A Reflection on the Security Proofs of Boneh-Franklin Identity-Based Encryption

Yu Chen

Institute of Software, School of Electronics Engineering and Computer Science
Peking University, Beijing, China
Email: cycosmic@gmail.com

Abstract. Boneh and Franklin constructed the first practical Identity-Based Encryption scheme (IBE) in 2001. They also defined a formal security model for IBE and proved their scheme (BF-IBE) to be secure in the random oracle model assuming the computational Bilinear Diffie-Hellman (CBDH) assumption holds. However, few years later, Galindo [1] pointed out a flawed step in its proof against adaptively chosen ciphertext attack (CCA) and claimed that the flaw can be fixed without changing the original scheme and the underlying assumption. In the same paper, Galindo provided a revised proof with a looser security reduction. Shortly afterwards, Nishioka [2] attempted to extend Galindo's idea to achieve a tighter security reduction. Unfortunately, we find that there are some lapses in their proofs, which make their proofs not infallible in the sense of CCA security for IBE setting. Zhang and Imai [3] proposed a another proof for BF-IBE in which the simulator simulates itself all the oracles. However, we show that there exists a inconspicuous lapse in the simulation of hash functions, which renders the simulator can not answer all the queries to the oracles coherently. In this paper, besides pointing out the lapses existed in the aforementioned proofs, we present a new proof for the CCA security of BF-IBE which relies on a stronger assumption, namely gap Bilinear Diffie-Hellman (GBDH) assumption.

Key words: identity-based encryption, security reduction, chosen ciphertext security

1 Introduction

Identity-Based Encryption (IBE) allows a party to encrypt a message using the recipient's identity as a public key. Such property simplifies key management and avoids the use of digital certificates. This can be very useful in applications such as email system where the recipient is often off-line and unable to present a public-key certificate while the sender encrypts a message.

Since Shamir proposed the concept of IBE in 1984 [4], various Identity-Based Signature (IBS) and Authentication (IBA) schemes have been proposed, but secure and fully functional IBE scheme was not found until Boneh and Franklin [5], Cocks [6] and Sakai *et al.* [7] presented three IBE schemes in 2001, respectively. Among those solutions, Boneh and Franklin's one happen to be the most practical one. In order to prove the security of BF-IBE, Boneh and Franklin [8] introduced new security definitions to fit the Identity-Based setting, then proved its security in the random oracle model assuming the hardness of computational Bilinear Diffie-Hellman problem [8]. For this reason, BF-IBE has received much attention and has had a great influence on later designs and analysis of cryptographic settings. z Numerous schemes [9] [10] [11] [12] [13] are based on BF-IBE scheme.

Fixed proofs about BF-IBE. The original security proof of BF-IBE was long believed correct until 2005, Galindo [1] pointed out a flawed step in the security reduction for CCA security. Galindo claimed that the flawed step could be fixed by his new security reduction without changing both the scheme and the underlying assumption if the efficiency of the security reduction is sacrificed. In the same year, Nishioka [2] enhanced Galindo's idea to provide another proof with tighter security reduction. In the same year, Zhang and Imai [3] proposed a new proof of BF-IBE, which was claimed essentially improved previously known results. Up to present, there is no doubt about the correctness of their fixed proofs.

1.1 Our contributions

Reflect previous proofs of BF-IBE. After re-examine the flawed step in BF-IBE's original CCA security proof exhibited in [8] and analyse why it fails, we point out the lapses in the subsequent revised security proofs proposed Galindo [1], Nishioka [2] and Zhang *et.al* [3], respectively. Galindo's proof and Nishioka's proof are similar. Both of their proofs begin with a doubtable assumption that the challenger know the number of queries which adversary will make at the beginning of the CCA game. In the challenge step, the behavior of adversary goes against the strict definition of a CCA adversary. In Zhang and Imai's proof [3], the simulator simulates itself all the oracles: the H_i oracles, extraction oracle, encryption oracle and decryption oracle. However, we find that the answers to the oracles are not coherent, which makes the simulation is not identical to the real attack in adversary's view.

Present a new proof of BF-IBE. Since BF-IBE scheme has been used as a primitive for numerous cryptographic protocols, the security of BF-IBE implies direct consequences for many other schemes [9] [10] [11] [12] [13]. It is necessary to provide a correct proof without flaws. Motivated by this, we provide a new proof of BF-IBE which employs the proof technique used in [14] [15]. We reduce the CCA security of BF-IBE directly to the underlying intractable problem without intermediate steps. We also remark that our security reduction is based on the gap Bilinea Diffie-Hellman (GBDH) assumption, which is a little stronger than the computational Bilinear Diffie-Hellman (CBDH) assumption used in the original proof.

1.2 Organization

In Section 2, we give the background information on security definitions and complexity assumptions. Section 3 briefly revisits BF-IBE scheme and its orginal security proof. In Section 4 we analysz the lapses in the subsequent revised proofs proposed by Galindo [1] and Nishioka [2], respectively. In Section 5 we point out an inconspicuous lapse in Zhang and Imai's proof [3]. Section 6 shows that if restricting the identities space to a finite set, then the full security of BF-IBE can be achieved from its *selective*-ID security. In Section 7 we present a new proof for BF-IBE relying on the GBDH assumption in the random oracle model. Finally, we conclude the paper in Section 8.

2 Preliminaries

We briefly review the groups equipped with efficiently computable bilinear maps. For more details, we recommend the reader to previous literature [8].

Bilinear Map. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups of prime order q. A map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is said as an admissible bilinear map if the following three properties hold.

1. Bilinear. $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1$ and all $a, b \in \mathbb{Z}_q^*$.
2. Non-degenerate. $e(P, P) \neq 1$.
3. Computable. There is an efficient algorithm to compute $e(P, Q)$ for any $P, Q \in \mathbb{G}_1$.

Bilinear Diffie-Hellman (BDH) Parameter Generator. A BDH parameter generator $\mathcal{G}$ is an algorithm which takes a security parameter $k \in \mathbb{Z}^+$ as input and outputs two groups of prime order q and an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. We describe it as $\mathcal{G}(1^k) \rightarrow (q, \mathbb{G}_1, \mathbb{G}_2, e)$.

2.1 Complexity Assumptions

Given groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order q, a blinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and a generator P of $\mathbb{G}_1$, we introduce three complexity assumptions as follows.

Computational Bilinear Diffie-Hellman Problem (CBDH). The CBDH problem [16] [5] is given the tuple (P, aP, bP, cP) for some $a, b, c \in \mathbb{Z}_q^*$, compute the $e(P, P)^{abc} \in \mathbb{G}_2$. An adversary $\mathcal{A}$ is said to have at least advantage ϵ in solving CBDH problem if Pr $\left[\mathcal{A}(P, aP, bP, cP) = e(P, P)^{abc}\right] \geq \epsilon$.

Decisional Bilinear Diffie-Hellman Problem (DBDH). For random $a, b, c, z \in \mathbb{Z}_q^*$ and a fair coin β. If $\beta = 1$ the challenger outputs a tuple $(P, aP, bP, cP, Z = e(P, P)^{abc}) \in D_1$. Else, it outputs a tuple $(P, aP, bP, cP, Z = e(P, P)^z) \in D_2$. The adversary is expected to output a guess β' of β. An adversary $\mathcal{A}$ is said to have at least an ϵ advantage in solving the DBDH problem if $\left|\Pr[\beta = \beta'] - \frac{1}{2}\right| \geq \epsilon$. Tuples from D_1 are denoted as "BDH" tuples in contrast to those from D_2 which will be called "random tuples". A DBDH oracle can determine whether a tuple (P, aP, bP, cP, Z) is a real "BDH" tuple.

Gap Bilinear Diffie-Hellman Problem (GBDH). The GBDH problem is given a CBDH challenge (P, aP, bP, cP), to compute $e(P, P)^{abc}$ with the help of a DBDH oracle.

2.2 Security Notions

Recall that an IBE scheme consists of four algorithms [4] [8]: Setup, Extract, Encrypt, and Decrypt. The Setup algorithm generates system parameters params and a master secret master-key. The Extract algorithm uses the master-key to generate the private key corresponding to a given identity. The Encrypt algorithm encrypts messages for a given identity (using the system parameters) and the Decrypt algorithm decrypts ciphertext using the private key. The message space is $\mathcal{M}$. The ciphertext space is $\mathcal{C}$.

Chosen Ciphertext Security for IBE. An IBE scheme $\mathcal{E}$ is said to be secure against adaptively chosen ciphertext attack (IND-ID-CCA) if no probabilistic polynomial time (PPT) algorithm $\mathcal{A}$ has a non-negligible advantage against the challenger in the following game:

Setup. The challenger takes the security parameter and runs the Setup algorithm. It gives the adversary the resulting system parameters and keeps the master secret to itself.

Phase 1. The adversary issues queries $q_1, \ldots, q_m$ where query q_i is one of:

- Extraction query $\langle \mathsf{ID}_i \rangle$. The challenger responds by running algorithm Extract to generate the private key d_i corresponding to ID_i. It sends d_i to the adversary $\mathcal{A}$.
- Decryption query $\langle \mathsf{ID}_i, C_i \rangle$. The challenger responds by running algorithm Extract to generate the private key d_i corresponding to ID_i. It then runs algorithm Decrypt to decrypt the ciphertext C_i using the private key d_i. It sends the resulting plaintext to the adversary $\mathcal{A}$.

