Anonymous Identity-Based Broadcast Encryption with Constant Decryption Complexity and Strong Security

Peng Xu[†], Jingnan Li[†], Wei Wang[‡], Hai Jin[†]
[†]Services Computing Technology and System Lab, Cluster and Grid Computing Lab, Big Data Technology and System Lab, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, 430074, China
[‡]Cyber-Physical-Social Systems Lab, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, 430074, China
{xupeng, jnli}@mail.hust.edu.cn, viviawangww@gmail.com, hjin@mail.hust.edu.cn

ABSTRACT

Anonymous Identity-Based Broadcast Encryption (AIBBE) allows a sender to broadcast a ciphertext to multi-receivers, and keeps receivers' anonymity. The existing AIBBE schemes fail to achieve efficient decryption or strong security, like the constant decryption complexity, the security under the adaptive attack, or the security in the standard model. Hence, we propose two new AIBBE schemes to overcome the drawbacks of previous schemes in the state-of-art. The biggest contribution in our work is the proposed AIBBE scheme with constant decryption complexity and the provable security under the adaptive attack in the standard model. This scheme should be the first one to obtain advantages in all above mentioned aspects, and has sufficient contribution in theory due to its strong security. We also propose another AIBBE scheme in the *Random Oracle* (RO) model, which is of sufficient interest in practice due to our experiment.

Keywords

Anonymous Identity-Based Broadcast Encryption, Constant Decryption Complexity, Adaptive Attack, Standard Model, Random Oracle Model

1. INTRODUCTION

With the increased determination of content owners to preserve copyright and the enhanced awareness of common users to protect their privacy, many multi-user applications, such as digital content distribution and pay-per-view, require a cryptographic mechanism to prevent unauthorized users and keep receiver anonymity. *Anonymous Identity-Based Broadcast Encryption* (AIBBE) seems to be an ideal solution to settle the issue mentioned above. AIBBE allows a sender to generate a broadcast ciphertext with multiple intended receivers' identities. Given the generated ciphertext, every intended receiver can decrypt out the contained plaintext. In general, a secure AIBBE scheme guarantees that no one except the intended receivers can learn anything about the plaintext, and know who are the intended receivers.

In recent years, the work of designing an efficient AIBBE scheme has gained an increasing interest both from the academic and industrial communities. As a result, some AIBBE schemes have been proposed. However, they have significant disadvantages in the aspect of performance or security. In general, the existing AIBBE schemes can be categorized into two types according to their mathematical characteristics. One type is based on Lagrange-polynomial, and another type is based on randomness-reuse. Suppose N is the number of the intended receivers of an AIBBE ciphertext. In the Lagrange-polynomial-based AIBBE schemes, such as the schemes in [1–5], it is unavoidable that their decryption algorithms need to compute a polynomial of degree N.

Similarly, the early proposed randomness-reuse-based AIBBE schemes, such as the schemes in [6–9], have the same decryption complexity as the Lagrange-polynomial-based AIBBE schemes. In contrast, the later proposed randomness-reuse-based AIBBE schemes, such as the schemes in [10,11], claim that they achieve the constant decryption complexity. But they do not show the details in their decryption algorithms. In our opinion, those details are very important, since they are relevant not only to decryption complexity but also to anonymity. More explanations will be given in the following content.

In the aspect of security, no exiting randomness-reuse-based AIBBE scheme can prove its security in the standard model or guarantee a strong anonymity. The former characteristic is obviously important for the theoretical research of AIBBE, since it avoids the unreasonable assumption in the *Random Oracle* (RO) model. The latter characteristic is also important, but never be observed by previous AIBBE schemes. Roughly, to encrypt a plaintext M, the existing randomness-reuse-based AIBBE schemes takes an identity set $\mathcal{I} = \{ID_1, \cdots, ID_N\}$ as input, and generates a ciphertext $\mathcal{C}$ having the general form

$$\mathcal{C} = (C_0, C_1 = E(ID_1, M), \cdots, C_N = E(ID_N, M)),$$

where C_0 is the part of a reused randomness, and C_i is a partial ciphertext corresponding to receiver ID_i. Given $\mathcal{C}$, every receiver ID_i retrieves his partial ciphertext C_i, and decrypts out plaintext M using his private key and C_0. It is clear that the generated ciphertext $\mathcal{C}$ implies the order of identities in set $\mathcal{I}$. Suppose that in the security proof, an adversary with

ASIA CCS '16, May 30-June 03, 2016, Xi'an, China
© 2016 ACM. ISBN 978-1-4503-4233-9/16/05...$15.00
DOI: http://dx.doi.org/10.1145/2897845.2897853

identity ID takes two identity sets as his challenge, and identity ID has different positions in those two sets. Then the adversary can distinguish the corresponding two challenge ciphertexts according to the different positions, where he can retrieve his partial ciphertexts from those two challenge ciphertexts.

In addition, most of existing randomness-reuse-based AIBBE schemes are provably secure (including anonymous) under the non-adaptive attack. The non-adaptive attack means that an adversary must choose two challenge identity sets at the beginning of an attack game, i.e. no information can be leveraged by the adversary to make his choice. In contrast, the adaptive attack allows an adversary to adaptively choose two challenge identity sets according to some historical information. It is obvious that the adaptive attack is stronger than the non-adaptive one. So far as we known, only the scheme in [8] is provably secure under the adaptive attack. But this scheme does not achieve the above strong anonymity.

1.1 Our Ideas

In this paper, our final goal is to design an AIBBE scheme with constant decryption complexity, and provable strong security (including anonymity) under the adaptive attack in the standard model. Note that we do not consider the security under chosen ciphertext attacks, since our proposed AIBBE schemes can be easily extended by some general methods [12–15] to achieve that security. In order to obtain those advantages, we adopt the following ideas.

To obtain the constant decryption complexity and the strong anonymity, we adopt the idea of dual randomness-reuse and a standard data structure called history-independent dictionary. In general, to encrypt a plaintext M with an identity set $\mathcal{I} = \{ID_1, \cdots, ID_N\}$, the idea of dual randomness-reuse allows our AIBBE scheme to generate two randomness-reused parts C_0 and C_0', and N label-ciphertext pairs $\{(I_i, C_i = E(ID_i, M)) | i \in [1, N]\}$; then $\{(I_i, C_i = E(ID_i, M)) | i \in [1, N]\}$ are stored in a history-independent dictionary $\mathcal{L}$, i.e. the position of C_i in $\mathcal{L}$ is independent with i; finally the generated ciphertext is $\mathcal{C} = (C_0, C_0', \mathcal{L})$. Given C_0 and the corresponding private key, receiver ID_i can compute label I_i, and retrieve C_i according to I_i from $\mathcal{L}$ in complexity $O(1)$; given C_0', receiver ID_i can further decrypt out plaintext M from C_i. It is clear that (1) if the complexity to compute label I_i and decrypt out plaintext M from C_i is constant, our AIBBE scheme achieves constant decryption complexity, and (2) due to the property of history-independency, dictionary $\mathcal{L}$ does not imply the order of identities in set $\mathcal{I}$.

To obtain the provable security under the adaptive attack in the standard model, we adopt a programmable hash function in the multilinear setting, which was introduced in [16]. In general, the security proof of AIBBE is a game between an adversary and a simulator. In the game, the simulator must forge an ideal AIBBE scheme, such that the ideal AIBBE scheme is indistinguishable with a real AIBBE scheme in the view of the adversary; the adversary can issue some queries to the simulator. Most of existing randomness-reuse-based AIBBE schemes can not prove their securities under the adaptive attack, since they can not forge an effective hash function to respond the queries of the adversary. In other words, their forged hash functions only have a negligible probability (which is related to N) to be effective. Fortunately, this problem can be avoided by the programmable hash function. Moreover, the programmable hash function needs not to be assumed as a random oracle in the security proof. This property allows our AIBBE scheme to be provably secure in the standard model.

1.2 Our Contributions

In this paper, we sequentially propose two AIBBE schemes. Both of them have constant decryption complexity and strong anonymity. The first AIBBE scheme is provably secure under the non-adaptive attack in the RO model. This scheme clearly presents our idea of dual randomness-reuse and the application of a history-independent dictionary. The experimental results show that this scheme is practical. Hence, it is of independent interest for the industrial community. The second AIBBE scheme is provably secure under the adaptive attack in the standard model. This scheme shows the application of the programmable hash function, and achieves our final goal of this paper.

1.3 Organization

The remainder is as follows. Section 2 gives some related definitions about AIBBE, and the definition of history-independent dictionary. Section 3 introduces our first AIBBE scheme, which is provably secure in the RO model, and shows that this scheme is practical by some numerical results. Section 4 introduces our second AIBBE scheme, which is provably secure in the standard model. More details about the related works are introduced in Section 5. Section 6 concludes this paper.

2. PRELIMINARY

In theory, an AIBBE scheme means an IBBE scheme with anonymity and semantic security. Hence, to define AIBBE, we need to define IBBE and its anonymity and semantic security. These definitions and the definition of history-independent dictionary are introduced in this section. Let $\gamma \xleftarrow{\$} \Re$ denote an element γ randomly sampled from $\Re$. The definitions are as follows.

Definition 1. ***(IBBE)**. Let $N \in \mathbb{N}$ be the maximal size of receiver set for one IBBE encryption. An IBBE scheme consists of the following four algorithms:*

- **Setup**$(1^k, \mathcal{ID})$: *Take as inputs a security parameter 1^k where $k \in \mathbb{N}$ and an identity space $\mathcal{ID}$, and output the master public key $\mathcal{MP}$ and the master secret key $\mathcal{MS}$;*
- **Extract**$(\mathcal{MS}, ID)$: *Take as inputs $\mathcal{MS}$ and an identity $ID \in \mathcal{ID}$, and output a private key SK_{ID} of ID;*
- **Enc**$(\mathcal{MP}, \mathcal{I}, M)$: *Take as inputs $\mathcal{MP}$, an identity set $\mathcal{I} = \{ID_1, ..., ID_N\}$ consisting of N intended receivers' identities (where $ID_i \in \mathcal{ID}$ with $i \in [1, N]$), and a plaintext M, and output a ciphertext $\mathcal{C}$;*
- **Dec**$(SK_{ID'}, \mathcal{C})$: *Take as inputs a private key $SK_{ID'}$ of identity ID' and a ciphertext $\mathcal{C}$, and output a plaintext M or $\perp$ otherwise ($\perp$ denotes a failed decryption).*

*In addition, an IBBE scheme must be consistent in the sense that for any $\mathcal{C} \leftarrow$ **Enc**$(\mathcal{MP}, \mathcal{I}, M)$ and $SK_{ID'} \leftarrow$ **Extract**$(\mathcal{MS}, ID')$, **Dec**$(SK_{ID'}, \mathcal{C}) = M$ holds if $ID' \in \mathcal{I}$, except with a negligible probability in the security parameter k.*

In the paradigmatic application of IBBE, a trusted *Key Generation Center* (KGC) initializes an IBBE scheme by running algorithm **Setup** and publishing the generated master public key, and generates private keys for all legal users by running algorithm **Extract**. A sender runs algorithm **Enc** to generate a broadcast ciphertext and sends the ciphertext to the intended receivers. When receiving the ciphertext, every intended receiver runs algorithm **Dec** to decrypt out the contained plaintext.

