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
cycosmic@gmail.com

Abstract. Boneh and Franklin constructed the first practical Identity-Based Encryption scheme (BF-IBE) [1] and proved its security based on the computational Bilinear Diffie-Hellman assumption (CBDH) in 2001. The correctness of its security proof was long believed to be correct until in 2005, Galindo [2] noticed a flawed step in the original proof. In the same paper, Galindo provided a new proof with a looser security reduction. Shortly afterwards, Nishioka [3] improved Galindo's proof to achieve a tighter security reduction. In the same year, Zhang and Imai [4] gave another proof of BF-IBE. Unfortunately, we find that none of their proofs is flawless. In this paper, besides identifying and fixing the lapses in previous proofs, we present two new proofs for the CCA security of BF-IBE. The first proof is proved via selective-identity security with imposing a natural constraint to the original scheme. The second proof is proved by directly reducing the security to a stronger assumption, namely the gap Bilinear Diffie-Hellman (GBDH) assumption.

Key words: IBE, provable security, security reduction, BDH assumption

1 Introduction

Since Shamir proposed the concept of IBE in 1984 [5], 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 [1], Cocks [6] and Sakai *et al.* [7] presented three IBE schemes in 2001, respectively. Among those solutions, Boneh and Franklin's one happens 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, namely indistinguishable chosen ciphertext attack for ID-based encryption (IND-ID-CCA), and 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. Numerous schemes [9] [10] [11] [12] [13] are based on BF-IBE.

The original proof of BF-IBE was never challenged until Galindo [2] pointed out a flawed step in the reduction for CCA security in 2005. 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 could be sacrificed. Subsequently, Nishioka [3] extended Galindo's idea to provide another proof, which can be viewed as a improved version of Galindo's proof. In the same year, Zhang and Imai [4] 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 contribution

Justify previous proofs of BF-IBE. We first re-examine the flawed steps in BF-IBE's original CCA security proof exhibited in [8] and analyze why it failed, then point out the lapses in the subsequent revised security proofs proposed by Galindo [2], Nishioka [3] and Zhang *et.al* [4], respectively. Galindo's proof and Nishioka's proof are similar. Compared to the original proof in [1], their simulation algorithm are straightforward. However, the simulation algorithm in Galindo's proof is not well defined itself. Nishioka improved Galindo's proof, but the probability of successful simulation is wrong. Zhang and Imai [4] reduced the security directly to the complexity assumption by making the simulator simulates itself all the oracles: the H_i oracles, extraction oracle, encryption oracle and decryption oracle. The issue is that Lemma 2 and Lemma 3 [4] are not accurate. After identifying the lapses in these proofs, we also show how to fix them.

Present two new proofs of BF-IBE. Inspired by the theorem proved by Boneh and Boyen [14], we prove the full security of BF-IBE via *selective*-ID security. The only issue is that we need to impose a constraint to the original scheme, i.e. identities must be bitstrings of a fixed length, not of arbitrary length. However, it is a natural requirement for practical use. On the other hand, we provide an elegant proof with tighter security reduction of BF-IBE which employs the proof technique used in [15] [16]. This new proof reduces the CCA security of BF-IBE directly to the underlying complex assumption without intermediate steps. We remark that our second proof is based on the gap Bilinear Diffie-Hellman (GBDH) assumption, which is stronger than the computational Bilinear Diffie-Hellman (CBDH) assumption used in the original proof.

Note. We remark that all the proofs in this paper are in the random oracle model [17]. In order to keep the consistency of notation, we replace some symbols and variables when quoting the related references. However, they can be easily known by comparing our description with the original papers [1] [2] [3] [4].

2 Complexity Assumptions

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

Computational Bilinear Diffie-Hellman Problem (CBDH). The CBDH problem [18] [1] 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). Given a CBDH challenge (P, aP, bP, cP), to compute $e(P, P)^{abc}$ with the help of a DBDH oracle.