These queries may be asked adaptively, that is, each query q_i may depend on the replies to $q_1, \ldots, q_{i-1}$.

Challenge. Once the adversary decides that Phase 1 is over it outputs two equal length plaintexts $M_0, M_1 \in \mathcal{M}$ and an identity ID on which it wishes to be challenged. The only constraint is that ID did not appear in any private key extraction query in Phase 1. The challenger picks a random bit $\beta \in \{0,1\}$ and sets $C = \mathsf{Encrypt}(\mathit{params}, \mathsf{ID}, M_\beta)$. It sends C as the challenge to the adversary.

Phase 2. The adversary issues more queries $q_{m+1}, \ldots, q_r$ where q_i is one of:

- Extraction query $\langle \mathsf{ID}_i \rangle \neq \mathsf{ID}$. Challenger responds as in Phase 1.
- Decryption query $\langle \mathsf{ID}_i, C_i \rangle \neq \langle \mathsf{ID}, C \rangle$. Challenger responds as in Phase 1.

These queries may be asked adaptively as in Phase 1.

Guess. Finally, the adversary outputs a guess $\beta' \in \{0,1\}$ and wins the game if $\beta = \beta'$.

We refer to such an adversary $\mathcal{A}$ as an IND-ID-CCA adversary. We define adversary $\mathcal{A}$'s advantage over the scheme $\mathcal{E}$ by $Adv^{\mathsf{CCA}}_{\mathcal{E},\mathcal{A}}(k) = \left|\Pr[c = c'] - \frac{1}{2}\right|$, where k is the security parameter. The probability is over the random bits used by the challenger and the adversary. Similarly, the IND-ID-CPA security notion can be defined by using a similar game as the one above but disallowing decryption queries. The advantage of an adversary $\mathcal{A}$ is defined by $Adv^{\mathsf{CPA}}_{\mathcal{E},\mathcal{A}}(k) = \left|\Pr[\beta = \beta'] - \frac{1}{2}\right|$.

Definition 2.1 *We say that an IBE scheme $\mathcal{E}$ is* IND-ID-CCA (IND-ID-CPA) *secure if for any probabilistic polynomial time* IND-ID-CCA (IND-ID-CPA) *adversary $\mathcal{A}$ the advantage $Adv^{\mathsf{CCA}}_{\mathcal{E},\mathcal{A}}(k)$ $\left(Adv^{\mathsf{CPA}}_{\mathcal{E},\mathcal{A}}(k)\right)$ is negligible.*

Selective-ID model. Boneh and Franklin [5] defined the adaptive chosen ciphertext security for IBE systems by the above game. We refer to it as full IBE security model. In this model, the adversary can issue both adaptive chosen private key queries and adaptive chosen ciphertext queries. Eventually, the adversary adaptively chooses the identity it wishes to attack and asks for a semantic security challenge for this identity. Canetti, Halevi, and Katz [17] [18] defined a slightly weaker security model, called *selective*-ID security model, in which the adversary must commit ahead of time (non-adaptively) to the identity it intends to attack. More precisely, it is defined using the following game:

Init. The adversary outputs an identity ID_{ch} where it wishes to be challenged.

Setup and **Phase 1** are same as in IND-ID-CCA game.

Phase 1. Same as in IND-ID-CCA game.

Challenge. Once the adversary decides that Phase 1 is over it outputs two equal length plaintexts M_0, M_1 on which it wishes to be challenged. The challenger picks a random bit $\beta \in \{0, 1\}$ and sets the challenge ciphertext to $C = \mathsf{Encrypt}(\mathsf{params}, \mathsf{ID}_{ch}, M_\beta)$. It sends C as the challenge to the adversary.

Phase 2 and **Guess** are same as in IND-ID-CCA game.

We refer to such an adversary $\mathcal{A}$ as an IND-sID-CCA adversary. The advantage of the adversary $\mathcal{A}$ is defined by $Adv_{\mathcal{E},\mathcal{A}}(k) = \left|\Pr[\beta = \beta'] - \frac{1}{2}\right|$, where the probability is over the random bit used by the challenger and the adversary.

Definition 2.2 *An IBE system $\mathcal{E}$ is* IND-sID-CCA *secure if for any PPT* IND-sID-CCA *adversary $\mathcal{A}$ the advantage $Adv_{\mathcal{E},\mathcal{A}}(k)$ is negligible.*

3 Boneh-Franklin's IBE Scheme

In this section, we briefly describe BF-IBE scheme [5] and examine the original proof. Boneh and Franklin named their full scheme as FullIdent. In order to make the presentation easier, they also define the BasicIdent and two public key encryption (PKE) scheme called BasicPub and $\mathsf{BasicPub}^{hy}$. BasicIdent which has only CPA security, is a simplified version of FullIdent, BasicPub is a PKE scheme derived from BasicIdent, and $\mathsf{BasicPub}^{hy}$ is a PKE scheme obtained by applying the Fujisaki-Okamoto conversion [19] to BasicPub. We first review the FullIdent in Figure 1.

A series of security reductions for FullIdent and BasicIdent follows the diagram below:

$$\begin{tikzcd}
\mathsf{FullIdent} & \mathsf{BasicPub}^{hy} & \mathsf{BasicPub} & \mathrm{CBDH} \\
\mathcal{A}(t,\epsilon) \arrow[u, dashed] \arrow[r, "\text{Reduce 4}"] & \mathcal{A}_3(t_3,\epsilon_3) \arrow[u, dashed] \arrow[r, "\text{Reduce 3}"] & \mathcal{A}_1(t_1,\epsilon_1) \arrow[u, dashed] \arrow[r, "\text{Reduce 1}"] & \mathcal{B}(t',\epsilon') \arrow[u, dashed] \\
 & \mathsf{BasicIdent} & \mathcal{A}_2(t_2,\epsilon_2) \arrow[l, dashed] \arrow[u, "\text{Reduce 2}"'] &
\end{tikzcd}$$

The following results are presented in [8]. Hereafter, let q_E, q_D, and q_{H_i} denote the number of extraction, decryption and H_i random oracle queries, respectively.

Reduction 1. Suppose there is an IND-CPA adversary $\mathcal{A}_1$ has the advantage $\epsilon(k)$ against BasicPub and $\mathcal{A}_1$ makes at most q_{H_2} queries to the random oracle H_2. Then there is an algorithm $\mathcal{B}$ that solves the CBDH problem with advantage at least $2\epsilon(k)/q_{H_2}$ in running time $O(time(\mathcal{A}_1))$.

BF-IBE(FullIdent)	
Setup(1^k): $s \leftarrow \mathbb{Z}_q^*$; $P_{pub} = sP$ $\mathsf{params} = (q, \mathbb{G}_1, \mathbb{G}_2, e, n, P, P_{pub}, H_i)$ $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n$, $H_3 : \{0,1\}^n \times \{0,1\}^n \rightarrow \mathbb{Z}_q^*$, $H_4 : \{0,1\}^n \rightarrow \{0,1\}^n$.	**Extract**(ID, params, master-key) $Q_{\mathsf{ID}} = H_1(\mathsf{ID})$ $d_{\mathsf{ID}} = sQ_{\mathsf{ID}}$.
Encrypt(ID, params, M) $Q_{\mathsf{ID}} = H_1(\mathsf{ID})$; $\sigma \leftarrow \{0,1\}^n$, $r = H_3(\sigma, M)$; $U = rP$; $V = \sigma \oplus H_2(e(P_{pub}, Q_{\mathsf{ID}})^r)$; $W = M \oplus H_4(\sigma)$; $C = \langle U, V, W \rangle$.	**Decrypt**(C, params, d_{ID}) Parse $C = \langle U, V, W \rangle$. If $U \notin \mathbb{G}_1$, return $\perp$ Compute $\sigma = V \oplus H_2(e(d_{\mathsf{ID}}, U))$. Compute $M = W \oplus H_4(\sigma)$. Set $r = H_3(\sigma, M)$. If $U \neq rP$, return $\perp$. Output M as the decryption of C.

Fig. 1. The algorithms of FullIdent

Reduction 2. Suppose there is an IND-ID-CPA adversary $\mathcal{A}_2$ that has advantage $\epsilon(k)$ against BasicIdent and makes at most q_E private key extraction queries, and at most q_{H_2} queries to the random oracle H_2. Then there is an IND-CPA adversary $\mathcal{A}_1$ against BasicPub with advantage at least $\epsilon(k)/e(1+q_E)$ in running time $O(time(\mathcal{A}_2))$. Here $e \approx 2.71$ is the base of the natural logarithm.

From Reduction 1 and Reduction 2, we get:

Result 1. BasicIdent is IND-ID-CPA secure assuming the CBDH is hard in groups generated by $\mathcal{G}$. Concretely, suppose there is an IND-ID-CPA adversary $\mathcal{A}_2$ that has advantage $\epsilon(k)$ against BasicIdent. If $\mathcal{A}_2$ makes at most $q_E > 0$ private key extraction queries and q_{H_2} hash queries to H_2. Then there is an algorithm $\mathcal{B}$ that solves CBDH with advantage at least $\frac{2\epsilon(k)}{e(1+q_E)\cdot q_{H_2}}$.