The anonymity of IBBE is defined as the *Anonymity under the adaptive-Multiple-IDs and Chosen Plaintext Attacks* (Anon-MID-CPA). It defines an attack game between a *Probabilistically Polynomial Time* (PPT) adversary and a challenger. In this game, the adversary queries the private keys of some identities, and then adaptively chooses two identity sets as his challenge. The challenger randomly chooses one of that two identity sets and generates a challenge ciphertext. We say that an IBBE scheme is Anon-MID-CPA secure if the adversary can not decide which one of that two identity sets was used to generate the challenge ciphertext. The Anon-MID-CPA security defines the strong anonymity, which was mentioned in Section 1, by allowing the adversary to query the identities belonging to the intersection of that two identity sets. The details of Anon-MID-CPA security are as follows.

Definition 2. ***(Anon-MID-CPA)****. An IBBE scheme is Anon-MID-CPA secure if any PPT adversary $\mathcal{A}$ has only a negligible advantage $Adv_{IBBE,\mathcal{A}}^{Anon\text{-}MID\text{-}CPA}$ to win in the following Anon-MID-CPA game:*

- ***Setup Phase****: A challenger sets up the IBBE scheme by running algorithm* **Setup** *to generate the master public-and-secret-keys pair* $(\mathcal{MP}, \mathcal{MS})$*, and sends* $\mathcal{MP}$ *to* $\mathcal{A}$*;*
- ***Query Phase 1****:* $\mathcal{A}$ *adaptively issues the following query multiple times.*
  - *Private Key Query* $\mathcal{Q}_{SK}(ID)$*: Given a queried identity* $ID \in \mathcal{ID}$*, the challenger returns a private key of identity* ID*;*
- ***Challenge Phase****:* $\mathcal{A}$ *sends two different challenge identity sets* $\mathcal{I}_0^*$ *and* $\mathcal{I}_1^*$ *and a plaintext* M *to the challenger, where* $\mathcal{I}_0^* = \{ID_1^{*0}, ..., ID_N^{*0}\}$ *and* $\mathcal{I}_1^* = \{ID_1^{*1}, ..., ID_N^{*1}\}$*. The challenger picks* $d \xleftarrow{\$} \{0,1\}$*, computes the challenge ciphertext* $\mathcal{C}_d^* = \mathbf{Enc}(\mathcal{MP}, \mathcal{I}_d^*, M)$*, and sends* $\mathcal{C}_d^*$ *to* $\mathcal{A}$*;*
- ***Query Phase 2****: This phase is the same with* ***Query Phase 1****. Note that both in* ***Query Phase 1*** *and* ***Query Phase 2****,* $\mathcal{A}$ *can not query the private keys corresponding to the challenge identities in* $\mathcal{I}_0^*$ *and* $\mathcal{I}_1^*$ *except the challenge identities in* $\mathcal{I}_0^* \cap \mathcal{I}_1^*$*;*
- ***Guess Phase****:* $\mathcal{A}$ *sends his guess* d' *to the challenger. We say that* $\mathcal{A}$ *wins if* $d' = d$*. Let* $Adv_{IBBE,\mathcal{A}}^{Anon\text{-}MID\text{-}CPA} = Pr[d' = d] - \frac{1}{2}$ *be the advantage of* $\mathcal{A}$ *to win in the above game.*

The anonymity of IBBE under the non-adaptive attack is defined as Anon-sMID-CPA. The corresponding attack game of Anon-sMID-CPA is the same with that of Anon-MID-CPA, except that an adversary in Anon-sMID-CPA must choose two challenge identity sets at the beginning of the attack game. Let $Adv_{IBBE,\mathcal{A}}^{\text{Anon-sMID-CPA}}$ denote the advantage of adversary $\mathcal{A}$ to win in this game.

The semantic security of IBBE is defined as the *Semantic Security under the adaptive-Multiple-IDs and Chosen Plaintext Attacks* (SS-MID-CPA). In the attack game defined in SS-MID-CPA, a PPT adversary queries the private keys of some identities, and then adaptively chooses a challenge identity set and two challenge plaintexts. A challenger randomly choose one of that two challenge plaintexts and generates a challenge ciphertext for the challenge identity set. We say that an IBBE scheme is SS-MID-CPA secure if the adversary can not decide which one of that two challenge plaintexts was used to generate the challenge ciphertext. Note that the adversary can not query the private keys of those challenge identities. The details of SS-MID-CPA security are as follows.

Definition 3. ***(SS-MID-CPA)****. An IBBE scheme is SS-MID-CPA secure if any PPT adversary $\mathcal{A}$ has only a negligible advantage $Adv_{IBBE,\mathcal{A}}^{SS\text{-}MID\text{-}CPA}$ to win in the following SS-MID-CPA game:*

- ***Setup Phase****: A challenger sets up the IBBE scheme by running algorithm* **Setup** *to generate the master public-and-secret-keys pair* $(\mathcal{MP}, \mathcal{MS})$*, and sends* $\mathcal{MP}$ *to* $\mathcal{A}$*;*
- ***Query Phase 1****:* $\mathcal{A}$ *adaptively issues the following query multiple times.*
  - *Private Key Query* $\mathcal{Q}_{SK}(ID)$*: Given a queried identity* $ID \in \mathcal{ID}$*, the challenger returns a private key of identity* ID*;*
- ***Challenge Phase****:* $\mathcal{A}$ *sends an identity set* $\mathcal{I}$ *and two challenge plaintexts* (M_0^*, M_1^*) *to the challenger, where* $\mathcal{I} = \{ID_1, ..., ID_N\}$ *and* $|M_0^*| = |M_1^*|$*. The challenger picks* $d \xleftarrow{\$} \{0,1\}$*, computes the challenge ciphertext* $\mathcal{C}_d^* = \mathbf{Enc}(\mathcal{MP}, \mathcal{I}, M_d^*)$*, and sends* $\mathcal{C}_d^*$ *to* $\mathcal{A}$*;*
- ***Query Phase 2****: This phase is the same with* ***Query Phase 1****. Note that both in* ***Query Phase 1*** *and* ***Query Phase 2****,* $\mathcal{A}$ *can not query the private keys corresponding to the identities in set* $\mathcal{I}$*;*
- ***Guess Phase****:* $\mathcal{A}$ *sends his guess* d' *to the challenger. We say that* $\mathcal{A}$ *wins if* $d' = d$*. Let* $Adv_{IBBE,\mathcal{A}}^{SS\text{-}MID\text{-}CPA} = Pr[d' = d] - \frac{1}{2}$ *be the advantage of* $\mathcal{A}$ *to win in the above game.*

The semantic security of IBBE under the non-adaptive attack is defined as SS-sMID-CPA. The corresponding attack game of SS-sMID-CPA is the same with that of SS-MID-CPA, except that an adversary in SS-sMID-CPA must choose a challenge identity set at the beginning of the attack game. Let $Adv_{IBBE,\mathcal{A}}^{\text{SS-sMID-CPA}}$ denote the advantage of adversary $\mathcal{A}$ to win in this game.

In addition to the above definitions about AIBBE, our proposed AIBBE schemes will employ a standard data structure dictionary and two operations on dictionary. We define those two operations as follows:

- **Creat**$(\mathcal{T})$: Take a list $\mathcal{T}$ of label-data pairs as input (where each label is unique), and return a dictionary $\mathcal{L}$;

- $\mathbf{Get}(\mathcal{L}, I)$: Take a dictionary $\mathcal{L}$ and a label I as inputs, return the corresponding data D if $(I, D) \in \mathcal{D}$, otherwise return $NULL$.

Note that the dictionary operation $\mathbf{Creat}(\mathcal{T})$ is history-independent [17]. It means that for any list $\mathcal{T}$ the distribution of $\mathcal{D} \leftarrow \mathbf{Creat}(\mathcal{T})$ is independent with the order of the label-data pairs in $\mathcal{T}$. In addition, the time complexity of operation **Get** is $O(1)$.

3. OUR FIRST AIBBE SCHEME

Our first AIBBE scheme will be constructed by a popular mathematical tool called bilinear map. Before the construction, some related mathematical preliminaries will be introduced. After the construction, we will prove that our first AIBBE scheme is Anon-sMID-CPA and SS-sMID-CPA secure based on the *Decisional Bilinear Diffie-Hellman* (DBDH) assumption [18] in the RO model, and shows that our first AIBBE scheme is practical by some numerical results.

Let $\mathbb{G}$ and $\mathbb{G}_1$ denote two multiplicative groups of prime order q. Let g be a generator of $\mathbb{G}$. A bilinear map $\hat{\mathbf{e}} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$ is an efficiently computable and non-degenerate function, with the bilinearity property $\hat{\mathbf{e}}(g^a, g^b) = \hat{\mathbf{e}}(g, g)^{ab}$, where $(a, b) \overset{\$}{\leftarrow} \mathbb{Z}_q^*$ and $\hat{\mathbf{e}}(g, g)$ is a generator of $\mathbb{G}_1$. Let $\mathbf{BGen}(1^k)$ be an efficient bilinear map generator that takes as input a security parameter 1^k and probabilistically outputs $(q, \mathbb{G}, \mathbb{G}_1, g, \hat{\mathbf{e}})$. Let an identity space be $\mathcal{ID} = \{0, 1\}^*$.

Our first AIBBE scheme is constructed as follows.

- $\mathbf{Setup}(1^k, \mathcal{ID})$: Take as inputs 1^k and $\mathcal{ID}$, compute $(q, \mathbb{G}, \mathbb{G}_1, g, \hat{\mathbf{e}}) \leftarrow \mathbf{BGen}(1^k)$, pick $s \overset{\$}{\leftarrow} \mathbb{Z}_q^*$, set $p \leftarrow g^s$, choose a cryptographic hash functions $\mathbf{H} : \{0, 1\}^* \rightarrow \mathbb{G}$, and output the master public-and-secret keys

  $$\mathcal{MP} = (q, \mathbb{G}, \mathbb{G}_1, g, \hat{\mathbf{e}}, p, \mathbf{H}, \mathcal{ID}) \text{ and } \mathcal{MS} = s;$$

- $\mathbf{Extract}(\mathcal{MS}, ID)$: Take as inputs $\mathcal{MS}$ and an identity $ID \in \mathcal{ID}$, and output a private key $SK_{ID} = \mathbf{H}(ID)^s$ of ID;

- $\mathbf{Enc}(\mathcal{MP}, \mathcal{I}, M)$: Take as inputs $\mathcal{MP}$, an identity set $\mathcal{I} = \{ID_1, ..., ID_N\}$ of multiple receivers, and a plaintext $M \in \mathbb{G}_1$, and do the following steps:

  1. Choose values $(r_1, r_2) \overset{\$}{\leftarrow} \mathbb{Z}_q^*$, initialize an empty list $\mathcal{T}$, and compute $C_0 = g^{r_1}$ and $C_0' = g^{r_2}$;
  2. Compute $I_i = \hat{\mathbf{e}}(p, \mathbf{H}(ID_i))^{r_1}$ and $C_i = \hat{\mathbf{e}}(p, \mathbf{H}(ID_i))^{r_2} \cdot M$, and add the label-data pair (I_i, C_i) into $\mathcal{T}$ for $i \in [1, N]$;
  3. Generate dictionary $\mathcal{L} \leftarrow \mathbf{Creat}(\mathcal{T})$, and output a ciphertext $\mathcal{C} = (C_0, C_0', \mathcal{L})$;

- $\mathbf{Dec}(SK_{ID'}, \mathcal{C})$: Take as inputs a private key $SK_{ID'}$ of identity ID' and a ciphertext $\mathcal{C}$, and do the following steps:

  1. Parse $\mathcal{C} = (C_0, C_0', \mathcal{L})$, compute $I' = \hat{\mathbf{e}}(C_0, SK_{ID'})$, and retrieve $C' \leftarrow \mathbf{Get}(\mathcal{L}, I')$;
  2. Output a plaintext $M = C' \cdot \hat{\mathbf{e}}(C_0', SK_{ID'})^{-1}$.

