrupted.
4. When $\mathcal{Z}$ sends (`Decrypt`, sid, c, ID) to corrupted party D (i.e., $\mathcal{S}$), $\mathcal{S}$ sends (`Decrypt`, sid, c, ID) to $\mathcal{A}$. $\mathcal{A}$ returns a reply (which may be $\perp$) to $\mathcal{S}$, which forwards $\mathcal{A}$'s reply to $\mathcal{Z}$.

In this case, $\mathcal{Z}$ cannot distinguish whether $(\mathcal{S}, \mathcal{F}_{\mathrm{IBE}})$ or $(\mathcal{A}, \pi_\Sigma)$, because the message returned by $\mathcal{S}$ (using $\mathcal{A}$) as D in the ideal world is the same as that returned by $\mathcal{A}$ as D in the real world, (`Encryption Algorithm`, sid, e) returned by $\mathcal{F}_{\mathrm{IBE}}$ is exactly the same as that returned by T in the real world, and (`Extracted`, sid, ID, D) returned by $\mathcal{F}_{\mathrm{IBE}}$ is exactly the same as that returned by D in the real world, and (`Ciphertext`, sid, c) returned by $\mathcal{F}_{\mathrm{IBE}}$ is exactly the same as that returned by E in the real world.

In Case 4, we can construct simulator $\mathcal{S}$ such that no $\mathcal{Z}$ can distinguish whether $(\mathcal{S}, \mathcal{F}_{\mathrm{IBE}})$ or $(\mathcal{A}, \pi_\Sigma)$ as follows:

1. When $\mathcal{Z}$ sends (`Setup`, sid, T) to corrupted party T (i.e., $\mathcal{S}$), $\mathcal{S}$ receives the message and sends it to $\mathcal{F}_{\mathrm{IBE}}$ on behalf of T and the simulated copy of $\mathcal{A}$, which returns a reply message (which may be $\perp$) to $\mathcal{S}$. When $\mathcal{S}$ receives (`Setup`, sid, T) from $\mathcal{F}_{\mathrm{IBE}}$, $\mathcal{S}$ sends $\mathcal{A}$'s reply to $\mathcal{F}_{\mathrm{IBE}}$. $\mathcal{S}$ sends $\mathcal{A}$'s reply to $\mathcal{Z}$.
2. When $\mathcal{Z}$ sends (`Extract`, sid, ID, D) to D, D forwards it to $\mathcal{F}_{\mathrm{IBE}}$. When $\mathcal{S}$ receives (`Corrupted Extract`, sid, ID, D) from $\mathcal{F}_{\mathrm{IBE}}$, $\mathcal{S}$ sends it to the simulated copy of $\mathcal{A}$, which returns a reply message (which may be $\perp$) to $\mathcal{S}$. If $\mathcal{A}$'s reply is decryption algorithm $\tilde{d}_{ID}$ (which may be inappropriate), $\mathcal{S}$ sends (`Corrupted Decrypt`, $sid, ID, \tilde{d}_{ID}$) to $\mathcal{F}_{\mathrm{IBE}}$. $\mathcal{F}_{\mathrm{IBE}}$ records $(ID, D, \tilde{d}_{ID}$, `corrputed`) and returns (`Extracted`, sid, ID, D). If $\mathcal{A}$'s reply is $\perp$, $\mathcal{S}$ sends it to $\mathcal{F}_{\mathrm{IBE}}$ and $\mathcal{F}_{\mathrm{IBE}}$ returns an error message.
3. When $\mathcal{Z}$ sends (`Encrypt`, sid, m, ID, e') to corrupted party E (i.e., $\mathcal{S}$), $\mathcal{S}$ receives the message and sends it to the simulated copy of $\mathcal{A}$, which replies to $\mathcal{S}$. $\mathcal{S}$ then returns $\mathcal{A}$'s reply (which may be $\perp$) to $\mathcal{Z}$.
4. When $\mathcal{Z}$ sends (`Decrypt`, sid, c, ID) to D, D forwards it to $\mathcal{F}_{\mathrm{IBE}}$. If the simulated copy of $\mathcal{A}$ output the decryption algorithm in step 2, $\mathcal{F}_{\mathrm{IBE}}$ returns (`Plaintext`, $sid, \tilde{d}_{ID}(c)$). Otherwise, $\mathcal{F}_{\mathrm{IBE}}$ outputs `not-recorded`, because $(ID, D, \tilde{d}_{ID}$, `corrputed`) is not recorded.