Reduction 3. Using the Fujisaki-Okamoto transformation Boneh and Franklin introduce BasicPub^{hy} which is IND-CCA secure. Suppose there is an IND-CCA adversary $\mathcal{A}_3$ that has advantage $\epsilon(k)$ against BasicPub^{hy} and makes at most q_D decryption queries, and at most q_{H_3}, q_{H_4} queries to the random oracles H_3, H_4 respectively. Then there exists an IND-CPA adversary $\mathcal{A}_1$ against BasicPub with advantage at least $[(\epsilon(k)+1)(1-2/q)^{q_D} - 1]/2(q_{H_3} + q_{H_4})$ in running time $O(time(\mathcal{A}_3))$.

Reduction 4. Suppose there is an IND-ID-CCA adversary $\mathcal{A}$ that has advantage $\epsilon(k)$ against FullIdent. Suppose $\mathcal{A}$ makes at most q_E private key extraction queries, at most q_D decryption queries, and at most q_{H_1} queries to the random oracle H_1. Then there exists an IND-CCA adversary $\mathcal{A}_3$ against BasicPub^{hy} with advantage at least $\epsilon(k)/e(1+q_E+q_D)$ in running time $O(time(\mathcal{A}))$.

From Reduction 1, Reduction 3 and Reduction 4, we have:

Result 2. FullIdent is IND-ID-CCA secure assuming CBDH is hard in groups generated by $\mathcal{G}$. Concretely, suppose there is an IND-ID-CPA adversary $\mathcal{A}$ that has advantage $\epsilon(k)$ against BasicIdent. If $\mathcal{A}$ makes at most $q_E > 0$ private key extraction queries, at most q_D decryption queries, and at most q_{H_2}, q_{H_3}, q_{H_4} hash queries to H_2, H_3, H_4, respectively. Then there is an algorithm $\mathcal{B}$ that solves CBDH with advantage at least $\left[\frac{\epsilon(k)}{e(1+q_E+q_D)+1}(1-2/q)^{q_D}-1\right]/q_{H_2}(q_{H_3}+q_{H_4})$.

3.1 Analysis of Reduction 4 in BF-IBE

The aim of Reduction 4 is constructing an $\mathsf{IND\text{-}CCA}$ adversary $\mathcal{A}_3$ against $\mathsf{BasicPub}^{hy}$ by interacting with an $\mathsf{IND\text{-}ID\text{-}CCA}$ adversary $\mathcal{A}$ against $\mathsf{FullIdent}$. Next we list two lapses in Reduction 4 of BF-IBE, which is the Lemma 4.6 in [8].

- **Issue 1**. In Phase 1, when $\mathcal{A}$ issues a decryption query $\langle \mathsf{ID}_i, C_i\rangle$, where $C_i = \langle U_i, V_i, W_i\rangle = \langle rP, \sigma \oplus H_2(e(Q_i, P_{pub})^r), M \oplus H_4(\sigma)\rangle$. According to the above algorithm, if $coin_i = 1$, $\mathcal{A}_3$ will modify C_i as $C_i' = \langle U_i', V_i', W_i'\rangle = \langle b_iU_i, V_i, W_i\rangle$ and then relay C_i' to its challenger. When the challenger decrypts C_i' using the private key d_{ID}, it does:
  1. Compute $V_i' \oplus H_2(e(d_{\mathsf{ID}}, U_i')) = V_i \oplus H_2(e(d_{\mathsf{ID}}, b_iU_i)) = \sigma \oplus H_2(e(Q_i, P_{pub})^r)$ $\oplus H_2(e(sQ_{\mathsf{ID}}, b_irP)) = \sigma$. This step recovers the random chosen $\sigma \in \{0,1\}^n$ exactly.
  2. Compute $W_i' \oplus H_4(\sigma) = W_i \oplus H_4(\sigma) = M \oplus H_4(\sigma) \oplus H_4(\sigma) = M$. This step recovers the original plaintext M exactly.
  3. Set $r = H_3(\sigma, M)$. Test whether $U_i' = rP$. Note that b_i is randomly chosen from $\mathbb{Z}_q^*$ and H_3 is a random oracle model not controlled by $\mathcal{A}_3$. These facts imply that the probability of $H_3(\sigma, M) \neq b_ir$ is $1-1/q$, and therefore challenger will reject the modified ciphertext with overwhelming probability as $\mathsf{BasicPub}^{hy}$ is $\mathsf{IND\text{-}CCA}$ secure.

  Thereby, $\mathcal{A}_3$ can not employ the decryption oracle of $\mathsf{BasicPub}^{hy}$ to answer decryption queries issued by $\mathcal{A}$ if the corresponding $coin_i = 1$.
- **Issue 2**. In the Challenge stage, $\mathcal{A}$ outputs ID_{ch} and two equal length M_0, M_1 on which it wishes to be challenged. $\mathcal{A}_3$ gives its challenger M_0, M_1 as the messages that it wishes to be challenged on. The challenger gives $\mathcal{A}_3$ the ciphertext $C = \langle U, V, W\rangle = \langle rP, \sigma \oplus H_2(e(Q_{\mathsf{ID}}, P_{pub})^r), M_c \oplus H_4(\sigma)\rangle$ such that C is the encryption of M_β for random $\beta \in \{0,1\}$. Let $\langle \mathsf{ID}_{ch}, Q, b, coin\rangle$ be the corresponding tuple on the H_1^{list}. According to the above algorithm, if $coin = 0$ $\mathcal{A}_3$ aborts the game and the attack fails, otherwise $\mathcal{A}_3$ will modify C to be $C' = \langle U', V', W'\rangle = \langle b^{-1}U, V, W\rangle$ and relays C' to $\mathcal{A}$ as the challenge ciphertext. Boneh and Franklin claimed that C' is also a proper $\mathsf{FullIdent}$ encryption result of M_c under the public key $\mathsf{ID}_{ch} = Q = bQ_{\mathsf{ID}}$. However, if C' is a valid ciphertext of M_c in $\mathsf{FullIdent}$, we have $r' = rb^{-1}$, $H_4(\sigma) = H_4(\sigma')$, $H_4(\sigma) = r$, $H_4(\sigma') = r'$. These facts imply that $b = 1$. Be aware of that b is randomly chosen from $\mathbb{Z}_q^*$, thereby the probability that C' is a valid ciphertext of M_β in $\mathsf{FullIdent}$ is $1 - 1/q$. On the other hand, for the same reason explained in Issue 1, $\mathcal{A}$ will reject the modified ciphertext with overwhelming probability.

Therefore, these two lapses render the Reduction 4 in the original proof invalid. By the way, Galindo only noticed Issue 1 in [1].

3.2 Flipping coin technique

In the proof of Reduction 2, Boneh and Franklin borrowed the technique from Coron's analysis of the Full Domain Hash signature scheme [20], which we refer to it as flipping coin technique. More precisely, $\mathcal{A}_1$ answers H_1 queries according to the result of flipping a coin when simulating the H_1 random oracle for $\mathcal{A}_2$, i.e. before answering a new H_1-query at ID_i, $\mathcal{A}_2$ will generate a random $coin \in \{0, 1\}$ with probability $\Pr[coin = 0] = \delta$,

- If $coin = 0$, return $Q_i = b_i P \in \mathbb{G}_1^*$.
- If $coin = 1$, return $Q_i = b_i Q_{\mathsf{ID}} \in \mathbb{G}_1^*$.

In Phase 1, only when $coin_i = 0$ could $\mathcal{A}_1$ answer the private key query $\langle \mathsf{ID}_i \rangle$ properly, because $Q_i = b_i P$ enables $\mathcal{A}_1$ to extract the private key as $d_i = b_i P_{pub}$. In the Challenge stage, only the case $coin_i = 1$ allows $\mathcal{A}_1$ to utilize $\mathcal{A}_2$'s guess to win the game, because $Q_i = b_i Q_{\mathsf{ID}}$ enables $\mathcal{A}_2$ to make use of the homomorphic relationship.

Waters [21] generalized such "flipping coin technique" as *Partitioning Reduction*: creating a reduction algorithm $\mathcal{B}$ that partitions the identity space into two parts (1) identities for which it can create private keys; (2) identities that it can use in the challenge phase. Simulator hopes that the extraction/decryption queries and the challenge identity fall favorably in the partition, then the simulation is identical to the real attack in the adversary's view and the attack succeeds. The partition of identity space is only determined by the simulator (the distribution of $coin$) and independent to adversary's particular behavior, which enables the possibility of perfect simulation computable.

When Boneh and Franklin applied this identical technique to Reduction 4, it does not work. Because both the challenger and the adversary $\mathcal{A}$ will check the validity of ciphertext, as pointed out in Issue 1 and Issue 2. The reason is that Fujisaki and Okamoto transformation [19] removes the homomorphic relationship between the ciphertext and its corresponding key, malformed ciphertext will be rejected with overwhelming probability, thereby the simulation fails.

4 Analysis of Galindo and Nishioka's Proofs

In this section we investigate the two subsequent revised proofs provided by Galindo and Nishioka, respectively.

4.1 Galindo's proof

Galindo [1] tried to fix the Reduction 4 by modifying the simulation method of random oracle H_1. His revised proof of Reduction 4 is shown as follows.
Setup. Same as BF-IBE's.