In practice, we usually have $k \in [160, 512]$. For a fixed k, each execution of bilinear map $\hat{\mathbf{e}}$ takes a constant time complexity. Hence, it is obvious that the time complexity of the above algorithm **Dec** is constant for a fixed k. In other words, the time complexity of the above algorithm **Dec** is independent with the number of receivers of the ciphertext $\mathcal{C}$.

3.1 Consistency

Roughly, suppose that in the above algorithm **Dec** we have $ID' = ID_i$ where $ID_i \in \mathcal{I}$. It is easy to find that $I' = I_i$ holds, since $I' = \hat{\mathbf{e}}(C_0, SK_{ID'}) = \hat{\mathbf{e}}(g^{r_1}, \mathbf{H}(ID')^s) = \hat{\mathbf{e}}(g^s, \mathbf{H}(ID'))^{r_1} = I_i$. Since C_i is the corresponding ciphertext of I_i, we have $C' = C_i$. Moreover, according to the definition of bilinear map, it is easy to prove that the above algorithm **Dec** can output the correct plaintext. Formally, we have Theorem 1 on consistency whose proof can be found in Appendix A.

THEOREM 1. *For any ciphertext $\mathcal{C} \leftarrow \mathbf{Enc}(\mathcal{MP}, \mathcal{I}, M)$ and private key $SK_{ID'} \leftarrow \mathbf{Extract}(\mathcal{MS}, ID')$, the above IBBE scheme has $\mathbf{Dec}(SK_{ID'}, \mathcal{C}) = M$ if $ID' \in \mathcal{I}$, except with a negligible probability in the security parameter k.*

3.2 Anon-sMID-CPA Security Proof

The Anon-sMID-CPA security of the above IBBE scheme relies on the DBDH assumption in $\mathbf{BGen}(1^k)$. The definition of DBDH assumption [18] is as follows.

DEFINITION 4. *(**The DBDH Assumption** [18]). The DB-DH problem in $\mathbf{BGen}(1^k) = (q, \mathbb{G}, \mathbb{G}_1, g, \hat{\mathbf{e}})$ is defined as the advantage of any PPT algorithm $\mathcal{B}$ to distinguish the tuples $(g^a, g^b, g^c, \hat{\mathbf{e}}(g, g)^{abc})$ and $(g^a, g^b, g^c, \hat{\mathbf{e}}(g, g)^y)$, where $(a, b, c, y) \overset{\$}{\leftarrow} \mathbb{Z}_q^{*4}$. Let $Adv_{\mathcal{B}}^{DBDH}(1^k) = Pr[\mathcal{B}(g^a, g^b, g^c, \hat{\mathbf{e}}(g, g)^{abc}) = 1] - Pr[\mathcal{B}(g^a, g^b, g^c, \hat{\mathbf{e}}(g, g)^y) = 1]$ be the advantage of algorithm $\mathcal{B}$ to solve the DBDH problem. We say that the DBDH assumption holds in $\mathbf{BGen}(1^k)$, if the advantage $Adv_{\mathcal{B}}^{DBDH}(1^k)$ is negligible in the parameter k.*

In the security proof of Anon-sMID-CPA, we prove that if there is an adversary who can break the Anon-sMID-CPA security of the above IBBE scheme in the RO model, then there is an algorithm which can solve the DBDH problem in $\mathbf{BGen}(1^k)$. Formally, we have Theorem 2 whose proof can be found in Appendix B.

THEOREM 2. *Let the hash function $\mathbf{H}$ be modeled as the random oracle $\mathcal{Q}_H(\cdot)$. Suppose a PPT adversary $\mathcal{A}$ wins in the Anon-sMID-CPA game of the above IBBE scheme with advantage $Adv_{IBBE,\mathcal{A}}^{Anon\text{-}sMID\text{-}CPA}$. Then there is a PPT algorithm $\mathcal{B}$ that solves the DBDH problem in $\mathbf{BGen}(1^k)$ also with advantage $Adv_{\mathcal{B}}^{DBDH}(1^k) = Adv_{IBBE,\mathcal{A}}^{Anon\text{-}sMID\text{-}CPA}$.*

Since the advantage $Adv_{\mathcal{B}}^{DBDH}(1^k)$ is widely recognized as a negligible value in practice, Theorem 2 implies that the advantage $Adv_{IBBE,\mathcal{A}}^{\text{Anon-sMID-CPA}}$ is also negligible. Hence, the above IBBE scheme is Anon-sMID-CPA secure.

3.3 SS-sMID-CPA Security Proof

In the security proof of SS-sMID-CPA, we prove that if there is an adversary who can break the SS-sMID-CPA security of the above IBBE scheme in the RO model, then there is an algorithm which can solve the DBDH problem in $\mathbf{BGen}(1^k)$. Formally, we have Theorem 3 whose proof can be found in Appendix C.

Theorem 3. *Let the hash function* $\mathbf{H}$ *be modeled as the random oracle* $\mathcal{Q}_H(\cdot)$. *Suppose a PPT adversary* $\mathcal{A}$ *wins in the SS-sMID-CPA game of the above IBBE scheme with advantage* $Adv^{SS\text{-}sMID\text{-}CPA}_{IBBE,\mathcal{A}}$. *Then there is a PPT algorithm* $\mathcal{B}$ *that solves the DBDH problem in* $\mathbf{BGen}(1^k)$ *also with advantage* $Adv^{DBDH}_{\mathcal{B}}(1^k) = Adv^{SS\text{-}sMID\text{-}CPA}_{IBBE,\mathcal{A}}$.

By the same reason as the Anon-sMID-CPA security of the above IBBE scheme, Theorem 3 implies that the above IBBE scheme is SS-sMID-CPA secure in practice.

3.4 Experiment

We coded our first AIBBE scheme, and tested the time cost of algorithm **Dec** to decrypt the ciphertexts of different number of receivers. Table 1 shows the system parameters including hardware, software and the chosen elliptic curve. We generated several ciphertexts by algorithm **Enc** for different $N \in [5, 100]$, and each ciphertext was decrypted by a randomly chosen receiver of the ciphertext using algorithm **Dec**. Figure 1 shows the time cost of algorithm **Dec** to decrypt those ciphertexts. It is clear that the time cost of algorithm **Dec** is independent with N, and almost the same for all ciphertexts. Hence, our first AIBBE scheme is practical.

Table 1: Configuration of System Parameters

Hardware	Intel Core 2 Duo CPU E5300 @ 2.60GHz
OS and Compiler	Windows XP, Microsoft Visual C++ 6.0
Program Library	MIRACL version 5.4.1
Parameters of bilinear map	
Elliptic Curve	$y^2 = x^3 + A \cdot x + B \cdot x$
Pentanomial Basis	$t^m + t^a + t^b + t^c + 1$
Base Field: 2^m	$m = 379$
A	1
B	1
Group Order: q	$2^m + 2^{(m+1)/2} + 1$
a	315
b	301
c	287
The default unit is decimal.	

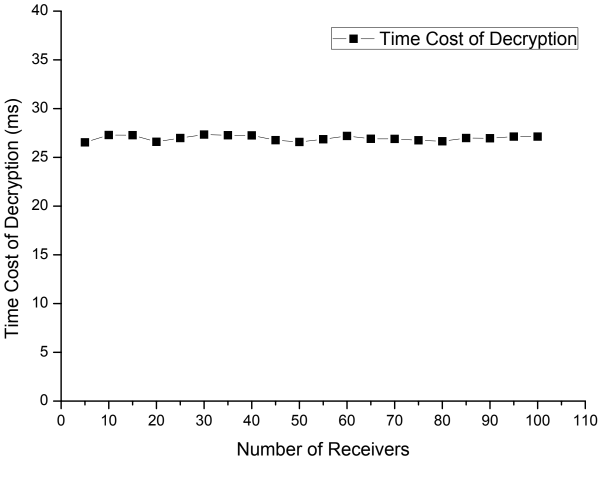

Figure 1: Time Cost of Decryption

4. OUR SECOND AIBBE SCHEME

Our second AIBBE scheme will be constructed by a new mathematical tool called multilinear map. Before the construction, multilinear map will be introduced. After the construction, we will prove that our second AIBBE scheme is Anon-MID-CPA and SS-MID-CPA secure in the standard model.

In [16], Freire et al. utilized the "approximation" of multilinear maps [19] to construct a programmable hash function in the multilinear setting (MPHF). To simplify the description of our second AIBBE scheme, we do not consider the "approximation" of multilinear maps. It means that we will leave out the functions that are the encoding of a group element, the re-randomization of an encoding and the extraction of an encoding. The following definitions will be used to construct our second AIBBE scheme and prove its security.

Definition 5. ***(Multilinear Maps** [16]).* *An* ℓ*-group system in multilinear setting consists of* ℓ *cyclic groups* $\mathbb{G}_1, \cdots, \mathbb{G}_\ell$ *of prime order* q*, along with bilinear maps* $\hat{\mathbf{e}}_{i,j} : \mathbb{G}_i \times \mathbb{G}_j \rightarrow \mathbb{G}_{i+j}$ *for all* $i, j \geq 1$ *with* $i + j \leq \ell$. *Let* g_i *be a generator of* $\mathbb{G}_i$. *The map* $\hat{\mathbf{e}}_{i,j}$ *satisfies* $\hat{\mathbf{e}}_{i,j}(g_i^a, g_j^b) = g_{i+j}^{ab}$ *(for all* $a, b \in \mathbb{Z}_q$*). When* i, j *are clear, we will simply write* $\hat{\mathbf{e}}$ *instead of* $\hat{\mathbf{e}}_{i,j}$. *It will also be convenient to abbreviate* $\hat{\mathbf{e}}(h_1, \cdots, h_j) = \hat{\mathbf{e}}(h_1, \hat{\mathbf{e}}(h_2, \cdots, \hat{\mathbf{e}}(h_{j-1}, h_j) \cdots))$ *for* $h_j \in \mathbb{G}_{i_j}$ *and* $i = (i_1 + i_2 + \cdots + i_j) \leq \ell$. *By induction, it is easy to see that this map is* j*-linear. Additionally, we define* $\hat{\mathbf{e}}(g) = g$. *Finally, it can also be useful to define the group* $\mathbb{G}_0 = \mathbb{Z}_q^+$ *of exponents to which this pairing family naturally extends. In the following, we will assume an* ℓ*-group system* $\mathbf{MPG}_\ell = \{\{\mathbb{G}_i\}_{i \in [1,\ell]}, q, \{\hat{\mathbf{e}}_{i,j}\}_{i,j \geq 1, i+j \leq \ell}\}$ *generated by a multilinear maps parameter generator* $\mathbf{MG}_\ell$ *on input a security parameter* 1^k.