3 Analysis of Reduction 4 in BF-IBE

We give some necessary knowledge of bilinear maps in Appendix A. The related security notion for IBE can be found in Appendix B. A brief review of BF-IBE (the definitions of FullIdent, BasicIdent, BasicPub, BasicPub^{hy}) and the original proofs (Reduction i) can be found in Appendix C. In this section, we re-examine the Reduction 4 for CCA security of FullIdent.

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

Issue 1. In Phase 1, when $\mathcal{A}$ issues a decryption query $\langle \mathsf{ID}, C\rangle$, where $C = \langle U, V, W\rangle = \langle rP, \sigma \oplus H_2(e(Q, P_{pub})^r), M \oplus H_4(\sigma)\rangle$. According to the simulation algorithm, if $coin = 1$ ($Q = bQ_{\mathsf{ID}}$), $\mathcal{B}$ modifies C as $C' = \langle U', V', W'\rangle = \langle bU, V, W\rangle$ and then relays C' to its challenger. When the challenger decrypts C' using the private key d_{ID}, it does:

1. Compute $V' \oplus H_2(e(d_{\mathsf{ID}}, U')) = V \oplus H_2(e(d_{\mathsf{ID}}, bU)) = \sigma \oplus H_2(e(Q, P_{pub})^r) \oplus H_2(e(sQ_{\mathsf{ID}}, brP)) = \sigma$. This step recovers the random chosen $\sigma \in \{0,1\}^n$ exactly.
2. Compute $W' \oplus H_4(\sigma) = W \oplus H_4(\sigma) = M \oplus H_4(\sigma) \oplus H_4(\sigma) = M$. This step recovers the original plaintext M exactly.
3. Set $r = H_3(\sigma, M)$ and test whether $U' = rP$. Note that b is randomly chosen from $\mathbb{Z}_q^*$ which means the probability of $r = r'$ ($r' = br$) is $1/q$. Thereby the challenger will reject the modified ciphertext with overwhelming probability.

Therefore, $\mathcal{B}$ can not employ the decryption oracle of BasicPub^{hy} to answer the decryption queries issued by $\mathcal{A}$ if the corresponding $coin = 1$.

Issue 2. In the challenge phase, $\mathcal{A}$ outputs ID_{ch} and M_0, M_1 on which it wishes to be challenged. $\mathcal{B}$ gives its challenger M_0, M_1 as the messages that it wishes to be challenged on. The challenger gives $\mathcal{B}$ the ciphertext $C = \langle U, V, W\rangle = \langle rP, \sigma \oplus H_2(e(Q_{\mathsf{ID}}, P_{pub})^r), M_\beta \oplus H_4(\sigma)\rangle$ such that C is the ciphertext 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 $coin = 0$, $\mathcal{B}$ aborts the game and the attack fails. Otherwise, $\mathcal{B}$ will modify C to $C' = \langle U', V', W'\rangle = \langle b^{-1}U, V, W\rangle$ and relay C' to $\mathcal{A}$ as the challenge ciphertext. Boneh and Franklin claimed that C' is also a proper FullIdent ciphertext of M_β under the public key $\mathsf{ID}_{ch} = Q = bQ_{\mathsf{ID}}$. However, if C' is a valid ciphertext of M_β, 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 FullIdent ciphertext of M_β is $1/q$. $\mathcal{A}$ will reject the modified ciphertext with overwhelming probability.

These two issues render the Reduction 4 in the original proof invalid. By the way, Galindo only pointed out Issue 1 in [2].

3.1 Flipping coin technique

Boneh and Franklin borrowed the technique from Coron's analysis of the Full Domain Hash signature scheme [19], which we refer to it as flipping coin technique. More precisely, $\mathcal{B}$ answers H_1-queries according to the result of flipping a $coin \in \{0,1\}$. If $coin = 0$, return $Q_i = b_i P$. If $coin = 1$, return $Q_i = b_i Q_{\mathsf{ID}}$. Note

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

"Flipping coin" can be viewed as a concrete technique of *Partitioning Reduction* methodology which was generalized by Waters [20]. The idea of *Partitioning Reduction* is 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. Boneh and Franklin successfully use "flipping coin" technique to prove the CPA security of BasicIdent (Reduction 2). But this technique can not be applied to Reduction 4. Because in FullIdent, Fujisaki and Okamoto transformation [21] removes the homomorphic relationship between the ciphertext and its corresponding key, malformed ciphertext (modified by simulator) will be rejected with overwhelming probability. Thus "flipping coin" technique cannot partition the identity space favorably as it does in proving Reduction 2.