H_1-**queries**. Before initializing H_1^{list}, $\mathcal{A}_3$ selects a random $j \leftarrow \{1, \ldots, q_{H_1}\}$. When $\mathcal{A}$ queries H_1 at ID_i, $\mathcal{A}_3$ responds as follows: if $i \neq j$, it picks $b_i \leftarrow \mathbb{Z}_q^*$ and sets $Q_i = b_i P$, adds $\langle \mathsf{ID}_i, Q_i, b_i \rangle$ to the list. If $i = j$, it sets $Q_i = Q_{\mathsf{ID}}$, adds $\langle \mathsf{ID}_i, Q_i, * \rangle$ to the list. Finally, $\mathcal{A}_3$ sends Q_i to $\mathcal{A}$.

Phase 1 - Extraction queries. When $\mathcal{A}$ asks for the private key of ID_i, $\mathcal{A}_3$ runs the above algorithm and gets $H_1(\mathsf{ID}_i) = Q_i$, where $\langle \mathsf{ID}_i, Q_i, b_i \rangle$ is the corresponding entry in H_1^{list}. If $i = j$, then $\mathcal{A}_3$ aborts the game. Otherwise, it sets $d_i = b_i P_{pub}$. Finally, $\mathcal{A}_3$ gives d_i to $\mathcal{A}$.

Phase 1 - Decryption queries. $\mathcal{A}_3$ answers to decryption query $\langle \mathsf{ID}_i, C_i \rangle$ as follows. It runs H_1-queries algorithm and let $\langle \mathsf{ID}_i, Q_i, b_i \rangle \in H_1^{list}$. If $i \neq j$, then $\mathcal{A}_3$ retrieves the private key d_i and decrypts C_i using the decryption algorithm. If $i = j$, then $Q_i = Q_{\mathsf{ID}}$, and the decryption of $\langle \mathsf{ID}_i, C_i \rangle$ is the same as the decryption of C_i under $\mathsf{BasicPub}^{hy}$. Then, $\mathcal{A}_3$ asks its challenger to decrypt C_j and relays the answer to $\mathcal{A}$.

Challenge. $\mathcal{A}$ outputs a public key ID_{ch} and two messages M_0, M_1 on which it wishes to be challenged. $\mathcal{A}_3$ proceeds as follows. If $\mathsf{ID}_{ch} \neq \mathsf{ID}_j$, it aborts the game and the attack against $\mathsf{BasicPub}^{hy}$ failed. Otherwise, it sends M_0, M_1 to its own challenger and gets back C, the encryption of M_β for a random bit $\beta \in \{0, 1\}$ under $\mathsf{BasicPub}^{hy}$. Finally, $\mathcal{A}_3$ relays C to $\mathcal{A}$, which is an also encryption of M_β under ID_{ch} for $\mathsf{FullIdent}$.

The **Phase 2** and **Guess** stage are identical to BF-IBE's.

In this game $\mathcal{A}_3$'s simulation can be aborted for two reasons: (1) in Phase 1 $\mathcal{A}$ issues the private key query of ID_j, or (2) in Challenge stage, the challenge identity $\mathsf{ID}_{ch} \neq \mathsf{ID}_j$. Note that $\mathcal{A}_3$ will not abort in Phase 2, since in this case $\mathcal{A}$ is not allowed to query the private key for $\mathsf{ID}_{ch} = \mathsf{ID}_j$.

Let $\mathcal{E}_1$ be the event that $\mathcal{A}_3$ aborts due to (1), $\mathcal{E}_2$ be the event that $\mathcal{A}_3$ aborts due to (2). The probability that $\mathcal{A}_3$ does not abort is Pr$[\neg\mathcal{E}_1 \wedge \neg\mathcal{E}_2]$ = Pr$[\neg\mathcal{E}_2 | \neg\mathcal{E}_1]$ Pr$[\neg\mathcal{E}_1]$.

Galindo deemed that the upper bound for Pr$[\mathcal{E}_1]$ was q_E / q_{H_1}, since the maximum number of private extraction queries is q_E; the lower bound for Pr$[\neg\mathcal{E}_2 | \neg\mathcal{E}_1]$, that is the probability that $\mathcal{A}$ choose ID_j as the challenge identity, was $1/q_{H_1}$. Therefore, he concluded

$$\text{Pr}[\mathcal{A}_3 \text{ does not abort}] \geq \frac{1}{q_{H_1}} \left(1 - \frac{q_E}{q_{H_1}} \right)$$

Now we point out three issues which may be overlooked in Galindo's proof.

- **Issue 1**. According to the definition of $\mathsf{IND\text{-}ID\text{-}CCA}$ game, q_{H_1} is unknown to the challenger $\mathcal{A}_3$ until the end of the game. So the execution of $\mathcal{A}_3$'s selecting a random $j \leftarrow \{1, \ldots, q_{H_1}\}$ at the beginning of simulation is questionable. In the other side, in order to provide a general and valid proof, the construction of $\mathcal{A}_3$ should be independent of the concrete behaviour of adversary $\mathcal{A}$, such as how many queires $\mathcal{A}$ issues. In a word, this issue make the proof does not hold in a general sense.
- **Issue 2**. Even Issue 1 could be ignored, here follows issue 2. In the challenge stage, when $\mathcal{A}$ outputs the target identity ID_{ch}, the simulator need to judge if $\mathsf{ID}_{ch} = \mathsf{ID}_j$. In fact, the exact number of H_1 queries that $\mathcal{A}$ issues in Phase 1 may differ from different adversaries in different simulations. Moreover, in some cases whether "ID_j" exists is unknown, thus the probability of "$\mathcal{A}_3$ does not abort" is immeasurable. For example, suppose the random $j = 10$ and an $\mathcal{A}$ issues only three H_1 queries in Phase 1, then the so called ID_j does not even exist. It is fair to say the simulation algorithm is not well defined.

- **Issue 3**. Even both Issue 1 and Issue 2 could be fixed, there is issue 3 following. The result of $\Pr[\neg\mathcal{E}_2|\neg\mathcal{E}_1] \geq 1/q_{H_1}$ is implied from the hypothesis that in the challenge stage the adversary $\mathcal{A}$ will randomly picks the target ID_{ch} from the current H_1^{list}. First, due to the same reason of Issue 2, whether the so called "ID_j" exists is a question, thereby the probability $\Pr[\mathsf{ID}_{ch} = \mathsf{ID}_j]$ is not well defined itself. Second, this goes against the definition of IND-ID-CCA which states that the target ID_{ch} can be chosen without any restriction, in particular outside the current H_1^{list}. Someone may argue that if the adversary $\mathcal{A}$ does not choose ID_{ch} from the current H_1^{list}, the advantage against the IND-ID-CCA game will be statistically closed to 0. Remember that $\mathcal{A}$ could issue the corresponding H_1-query in Phase 2. Besides, there is no evidence guarantees that the adversary $\mathcal{A}$ will choose the target identity uniformly from either inside or outside the current H_1^{list}.

From the above analyses, we think the proof proposed by Galindo is not infallible.

4.2 Nishioka's proof

In IndoCrypt 2005, Nishioka gave a new proof for the security of BF-IBE scheme in [2], claimed that it has a tighter security reduction than had been previously believed. Realizing that there are some problems in Galindo's proof, Nishioka claimed that Galindo's proof could be revised by his new proof. Unfortunately, we think the new proof shares the similar fundamental problems as Galindo's proof.

Nishioka's proof for Reduction 4 is similar to Galindo's proof except three minor alterations. The first alteration is in the simulation of H_1-queries: in [1] $\mathcal{A}_3$ selects a random $j \in \{1, \ldots, 1 + q_{H_1}\}$, while in [2] $\mathcal{A}_3$ selects a random $j \in \{1, \ldots, 1 + q_{H_1} + q_D\}$. The second alteration is in [2] $\mathcal{A}_3$ maintains two lists named as H_1^{list1} and H_1^{list2}, where L_1 and L_2 are their corresponding size. H_1^{list1} is used to save $\mathcal{A}_3$'s responses to H_1-queries and decryption queries, while H_1^{list2} is used to save $\mathcal{A}_3$'s responses to extraction queries. The last alternation is replacing $\mathsf{ID}_i = \mathsf{ID}_j$ with $i = L_1 - 1$ in the challenge stage. The rest parts are identical to Galindo's proof.

In order to compute the probability that $\mathcal{A}_3$ does not abort during the simulation, Nishioka defines $\mathcal{E}_1$ as the event that $\mathcal{A}_3$ issues a private key query ID_j which corresponds to the tuple $\langle \mathsf{ID}_j, Q_{\mathsf{ID}}, * \rangle$ on H_1^{list1} during Phase 1 or 2, and defines $\mathcal{E}_2$ as the event that $\mathcal{A}$ sets the challenge identity ID_{ch} that does not correspond to the tuple $\langle \mathsf{ID}_j, Q_{\mathsf{ID}}, * \rangle$ on H_1^{list1}. Then Nishioka claimed that $\Pr[\mathcal{A}_3 \text{ does not abort}] = \Pr[\neg\mathcal{E}_1 \wedge \neg\mathcal{E}_2] = \Pr[\neg\mathcal{E}_2] \geq 1/(1 + q_{H_1} + q_D)$ (Equation. 2 in Section 3.2 in [2]). We summarise the lapses in Nishioka's proof as follows.