In this case, $\mathcal{Z}$ cannot distinguish whether $(\mathcal{S}, \mathcal{F}_{\mathrm{IBE}})$ or $(\mathcal{A}, \pi_\Sigma)$, because the message returned by $\mathcal{S}$ (using $\mathcal{A}$) as T and E in the ideal world is the same as that returned by $\mathcal{A}$ as T and E in the real world, and `not-recorded` or (`Plaintext`, $sid, \tilde{d}_{ID}(c)$) returned by $\mathcal{F}_{\mathrm{IBE}}$ is exactly the same as that returned by D in the real world.

In Case 5, we can construct simulator $\mathcal{S}$ such that no $\mathcal{Z}$ can distinguish whether $(\mathcal{S}, \mathcal{F}_{\mathrm{IBE}})$ or $(\mathcal{A}, \pi_\Sigma)$ as follows:

1. When $\mathcal{Z}$ sends (`Setup`, sid, T) to corrupted party T (i.e., $\mathcal{S}$), $\mathcal{S}$ receives the message and sends it to $\mathcal{F}_{\mathrm{IBE}}$ on behalf of T and the simulated copy of $\mathcal{A}$, which returns a reply message (which may be $\perp$) to $\mathcal{S}$. When $\mathcal{S}$ receives (`Setup`, sid, T) from $\mathcal{F}_{\mathrm{IBE}}$, $\mathcal{S}$ sends $\mathcal{A}$'s reply to $\mathcal{F}_{\mathrm{IBE}}$. $\mathcal{S}$ sends $\mathcal{A}$'s reply to $\mathcal{Z}$.
2. When $\mathcal{Z}$ sends (`Extract`, sid, ID, D) to corrupted D (i.e., $\mathcal{S}$), $\mathcal{S}$ sends it to the simulated copy of $\mathcal{A}$. If $\mathcal{A}$

requests the private key, forwards it to $\mathcal{F}_{\mathrm{IBE}}$. When $\mathcal{S}$ receives (Corrupted Extract, sid, ID, D) from $\mathcal{F}_{\mathrm{IBE}}$, $\mathcal{S}$ sends it to the simulated copy of $\mathcal{A}$, which returns a reply message (which may be $\perp$) to $\mathcal{S}$. If $\mathcal{A}$'s reply is decryption algorithm $\tilde{d}_{ID}$ (which may be inappropriate), $\mathcal{S}$ sends (Corrupted Decrypt, $sid, ID, \tilde{d}_{ID}$) to $\mathcal{F}_{\mathrm{IBE}}$. $\mathcal{F}_{\mathrm{IBE}}$ records ($ID, D, \tilde{d}_{ID}$, corrupted) and returns (Extracted, sid, ID, D). If $\mathcal{A}$'s reply is $\perp$, $\mathcal{S}$ sends it to $\mathcal{F}_{\mathrm{IBE}}$ and $\mathcal{F}_{\mathrm{IBE}}$ returns an error message.

3. When $\mathcal{Z}$ sends (Encrypt, sid, m, ID, e') to E, E forwards it to $\mathcal{F}_{\mathrm{IBE}}$. $\mathcal{F}_{\mathrm{IBE}}$ generates $c = e'(ID, m)$ and returns (Ciphertext, sid, c) to E, since T (i.e., $\mathcal{S}$) sent no (Setup, sid, T) to $\mathcal{F}_{\mathrm{IBE}}$, which records nothing as encryption algorithm e.
4. When $\mathcal{Z}$ sends (Decrypt, sid, c, ID) to corrupted party D (i.e., $\mathcal{S}$), $\mathcal{S}$ sends (Decrypt, sid, c, ID) to $\mathcal{A}$. $\mathcal{A}$ returns a reply (which may be $\perp$) to $\mathcal{S}$, which forwards $\mathcal{A}$'s reply to $\mathcal{Z}$.