Definition 6. ***(The** ℓ**-Multilinear Decisional Diffie-Hellman (MDDH) Assumption** [16]).* *Given* $(g, g^{x_1}, \cdots, g^{x_{\ell+1}})$ *(for* $g \xleftarrow{\$} \mathbb{G}_1$ *and uniform exponents* x_i*), the* ℓ*-MDDH assumption is that the element* $\hat{\mathbf{e}}(g^{x_1}, \cdots, g^{x_\ell})^{x_{\ell+1}} \in \mathbb{G}_\ell$ *is computationally indistinguishable from a uniform* $\mathbb{G}_\ell$*-element.*

Definition 7. ***(Group hash function** [16]).* *A group hash function* $\mathbf{H}$ *into* $\mathbb{G}$ *consists of two polynomial-time algorithms: the probabilistic algorithm* $\mathbf{HGen}(1^k)$ *output a key* hk*, and* $\mathbf{HEval}(hk, X)$ *(for a key* hk *and* $X \in \{0,1\}^k$*) deterministically outputs an image* $\mathbf{H}_{hk}(X) \in \mathbb{G}$.

Definition 8. ***(MPHF** [16]).* *Assume an* ℓ'*-group system* $\mathbf{MPG}_{\ell'}$ *as generated by* $\mathbf{MG}_{\ell'}(1^k)$. *Let* $\mathbf{H}$ *be a group hash function into* $\mathbb{G}_\ell$ $(\ell \leq \ell')$*, and let* $m, n \in \mathbb{N}$. *We say that* $\mathbf{H}$ *is an* (m, n)*-programmable hash function in the multilinear setting ((*m, n*)-MPHF) if there are PPT algorithms* **TGen** *and* **TEval** *as follows.*

- $\mathbf{TGen}(1^k, c_1, \cdots, c_l, h)$ *(for* $c_i, h \in \mathbb{G}_1$ *and* $h \neq 1$*) outputs a key* hk *and a trapdoor* td. *We require that for all* c_i *and* h*, that distribution of* hk *is statistically close to the output of* **HGen**.

- $\mathbf{TEval}(td, X)$ *(for a trapdoor* td *and* $X \in \{0,1\}^k$*) deterministically outputs* $a_X \in \mathbb{Z}_q^*$ *and* $B_X \in \mathbb{G}_{\ell-1}$ *with* $\mathbf{H}_{hk}(X) = \hat{\mathbf{e}}(c_1, \cdots, c_\ell)^{a_X} \cdot \hat{\mathbf{e}}(B_X, h)$. *We require that there is a polynomial* $p(k)$ *such that for all* hk

and $X_1, \cdots, X_m, Z_1, \cdots, Z_n \in \{0,1\}^k$ *with* $\{X_i\}_i \cap \{Z_j\}_j = \emptyset$,

$$P_{hk,\{X_i\},\{Z_j\}} = Pr[(a_{X_1} = \cdots = a_{X_m} = 0) \\ \wedge (a_{Z_1} = \cdots = a_{Z_n} \neq 0)] \geq 1/p(k),$$

where the probability is over possible trapdoors td output by **TGen** *along with the given hk. Furthermore, we require that* $P_{hk,\{X_i\},\{Z_j\}}$ *is close to statistically independent of hk. (Formally, we have* $|P_{hk,\{X_i\},\{Z_j\}} - P_{hk',\{X_i\},\{Z_j\}}| \leq v(k)$ *for all hk and hk′ in the range of* **TGen***, all* $\{X_i\}$ *and* $\{Z_j\}$*, and negligible* $v(k)$*.)*

We say that $\mathbf{H}$ *is a* (poly, n)*-MPHF if it is a* $(q(k), n)$*-MPHF for every polynomial* $q(k)$*. Note that* **TEval** *algorithm of an MPHF into* $\mathbb{G}_1$ *yields* $B_X \in \mathbf{G}_0$*, i.e., exponents* B_X*.*

Let an identity space be $\mathcal{ID} = \{0,1\}^k$. Our second AIBBE scheme is constructed as follows.

- **Setup**$(1^k, \mathcal{ID})$: Take as input a security parameter 1^k and space $\mathcal{ID}$, generate an $(\ell+1)$-group system $\mathbf{MPG}_{\ell+1} = \{\{\mathbb{G}_i\}_{i\in[1,\ell+1]}, q, \{\hat{\mathbf{e}}_{i,j}\}_{i,j\geq 1, i+j\leq \ell+1}\} \leftarrow \mathbf{MG}_{\ell+1}(1^k)$, generate a (poly, N)-MPHF $\mathbf{H}$ into $\mathbb{G}_\ell$ and $hk \leftarrow \mathbf{HGen}(1^k)$, pick $g \xleftarrow{\$} \mathbb{G}_1$ and $s \xleftarrow{\$} \mathbb{Z}_q$, set $p \leftarrow g^s$, and output the master public-and-secret keys
  $$\mathcal{MP} = (\mathbf{MPG}_{\ell+1}, g, p, \mathbf{H}, hk, \mathcal{ID}) \text{ and } \mathcal{MS} = (hk, s);$$
- **Extract**$(\mathcal{MS}, ID)$: Take as inputs $\mathcal{MS}$ and an identity $ID \in \mathcal{ID}$, and output a decryption key $SK_{ID} = \mathbf{H}_{\text{hk}}(ID)^s$ of ID.
- **Enc**$(\mathcal{MP}, \mathcal{I}, M)$: Take as inputs $\mathcal{MP}$, an identity set $\mathcal{I} = \{ID_1, ..., ID_N\}$ of multiple receivers, and a plaintext $M \in \mathbb{G}_{\ell+1}$, and do the following steps:
  1. Choose values $(r_1, r_2) \xleftarrow{\$} \mathbb{Z}_q^*$, initialize an empty list $\mathcal{T}$, and compute $C_0 = g^{r_1}$ and $C_0' = g^{r_2}$;
  2. Set $I_i = \hat{\mathbf{e}}(p, \mathbf{H}_{hk}(ID_i))^{r_1}$ and $C_i = \hat{\mathbf{e}}(p, \mathbf{H}_{hk}(ID_i))^{r_2} \cdot M$, and add the label-data pair (I_i, C_i) into $\mathcal{T}$ for $i \in [1, N]$;
  3. Generate dictionary $\mathcal{L} \leftarrow \mathbf{Creat}(\mathcal{T})$, and output ciphertext $\mathcal{C} = (C_0, C_0', \mathcal{L})$;
- **Dec**$(SK_{ID'}, \mathcal{C})$: Take as inputs a private key $SK_{ID'}$ of identity ID' and a ciphertext $\mathcal{C}$, and do the following steps:
  1. Parse $\mathcal{C} = (C_0, C_0', \mathcal{L})$, compute $I' = \hat{\mathbf{e}}(C_0, SK_{ID'})$, and retrieve $C' \leftarrow \mathbf{Get}(\mathcal{L}, I')$;
  2. Output a plaintext $M = C' \cdot \hat{\mathbf{e}}(C_0', SK_{ID'})^{-1}$.

4.1 Consistency

Roughly, suppose that in the above algorithm **Dec** we have $ID' = ID_i$ where $ID_i \in \mathcal{I}$. It is easy to find that $I' = I_i$ holds, since $I' = \hat{\mathbf{e}}(C_0, SK_{ID'}) = \hat{\mathbf{e}}(g^{r_1}, \mathbf{H}_{hk}(ID')^s) = \hat{\mathbf{e}}(g^s, \mathbf{H}_{hk}(ID'))^{r_1} = I_i$. Since C_i is the corresponding ciphertext of I_i, we have $C' = C_i$. Moreover, according to the definition of multilinear maps, it is easy to prove that the above algorithm **Dec** can output the correct plaintext. Formally, we have Theorem 4 on consistency.

THEOREM 4. *For any* $\mathcal{C} \leftarrow \mathbf{Enc}(\mathcal{MP}, \mathcal{I}, M)$ *and* $SK_{ID'} \leftarrow \mathbf{Extract}(\mathcal{MS}, ID')$*, the above IBBE scheme has* $\mathbf{Dec}(SK_{ID'}, \mathcal{C}) = M$ *if* $ID' \in \mathcal{I}$*, except with a negligible probability in the security parameter k.*

PROOF. Without loss of generality, let $\mathcal{I} = \{ID_1, \cdots, ID_N\}$, $ID' = ID_i$, and (I_i, C_i) be the label-data pair generated by algorithm $\mathcal{C} \leftarrow \mathbf{Enc}(\mathcal{MP}, \mathcal{I}, M)$, where $i \in [1, N]$, $I_i = \hat{\mathbf{e}}(p, \mathbf{H}_{hk}(ID_i))^{r_1}$ and $C_i = \hat{\mathbf{e}}(p, \mathbf{H}_{hk}(ID_i))^{r_2} \cdot M$. Parse $\mathcal{C} = (C_0, C_0', \mathcal{L})$. Algorithm $\mathbf{Dec}(SK_{ID'}, \mathcal{C})$ shows that $I' = \hat{\mathbf{e}}(C_0, SK_{ID'}) = \hat{\mathbf{e}}(g^{r_1}, \mathbf{H}_{hk}(ID')^s) = \hat{\mathbf{e}}(g^s, \mathbf{H}_{hk}(ID'))^{r_1} = \hat{\mathbf{e}}(p, \mathbf{H}_{hk}(ID'))^{r_1} = I_i$ according to the definition of multilinear maps. Hence, we have $C' = C_i$, except with a negligible probability in the security parameter k. According to equation $C' \cdot \hat{\mathbf{e}}(C_0', SK_{ID'})^{-1} = \hat{\mathbf{e}}(p, \mathbf{H}_{hk}(ID_i))^{r_2} \cdot M \cdot \hat{\mathbf{e}}(C_0', SK_{ID'})^{-1} = M$, we can prove that $\mathbf{Dec}(SK_{ID'}, \mathcal{C}) = M$. □

4.2 Anon-MID-CPA Security

In the security proof of Anon-MID-CPA, we prove that if there is an adversary who can break the Anon-MID-CPA security of the above IBBE scheme in the standard model, then there is an algorithm which can break the $(\ell+1)$-MDDH assumption in $\mathbf{MG}_{\ell+1}(1^k)$. Formally, we have the following theorem.