4 Analysis of Galindo and Nishioka's Proofs

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

4.1 Galindo's proof

Galindo [2] tried to fix Reduction 4 by modifying the simulation algorithm of random oracle H_1. We briefly show his revised proof of Reduction 4 as follows.
Setup. Same as in BF-IBE.

H_1-**queries**. Before initializing H_1^{list}, $\mathcal{B}$ selects a random $j \leftarrow \{1, \ldots, q_{H_1}\}$. When $\mathcal{A}$ queries H_1 at ID_i, $\mathcal{B}$ 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, $\mathcal{B}$ sends Q_i to $\mathcal{A}$.

Phase 1 - Private key queries. When $\mathcal{A}$ asks for the private key of ID_i, $\mathcal{B}$ 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 $\mathcal{B}$ aborts the game. Otherwise, it sets $d_i = b_i P_{pub}$. Finally, $\mathcal{B}$ gives d_i to $\mathcal{A}$.

Phase 1 - Decryption queries. $\mathcal{B}$ answers to the 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 $\mathcal{B}$ 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 $\mathcal{B}$ asks its challenger to decrypt C_j and relays the answer to $\mathcal{A}$.

Challenge. $\mathcal{A}$ outputs the target identity ID_{ch} and two messages M_0, M_1 on which it wishes to be challenged. $\mathcal{B}$ 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 ciphertext of M_β for a random bit $\beta \in \{0, 1\}$ under $\mathsf{BasicPub}^{hy}$. Finally, $\mathcal{B}$ relays C to $\mathcal{A}$, which is an also valid FullIdent ciphertext of M_β under ID_{ch}.

The **Phase 2** and **Guess** phase are same as in BF-IBE.
In the game $\mathcal{B}$ may aborts due to 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 $\mathcal{B}$ will not abort in Phase 2, since $\mathcal{A}$ is not allowed to query the private key for $\mathsf{ID}_{ch} = \mathsf{ID}_j$ in Phase 2.

Let $\mathcal{E}_1$ be the event that $\mathcal{B}$ aborts due to (1), $\mathcal{E}_2$ be the event that $\mathcal{B}$ aborts due to (2). The probability that $\mathcal{B}$ 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}, where q_E is the maximum number of private extraction queries; 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 $1/q_{H_1}$. He concluded that $\Pr[\mathcal{B}$ does not abort$] \geq 1/q_{H_1} \cdot (1 - q_E/q_{H_1})$.

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

Issue 1. In the challenge phase, 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 in different simulations, so whether "ID_j" exists is unknown. Thus the probability of "$\mathcal{B}$ does not abort" is immeasurable. For example, suppose $\mathcal{B}$ picks $j = 10$ in the beginning of the simulation while $\mathcal{A}$ only issues three H_1 queries in Phase 1, then the so called ID_j does not even exist. At least, it is fair to say the simulation algorithm is not well defined.

Issue 2. Even Issue 1 could be ignored, here follows issue 2. 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 phase the adversary $\mathcal{A}$ randomly picks the target identity from the current H_1^{list}. This goes against the definition of IND-ID-CCA game which states that the target identity ID_{ch} can be chosen without any restriction, in particular outside the current H_1^{list}. Someone may argue that if $\mathcal{A}$ does not choose ID_{ch} from the current H_1^{list}, its advantage against the IND-ID-CCA game will be statistically closed to 0. Note that $\mathcal{A}$ could issue the H_1-query for ID_{ch} 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}. So $\Pr[\neg\mathcal{E}_2|\neg\mathcal{E}_1]$ is immeasurable.