In this case, $\mathcal{Z}$ cannot distinguish whether $(\mathcal{S}, \mathcal{F}_{\mathrm{IBE}})$ or $(\mathcal{A}, \pi_{\Sigma})$, because the message returned by $\mathcal{S}$ (using $\mathcal{A}$) as T and D in the ideal world is the same as that returned by $\mathcal{A}$ as T and D in the real world, and (Ciphertext, sid, c) returned by $\mathcal{F}_{\mathrm{IBE}}$ is exactly the same as that returned by E in the real world.

In Case 6, we can construct simulator $\mathcal{S}$ such that no $\mathcal{Z}$ can distinguish whether $(\mathcal{S}, \mathcal{F}_{\mathrm{IBE}})$ or $(\mathcal{A}, \pi_{\Sigma})$ as follows:

1. When $\mathcal{Z}$ sends (Setup, sid, T) to T, T forwards it to $\mathcal{F}_{\mathrm{IBE}}$. $\mathcal{F}_{\mathrm{IBE}}$ sends (Setup, sid, T) to $\mathcal{S}$, $\mathcal{S}$ computes (PK, MK) by running algorithm Set, and generates x, e and d, where $x = \mathsf{Xtr}(PK, MK, \cdot)$, $e = \mathsf{Enc}(PK, \cdot, \cdot)$ and $d = \mathsf{Dec}(PK, \cdot, \cdot)$. $\mathcal{S}$ returns (Algorithms, sid, x, e, d) to $\mathcal{F}_{\mathrm{IBE}}$.
2. When $\mathcal{Z}$ sends (Extract, sid, ID, D) to corrupted D (i.e., $\mathcal{S}$), $\mathcal{S}$ sends it to the simulated copy of $\mathcal{A}$. If $\mathcal{A}$ requests the private key, forwards it to $\mathcal{F}_{\mathrm{IBE}}$ and returns $dk_{ID} = x(ID)$. $\mathcal{F}_{\mathrm{IBE}}$ records (ID, D) and returns (Extracted, sid, ID, D).
3. When $\mathcal{Z}$ sends (Encrypt, sid, m, ID, e') to corrupted party E (i.e., $\mathcal{S}$), $\mathcal{S}$ receives the message and sends it to the simulated copy of $\mathcal{A}$, which replies to $\mathcal{S}$. $\mathcal{S}$ then returns $\mathcal{A}$'s reply (which may be $\perp$) to $\mathcal{Z}$.
4. When $\mathcal{Z}$ sends (Decrypt, sid, c, ID) to corrupted party D (i.e., $\mathcal{S}$), $\mathcal{S}$ sends (Decrypt, sid, c, ID) to $\mathcal{A}$. $\mathcal{A}$ returns a reply (which may be $\perp$) to $\mathcal{S}$, which forwards $\mathcal{A}$'s reply to $\mathcal{Z}$.

In this case, $\mathcal{Z}$ cannot distinguish whether $(\mathcal{S}, \mathcal{F}_{\mathrm{IBE}})$ or $(\mathcal{A}, \pi_{\Sigma})$, because the message returned by $\mathcal{S}$ (using $\mathcal{A}$) as E and D in the ideal world is the same as that returned by $\mathcal{A}$ as E and D in the real world, (Encryption Algorithm, sid, e) returned by $\mathcal{F}_{\mathrm{IBE}}$ is exactly the same as that returned by D in the real world, and (Extracted, sid, ID, D) returned by $\mathcal{F}_{\mathrm{IBE}}$ is exactly the same as that returned by T in the real world.

Thus, $\mathcal{Z}$ cannot distinguish $(\mathcal{S}, \mathcal{F}_{\mathrm{IBE}})$ or $(\mathcal{A}, \pi_{\Sigma})$ in Cases 1, 2, 3, 4, 5, and 6. Hereafter, we consider only Case B.