THEOREM 5. *Assume the above IBBE scheme is implemented in an* $(\ell+1)$*-group system, and with a* (poly, N)*-MPHF* $\mathbf{H}$ *into* $\mathbb{G}_\ell$*. Then, under the* $(\ell+1)$*-MDDH assumption, the IBBE scheme is Anon-MID-CPA secure.*

PROOF. Suppose a PPT adversary $\mathcal{A}$ wins in the Anon-MID-CPA game of the above IBBE scheme with advantage $Adv_{IBBE,\mathcal{A}}^{\text{Anon-MID-CPA}}$, in which $\mathcal{A}$ makes at most q_p queries to oracle $\mathcal{Q}_{SK}^{IBBE}(\cdot)$. To prove this theorem, we will construct a PPT algorithm $\mathcal{B}$ to play the Anon-MID-CPA game with adversary $\mathcal{A}$ and take advantage of $\mathcal{A}$ to break the $(\ell+1)$-MDDH assumption in $\mathbf{MG}_{\ell+1}(1^k)$. The constructed algorithm $\mathcal{B}$ is as follows.

- **Setup Phase**: Algorithm $\mathcal{B}$ gets as input an $(\ell+1)$-group system $\mathbf{MPG}_{\ell+1}$ with security parameter 1^k and group elements g, g^{x_1}, $\cdots$, $g^{x_{\ell+1}}$, $g^{x_{\ell+2}} \in \mathbb{G}_1$ and $S \in \mathbb{G}_{\ell+1}$, where either $S = \hat{\mathbf{e}}(g^{x_1}, \cdots, g^{x_{\ell+1}})^{x_{\ell+2}}$ (i.e., S is real) or $S \in \mathbb{G}_{\ell+1}$ uniformly (i.e., S is random). $\mathcal{B}$ generates a (q_p, N)-MPHF $\mathbf{H}$ into $\mathbb{G}_\ell$, sets up the master public key as $\mathcal{MP} = (\mathbf{MPG}_{\ell+1}, g, p = g^{x_{\ell+1}}, \mathbf{H}, hk, \mathcal{ID})$ where $(hk, td) \leftarrow \mathbf{TGen}(1^k, g^{x_1}, \cdots, g^{x_\ell}, g)$, and sends $\mathcal{MP}$ to adversary $\mathcal{A}$. Here, we use the **TGen** and **TEval** algorithms of the (q_p, N)-MPHF property of $\mathbf{H}$.
- **Query Phase 1**: Adversary $\mathcal{A}$ adaptively issues the following query multiple times.
  - Decryption Key Query $\mathcal{Q}_{SK}(ID)$: Taking as input an identity $ID \in \mathcal{ID}$, algorithm $\mathcal{B}$ does the following steps:
    1. Compute $\mathbf{TEval}(td, ID) = (a_{ID}, B_{ID}) \in \mathbb{Z}_q^* \times \mathbb{G}_{\ell-1}$;
    2. If $a_{ID} = 0$, return private key $SK_{ID} = \hat{\mathbf{e}}(B_{ID}, p)$; otherwise, abort and output $\perp$;

    Note that we have $SK_{ID} = \hat{\mathbf{e}}(B_{ID}, p) = \hat{\mathbf{e}}(B_{ID}, g)^{x_{\ell+1}} = \mathbf{H}_{hk}(ID)^{x_{\ell+1}}$. So $\mathcal{B}$ can answer a $\mathcal{Q}_{SK}(ID)$ query of $\mathcal{A}$ for identity ID precisely when $a_{ID} = 0$.

- **Challenge Phase**: Adversary $\mathcal{A}$ sends two different challenge identity sets $\mathcal{I}_0^* = \{ID_1^{*0}, \cdots, ID_N^{*0}\}$ and $\mathcal{I}_1^* = \{ID_1^{*1}, \cdots, ID_N^{*1}\}$ and a plaintexts $M \in \mathbb{G}_{\ell+1}$ to algorithm $\mathcal{B}$; $\mathcal{B}$ picks $d \xleftarrow{\$} \{0,1\}$, and does the following steps:

  1. For $ID_i^{*d} \in \mathcal{I}_0^* \bigcap \mathcal{I}_1^*$, compute $\mathbf{TEval}(td, ID_i^{*d}) = (\, a_{ID_i^{*d}}, B_{ID_i^{*d}})$, and if $a_{ID_i^{*d}} \neq 0$, abort and output $\perp$;
  2. For $ID_i^{*d} \in \mathcal{I}_d^* - \mathcal{I}_0^* \bigcap \mathcal{I}_1^*$, compute $\mathbf{TEval}(td, ID_i^{*d}) = (a_{ID_i^{*d}}, B_{ID_i^{*d}})$, and if $a_{ID_i^{*d}} = 0$, abort and output $\perp$;
  3. Choose a value $r \xleftarrow{\$} \mathbb{Z}_q^*$, initialize an empty list $\mathcal{T}$, and compute $C_0 = g^{x_{\ell+2}}$ and $C_0' = g^{r \cdot x_{\ell+2}}$;
  4. For $ID_i^{*d} \in \mathcal{I}_d^*$, compute $I_i = S^{a_{ID_i^{*d}}} \cdot \hat{\mathbf{e}}(B_{ID_i^{*d}}, g^{x_{\ell+1}}, g^{x_{\ell+2}})$ and $C_i = I_i^r \cdot M$, and add the label-data pair (I_i, C_i) into $\mathcal{T}$;
  5. Generate dictionary $\mathcal{L} \leftarrow \mathbf{Creat}(\mathcal{T})$, and send a challenge ciphertext $\mathcal{C}_d^* = (C_0, C_0', \mathcal{L})$ to adversary $\mathcal{A}$.

  Suppose algorithm $\mathcal{B}$ does not abort in this phase (i.e., $a_{ID_i^{*d}} = 0$ for all $a_{ID_i^{*d}} \in \mathcal{I}_0^* \bigcap \mathcal{I}_1^*$, and $a_{ID_i^{*d}} \neq 0$ for all $a_{ID_i^{*d}} \in \mathcal{I}_d^* - \mathcal{I}_0^* \bigcap \mathcal{I}_1^*$). We have $\mathbf{H}_{hk}(ID_i^{*d}) = \hat{\mathbf{e}}(g^{x_1}, \cdots, g^{x_\ell})^{a_{ID_i^{*d}}} \cdot \hat{\mathbf{e}}(B_{ID_i^{*d}}, g)$. If $S = \hat{\mathbf{e}}(g^{x_1}, \cdots, g^{x_{\ell+1}})^{x_{\ell+2}}$, we have $I_i = S^{a_{ID_i^{*d}}} \cdot \hat{\mathbf{e}}(B_{ID_i^{*d}}, g^{x_{\ell+1}}, g^{x_{\ell+2}}) = \hat{\mathbf{e}}(\mathbf{H}_{hk}(ID_i^{*d}), g^{x_{\ell+1}})^{x_{\ell+2}}$, which implies that the generated (I_i, C_i) is valid in this case. Otherwise, the generated (I_i, C_i) is uniformly distributed in their ranges and independent with any identity in sets $\mathcal{I}_0^*$ and $\mathcal{I}_1^*$.

- **Query Phase 2**: This phase is the same as **Query Phase 2**. Note that both in **Query Phase 1** and **Query Phase 2**, adversary $\mathcal{A}$ cannot query the decryption keys of the challenge identities in set $\mathcal{I}_0^* \bigcup \mathcal{I}_1^* - \mathcal{I}_0^* \bigcap \mathcal{I}_1^*$.

- **Guess Phase**: Adversary $\mathcal{A}$ sends a guess $\hat{d}'$ to algorithm $\mathcal{B}$. Let $\overline{Abort'}$ denote the event that $\mathcal{B}$ does not abort in all previous phases. Let $\{ID_1, \cdots, ID_{q_p}\}$ be the set of all queried IDs by $\mathcal{A}$ both in **Query Phase 1** and **Query Phase 2**. Let $\mathcal{I} = \{ID_1, \cdots, ID_{q_p}\} \bigcup (\mathcal{I}_0^* \bigcup \mathcal{I}_1^* - \mathcal{I}_0^* \bigcap \mathcal{I}_1^*)$, where $|\mathcal{I}_0^* \bigcup \mathcal{I}_1^* - \mathcal{I}_0^* \bigcap \mathcal{I}_1^*| \leq N$. Let $P_{\mathcal{I}} = Pr[\overline{Abort'}|\mathcal{I}]$, which will be decided later. As in [16, 20], $\mathcal{B}$ "artificially" aborts with probability $1 - 1/(P_{\mathcal{I}} \cdot p(k))$ for the polynomial $p(k)$ from Definition 8 and outputs $\perp$. If it does not abort, $\mathcal{B}$ uses the guess of $\mathcal{A}$. This means that if $d = d'$, $\mathcal{B}$ outputs 1, otherwise it outputs 0.

In **Guess Phase**, $\mathcal{B}$ did not directly use the guess of $\mathcal{A}$, since event $\overline{Abort'}$ might not be independent of the identities in $\mathcal{I}$. So $\mathcal{B}$ "artificially" aborts to achieve the independence. Let $\overline{Abort}$ be the event that $\mathcal{B}$ does not abort in the above game. We have that $Pr[\overline{Abort}] = 1 - Pr[Abort'|\mathcal{I}] - Pr[\overline{Abort'}|\mathcal{I}] \cdot (1 - 1/(P_{\mathcal{I}} \cdot p(k))) = 1/p(k)$. Hence, we have $Pr[\mathcal{B} = 1|S \text{ is real}] = Pr[\overline{Abort}] \cdot (\frac{1}{2} + Adv_{IBBE,\mathcal{A}}^{\text{Anon-MID-CPA}})$ and $Pr[\mathcal{B} = 1|S \text{ is random}] = Pr[\overline{Abort}] \cdot \frac{1}{2}$, where $\frac{1}{2} + Adv_{IBBE,\mathcal{A}}^{\text{Anon-MID-CPA}}$ is the probability that $\mathcal{A}$ succeeds in the Anon-MID-CPA game of IBBE. Further, we have $Pr[\mathcal{B} = 1|S \text{ is real}] - Pr[\mathcal{B} = 1|S \text{ is random}] = \frac{1}{p(k)} \cdot Adv_{IBBE,\mathcal{A}}^{\text{Anon-MID-CPA}}$. Hence, $\mathcal{B}$ breaks the $(\ell+1)$-MDDH assumption if $\mathcal{A}$ breaks the Anon-MID-CPA security of the proposed IBBE scheme.

Finally, to evaluate $P_{\mathcal{I}}$, we can only approximate it (up to an inversely polynomial error, by running **TEval** with freshly generated keys sufficiently often), which introduces an additional error term in the analysis. We refer to [20] for details on this evaluation. □

4.3 SS-MID-CPA Security

In the security proof of SS-MID-CPA, we prove that if there is an adversary who can break the SS-MID-CPA security of the above IBBE scheme in the standard model, then there is an algorithm which can break the $(\ell+1)$-MDDH assumption in $\mathbf{MG}_{\ell+1}(1^k)$. Formally, we have the following theorem.