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

4.2 Nishioka's proof

Realizing that there are some problems in Galindo's proof, Nishioka [3] gave an imporved proof of Reduction 4. Nishioka's proof is similar to Galindo's proof except three modifications. The first modification is in the simulation of H_1-oracle: in [2] $\mathcal{B}$ randomly selects $j \in \{1, \ldots, 1 + q_{H_1}\}$, while in [3] $\mathcal{B}$ randomly selects $j \in \{1, \ldots, 1 + q_{H_1} + q_D\}$. The second modification is in [3] $\mathcal{B}$ maintains two lists named as H_1^{list1} and H_1^{list2}, where $l_1 = \#H_1^{list1}$ and $l_2 = \#H_1^{list2}$. H_1^{list1} is used to record $\mathcal{B}$'s responses to H_1-queries and decryption queries, while H_1^{list2} is used to record $\mathcal{B}$'s responses to extraction queries. The third modification is that Nishioka refine the simulation algorithm in the challenge phase. The details of Nishioka's proof are omitted here. We remark that the last modification make the simulation algorithm well defined and the probability of successful computation computable.

In order to compute the probability that $\mathcal{B}$ does not abort during the simulation, Nishioka defines $\mathcal{E}_1$ as the event that $\mathcal{B}$ 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 Phase 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 $\langle j, \mathsf{ID}_j, Q, * \rangle$ on the H_1^{list1}. Nishioka claimed that $\Pr[\mathcal{B}$ 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 [3]). At last, he concludes the advantage against the CBDH problem is $2\epsilon/q_{H_2}(1 + q_{H_1} + q_D)$ (Equation (6) in Section 3.2 in [3]).

Issue 1. The computation of $\Pr[\mathcal{B} \text{ does not abort}] = \Pr[\neg\mathcal{E}_1 \wedge \neg\mathcal{E}_2] = \Pr[\neg\mathcal{E}_2] \geq 1/(1+q_{H_1}+q_D)$ is not correct. Nishioka negelected to count in $\mathcal{E}_1$. Actually,

$$\Pr[\mathcal{B} \text{ does not abort}] = \Pr[\neg\mathcal{E}_1 \wedge \neg\mathcal{E}_2] = \Pr[\neg\mathcal{E}_1]\Pr[\neg\mathcal{E}_2|\neg\mathcal{E}_1] \geq \left(1 - \frac{q_E}{1+q_{H_1}+q_D}\right) \cdot \frac{1}{1+q_{H_1}+q_D}$$

By the way, there is no proof for $\Pr[\neg\mathcal{E}_2|\neg\mathcal{E}_1] = 1/(1+q_{H_1}+q_D)$ in the original paper [3]. Considering this result is not as obvious as it looks like, we provide a strict proof for it in Appendix D.

Issue 2. For the reason of Issue 1, Equation (6) in [3] is also not accurate. Nishioka also described a tighter Reduction 3 in [3], but there is also no proof available in his paper. Here we still use the results of Reduction 3 and Reduction 1 in the original paper [8], we fix Equation (6) as

$$Adv_{\mathcal{B}} = \frac{\epsilon}{(1+q_{H_1}+q_D)q_{H_2}(q_{H_3}+q_{H_4})}\left(1 - \frac{q_E}{1+q_{H_1}+q_D}\right)$$

Apparently, this leads to a looser but not a tighter security reduction as the author claimed.

Remark 1. Galindo and Nishioka abandoned the "flipping coin" technique used in [8]. They used a straightforward simulation algorithm to partition the identity space. The efficiency of such partition is $\frac{1}{q_{H_1}}\left(1 - \frac{q_E}{q_{H_1}}\right)$, which also indicates the tightness of Reduction 4.

5 Zhang and Imai's proof

Zhang and Imai gave a new proof of the BF-IBE in [4]. They reduced the CCA security of FullIdent directly to the CBDH problem, not the IND-CCA security of $\mathsf{BasicPub}^{hy}$. We first have a glance at their proof.