- **Issue 1**. The execution of $\mathcal{A}_3$'s selecting a random $j \leftarrow \{1, \ldots, 1 + q_{H_1} + q_D\}$ when initializing H_1^{list1} is doubtable. The reason is same as Issue 1 of Galindo's proof.
- **Issue 2**. The computation of $\Pr[\mathcal{A}_3 \text{ does not abort}] = \Pr[\neg\mathcal{E}_1 \wedge \neg\mathcal{E}_2] = \Pr[\neg\mathcal{E}_2] \geq 1/(1 + q_{H_1} + q_D)$ is not correct. In fact,

$$\Pr[\mathcal{A}_3 \text{ does not abort}] = \Pr[\neg\mathcal{E}_1 \wedge \neg\mathcal{E}_2] = \Pr[\neg\mathcal{E}_1]\Pr[\neg\mathcal{E}_2|\neg\mathcal{E}_1]$$

where

$$\Pr[\neg\mathcal{E}_2|\neg\mathcal{E}_1] = \sum_{i}^{1+q_{H_1}+q_D} \Pr[j=i]\Pr[\mathcal{B} \text{ answers } H_1(\mathsf{ID}_{ch}) \text{ with } Q_{\mathsf{ID}})|j=i]$$
$$= \frac{1}{1+q_{H_1}+q_D} \sum_{i}^{1+q_{H_1}+q_D} \Pr[\mathcal{B} \text{ answers } H_1(\mathsf{ID}_{ch}) \text{ with } Q_{\mathsf{ID}}|j=i]$$

For any fixed i, Pr[$\mathcal{B}$ answers $H_1(\mathsf{ID}_{ch})$ with $Q_{\mathsf{ID}}|j=i$] is immeasurable, thereby the probability of Pr[$\mathcal{A}_3$ does not abort] is immeasurable.

These two issues render Nishioka's proof for Reduction 4 not correct.

Remark 1. Galindo and Nishioka abandoned the "flip coin technique" by straightforward simulation. However, from the above analysis we find that straightforward simulation makes the the probability of perfect simulation immeasurable. Their proofs can not be fixed even by maximising the values of q_{H_i}, q_E and q_D in the setup phase.

5 Zhang and Imai's proof

Zhang and Imai gave a new proof of the BF-IBE in [3]. The main difference lies in that they directly reduce the CCA security of BF-IBE to the underlying CBDH problem, not the IND-CCA security of BasicPub^{hy}. However, we find in their proof, the simulator fails to simulate "properly", which means the IND-ID-CCA adversary $\mathcal{A}$ could distinguish the simulation from real attacks. Before we point out the concrete issues, we first have a glance at their proof.

$\mathcal{B}$ is given the a CDBH instance $(P, aP, bP, cP) \in (\mathbb{G}_1)^4$ whose goal is to output $e(P,P)^{abc}$. $\mathcal{B}$ simulates all the H_i functions.

Setup. Same as in BF-IBE.

H_1**-queries**. Same as in BF-IBE, except replace Q_{ID} with P_2.

H_2**,** H_3**,** H_4**-queries**. $\mathcal{B}$ proceeds H_2, H_3 and H_4 queries using the same method: when a H_i-query comes, if there is an such entry on H_i-list, $\mathcal{B}$ returns the corresponding result to $\mathcal{A}$; otherwise, $\mathcal{B}$ chooses a random value for the query and adds it into H_i-list.

Extraction queries. Same as in BF-IBE.

Decryption queries. When a query $(\mathsf{ID}, C = \langle U, V, W\rangle)$ comes, $\mathcal{B}$ searches H_1-list for (ID), H_2-list for (t), H_3-list for a tuple (σ, M) and H_4-list for (σ) such that $(\mathsf{ID}, M, r, t, \sigma)$ such that satisfying below equations: 1) $Q_{\mathsf{ID}} = H_1(\mathsf{ID})$; 2) $r = H_3(\sigma, M)$ and $U = rP$; 3) $t = e(P_{pub}, Q_{\mathsf{ID}})^r$ and $V = \sigma \oplus H_2(t)$; 4) $W = M \oplus H_4(\sigma)$. If there exists such an M and associated $(\mathsf{ID}, \sigma, r, t)$ in those lists, $\mathcal{B}$ returns M to $\mathcal{A}$ as the answer. Otherwise, $\mathcal{B}$ returns "reject" to $\mathcal{B}$.

Challenge. On $\mathcal{A}$'s input ID and M_0, M_1 , let the corresponding tuple in H_1^{list} is $(\mathsf{ID}, Q_{\mathsf{ID}}, s, coin)$. If $coin = 0$, $\mathcal{B}$ aborts the simulation; otherwise, $\mathcal{B}$ chooses a random $v^* \in \{0,1\}^n$, $\beta \in \{0,1\}$ and sets $V = M_\beta \oplus v^*$ and $W = \{0,1\}^n$. Especially, $\mathcal{B}$ sets $U = P3^{s^{-1}}$ and returns $C = \langle U, V, W\rangle$ to $\mathcal{A}$ as the challenge ciphertext.

$\mathcal{B}$ keeps interacting with $\mathcal{A}$ until $\mathcal{A}$ halts or aborts. Finally, when $\mathcal{A}$ terminates, $\mathcal{B}$ chooses an arbitrary t from H_2-list and computes $t^{s^{-1}}$ as its answer to the CDBH problem. This completes the decryption of $\mathcal{B}$.

Next, we point out the lapses in their proof.

- **Issue 1**. First it is obliged to correct two typos in the above proof. (1) According to the encryption algorithm, $V = M_\beta \oplus v^*$ should be corrected as $V = \sigma \oplus v^*$. (2) The authors set $U = (P_3)^{s^{-1}}$, then $\mathcal{B}$ should answers t as its answer to the CDBH problem but not $t^{s^{-1}}$. It is easy to verify that when $Q_{\mathsf{ID}} = P_2 = sbP$, the associated $d_{\mathsf{ID}} = sabP$, therefore $t = e(U, d_{\mathsf{ID}}) = e(s^{-1}cP, sabP) = e(P, P)^{abc}$ is exactly the answer we need.
- **Issue 2**. $\mathcal{B}$ should answer all extraction queries and H_i-queries "properly", and returns the "proper" challenge ciphertext, which means $\mathcal{B}$ should simulates the real attack scenario perfectly. In the challenge stage, when $\mathcal{A}$ submits the target identity ID and two messages M_0, M_1, $\mathcal{B}$ is expected to return a valid ciphertext of M_β. In order to do so, $\mathcal{B}$ need to pick a random σ and query H_3-oracle for $r = H_3(\sigma, M_\beta)$, then queries the H_2-oracle for $v = H_2(e(P_{pub}, Q_{\mathsf{ID}})^r)$, at last query H_4-oracle for $H_4(\sigma)$. The key point is $\mathcal{B}$ should manage to make $r = c$ $(U = P_3)$ and at the same time ensure all the queries are indistinguishable in $\mathcal{A}$'s view. Zhang and Imai generated the challenge ciphertext by implicitly assigning $H_2(e(P_{pub}, Q_{\mathsf{ID}})^{r^*})$ with a random $v^* \in \mathbb{Z}_q$ and assigning $H_4(\sigma^*)$ with random $w^* \in \{0,1\}^n$, thus implicitly means that the underlying σ^* must satisfy $H_3(\sigma^*, M_\beta) = r^*$ and $H_4(\sigma^*) = w^*$. However, the ciphertext is not a valid encryption result of M_β. Note that both $e(P_{pub}, Q_{\mathsf{ID}})^{r^*}$ and σ^* are unknown to $\mathcal{B}$, thus renders $\mathcal{B}$' simulation for H_2, H_3 and H_4 are not coherent in the game. For example, if $\mathcal{A}$ explicitly issues a query $e(P_{pub}, Q_{\mathsf{ID}})^{r^*}$ to H_2-oracle, (σ^*, M_β) to H_3-oracle and σ^* to H_4-oracle (either in Phase 1 or Phase 2), then $\mathcal{B}$ actually assigns two different value for the same input with overwhelming probability, which goes against the definition of random oracle model and makes simulation distinguishable from real attack.
- **Issue 3**. In the simulation of decryption oracle, $\mathcal{B}$ answers the decryption queries by searching all H_i-lists. Let alone the low efficiency it causes, the authors think for every valid ciphertext C, there must have existed corresponding H_i-queries records in H_i-lists. In other words, they think it is impossible (with probability less than $1/2^n$) for an attacker to obtain a valid ciphertext without making corresponding queries. We have to argue that this hypothesis is too strong. In real attack, it is easy for an attacker to obtain some valid ciphertexts by eavesdropping.

These issues make their proof not infallible.

We summarise the guidelines that a valid IND-ID-CCA proof should follows.

- The construction of the simulator should be general. More exactly, the simulation algorithm should be independent of the adversary's particular behavior, such as the exact number random oracle queries, extraction/decryption queries an adversary makes in Phase 1 and Phase 2.
- The adversary must be handled strictly according to the definition of IND-ID-CCA game, no extra hypothesis should be imposed on it, such as how does the adver-

sary choose the target identity? From which set or the choices comply to what probabilistic distribution?