THEOREM 6. *Assume the above IBBE scheme is implemented in an $(\ell+1)$-group system, and with a (poly, N)-MPHF $\mathbf{H}$ into $\mathbb{G}_\ell$. Then, under the $(\ell+1)$-MDDH assumption, the IBBE scheme is SS-MID-CPA secure.*

PROOF. Suppose a PPT adversary $\mathcal{A}$ wins in the SS-MID-CPA game of the above IBBE scheme with advantage $Adv_{IBBE,\mathcal{A}}^{\text{SS-MID-CPA}}$, in which $\mathcal{A}$ makes at most q_p queries to oracle $\mathcal{Q}_{SK}^{IBBE}(\cdot)$. To prove this theorem, we will construct a PPT algorithm $\mathcal{B}$ to play the SS-MID-CPA game with adversary $\mathcal{A}$ and take advantage of $\mathcal{A}$ to break the $(\ell+1)$-MDDH assumption in $\mathbf{MG}_{\ell+1}(1^k)$. The constructed algorithm $\mathcal{B}$ is as follows.

- **Setup Phase**: Algorithm $\mathcal{B}$ gets as input an $(\ell+1)$-group system $\mathbf{MPG}_{\ell+1}$ with security parameter 1^k and group elements g, g^{x_1}, $\cdots$, $g^{x_{\ell+1}}$, $g^{x_{\ell+2}} \in \mathbb{G}_1$ and $S \in \mathbb{G}_{\ell+1}$, where either $S = \hat{\mathbf{e}}(g^{x_1}, \cdots, g^{x_{\ell+1}})^{x_{\ell+2}}$ (i.e., S is real) or $S \in \mathbb{G}_{\ell+1}$ uniformly (i.e., S is random). $\mathcal{B}$ generates a (q_p, N)-MPHF $\mathbf{H}$ into $\mathbb{G}_\ell$, sets up the master public key as $\mathcal{MP} = (\mathbf{MPG}_{\ell+1}, g, p = g^{x_{\ell+1}}, \mathbf{H}, hk, \mathcal{ID})$ where $(hk, td) \leftarrow \mathbf{TGen}(1^k, g^{x_1}, \cdots, g^{x_\ell}, g)$, and sends $\mathcal{MP}$ to adversary $\mathcal{A}$. Here, we use the **TGen** and **TEval** algorithms of the (q_p, N)-MPHF property of $\mathbf{H}$.

- **Query Phase 1**: Adversary $\mathcal{A}$ adaptively issues the following query multiple times.

  – Decryption Key Query $\mathcal{Q}_{SK}(ID)$: Taking as input an identity $ID \in \mathcal{ID}$, algorithm $\mathcal{B}$ does the following steps:

    1. Compute $\mathbf{TEval}(td, ID) = (a_{ID}, B_{ID}) \in \mathbb{Z}_q^* \times \mathbb{G}_{\ell-1}$;
    2. If $a_{ID} = 0$, return private key $SK_{ID} = \hat{\mathbf{e}}(B_{ID}, p)$; otherwise, abort and output $\perp$;

    Note that we have $SK_{ID} = \hat{\mathbf{e}}(B_{ID}, p) = \hat{\mathbf{e}}(B_{ID}, g)^{x_{\ell+1}} = \mathbf{H}_{hk}(ID)^{x_{\ell+1}}$. So $\mathcal{B}$ can answer a $\mathcal{Q}_{SK}(ID)$ query of $\mathcal{A}$ for identity ID precisely when $a_{ID} = 0$.

- **Challenge Phase**: Adversary $\mathcal{A}$ sends a challenge identity set $\mathcal{I}^* = \{ID_1^*, \cdots, ID_N^*\}$ and two challenge plaintexts $M_0^* \in \mathbb{G}_{\ell+1}$ and $M_1^* \in \mathbb{G}_{\ell+1}$ to algorithm $\mathcal{B}$; $\mathcal{B}$ picks $d \xleftarrow{\$} \{0,1\}$, and does the following steps:

1. For $ID_i^* \in \mathcal{I}^*$, compute $\mathbf{TEval}(td, ID_i^*) = (a_{ID_i^*}, B_{ID_i^*})$, and if $a_{ID_i^*} = 0$, abort and output $\perp$;
2. Choose a values $r \overset{\$}{\leftarrow} \mathbb{Z}_q^*$, initialize an empty list $\mathcal{T}$, and compute $C_0 = g^{x_{\ell+2}}$ and $C_0' = g^{r \cdot x_{\ell+2}}$;
3. For $ID_i^* \in \mathcal{I}^*$, compute $I_i = S^{a_{ID_i^*}} \cdot \hat{\mathbf{e}}(B_{ID_i^*}, g^{x_{\ell+1}}, g^{x_{\ell+2}})$ and $C_i = I_i^r \cdot M_d^*$, and add the label-data pair (I_i, C_i) into $\mathcal{T}$;
4. Generating dictionary $\mathcal{L} \leftarrow \mathbf{Creat}(\mathcal{T})$, and send a challenge ciphertext $\mathcal{C}_d^* = (C_0, C_0', \mathcal{L})$ to adversary $\mathcal{A}$.

Suppose algorithm $\mathcal{B}$ does not abort in this phase (i.e., $a_{ID_i^*} \neq 0$ for all $a_{ID_i^*} \in \mathcal{I}^*$). We have $\mathbf{H}_{hk}(ID_i^*) = \hat{\mathbf{e}}(g^{x_1}, \cdots, g^{x_\ell})^{a_{ID_i^*}} \cdot \hat{\mathbf{e}}(B_{ID_i^*}, g)$. If $S = \hat{\mathbf{e}}(g^{x_1}, \cdots, g^{x_{\ell+1}})^{x_{\ell+2}}$, we have $I_i = S^{a_{ID_i^*}} \cdot \hat{\mathbf{e}}(B_{ID_i^*}, g^{x_{\ell+1}}, g^{x_{\ell+2}}) = \hat{\mathbf{e}}(\mathbf{H}_{hk}(ID_i^*), g^{x_{\ell+1}})^{x_{\ell+2}}$, which implies that the generated (I_i, C_i) is valid in this case. Otherwise, the generated (I_i, C_i) is uniformly distributed in their ranges.

- **Query Phase 2**: This phase is the same as **Query Phase 2**. Note that both in **Query Phase 1** and **Query Phase 2**, adversary $\mathcal{A}$ cannot query the decryption keys of the challenge identities in $\mathcal{I}^*$.
- **Guess Phase**: Adversary $\mathcal{A}$ sends a guess $\hat{d'}$ to algorithm $\mathcal{B}$. Let $\overline{Abort'}$ denote the event that $\mathcal{B}$ does not abort in all previous phases. Let $\mathcal{I} = \{ID_1, \cdots, ID_{q_p}, ID_1^*, \cdots, ID_N^*\}$ be the set of all queried IDs by $\mathcal{A}$ both in **Query Phase 1** and **Query Phase 2** and all challenge identities. Let $P_{\mathcal{I}} = Pr[\overline{Abort'}|\mathcal{I}]$, which will be decided later. As in [16,20], $\mathcal{B}$ "artificially" aborts with probability $1 - 1/(P_{\mathcal{I}} \cdot p(k))$ for the polynomial $p(k)$ from Definition 8 and outputs $\perp$. If it does not abort, $\mathcal{B}$ uses the guess of $\mathcal{A}$. This means that if $d = d'$, $\mathcal{B}$ outputs 1, otherwise it outputs 0.

In **Guess Phase**, $\mathcal{B}$ did not directly use the guess of $\mathcal{A}$, since event $\overline{Abort'}$ might not be independent of the identities in $\mathcal{I}$. So $\mathcal{B}$ "artificially" aborts to achieve the independence. Let $\overline{Abort}$ be the event that $\mathcal{B}$ does not abort in the above game. We have that $Pr[\overline{Abort}] = 1 - Pr[Abort'|\mathcal{I}] - Pr[\overline{Abort'}|\mathcal{I}] \cdot (1 - 1/(P_{\mathcal{I}} \cdot p(k))) = 1/p(k)$. Hence, we have $Pr[\mathcal{B} = 1|S \text{ is real}] = Pr[\overline{Abort}] \cdot (\frac{1}{2} + Adv_{IBBE,\mathcal{A}}^{\text{SS-MID-CPA}})$ and $Pr[\mathcal{B} = 1|S \text{ is random}] = Pr[\overline{Abort}] \cdot \frac{1}{2}$, where $\frac{1}{2} + Adv_{IBBE,\mathcal{A}}^{\text{SS-MID-CPA}}$ is the probability that $\mathcal{A}$ succeeds in the SS-MID-CPA game of IBBE. Further, we have $Pr[\mathcal{B} = 1|S \text{ is real}] - Pr[\mathcal{B} = 1|S \text{ is random}] = \frac{1}{p(k)} \cdot Adv_{IBBE,\mathcal{A}}^{\text{SS-MID-CPA}}$. Hence, $\mathcal{B}$ breaks the $(\ell + 1)$-MDDH assumption if $\mathcal{A}$ breaks the SS-MID-CPA security of the proposed IBBE scheme.

Finally, to evaluate $P_{\mathcal{I}}$, we can only approximate it (up to an inversely polynomial error, by running **TEval** with freshly generated keys sufficiently often), which introduces an additional error term in the analysis. We refer to [20] for details on this evaluation. □

5. RELATED WORKS

AIBBE has been extensively investigated in recent years. Several AIBBE schemes were proposed. Table 2 compares the existing AIBBE schemes and our proposed AIBBE schemes. More details are as follows. In 2010, Fan et al. [1] proposed the first AIBBE scheme (called FHH'10 in our paper). However, Wang et al. [2] and Chien [6] respectively demonstrated that FHH'10 scheme fails to provide anonymity under the inside attack. In other words, an intended receiver of a ciphertext generated by FHH'10 scheme can extract the identities of the other intended receivers. Hence, Wang et al. [2] proposed an AIBBE scheme (called WZXQ'12 in our paper) to improve FHH'10 scheme. Unfortunately, Zhang et al. [3] presented that WZXQ'12 scheme is still unable to provide anonymity, and proposed a new AIBBE scheme. Tseng et al. [4] proposed an AIBBE scheme with the provable security under chosen ciphertext attacks. Ren et al. [5] proposed an AIBBE scheme based on asymmetric bilinear groups, which is secure under the adaptive attack in the standard model.

In 2012, Chien [6] also proposed a randomness-reuse-based AIBBE scheme (called C'12 in our paper) to improve FHH'10 scheme. But Wang [21] pointed out that C'12 scheme does not satisfy the indistinguishability of encryptions under the non-adaptive and chosen ciphertext attacks. Hur et al. [7] proposed an AIBBE scheme to reduce decryption cost, but they did not give formal security proof. To protect anonymity and reduce decryption cost, Cui et al. [8] introduced a new concept called server-aided identity-based anonymous broadcast encryption scheme. In 2013, Zhang et al. [10] proposed an AIBBE scheme (called ZT'13 in our paper) with the constant decryption complexity. But, Zhang et al. [9] pointed out that ZT'13 scheme did not achieve anonymity, and proposed an improved AIBBE scheme. Tseng et al. [11] proposed an AIBBE scheme with the constant decryption complexity and the anonymity under the non-adaptive attacks in the RO model.

In addition, Zhang et al. [22] constructed an AIBBE scheme with the provable security under the adaptive attack in the standard model. But this scheme has the number of public parameters linear with the total number of users. Hence, it fails to obtain the advantage of identity-based cryptography.