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

Setup. Same as in BF-IBE.

H_1**-queries**. Same as in BF-IBE, except replacing 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 such an 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 $\langle \mathsf{ID}, C\rangle$ comes, where $C = \langle U, V, W\rangle$, $\mathcal{B}$ 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 $(\mathsf{ID}, M, r, t, \sigma)$ satisfies the below equations: 1) $Q_{\mathsf{ID}} = H_1(\mathsf{ID})$; 2) $r = H_3(\sigma, M)$ and $c_1 = rP$; 3) $t = e(P_{pub}, Q_{\mathsf{ID}})^r$ and $c_2 = \sigma \oplus H_2(t)$; 4) $c_3 = M \oplus H_4(\sigma)$. If there exists such an M and associated $(\mathsf{ID}, \sigma, r, t)$ in those lists, $\mathcal{B}$ returns M to $\mathcal{A}$ as its 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} be $(\mathsf{ID}, Q_{\mathsf{ID}}, s, coin)$. If $coin = 0$, $\mathcal{B}$ aborts the simulation. Otherwise, $\mathcal{B}$ chooses random $v^* \in \{0,1\}^n$, $\beta \in \{0,1\}$ and sets $U^* = s^{-1}P_3$, $V^* = M_\beta \oplus v^*$, $W^* = \{0,1\}^n$. $\mathcal{B}$ returns $C = \langle U^*, V^*, W^*\rangle$ to $\mathcal{A}$ as the challenge ciphertext.

In Phase 2, $\mathcal{B}$ responds to private key extraction queries and decryption queries the same way as it did in Phase 1. After $\mathcal{A}$ submits its answer, $\mathcal{B}$ 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 algorithm $\mathcal{B}$.

In Lemma 2 they claimed that the probability of perfect decryption oracle simulation is at least $(1-2^{-n})^{q_{H_3}}(1-2^{-n})^{q_{H_4}}(1-1/q)^{q_D}$.

Next, we point out the lapses in their proof.

Issue 1. First there are several typos and irregular notations in their poof. Here we just point out three important bugs. 1) The multiplicative inverse of s should be denoted as s^{-1}, not $-s$; 2) According to the encryption algorithm, $V^* = M_\beta \oplus v^*$, $W^* = \{0,1\}^n$ should be corrected as $V^* = \sigma^* \oplus H_2(g_{\mathsf{ID}}^{r^*})$ and $W^* = M_\beta \oplus \omega^*$; 3) Since the authors set $U^* = s^{-1}P_3$, then $\mathcal{B}$ should answers t as its answer to the CDBH problem but not $t^{s^{-1}}$. Because 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. In the challenge phase, Zhang and Imai generated the challenge ciphertext by implicitly assigning $H_2(e(P_{pub}, Q_{\mathsf{ID}})^{r^*})$ with a random $v^* \in \{0,1\}^*$ 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 $H_4(\sigma^*) = w^*$. Note that $e(P_{pub}, Q_{\mathsf{ID}})^{r^*}$ and σ^* are unknown to $\mathcal{B}$, thus $\mathcal{B}$' simulation for H_2, H_3 and H_4 may be not coherent through the game. If $\mathcal{A}$ explicitly issues 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), $\mathcal{B}$ run a risk of assigning two different values for the same input for H_2, H_3 and H_4 with overwhelming probability, which goes against the definition of random oracle and makes the simulation distinguishable in $\mathcal{A}$'s view from real attack. The result of Lemma 2 in [4] is not accurate for they neglected to count in the event of querying $e(P_{pub}, Q_{\mathsf{ID}})^{r^*}$ to H_2-oracle and didn't compute the probability of querying (σ^*, M_β) to H_3-oracle precisely. It should be corrected as $(1-1/q)^{q_{H_2}}(1-2^{-2n})^{q_{H_3}}(1-2^{-n})^{q_{H_4}}(1-1/q)^{q_D}$.