– The simulation algorithm must ensure the probability of perfect simulation to be computable, thereby the advantage of the simulator against the underlying problem is measurable. Otherwise the security reduction is meaningless.
– The simulation should be identical to real attack in the adversary view. In the random oracle model, the simulator should simulate all the random oracle models coherently.

6 IND-sID-CCA implies IND-ID-CCA

This section shows that if imposing a little constraint to $\mathsf{FullIdent}$, then we can obtain its fully security based on its *selective*-ID security.

In [22] Boneh and Boyen proved the following theorem which quantifies the relationship between *selective*-ID IBE security and fully IBE security in the random oracle model.

Theorem 6.1 *Let $\mathcal{E}$ be a (t, q_E, ϵ) selective-ID secure IBE. Suppose identities in $\mathcal{E}$ are n-bits long. Let H be a hash function $H : \{0,1\}^* \to \{0,1\}^n$ modeled as a random oracle. H converts $\mathcal{E}$ to $\mathcal{E}_H$ by the process of hashing the identity ID with H before using ID. Then $\mathcal{E}_H$ is a (t, q_E, ϵ') fully secure IBE (in the random oracle model) for $\epsilon' \approx q_H \cdot \epsilon$, where q_H is the maximum number of oracle calls to H that the adversary can make.*

This theorem inspires us to prove $\mathsf{IND\text{-}ID\text{-}CCA}$ security via $\mathsf{IND\text{-}sID\text{-}CCA}$ security. Next we first prove that BF-IBE in secure in *selective*-ID model, then achieve the security in the full model by applying Theorem 6.1.

Theorem 6.2 *Let H_1 be a random oracle. Then $\mathsf{FullIdent}$ is $\mathsf{IND\text{-}sID\text{-}CCA}$ secure assuming the CBDH assumption is hard in groups generated by $\mathcal{G}$. Concretely, suppose there is an $\mathsf{IND\text{-}sID\text{-}CCA}$ adversary $\mathcal{A}$ that has advantage $\epsilon(k)$ against the $\mathsf{FullIdent}$. Then there is an $\mathsf{IND\text{-}CCA}$ adversary $\mathcal{A}_3$ that has advantage $\epsilon(k)$ against $\mathsf{BasicPub}^{hy}$. Its running time is $O(time(\mathcal{A}))$.*

Proof. We construct an $\mathsf{IND\text{-}CCA}$ adversary $\mathcal{A}_3$ that uses $\mathcal{A}$ to gain advantage against $\mathsf{BasicPub}^{hy}$. The game starts with the challenger first generates the public key $K_{pub} = \langle q, \mathbb{G}_1, \mathbb{G}_2, e, n, P, P_{pub}, Q_{\mathsf{ID}}, H_2, H_3, H_4 \rangle$ and a private key $d_{\mathsf{ID}} = sQ_{\mathsf{ID}}$. The challenger gives K_{pub} to algorithm $\mathcal{A}_3$. $\mathcal{A}_3$ mounts an $\mathsf{IND\text{-}CCA}$ attack on the the key K_{pub} using the help of algorithm $\mathcal{A}$. $\mathcal{A}_3$ interacts with $\mathcal{A}$ as follows.

Init. $\mathcal{A}$ outputs an identity ID_{ch} where it wishes to be challenged.

Setup. Same as BF-IBE's.

H_1-**queries**. To respond to H_1 queries, $\mathcal{A}_3$ maintains a list of tuples $\langle \mathsf{ID}_i, Q_i, b_i \rangle$ which is referred as H_1^{list}. The list is initially empty. When $\mathcal{A}$ queries H_1 at a point ID_i, $\mathcal{A}_3$ responds as follows:

1. If the query ID_i already appears on the H_1^{list} in a tuple $\langle \mathsf{ID}_i, Q_i, b_i \rangle$ then $\mathcal{A}_3$ responds with $H_1(\mathsf{ID}_i) = Q_i$.
2. Otherwise, if $\mathsf{ID}_i = \mathsf{ID}_{ch}$, $\mathcal{A}_3$ sets $b_i = *$ and $Q_i = Q_{\mathsf{ID}}$; else $\mathcal{A}_3$ generates a random $b_i \in \mathbb{Z}_q^*$ and computes $Q_i = b_i P$.
3. $\mathcal{A}_3$ adds the tuple $\langle \mathsf{ID}_i, Q_i, b_i \rangle$ to H_1^{list} and responds to $\mathcal{A}$ with $H_1(\mathsf{ID}_i) = Q_i$.

Phase 1 - Extraction queries. When $\mathcal{A}$ asks for the private key associate to ID_i, $\mathcal{A}_3$ runs the above algorithm and gets $H_1(\mathsf{ID}_i) = Q_i$, where $\langle \mathsf{ID}_i, Q_i, b_i \rangle$ is the corresponding entry in H_1^{list}. Observing that $Q_i = b_i P$, therefore the corresponding private key is $d_i = b_i P_{pub}$. Finally, $\mathcal{A}_3$ gives d_i to $\mathcal{A}$. The request $\langle \mathsf{ID}_{ch} \rangle$ will be denied.

Phase 1 - Decryption queries. Let $\langle \mathsf{ID}_i, C_i \rangle$ be a decryption query issued by algorithm $\mathcal{A}$. Let $C_i = \langle U_i, V_i, W_i \rangle$. When $\mathsf{ID}_i \neq \mathsf{ID}_{ch}$, $\mathcal{A}_3$ runs H_1-queries algorithm and let $\langle \mathsf{ID}_i, Q_i, b_i \rangle \in H_1^{list}$, then retrieves the private key d_i and decrypts C_i using the decryption algorithm. If $\mathsf{ID}_i = \mathsf{ID}_{ch}$, $\mathcal{A}_3$ relays the decryption query with the ciphertext $\langle U_i, V_i, W_i \rangle$ to the challenger and relays the challenger's response back to $\mathcal{A}$.

Challenge. Once $\mathcal{A}$ decides that Phase 1 is over and outputs two messages M_0, M_1 which it wishes to be challenged on. $\mathcal{A}_3$ responds as follows: first $\mathcal{A}_3$ gives its challenger the message M_0, M_1. The challenger responds with a $\mathsf{BasicPub}^{hy}$ ciphertext $C = \langle U, V, W \rangle$ such that C is the encryption of M_β for a random $\beta \in \{0, 1\}$. Next, $\mathcal{A}_3$ responds to $\mathcal{A}$ with the challenge C.

Phase 2 - Private key queries. $\mathcal{A}_3$ responds to the extraction queries in the same way as it did in Phase 1.

Phase 2 - Decryption queries. $\mathcal{A}_3$ responds to the decryption queries in the same way as it did in Phase 1 except that $\langle \mathsf{ID}_i, C_i \rangle = \langle \mathsf{ID}_{ch}, C \rangle$ is denied.

Guess. Eventually, adversary $\mathcal{A}$ outputs a guess β' for β. Algorithm $\mathcal{A}_3$ outputs β' as its guess for β.

All the responses to H_1-queries are as in real attack since each response is uniformly and independently distributed in $\mathbb{G}_1^*$. All the responses to private key extraction queries and decryption queries are valid. So algorithm $\mathcal{A}_3$ would not abort during the simulation. By definition of algorithm $\mathcal{A}$, we have that $\left|\Pr[c = c'] - \frac{1}{2}\right| \geq \epsilon(k)$. Note that $\Pr[\mathcal{A}_3 \text{ does not abort}] = 1$, this shows that $\mathcal{A}_3$'s advantage against $\mathsf{BasicPub}^{hy}$ is at least $\epsilon(k)$ as required. $\square$

As to BF-IBE's $\mathsf{FullIdent}$ scheme, if we first hash arbitrary identities in $\{0,1\}^*$ to binary strings of length n using a collision resistant function with n-bits output (such as SHA-1 whose output is 160 bits). Taking $n = 160$ as the length of identities in IBE system is a natural choice. We denote the resulting scheme as $\mathsf{FullIdent}_H$. We note that the *selective*-ID security of an IBE system is not weakened if additional restrictions on the identities are imposed (indeed, this only tightens the constraints on the adversary and relaxes those on the simulator). Thus $\mathsf{FullIdent}_H$ is also *selective*-ID secure according to Theorem 6.2. Finally, as a straightforward result of Theorem 6.1, we conclude $\mathsf{FullIdent}_H$ is fully secure in the random oracle model.

Remark 2. This proof works but is not a satisfying one, because we prove it by imposing a constraint to the original scheme.

7 The New proof of BF-IBE

The IND-ID-CCA security of BF-IBE was proven via Reduction 4, Reduction 3 and Reduction1 in the original paper [8] [1] [2]. However, this happens to be the reason why their proofs failed: the IND-ID-CCA security of FullIdent and the IND-CCA security of $\mathsf{BasicPub}^{hy}$ are not meaningfully linked. Zhang and Imai [3] realized this and tried to reduce the security directly to the underlying hard problem. Unfortunately, they failed because they cannot answer all the queries coherently. Inspired by the proof technique used in [15], we can use decisional oracle $\mathcal{O}_{DBDH}(\cdot)$ to ensure all the responses to queries coherent.

In this section, we give a new proof of BF-IBE based on the GBDH assumption in the random oracle model. We directly reduce the security of BF-IBE to the intractability of GBDH problem and only require H_1, H_2, H_3 to be random oracles.