6. CONCLUSION

This paper proposes two AIBBE schemes in the RO and standard models respectively. Both schemes employ the ideas of dual randomness-reuse and history-independent dictionary to achieve the constant decryption complexity and the strong anonymity. Hence, they allow a receiver to decrypt a ciphertext with the complexity independent with the number of receivers of the ciphertext, keep receivers' anonymity of a ciphertext even under the inside attack. In addition, the proposed AIBBE scheme in the standard model is secure under the adaptive attack. So far as we known, this scheme is the first one to achieve the constant decryption complexity, and the strongest security compared with previous works.

7. ACKNOWLEDGMENTS

The authors would like to thank the reviewers for their valuable suggestions that help to improve the paper greatly. The first author is partly supported by the National Natural Science Foundation of China under grant no. 61472156 and the National Program on Key Basic Research Project (973 Program) under grant no. 2014CB340600.

Table 2: Comparisions in the Aspects of Security and Decryption Complexity

Scheme	Type	Adaptive	Model	Strong Anonymity	Decryption Complexity
FHH'10 [1]	Lagrange-Polynomial	sMID	RO	Achieved	$O(n)$
WZXQ'12 [2]		sMID	RO	Achieved	
ZX'12 [3]		No Proof			
THC'14 [4]		sMID	RO	Achieved	
RNZ'14 [5]		MID	Standard	Achieved	
C'12 [6]	Randomness-Reuse	No Proof			
HPH'12 [7]		sMID	RO	Failed	
CMG'13 [8]		MID	RO	Failed	
ZM'15 [9]		sMID	RO	Failed	
ZT'13 [10]		sMID	RO	Failed	$O(1)$
TTHC'14 [11]		sMID	RO	Failed	
Our First AIBBE		sMID	RO	Achieved	
Our Second AIBBE		MID	Standard	Achieved	

8. REFERENCES

[1] C. I. Fan, L. Y. Huang, and P. H. Ho. Anonymous Multireceiver Identity-Based Encryption. *IEEE Transactions on Computers,* 59(9):1239-1249, 2010.

[2] H. Wang, Y. C. Zhang, H. Xiong, and B. Qin. Cryptanalysis and improvements of an anonymous multi-receiver identity-based encryption scheme. *IET Information Security,* 6(1):20-27, 2012.

[3] J. Zhang, and Y. Xu. Comment on Anonymous Multi-receiver Identity-Based Encryption Scheme. In Proceedings of *2012 Fourth International Conference on Intelligent Networking and Collaborative Systems, INCoS 2012*, pages 473-476. IEEE, 2012.

[4] Y. M. Tseng, Y. H. Huang, and H. J. Chang. Privacy-preserving multireceiver ID-based encryption with provable security. *International Journal of Communication Systems,* 27(7):1034-1050, 2014.

[5] Y. Ren, Z. Niu, and X. Zhang. Fully Anonymous Identity-based Broadcast Encryption without Random Oracles. *International Journal of Network Security,* 16(4):256-264, 2014.

[6] H. Y. Chien. Improved Anonymous Multi-receiver Identity-Based Encryption. *The Computer Journal,* 55(4):439-446, 2012.

[7] J. Hur, C. Park, and S. Hwang. Privacy-preserving identity-based broadcast encryption. *Information Fusion,* 13(4):296-303, 2012.

[8] H. Cui, Y. Mu, and F. Guo. Server-aided identity-based anonymous broadcast encryption. *International Journal of Security and Networks,* 8(1):29-39, 2013.

[9] J. Zhang, and J. Mao. An improved anonymous multi-receiver identity-based encryption scheme. *International Journal of Communication Systems,* 28(4):645-658, 2015.

[10] M. Zhang, and T. Takagi. Efficient Constructions of Anonymous Multireceiver Encryption Protocol and Their Deployment in Group E-mail Systems With Privacy Preservation. *IEEE Systems Journal,* 7(3):410-419, 2013.

[11] Y. M. Tseng, T. T. Tsai, S. S. Huang, and H. Y. Chien. Efficient anonymous multi-receiver ID-based encryption with constant decryption cost. In *Proceedings of 2014 International Conference on Information Science, Electronics and Electrical Engineering (ISEEE),* pages 131-137. IEEE, 2014.

[12] C. Rackoff, and D. R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Advances in Cryptology - CRYPTO '91,* LNCS, volume 576, pages 433-444. Springer, 1992.

[13] E. Fujisaki, and T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. In *Advances in Cryptology - CRYPTO '99,* LNCS, volume 1666, pages 537-554. Springer, 1999.

[14] E. Fujisaki, and T. Okamoto. How to Enhance the Security of Public-Key Encryption at Minimum Cost. In *Proceedings of Public Key Cryptography 1999*, LNCS, volume 1560, pages 53-68. Springer, 1999.

[15] R. Canetti, S. Halevi, and J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. In *Advances in Cryptology - EUROCRYPT 2004,* LNCS, volume 3027, pages 207-222. Springer, 2004.

[16] E. S. V. Freire, D. Hofheinz, K. G. Paterson, and C. Striecks. Programmable Hash Functions in the Multilinear Setting. In *Advances in Cryptology - CRYPTO 2013,* LNCS, volume 8042, pages 513-530. Springer, 2013.

[17] D. Cash, J. Jaeger, S. Jarecki, C. S. Jutla, H. Krawczyk, M. C. Rosu, and M. Steiner. Dynamic Searchable Encryption in Very-Large Databases: Data Structures and Implementation. In *Proceedings of 21st Annual Network and Distributed System Security Symposium, NDSS 2014,* 2014.

[18] D. Boneh, and X. Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In *Advances in Cryptology - EUROCRYPT 2004,* LNCS, volume 3027, pages 223-238. Springer, 2004.

[19] S. Garg, C. Gentry, and S. Halevi. Candidate Multilinear Maps from Ideal Lattices. In *Advances in Cryptology - EUROCRYPT 2013*, LNCS, volume 7881, pages 1-17. Springer, 2013.

[20] B. Waters. Efficient Identity-Based Encryption Without Random Oracles. In *Advances in Cryptology - EUROCRYPT 2005,* LNCS, volume 3494, pages 114-127. Springer, 2005.

[21] H. Wang. Insecurity of "Improved Anonymous Multi-Receiver Identity-Based Encryption". *The Computer Journal,* 57(4):636-638, 2014.

[22] L. Zhang, Q. Wu, and Y. Mu. Anonymous

Identity-Based Broadcast Encryption with Adaptive Security. In *Proceedings of 5th International Symposium on Cyberspace Safety and Security, CSS 2013,* LNCS, volume 8300, pages 258-271. Springer, 2013.

APPENDIX

A. THE PROOF OF THEOREM 1

PROOF. Without loss of generality, let $\mathcal{I} = \{ID_1, \cdots, ID_N\}$, $ID' = ID_i$, and (I_i, C_i) be the label-data pair generated by algorithm $\mathcal{C} \leftarrow \mathbf{Enc}(\mathcal{MP}, \mathcal{I}, M)$, where $i \in [1, N]$, $I_i = \hat{\mathbf{e}}(p, \mathbf{H}(ID_i))^{r_1}$ and $C_i = \hat{\mathbf{e}}(p, \mathbf{H}(ID_i))^{r_2} \cdot M$. Parse $\mathcal{C} = (C_0, C_0', \mathcal{L})$. Algorithm $\mathbf{Dec}(SK_{ID'}, \mathcal{C})$ shows that $I' = \hat{\mathbf{e}}(C_0, SK_{ID'}) = \hat{\mathbf{e}}(g^{r_1}, \mathbf{H}(ID')^s) = \hat{\mathbf{e}}(g^s, \mathbf{H}(ID'))^{r_1} = \hat{\mathbf{e}}(p, \mathbf{H}(ID'))^{r_1} = I_i$ according to the definition of bilinear map. Hence, we have $C' = C_i$, except with a negligible probability in the security parameter k. According to equation $C' \cdot \hat{\mathbf{e}}(C_0', SK_{ID'})^{-1} = \hat{\mathbf{e}}(p, \mathbf{H}(ID_i))^{r_2} \cdot M \cdot \hat{\mathbf{e}}(C_0', SK_{ID'})^{-1} = M$, we can prove that $\mathbf{Dec}(SK_{ID'}, \mathcal{C}) = M$. □

B. THE PROOF OF THEOREM 2

PROOF. To prove this theorem, we will construct a PPT algorithm $\mathcal{B}$ that plays the Anon-sMID-CPA game with adversary $\mathcal{A}$ and utilizes the capability of $\mathcal{A}$ to solve the DBDH problem in $\mathbf{BGen}(1^k)$ with advantage $Adv_{IBBE,\mathcal{A}}^{\text{Anon-sMID-CPA}}$. The constructed algorithm $\mathcal{B}$ in the Anon-sMID-CPA game is as follows.

- **Setup Phase**: Adversary $\mathcal{A}$ sends two challenge identity sets $(\mathcal{I}_0^*, \mathcal{I}_1^*)$ to the challenger, where $\mathcal{I}_0^* = \{ID_1^{*0}, ..., ID_N^{*0}\}$ and $\mathcal{I}_1^* = \{ID_1^{*1}, ..., ID_N^{*1}\}$. Algorithm $\mathcal{B}$ takes as inputs $(q, \mathbb{G}, \mathbb{G}_1, g, \hat{\mathbf{e}}, g^a, g^b, g^c, Z)$ (where Z equals either $\hat{\mathbf{e}}(g, g)^{abc}$ or $\hat{\mathbf{e}}(g, g)^y$) and identity space $\mathcal{ID} = \{0, 1\}^*$, initializes a list $\mathbf{HList} = \emptyset \subseteq \mathcal{ID} \times \mathbb{G} \times \mathbb{Z}_q^*$, and sends the master public key $\mathcal{MP} = (q, \mathbb{G}, \mathbb{G}_1, g, \hat{\mathbf{e}}, p = g^a, \mathcal{ID})$ to adversary $\mathcal{A}$;
- **Query Phase 1**: Adversary $\mathcal{A}$ adaptively issues the following queries multiple times. To simplify the description of this phase, we suppose that $\mathcal{A}$ never issues the same queries both in the following two kinds of queries.
  - Hash Query $\mathcal{Q}_H(ID)$: Given a queried identity $ID \in \mathcal{ID}$, algorithm $\mathcal{B}$ does the following steps:
    1. Pick $x \xleftarrow{\$} \mathbb{Z}_q^*$;
    2. If the queried identity $ID \in \mathcal{I}_0^* \bigcup \mathcal{I}_1^*$ and $ID \notin \mathcal{I}_{\prime}^* \bigcap \mathcal{I}^*$, add tuple $(ID, g^{c \cdot x}, x)$ into $\mathbf{HList}$, and return $g^{c \cdot x}$ to $\mathcal{A}$;
    3. Otherwise add tuple (ID, g^x, x) into $\mathbf{HList}$, and return g^x to $\mathcal{A}$;
  - Decryption Key Query $\mathcal{Q}_{SK}(ID)$: Given a queried identity $ID \in \mathcal{ID}$, algorithm $\mathcal{B}$ does the following steps:
    1. If $(ID, *, *) \notin \mathbf{HList}$, query $\mathcal{Q}_H(ID)$;
    2. According to ID, retrieve (ID, X, x) from $\mathbf{HList}$;
    3. Return private key $g^{a \cdot x}$ to $\mathcal{A}$;
- **Challenge Phase**: Adversary $\mathcal{A}$ sends a plaintext M to algorithm $\mathcal{B}$. $\mathcal{B}$ picks $d \xleftarrow{\$} \{0, 1\}$, and does the following steps:
  1. Choose value $r \xleftarrow{\$} \mathbb{Z}_q^*$, initialize an empty array $\mathcal{L}$, and compute $C_0 = g^b$ and $C_0' = g^{b \cdot r}$;
  2. For $i \in [1, N]$, retrieve tuple (ID_i^{*d}, X, x) from $\mathbf{HList}$ according to identity ID_i^{*d}, compute $I_i = Z^x$ and $C_i = Z^{x \cdot r} \cdot M$ if $ID_i^d \notin \mathcal{I}_0^* \bigcap \mathcal{I}_1^*$, otherwise compute $I_i = \hat{\mathbf{e}}(g^a, g^b)^x$ and $C_i = \hat{\mathbf{e}}(g^a, g^b)^{x \cdot r} \cdot M$, and finally set $\mathcal{L}[I_i] = C_i$;
  3. Return challenge ciphertext $\mathcal{C}_d^* = (C_0, C_0', \mathcal{L})$ to $\mathcal{A}$;
- **Query Phase 2**: This phase is the same as **Query Phase 1**. Note that both in **Query Phase 1** and **Query Phase 2**, $\mathcal{A}$ can not query the private keys corresponding to the challenge identities in $\mathcal{I}_0^*$ and $\mathcal{I}_1^*$ except the challenge identities in $\mathcal{I}_0^* \cap \mathcal{I}_1^*$;
- **Guess Phase**: Adversary $\mathcal{A}$ sends a guess d' to algorithm $\mathcal{B}$. If $d = d'$, $\mathcal{B}$ outputs 1; otherwise, outputs 0.