Issue 3. If $\mathcal{B}$ does not aborts during the simulation, then the probability of $\mathcal{B}$ succeeds in solving CBDH problem should be ϵ/q_{H_2} but not $1/(q_{H_2}+q_{H_3}+q_{H_4})$ in Lemma 3 [4].

Remark 2. The simulator algorithm cannot simulate coherently throughout the simulation (the answers to decryption queries and H_i queries may contradict each other), which not only lower the probability of successful simulation, but also raise the time complexity of simulation (in the simulation of decryption oracle, $\mathcal{B}$ must to search all the tuples in H_i-lists to answer a decryption query).

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

Boneh and Boyen [14] proved the following theorem which quantifies the relationship between *selective*-ID IBE and fully IBE 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 H queries that the adversary can make.*

This theorem hints us to prove IND-ID-CCA security via IND-sID-CCA security by imposing a minor constraint to FullIdent. Next we first prove that FullIdent is *selective*-ID secure, then prove it is also fully secure by applying Theorem 6.1.

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

Proof. We construct an IND-CCA adversary $\mathcal{B}$ 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 $\mathcal{B}$. $\mathcal{B}$ mounts an IND-CCA attack on the the key K_{pub} using the help of algorithm $\mathcal{A}$. $\mathcal{B}$ interacts with $\mathcal{A}$ as follows.
Init. $\mathcal{A}$ outputs an identity ID_{ch} where it wishes to be challenged.

Setup. Same as in BF-IBE.

H_1-**queries**. To respond to H_1 queries, $\mathcal{B}$ maintains a list of tuples $\langle \mathsf{ID}_i, Q_i, b_i\rangle$ which is referred to as H_1^{list}. The list is initially empty. When $\mathcal{A}$ queries H_1 at point ID_i, $\mathcal{B}$ 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 $\mathcal{B}$ responds with $H_1(\mathsf{ID}_i) = Q_i$.
2. Otherwise, if $\mathsf{ID}_i = \mathsf{ID}_{ch}$, $\mathcal{B}$ sets $b_i = *$ and $Q_i = Q_{\mathsf{ID}}$; else $\mathcal{B}$ generates a random $b_i \in \mathbb{Z}_q^*$ and computes $Q_i = b_i P$.
3. $\mathcal{B}$ 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 - Private key queries. When $\mathcal{A}$ asks for the private key associate to ID_i, $\mathcal{B}$ runs the above algorithm and obtains $H_1(\mathsf{ID}_i) = Q_i$, 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 $d_i = b_i P_{pub}$. Finally, $\mathcal{B}$ 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 $\mathsf{ID}_i \neq \mathsf{ID}_{ch}$, $\mathcal{B}$ 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 $\mathsf{ID}_i = \mathsf{ID}_{ch}$, $\mathcal{B}$ 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. $\mathcal{B}$ responds as follows: first $\mathcal{B}$ gives its challenger the message M_0, M_1. The challenger responds with a $\mathsf{BasicPub}^{hy}$ ciphertext $C = \langle U, V, W\rangle$ which is also a valid FullIdent ciphertext of M_β for a random $\beta \in \{0, 1\}$. Next, $\mathcal{B}$ responds to $\mathcal{A}$ with the challenge C.

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

Phase 2 - Decryption queries. $\mathcal{B}$ responds to the decryption queries 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 $\mathcal{B}$ 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 $\mathcal{B}$ would not abort during the simulation. By the definition of algorithm $\mathcal{A}$, we have that

$\left|\Pr[\beta = \beta'] - \frac{1}{2}\right| \geq \epsilon$. Note that $\Pr[\mathcal{B}$ does not abort$] = 1$, this shows that $\mathcal{B}$'s advantage against $\mathsf{BasicPub}^{hy}$ is at least ϵ as required. □

Remark 3. Interestingly, we noticed that the IND-sID-CCA in IBE setting is analogus notion to IND-CCA notion. That's why *selective*-ID security could be always tightly reduced to the security of the underlying public key scheme.