Theorem 7.1 *Let the hash functions H_1, H_2 and H_3 be random oracles. Then* FullIdent *is chosen ciphertext secure assuming GBDH is hard in groups generated by $\mathcal{G}$. Concretely, suppose there is an* IND-ID-CCA *adversary $\mathcal{A}$ that has advantage $\epsilon(k)$ and $\mathcal{A}$ makes at most q_E extraction queries, at most q_D decryption queires, and at most q_{H_i} queries to H_i oracles, respectively. Then there is a GBDH algorithm $\mathcal{B}$ has advantage*

$$Adv_{\mathcal{B}}(k) \geq \frac{\epsilon(k)}{e(1+q_E)}\left(1 - \frac{q_{H_3}}{2^n}\right)$$

in running time $O(time(\mathcal{A}))$.

Here e is the base of natural logarithm, n is the message size. Our aim is construct a GBDH adversary $\mathcal{B}$ with the help of an IND-ID-CCA adversary $\mathcal{A}$.

Proof. Suppose $\mathcal{B}$ is given a instance $(P, aP, bP, cP, \mathcal{O}_{DBDH})$ of the GBDH problem where $\mathcal{O}_{DBDH}(\cdot)$ is a decisional oracle to judge whether (P, aP, bP, cP, Z) is a valid BDH tuple. $\mathcal{B}$ is expected to output $T \in \mathbb{G}_2$ satisfying $T = e(P, P)^{abc}$.

Setup. $\mathcal{B}$ gives $\mathcal{A}$ $\mathsf{params} = \langle q, \mathbb{G}_1, \mathbb{G}_2, e, n, P, P_{pub}, H_1, H_2, H_3, H_4\rangle$ as the system parameters, where n is the length of plaintext, and H_1, H_2 and H_3 are random oracles controlled by $\mathcal{B}$. $\mathcal{B}$ sets P_{pub} as aP.

Phase 1- H_1 queries. $\mathcal{B}$ maintains a list L_1 which contains tuples $(\mathsf{ID}_j, Q_j, s_j, coin_j)$. When a query $\langle \mathsf{ID}_i \rangle$ comes, if there is already an entry $(\mathsf{ID}_i, Q_i, s_i, coin_i)$ in L_1, $\mathcal{B}$ replies it with Q_i. Otherwise, $\mathcal{B}$ flips a biased coin with $\Pr[coin = 0] = \delta$ (δ will be decided later), picks a random $s \in \mathbb{Z}_q^*$; if $coin = 0$ computes $Q_i = sP$, else computes $Q = sbP$. $\mathcal{B}$ adds the tuple $(\mathsf{ID}_i, Q_i, s, coin)$ to the L_1 and responds to $\mathcal{A}$ with $H_1(\mathsf{ID}_i) = Q_i$.

Phase 1- H_2 queries. H_2 hashes an element $\omega \in \mathbb{G}_2$ to a value $h \in \{0,1\}^n$. According to the proof technique already used in [14] [15], these queries are processed using two lists L_2 and L_2' which are initially empty:

- L_2 contains tuples (ω, h_2) which indicates a hash value $h_2 \in \{0,1\}^n$ was previously assigned to ω.

- L'_2 contains tuples (Q, U, ω^*, h'_2) which means $\mathcal{B}$ has implicitly assigned a hash value $h'_2 \in \{0,1\}^n$ to some ω^* satisfying $\mathcal{O}_{DBDH}(P, P_{pub}, Q, U, \omega^*) = 1$, although ω^* is unknown yet.

More precisely, when $\mathcal{A}$ submits a query ω to $H_2(\cdot)$,

- $\mathcal{B}$ first checks if there is an entry (ω, h_2) in L_2 list. If it does, h_2 is returned to $\mathcal{A}$.
- Else, for every tuple (Q, U) in L'_2, $\mathcal{B}$ submits $(P, P_{pub}, Q, U, \omega)$ to the $\mathcal{O}_{DBDH}(\cdot)$ oracle to decide whether it is a valid BDH tuple. If it is for some existing entry (Q, U, ω^*, h_2), $\mathcal{B}$ adds (ω, h_2) to L_2 and deletes the entry from L'_2. ($\mathcal{B}$ processes in this way in order to behave coherently. Otherwise $\mathcal{B}$ will run a risk of explicitly assigning two different h_2 for the same ω.) If there is no such entry in L'_2 satisfying $(P, P_{pub}, Q, U, \omega)$ is a valid BDH tuple, $\mathcal{B}$ assigns a random $h_2 \in \{0,1\}^n$ to ω, adds (ω, h_2) into L_2. At last, $\mathcal{B}$ returns h_2 to $\mathcal{A}$.

Phase 1- H_3 queries. $\mathcal{B}$ maintains a list contains tuples (σ, M, h_3). We refer to the list as L_3, which is initially empty. When a query (σ, M) comes, if there is an entry (σ, M, h_3) on H_3 list, $\mathcal{B}$ returns h_3 to $\mathcal{A}$; otherwise, $\mathcal{B}$ choose $h_3 \in \mathbb{Z}_q^*$, returns h_3 to $\mathcal{A}$ and adds (σ, M, h_3) to L_3.

Phase 1- Private key queries. When a private key query $\langle \mathsf{ID}_i \rangle$ comes (we can assume ID has already in L_1 list), B find the corresponding tuple $(\mathsf{ID}_i, Q_i, s_i, coin_i)$ in L_1. If $coin_i = 1$, $\mathcal{B}$ reports "abort" and quits the simulation. If $coin_i = 0$, $\mathcal{B}$ sets $d_i = aQ_i = s_i P_{pub} = s_i aP$ which is a valid private key for ID_i, and then returns d_i to $\mathcal{A}$.

Phase 1- Decryption queries. When a query (ID, C) comes. $\mathcal{B}$ searches in L_1 for $Q = H_1(\mathsf{ID})$.

- If the associated $coin = 0$, $\mathcal{B}$ obtains the private key for ID. Then use the private key to respond to the decryption query.
- If $coin = 1$, $\mathcal{B}$ searches ω in L_2 which satisfying $\mathcal{O}(P, P_{pub}, Q, U, \omega) = 1$. If ω_j is such an entry, computes $\sigma = V \oplus h_{2,j}$ ($h_{2,j} = H_2(\omega_j)$) and responds the query according to the decryption algorithm. If there isn't, for every entry (Q_i, U_i) in L'_2, $\mathcal{B}$ checks whether $e(Q, U) = e(Q_i, U_i)$. If (Q_j, U_j) is such an entry, computes $\sigma = V \oplus h'_{2,j}$ and responds the query according to the decryption algorithm. ($e(Q, U) = e(Q_i, U_i)$ indicates that the underlying ω is same, for $e^3(P_{pub}, Q, U) = e^3(P_{pub}, Q_i, U_i)$. The notation e^3 is defined as $e^3(aP, bP, cP) = e(P, P)^{abc}$.)
- Otherwise, $\mathcal{B}$ randomly chooses a $h'_2 \in \{0,1\}^n$ and adds (Q, U, ω^*, h'_2) in L'_2. ω^* is an unknown value which satisfies $\mathcal{O}_{DBDH}(P, P_{pub}, Q, U, \omega^*) = 1$. $\mathcal{B}$ computes $\sigma = V \oplus h'_2$ and carries on the decryption algorithm to respond the query.

  (In this way, $\mathcal{B}$ can always answers the decryption queries coherently.)

Challenge. Once $\mathcal{A}$ decides that Phase 1 is over it outputs two messages M_0, M_1 and an target identity ID_{ch} on which it wishes to be challenged. Let $(\mathsf{ID}, Q, s, coin)$ be the corresponding entry in L_1. If $coin = 0$, $\mathcal{B}$ aborts and reports "failure", because $\mathcal{A}$ is of no help in $\mathcal{B}$'s endeavor in such a situation. Otherwise, let $\beta \in \{0,1\}$ be a random bit, $\mathcal{B}$ sets $U = cP$, picks a random $\sigma^* \in \{0,1\}^n$ which is not in the current L_3 list, thus implicitly implies $H_3(\sigma^*, M_\beta) = c$, although c is unknown. In order to simulate perfectly, $\mathcal{B}$ obtains the hash value of $H_2(e(P_{pub}, Q)^c)$ in the following steps.

- Check whether L_2 contains an entry which satisfies $\mathcal{O}_{OBDH}(P, P_{pub}, Q, U, \omega_j) = 1$. If it does, set the hash value of $H_2(e(P_{pub}, Q)^c)$ as $h_{2,j} = H_2(\omega_j)$.
- Else check whether L'_2 contains an entry satisfying $e(Q_j, U_j) = e(Q, U)$. If it does, set the hash value of $H_2(e(P_{pub}, Q)^c)$ as $h'_{2,j} = H_2(\omega^*_j)$.
- Otherwise, $\mathcal{B}$ chooses a random $h'_2 \in \{0,1\}^n$ and adds (Q, U, ω^*, h'_2) into L'_2. Set the hash value of $H_2(e(P_{pub}, Q)^c)$ as h'_2.

$\mathcal{B}$ computes $V = M_\beta \oplus H_2(e(P_{pub}, Q)^c)$ and $W = M \oplus H_4(\sigma^*)$. Finally, $\mathcal{B}$ responds $\mathcal{A}$ with ciphertext $C = \langle U, V, W \rangle$.