Recall that $\mathcal{A}$ takes three types of messages from $\mathcal{Z}$: either to corrupt parties, or to report on messages sent in the protocol, or to deliver some messages. There are no party corruption instructions, since we are dealing with non-adaptive adversaries. However, $\mathcal{Z}$ may request some corrupted parties to reveal their private keys, so $\mathcal{A}$ need report private keys to $\mathcal{Z}$.

Thus, the activity of $\mathcal{S}$ is to provide the algorithms to $\mathcal{F}_{\mathrm{IBE}}$ and to report private keys. Since $\mathcal{Z}$ succeeds in distinguishing for any $\mathcal{S}$, it also succeeds for the following specific $\mathcal{S}$. Simulator $\mathcal{S}$ acts as follows:

When $\mathcal{S}$ receives message (Setup, sid, T) from $\mathcal{F}_{\mathrm{IBE}}$, it runs setup algorithm Set, obtains system parameters PK and master-key MK, and returns $x = \mathsf{Xtr}(PK, MK, \cdot)$, $e = \mathsf{Enc}(PK, \cdot, \cdot)$ and $d = \mathsf{Dec}(PK, \cdot, \cdot)$ to $\mathcal{F}_{\mathrm{IBE}}$.

When $\mathcal{Z}$ requests private keys, $\mathcal{S}$ returns them by using extraction algorithm $x = \mathsf{Xtr}(PK, MK, \cdot)$.

We consider the case where setup party T, some encryptor E and some decryptor D are uncorrupted and assume for contradiction that there is environment $\mathcal{Z}$ that can distinguish whether $(\mathcal{S}, \mathcal{F}_{\mathrm{IBE}})$ or $(\mathcal{A}, \pi_{\Sigma})$. We now prove that we can construct adversary G that breaks IND-ID-CCA2 security by using environment $\mathcal{Z}$. More precisely, we assume that there is real life adversary $\mathcal{A}$ such that for any ideal process adversary $\mathcal{S}$, there exists environment $\mathcal{Z}$ such that for fixed value k of security parameter and fixed input z for $\mathcal{Z}$,

$$|\mathtt{IDEAL}_{\mathcal{F}_{\mathrm{IBE}},\mathcal{S},\mathcal{Z}}(k, z) - \mathtt{REAL}_{\pi_{\Sigma},\mathcal{A},\mathcal{Z}}(k, z)| > \nu(k)$$

We then show that there exists G_h whose advantage $\mathbf{Adv}^{\text{ind-id-cca2}}_{\Sigma,G_h}(k) > \nu(k)/l$ in the IND-ID-CCA2 game, where *l is the total number of messages that were encrypted by uncorrupted party's ID* (Extract has already executed) throughout the running of the system and $h \in \{1, \ldots, l\}$. G_h is given system parameters PK, and is allowed to query $\mathcal{XO}_i$, $\mathcal{EO}_i$ and $\mathcal{DO}_i$ (as in Preliminaries). G_h runs $\mathcal{Z}$ on the following simulated interaction with a system running $\pi_{\Sigma}/\mathcal{F}_{\mathrm{IBE}}$. Let (m_j, ID_j) denote the jth pair of message and ID that $\mathcal{Z}$ activates some party with (Encrypt, sid, m_j, ID_j, e) in this simulation. Note that ID_j is uncorrupted party's ID and the private key for ID_j has already extracted.