We modify FullIdent by hashing 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 $\mathsf{FullIdent}_H$. Note that the *selective*-ID security 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. Finally, as a straightforward result of Theorem 6.1, we conclude that $\mathsf{FullIdent}_H$ is fully secure in the random oracle model.

7 The New proof of BF-IBE

The security proofs proposed by Boneh and Franklin [1], Galindo [2] and Nishioka [3] are of the same class. There are two issues about this kind of proof. First, the IND-ID-CCA security is achieved via a series security reductions (see the diagram in Appendix C). These intermediate reductions may make the final security reduction looser. Second, unlike IND-sID-CCA, IND-ID-CCA is stronger than IND-CCA. In IND-ID-CCA game, the adversary can determine the identity it wishes to attack in the challenge phase, while in IND-CCA game, the adversary must declare the public key it will attack in the beginning of the game. So IND-ID-CCA security of FullIdent and the IND-CCA security of $\mathsf{BasicPub}^{hy}$ are not meaningfully linked, which make it is hard to provide a simple and elegant proof.

Zhang and Imai [4] realized this and reduced the security directly to the underlying complex assumption. However, their proof is not perfect because the simulator algorithm cannot guarantee the simulation is coherent throughout the game. The proof technique used in [16] shows that with the help of decisional oracle $\mathcal{O}_{DBDH}$, the simulator can answer all the queries coherently throughout the simulation.

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 GBDH assumption 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 ϵ and $\mathcal{A}$ makes at most q_E extraction queries, at most q_D decryption queries, and at most q_{H_i} queries to H_i oracles, respectively. Then there is a GBDH algorithm $\mathcal{B}$ has advantage*

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

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 $\mathcal{B}$ by interacting with IND-ID-CCA adversary $\mathcal{A}$.

Proof. Suppose $\mathcal{B}$ is given a instance $(P, aP, bP, cP, \mathcal{O}_{DBDH})$ of the GBDH 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 $T = e(P,P)^{abc} \in \mathbb{G}_2$.

Setup. $\mathcal{B}$ gives $\mathcal{A}$ $\langle q, \mathbb{G}_1, \mathbb{G}_2, e, n, P, P_{pub}, H_1, H_2, H_3, H_4 \rangle$ 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 determined 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 [15] [16], 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' ($\mathcal{B}$ proceeds 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 an entry in L_2' satisfying that $(P, P_{pub}, Q, U, \omega)$ 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 this 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, $\mathcal{B}$ randomly picks $h_3 \in \mathbb{Z}_q^*$, 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 always assume ID_i has already in L_1 list, if not, we can generate the tuple according to H_1 algorithm), $\mathcal{B}$ find out 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 algo-

rithm. If there does not exist such an ω_j, 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 responds the query according to the decryption algorithm.

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 coherently, $\mathcal{B}$ generates the hash value of $H_2(e(P_{pub},Q)^c)$ in the following steps.

1. Check whether L_2 contains an entry that 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 $h_{2,j} = H_2(\omega_j)$.
2. 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 $h_{2,j}' = H_2(\omega_j^*)$.
3. Otherwise, $\mathcal{B}$ randomly picks $h_2' \in \{0,1\}^n$ 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 the 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 (σ^*, M_β), it reports "failure" and terminates (the reason of $\mathcal{B}$ has to abort in this case is it does not know the right value for $H_3(\sigma^*, M_\beta)$, because c is unknown to $\mathcal{B}$). In other cases, $\mathcal{B}$ 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 denote the event that $\mathcal{A}$ issues query $e(d_{\mathsf{ID}},U)$ to H_2-oracle as AskH_2. According to the definition of $\mathcal{A}$, AskH_2 must occur with a overwhelming probability. Otherwise, $\mathcal{A}$'s guess β' gives no information for β, which contradicts that $\mathcal{A}$ has non-negligible advantage in guessing β. 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. From the above analysis we also know that if AskH_2 occurs, the attack to GBDH problem succeeds. Therefore, the probability of $\mathcal{B}$ finding the wanted tuple (ω, h_2) in L_2 is at least ϵ.