Phase 2- Private key queries. $\mathcal{B}$ responds to private key queries in the same way as it did in Phase 1 except disallowing the query $\langle \mathsf{ID}_{ch} \rangle$.

Phase 2- Decryption queries. $\mathcal{B}$ responds to decryption queries in the same way as it did in Phase 1 except disallowing the query $\langle \mathsf{ID}_{ch}, C \rangle$.

Phase 2- H_i queries. $\mathcal{B}$ responds to H_1 and H_2 queries identically as it did in Phase 1. For H_3-oracle, when $\mathcal{B}$ comes with a query $(\sigma, M) = (\sigma^*, M_\beta)$, it reports "failure" and terminates. (The reason of $\mathcal{B}$ has to abort in this case is it does not know the value $r = H_3(\sigma^*, M_\beta)$). Else $\mathcal{B}$ responds to H_3 queries the same way as it did in Phase 1.

We denotes the event that $\mathcal{A}$ issues (σ^*, M_β) query to H_3 oracle as AskH_3.

Guess. Eventually $\mathcal{A}$ outputs a guess β' for β, then $\mathcal{B}$ terminates the IND-ID-CCA game.

When the game between $\mathcal{A}$ and $\mathcal{B}$ terminates, no matter what the reason is, $\mathcal{B}$ searches the entry (ω, h_2) in L_2 which satisfying $\mathcal{O}_{DBDH}(P, P_{pub}, Q, U, \omega) = 1$ and computes $(\omega)^{s^{-1}}$ as its answer to the GBDH problem. It is easy to verify the correctness observing that $\omega = e(d_{\mathsf{ID}}, U) = e(sabP, cP) = e(P, P)^{abcs}$.

Claim. We denotes the event that $\mathcal{A}$ issues $e(d_{\mathsf{ID}}, U)$ query to H_2-oracle as AskH_2. From the above analysis we know that as soon as AskH_2 occurs, the attack to GBDH problem succeeds. If algorithm $\mathcal{B}$ does not abort during the simulation before AskH_2 occurs then $\mathcal{A}$'s view is identical to its view in the real attack, because $\mathcal{B}$ simulates H_i-oracles coherently and all the responses to extraction queries and decryption queries are valid. On the other hand, $\mathcal{A}$ guesses the right $\beta' = \beta$ means AskH_2 must have occured. Therefore, according to the definition of $\mathcal{A}$, the probability of $\mathcal{B}$ finding the wanted tuple (ω, h_2) in L_2 is at least ϵ.

Note that different from other similar proofs, $\mathcal{B}$ can gain the advantage against the underlying GBDH problem even before $\mathcal{A}$ outputs its final guess. It suffices to compute the probability of $\mathcal{B}$ does not abort before AskH_2 occurs. We denote such probability as $\mathsf{Pr}[\mathsf{Success}]$.

$\mathcal{B}$ may terminates before AskH_2 occurs for the following three events.

1. $\mathcal{E}_1$ is the event that $\mathcal{A}$ issues private key queries while the corresponding $coin = 1$ during Phase 1 and Phase 2.
2. $\mathcal{E}_2$ is the event that $\mathcal{A}$ chooses the target ID_{ch} while the corresponding $coin = 0$ in the Challenge phase.
3. $\mathcal{E}_3$ is the event that AskH_3 occurs before AskH_2 occurs. ($\mathcal{B}$ is unable to extract the underlying hash value $r = c$).

According to the decryption algorithm, if AskH_2 happens, the corresponding AskH_3 follows with high probability. On the contrary, the probability of that AskH_3 happens before the AskH_2 is less than $q_{H_3}/2^n$, where 2^n is the cardinal of σ space. Because the chance that a random string σ equals to σ^* is at most $1/2^n$ and this happens at most q_{H_3} times. Note that AskH_3 will lead to the termination of the game, but AskH_2 has occurred before it with overwhelmingly probability.

Combines all above, the probability of perfect simulation before AskH_2 occurs is

$$\Pr[\mathsf{Success}] = \Pr[\neg\mathcal{E}_1 \wedge \neg\mathcal{E}_2 \wedge \neg\mathcal{E}_3] = \delta^{q_E}(1-\delta)\left(1 - \frac{q_{H_3}}{2^n}\right)$$

Using the same mathematical technique in [8], the lower bound is maximized at $\delta_{opt} = 1 - 1/(q_E + 1)$, thus

$$\Pr[\mathsf{Success}] \geq \frac{1}{e(1+q_E)}\left(1 - \frac{q_{H_3}}{2^n}\right)$$

The bound on time complexity can be verified easily. This proves the result as required. □

In order to answer decryption queries coherently, $\mathcal{B}$ has to call the $\mathcal{O}_{DBDH}(\cdot)$-oracle at most $q_D q_{H_2}$ times. In order to return a proper and valid ciphertext, $\mathcal{B}$ has to call the $\mathcal{O}_{DBDH}(\cdot)$-oracle at most q_{H_2} times. If we add (Q, U) as two extra inputs to H_2 function, i.e. replace $H_2(e(P_{pub}, Q_{\mathsf{ID}})^r)$ with $H_2(Q, U, e(P_{pub}, Q_{\mathsf{ID}})^r))$ in the encryption algorithm and replace $H_2(e(d_{\mathsf{ID}}, U))$ as $H_2(Q, U, e(d_{\mathsf{ID}}, U))$ in the decryption algorithm, we can save $(q_{H_2} + q_D q_{H_2})$ times call to $\mathcal{O}_{DBDH}(\cdot)$-oracle. A similar observation was made by Cramer and Shoup [14] in their security proof of the Hashed ElGamal KEM.

8 Conclusion

In this paper, we point out the flaws in some previous proofs of BF-IBE. We notice that by restricting all the identities of BF-IBE are n-bits long, we can prove its full security based on its *selective*-ID security. Besides, we give a new proof for BF-IBE based on the GBDH problem in the random oracle model. However, we think how to provide an elegant proof of BF-IBE relying on the original CBDH assumption is still a interesting problem.

References

1. David Galindo: Boneh-franklin identity based encryption revisited. Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005 Proceedings **3580** (2005) 791–802
2. Nishioka, M.: Reconsideration on the security of the boneh-franklin identity-based encryption scheme. Progress in Cryptology - INDOCRYPT 2005, 6th International Conference on Cryptology in India **3797** (2005) 270–282

3. Rui, Z., Imai, H.: Improvements on security proofs of some identity based encryption schemes. Information Security and Cryptology, First SKLOIS Conference **3822** (2005) 28–41
4. Shamir, A.: Identity-based cryptosystems and signatures schemes. Advances in Cryptology - Crypto 1984 **196** (1984) 47–53
5. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. Advances in Cryptology - CRYPTO 2001 **2139** (2001) 213–229
6. Clifford Cocks: An Indentity Based Encryption Scheme Based on Quadratic Residues. Institute of Mathematics and Its Applications International Conference on Cryptigraphy and Coding - Proceedinds of IMA 2001 **2260** (2001) 360–363
7. Ryuichi Sakai, Kiyoshi Ohgishi, M.K.: Cryptosystems based on pairing. The 2001 Symposium on Cryptography and Information Security, Japan **45** (2001) 26–28
8. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. SIAM J. of Computing **32** (2003) 586–615
9. Alriyami, S.S., Paterson, K.G., Holloway, R.: Certificateless public key cryptography. Advances in Cryptology - Asiacrypt 2003 **2894** (2003) 452–473
10. Boyen, X.: Multipurpose identity-based signcryption - a swiss army knife for identity-based cryptography. Advances in Cryptology - CRYPTO 2003 **2729** (2003) 383–399
11. Gentry, C.: Certificate-based encryption and the certificate revocation problem. **2656** (2003) 272–293
12. Gentry, C., Silverberg, A.: Hierarchical id-based cryptography. Advances in Cryptology - ASIACRYPT 2002 **2501** (2002) 548–566
13. Horwitz, J., Lynn, B.: Toward hierarchical identity-based encryption. Advances in Cryptology - Eurocrypt 2002 **2322** (2002)
14. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing **33** (2001) 167–226
15. Libert, B., Quisquater, J.J.: Identity based encryption without redundancy. Proceedings of ACNS 2005 **3531** 285–300
16. Joux, A.: A one round protocol for triparitite diffie-hellman. Algorithmic Number Theory, 4th International Symposium **1838** (2000) 385–394
17. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. Advances in Cryptology - Eurocrypt 2003 **2656** (2003) 255–271
18. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identitybased encryption. Advances in Cryptology - Eurocrypt 2004 **3027** (2004) 207–222
19. Eiichiro Fujisaki and Tatsuaki Okamoto: Secure integration of asymmetric and symmetric encryption schemes. (1999) 537–554
20. Coron, J.S.: On the exact security of full domain hash. CRYPTO 2000 **1880** (2000) 229–235
21. Waters, B.: Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. CRYPTO '09: Proceedings of the 29th Annual International Cryptology Conference on Advances in Cryptology (2009) 619–636
22. Boneh, D., Boyen, X.: Efficient selective-id secure identity based encryption without random oracles. Proceedings of Eurocrypt 2004 **3027** (2004) 223–238