1. When $\mathcal{Z}$ activates some party T with input (Setup, sid, T), G_h lets T output value e calculated from PK.
2. When $\mathcal{Z}$ activates some party P with input (Extract, sid, ID, P), G_h lets P output message (Extracted, sid, ID, P) from G_h's input. If P is corrupted and $\mathcal{Z}$ requests P's private key, then G_h queries $\mathcal{XO}_i$ on ID, obtains value u and lets P return u to $\mathcal{Z}$. This is perfect simulation, so $\mathcal{Z}$ cannot distinguish $(\mathcal{S}, \mathcal{F}_{\mathrm{IBE}})$ or $(\mathcal{A}, \pi_{\Sigma})$ in this step.
3. For the first $h - 1$ times that $\mathcal{Z}$ asks encryptor E to encrypt some message, m_j, G_h lets E return $c_j = e(ID_j, m_j)$.
4. The h-th time that $\mathcal{Z}$ asks E to encrypt message, m_h by ID^*, G_h queries encryption oracle $\mathcal{EO}_i$ with the pair of messages (m_h, μ), where $\mu \in \mathsf{M}$ is the fixed message,

and obtains target ciphertext c_h. It then hands c_h to $\mathcal{Z}$ as the encryption of m_h. That is, $c_h = \mathsf{Enc}(PK, ID^*, m_h)$ $(b = 0)$ or $c_h = \mathsf{Enc}(PK, ID^*, \mu)$ $(b = 1)$.

5. For the remaining $l - h$ times that $\mathcal{Z}$ asks E to encrypt some message, m_j, G_h lets E return $c_j = \mathsf{Enc}(PK, ID_j, \mu)$.
6. Whenever decryptor D is activated with input $(\mathtt{Decrypt}, sid, c, ID)$ where $c = c_j$ and $ID = ID_j$ for some j, G_h lets D return the corresponding plaintext m_j. If c is different from all c_j's and ID_j is extracted, G_h queries $\mathcal{DO}_i$ on (ID, c), obtains value v, and lets D return v to $\mathcal{Z}$. If c is different from all c_j's and ID_j is not extracted, G_h lets D output `not-recorded`. This is perfect simulation, so $\mathcal{Z}$ cannot distinguish $(\mathcal{S}, \mathcal{F}_{\mathrm{IBE}})$ or $(\mathcal{A}, \pi_\Sigma)$ in this step.
7. When $\mathcal{Z}$ halts, G_h outputs whatever $\mathcal{Z}$ outputs and halts.

Notice that $\mathcal{Z}$ cannot distinguish $(\mathcal{S}, \mathcal{F}_{\mathrm{IBE}})$ or $(\mathcal{A}, \pi_\Sigma)$ by activating E with $(\mathtt{Encrypt}, sid, m, ID, e)$ before activating T with $(\mathtt{Extract}, sid, ID, P_h)$, because in this case, $c = e(ID, m)$ in both the real and the ideal world.

We apply a standard hybrid argument for analyzing the success probability of G_h. For $j \in \{0, \ldots, l\}$, let $\mathtt{Env}_j$ be an event that $\mathcal{Z}$ interacts with $\mathcal{S}$ in the ideal process, with the exception that the first j ciphertexts are computed as an encryption of the real plaintexts, rather than encryptions of μ. The replies to $\mathcal{Z}$ from setup party T and decryptor D are the same as those shown in step 1, 2 and 6 above. Let H_j be $\Pr[\mathcal{Z} \to 1 | \mathtt{Env}_j]$.

Notice that in steps 2 and 6, $\mathcal{Z}$ cannot tell whether it is interacting with $\mathcal{A}$ and π_Σ or with $\mathcal{S}$ in the ideal process for $\mathcal{F}_{\mathrm{IBE}}$, because G_h offers perfect simulation.

It is easy to see that H_0 is identical to the probability that $\mathcal{Z}$ outputs 1 in the ideal process, and that H_l is identical to the probability that $\mathcal{Z}$ outputs 1 in the real life model. Furthermore, in a run of G_h, if value c_h that G_h obtains from its encryption oracle is encryption m_h, the probability that $\mathcal{Z}$ outputs 1 is identical to H_{h-1}. If c_h is an encryption of μ, the probability that $\mathcal{Z}$ outputs 1 is identical to H_h. Details follow:

$$
\begin{aligned}
H_0 &= \mathtt{IDEAL}_{\mathcal{F}_{\mathrm{IBE}}, \mathcal{S}, \mathcal{Z}}(k, z) \\
H_l &= \mathtt{REAL}_{\pi_\Sigma, \mathcal{A}, \mathcal{Z}}(k, z) \\
H_h &= \Pr[G_h \to 1 | c_h = \mathsf{Enc}(PK, ID^*, \mu)] \\
H_{h-1} &= \Pr[G_h \to 1 | c_h = \mathsf{Enc}(PK, ID^*, m_h)]
\end{aligned}
$$

$$
\begin{aligned}
\sum_{i=1}^{l} |H_{i-1} - H_i| &\geq |\sum_{i=1}^{l} (H_{i-1} - H_i)| \\
&= |H_0 - H_l| \\
&= |\mathtt{IDEAL}_{\mathcal{F}_{\mathrm{IBE}}, \mathcal{S}, \mathcal{Z}}(k, z) - \mathtt{REAL}_{\pi_\Sigma, \mathcal{A}, \mathcal{Z}}(k, z)| \\
&> \nu(k)
\end{aligned}
$$

Therefore, there exists some $h \in \{1, \ldots, l\}$ such that $|H_{h-1} - H_h| > \nu(k)/l$. Here, w.l.o.g, let $H_{h-1} - H_h > \nu(k)/l$, since if $H_h - H_{h-1} > \nu(k)/l$ for $\mathcal{Z}$, we can obtain $H_{h-1} - H_h > \nu(k)/l$ for $\mathcal{Z}^*$, where $\mathcal{Z}^*$ outputs the opposite of $\mathcal{Z}$'s output bit.

We have the advantage of adversary G_h as follows:

$$
\begin{aligned}
&\mathbf{Adv}^{\text{ind-id-cca2}}_{\Sigma, G_h}(k) \\
&= \Pr[\mathbf{Exp}^{\text{ind-id-cca2-1}}_{\Sigma, \mathcal{A}}(k) = 1] \\
&\quad - \Pr[\mathbf{Exp}^{\text{ind-id-cca2-0}}_{\Sigma, \mathcal{A}}(k) = 1] \\
&= \Pr[G_h \to 1 | c_h = \mathsf{Enc}(PK, ID^*, \mu)] \\
&\quad - \Pr[G_h \to 1 | c_h = \mathsf{Enc}(PK, ID^*, m_h)] \\
&= H_h - H_{h-1} > \nu(k)/l
\end{aligned}
$$

That is, G_h has non-negligible advantage in k since l is polynomially bounded in k. □

References

[1] N. Attrapadung, Y. Cui, D. Galindo, G. Hanaoka, I. Hasuo, H. Imai, K. Matsuura, P. Yang, and R. Zhang, "Relations among notions of security for identity based encryption schemes," Proc. LATIN'06, 3887 of LNCS, pp.130–141, 2006.

[2] B. Barak, R. Canetti, Y. Lindell, R. Pass, and T. Rabin, "Secure computation without authentication," Proc. CRYPTO'05, 3621 of LNCS, pp.361–377, 2005.

[3] B. Barak, R. Canetti, J.B. Nielsen, and R. Pass, "Universally composable protocols with relaxed set-up assumptions," Proc. FOCS'04, pp.186–195, 2004.

[4] D. Boneh and X. Boyen, "Efficient selective-id secure identity based encryption without random oracles," Proc. EUROCRYPT'04, 3027 of LNCS, pp.223–238, 2004.

[5] D. Boneh and X. Boyen, "Secure identity based encryption without random oracles," Proc. CRYPTO'04, 3152 of LNCS, pp.443–459, 2004.

[6] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," Proc. CRYPTO'01, 2139 of LNCS, pp.213–229, 2001.

[7] D. Boneh and J. Katz, "Improved efficiency for cca-secure cryptosystems built using identity based encryption," Proc. CT-RSA'05, 3376 of LNCS, pp.87–103, 2005.

[8] R. Canetti, "Universally composable security: A new paradigm for cryptograpic protocols," Proc. FOCS'01, pp.136–145, 2001. Current Full Version Available at Cryptology ePrint Archive, Report 2000/067 http://eprint.iacr.org/

[9] R. Canetti, "Universally composable signatures, certification, and authenticated communication," Proc. 17th Computer Security Foundations Workshop, pp.219–233, 2004.