Next, we will prove that the above game is computationally indistinguishable with a real one in the view of adversary $\mathcal{A}$ when $Z = \hat{\mathbf{e}}(g, g)^{abc}$, and adversary $\mathcal{A}$ has the negligible advantage to win in the above game when $Z = \hat{\mathbf{e}}(g, g)^y$. These proofs will finally demonstrate this theorem.

In the above game, it is easy to verify that all generated private keys are valid. When $Z = \hat{\mathbf{e}}(g, g)^{abc}$, the generated challenge ciphertext is a real one. Hence, in the view of adversary $\mathcal{A}$, the above game in this case is computationally indistinguishable with a real one. When $Z = \hat{\mathbf{e}}(g, g)^y$, the generated challenge ciphertext is only dependent with the challenge identities in $\mathcal{I}_0^* \bigcap \mathcal{I}_1^*$. But it is independent with the location of these challenge identities respectively in sets $\mathcal{I}_0^*$ and $\mathcal{I}_1^*$, since the array $\mathcal{L}$ in the challenge ciphertext is history-independent. Hence, adversary $\mathcal{A}$ has the negligible advantage to win in the above game when $Z = \hat{\mathbf{e}}(g, g)^y$. Summarily, we have the following equation.

$$\begin{aligned}
& Adv_{\mathcal{B}}^{DBDH}(1^k) \\
& = Pr[\mathcal{B} = 1 | Z = \hat{\mathbf{e}}(g, g)^{abc}] - Pr[\mathcal{B} = 1 | Z = \hat{\mathbf{e}}(g, g)^y] \\
& = Pr[d = d' | Z = \hat{\mathbf{e}}(g, g)^{abc}] - Pr[d = d' | Z = \hat{\mathbf{e}}(g, g)^y] \\
& = Adv_{IBBE,\mathcal{A}}^{\text{Anon-sMID-CPA}} + \frac{1}{2} - \frac{1}{2} = Adv_{IBBE,\mathcal{A}}^{\text{Anon-sMID-CPA}}
\end{aligned}$$

In addition, it is clear that algorithm $\mathcal{B}$ is a PPT algorithm, if adversary $\mathcal{A}$ is a PPT adversary. In conclusion, if a PPT adversary $\mathcal{A}$ wins in the Anon-sMID-CPA game of the above IBBE scheme with advantage $Adv_{IBBE,\mathcal{A}}^{\text{Anon-sMID-CPA}}$, there is a PPT algorithm $\mathcal{B}$ that solves the DBDH problem in $\mathbf{BGen}(1^k)$ with $Adv_{\mathcal{B}}^{DBDH}(1^k) = Adv_{IBBE,\mathcal{A}}^{\text{Anon-sMID-CPA}}$. □

C. THE PROOF OF THEOREM 3

PROOF. To prove this theorem, we will construct a PPT algorithm $\mathcal{B}$ that plays the SS-sMID-CPA game with adversary $\mathcal{A}$ and utilizes the capability of $\mathcal{A}$ to solve the DBDH problem in $\mathbf{BGen}(1^k)$ with advantage $Adv_{IBBE,\mathcal{A}}^{\text{SS-sMID-CPA}}$. The constructed algorithm $\mathcal{B}$ in the SS-sMID-CPA game is as follows.

- **Setup Phase**: Adversary $\mathcal{A}$ sends an identity set $\mathcal{I} = \{ID_1, ..., ID_N\}$ to algorithm $\mathcal{B}$. Then $\mathcal{B}$ takes as inputs $(q, \mathbb{G}, \mathbb{G}_1, g, \hat{\mathbf{e}}, g^a, g^b, g^c, Z)$ (where Z equals either $\hat{\mathbf{e}}(g, g)^{abc}$ or $\hat{\mathbf{e}}(g, g)^y$) and identity space $\mathcal{ID} = \{0, 1\}^*$, initializes a list $\mathbf{HList} = \emptyset \subseteq \mathcal{ID} \times \mathbb{G} \times \mathbb{Z}_q^*$, and sends the master public key $\mathcal{MP} = (q, \mathbb{G}, \mathbb{G}_1, g, \hat{\mathbf{e}}, p = g^a, \mathcal{ID})$ to adversary $\mathcal{A}$;

- **Query Phase 1**: Adversary $\mathcal{A}$ adaptively issues the following queries multiple times. To simplify the description of this phase, we suppose that $\mathcal{A}$ never issues the same queries both in the following two kinds of queries.
  - Hash Query $\mathcal{Q}_H(ID)$: Given a queried identity $ID \in \mathcal{ID}$, algorithm $\mathcal{B}$ does the following steps:
    1. Pick $x \overset{\$}{\leftarrow} \mathbb{Z}_q^*$;
    2. If the queried identity $ID \in \mathcal{I}$, add tuple (ID, $g^{c\cdot x}, x$) into **HList**, and return $g^{c\cdot x}$ to $\mathcal{A}$;
    3. Otherwise add tuple (ID, g^x, x) into **HList**, and return g^x to $\mathcal{A}$;
  - Decryption Key Query $\mathcal{Q}_{SK}(ID)$: Given a queried identity $ID \in \mathcal{ID}$, algorithm $\mathcal{B}$ does the following steps:
    1. If $(ID, *, *) \notin$ **HList**, query $\mathcal{Q}_H(ID)$;
    2. According to ID, retrieve (ID, X, x) from **HList**;
    3. Return private key $g^{a\cdot x}$ to $\mathcal{A}$;
- **Challenge Phase**: Adversary $\mathcal{A}$ sends two challenge plaintexts (M_0^*, M_1^*) with $|M_0^*| = |M_1^*|$ to algorithm $\mathcal{B}$. Then $\mathcal{B}$ picks $d \overset{\$}{\leftarrow} \{0,1\}$, and does the following steps:
  1. Choose value $r \overset{\$}{\leftarrow} \mathbb{Z}_q^*$, initialize an empty array $\mathcal{L}$, and compute $C_0 = g^b$ and $C_0' = g^{b\cdot r}$;
  2. For $i \in [1, N]$, retrieve tuple (ID_i, X, x) from **HList** according to identity ID_i, compute $I_i = Z^x$ and $C_i = Z^{x\cdot r} \cdot M_d^*$, and set $\mathcal{L}[I_i] = C_i$;
  3. Return challenge ciphertext $\mathcal{C}_d^* = (C_0, C_0', \mathcal{L})$ to $\mathcal{A}$;
- **Query Phase 2**: This phase is the same as **Query Phase 1**. Note that both in **Query Phase 1** and **Query Phase 2**, $\mathcal{A}$ can not query the private keys corresponding to the identities in set $\mathcal{I}$;
- **Guess Phase**: Adversary $\mathcal{A}$ sends a guess d' to algorithm $\mathcal{B}$. If $d = d'$, $\mathcal{B}$ outputs 1; otherwise, outputs 0.

Next, we will prove that the above game is computationally indistinguishable with a real one in the view of adversary $\mathcal{A}$ when $Z = \hat{\mathbf{e}}(g,g)^{abc}$, and adversary $\mathcal{A}$ has the negligible advantage to win in the above game when $Z = \hat{\mathbf{e}}(g,g)^y$. These proofs will finally demonstrate this theorem.

In the above game, it is easy to verify that all generated private keys are valid. When $Z = \hat{\mathbf{e}}(g,g)^{abc}$, the generated challenge ciphertext is a real one. Hence, in the view of adversary $\mathcal{A}$, the above game in this case is computationally indistinguishable with a real one. When $Z = \hat{\mathbf{e}}(g,g)^y$, the generated challenge ciphertext is independent with all identities in set $\mathcal{I}$. The generated challenge ciphertext has the same distribution regardless of the choice of d. Hence, adversary $\mathcal{A}$ has the negligible advantage to win in the above game when $Z = \hat{\mathbf{e}}(g,g)^y$. Summarily, we have the following equation.

$$\begin{aligned}
&Adv_{\mathcal{B}}^{DBDH}(1^k)\\
&= Pr[\mathcal{B} = 1 | Z = \hat{\mathbf{e}}(g,g)^{abc}] - Pr[\mathcal{B} = 1 | Z = \hat{\mathbf{e}}(g,g)^y]\\
&= Pr[d = d' | Z = \hat{\mathbf{e}}(g,g)^{abc}] - Pr[d = d' | Z = \hat{\mathbf{e}}(g,g)^y]\\
&= Adv_{IBBE,\mathcal{A}}^{\text{SS-sMID-CPA}} + \frac{1}{2} - \frac{1}{2} = Adv_{IBBE,\mathcal{A}}^{\text{SS-sMID-CPA}}
\end{aligned}$$

In addition, it is clear that algorithm $\mathcal{B}$ is a PPT algorithm, if adversary $\mathcal{A}$ is a PPT adversary. In conclusion, if a PPT adversary $\mathcal{A}$ wins in the SS-sMID-CPA game of the above IBBE scheme with advantage $Adv_{IBBE,\mathcal{A}}^{\text{SS-sMID-CPA}}$, there is a PPT algorithm $\mathcal{B}$ that solves the DBDH problem in **BGen**(1^k) with $Adv_{\mathcal{B}}^{DBDH}(1^k) = Adv_{IBBE,\mathcal{A}}^{\text{SS-sMID-CPA}}$. □