[10] R. Canetti and M. Fischlin, "Universally composable commitments," Proc. CRYPTO'01, 2139 of LNCS, pp.19–40, 2001.

[11] R. Canetti, S. Halevi, and J. Katz, "A forward-secure public-key encryption scheme," Proc. EUROCRYPT'03, 2656 of LNCS, pp.255–271, 2003.

[12] R. Canetti, S. Halevi, and J. Katz, "Chosen-ciphertext security from identity-based encryption," Proc. EUROCRYPT'04, 3027 of LNCS, pp.207–222, 2004.

[13] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," Proc. EUROCRYPT'02, 2332 of LNCS, pp.337–351, 2002.

[14] R. Canetti, E. Kushilevitz, and Y. Lindell, "On the limitations of universally composable two-party computation without set-up assumptions," Proc. EUROCRYPT'03, 2656 of LNCS, pp.68–86, 2003.

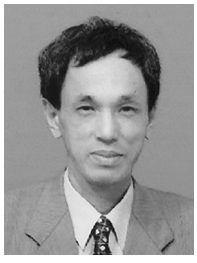
[15] R. Canetti, Y. Lindell, R. Ostrovsky, and A. Sahai, "Universally composable two-party and multi-party secure computation," Proc. STOC'02, pp.494–503, 2002.

[16] R. Canetti and T. Rabin, "Universal composition with joint state," Proc. CRYPTO'03, 2729 of LNCS, pp.265–281, 2003.

[17] I. Damgård and J.B. Nielsen, "Perfect hiding and perfect binding

universally composable commitment schemes with constant expansion factor," Proc. CRYPTO'02, 2442 of LNCS, pp.581–596, 2002.
[18] A. Datta, A. Derek, J.C. Mitchell, A. Ramanathan, and A. Scedrov, "Games and the impossibility of realizable ideal functionality," Proc. of TCC'06, 3876 of LNCS, pp.360–379, 2006.
[19] W. Nagao, Y. Manabe, and T. Okamoto, "On the equivalence of several security notions of key encapsulation mechanism," Cryptology ePrint Archive, Report 2006/268, 2006. http://eprint.iacr.org/
[20] M. Prabhakaran and A. Sahai, "New notions of security: Achieving universal composability without trusted setup," Proc. STOC'04, pp.242–251, 2004.
[21] A. Shamir, "Identity-based cryptosystems and signature schemes," Proc. CRYPTO'84, 196 of LNCS, pp.47–53, 1984.
[22] B. Waters "Efficient identity-based encryption without random oracles," Proc. EUROCRYPT'05, 3494 of LNCS, pp.114–127, 2005.

Ryo Nishimaki received the B.E., M.I., degrees from Kyoto University, Kyoto Japan in 2005 and 2007, respectively. In 2007, he joined NTT Information Sharing Platform Laboratories and he is also a doctor course student of Tokyo Institute of Technology. His research interests are theory of cryptography and its relation to computational complexity.



Yoshifumi Manabe received the B.E., M.E., and Dr.E. degrees from Osaka University, Osaka, Japan, in 1983, 1985, and 1993, respectively. In 1985, he joined Nippon Telegraph and Telephone Corporation. Currently, he is a senior research scientist, supervisor of NTT Communication Science Laboratories. His research interests include distributed algorithms, cryptography, and operating systems. He has been a guest associate professor of Kyoto University since 2001. He is a member of ACM, IPSJ, and IEEE.

Tatsuaki Okamoto received the B.E., M.E., and Dr.E. degrees from the University of Tokyo, Tokyo, Japan, in 1976, 1978, and 1988, respectively. He is a Fellow of NTT Information Sharing Platform Laboratories. He is presently engaged in research on cryptography and information security. Dr. Okamoto is a director of the Japan Society for Industrial and Applied Mathematics, and a guest professor of Kyoto University.