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

$\mathcal{B}$ may terminates before AskH_2 occurs due to 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 $q_{H_3}/2^n$ (2^n is the cardinality 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. Thus we have $\Pr[\neg\mathcal{E}_3] = 1 - q_{H_3}/2^n$. Combine all above and notice that $\mathcal{E}_1$, $\mathcal{E}_2$ and $\mathcal{E}_3$ are independent, we have 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 $\Pr[\mathsf{Success}] \geq 1/e(1+q_E) \cdot (1 - q_{H_3}/2^n)$. The bound on time complexity can be verified easily. This proves the result as required. □

Remark 4. In order to answer the decryption queries coherently, $\mathcal{B}$ has to call the $\mathcal{O}_{DBDH}$-oracle at most $q_D q_{H_2}$ times. In order to return a proper and valid ciphertext, $\mathcal{B}$ has to call the $\mathcal{O}_{DBDH}$-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 $\mathcal{O}_{DBDH}$-oracle. A similar observation was made by Cramer and Shoup [15] in their security proof of the Hashed ElGamal KEM.

At last, we provide a comparison with other proofs in Table 1.

8 Conclusion

In this paper, we point out and fix the lapses in previous proofs of BF-IBE. We show that if restricting all the identities to be n-bits long, we can prove the full security of BF-IBE via its *selective*-ID security. Besides, we give an elegant proof with tight security reduction of BF-IBE based on the GBDH assumption in the random oracle model. We think how to provide a tighter proof of BF-IBE based on the original CBDH assumption is still an interesting problem.

Proof	Assumption	Reduction factor
Nishioka's proof (fixed)	CBDH	$\frac{1}{(1+q_{H_1}+q_D)q_{H_2}(q_{H_3}+q_{H_4})}\left(1-\frac{q_E}{1+q_{H_1}+q_D}\right)$
Zhang and Imai's proof (fixed)	CBDH	$\frac{1}{e(1+q_E)q_{H_2}}$
Our first proof[1]	CBDH	$\frac{1}{q_{H_2}(q_{H_3}+q_{H_4})}$
Our second proof	GBDH	$\frac{1}{e(1+q_E)}$

[1] Our first proof is dedicated to $\mathsf{FullIdent}_H$, not the original FullIdent.

Table 1. Comparison of security proofs

References

1. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. Advances in Cryptology - CRYPTO 2001 **2139** (2001) 213–229
2. David Galindo: Boneh-franklin identity based encryption revisited. Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005 Proceedings **3580** (2005) 791–802
3. 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
4. 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
5. Shamir, A.: Identity-based cryptosystems and signatures schemes. Advances in Cryptology - Crypto 1984 **196** (1984) 47–53
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. Boneh, D., Boyen, X.: Efficient selective-id secure identity based encryption without random oracles. Proceedings of Eurocrypt 2004 **3027** (2004) 223–238
15. 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
16. Libert, B., Quisquater, J.J.: Identity based encryption without redundancy. Proceedings of ACNS 2005 **3531** 285–300
17. Mihir Bellare and Phillip Rogaway: Random oracles are practical: A paradigm for designing efficient protocols. (1995) 62–73
18. Joux, A.: A one round protocol for triparitite diffie-hellman. Algorithmic Number Theory, 4th International Symposium **1838** (2000) 385–394
19. Coron, J.S.: On the exact security of full domain hash. CRYPTO 2000 **1880** (2000) 229–235
20. 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
21. Eiichiro Fujisaki and Tatsuaki Okamoto: Secure integration of asymmetric and symmetric encryption schemes. (1999) 537–554

22. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. Advances in Cryptology - Eurocrypt 2003 **2656** (2003) 255–271
23. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identitybased encryption. Advances in Cryptology - Eurocrypt 2004 **3027** (2004) 207–222

A 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 $\mathcal{G}(1^k) \rightarrow (q, \mathbb{G}_1, \mathbb{G}_2, e)$.

B Security Notions

Recall that an IBE scheme consists of four algorithms [5] [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 [1] 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 [22] [23] 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.*

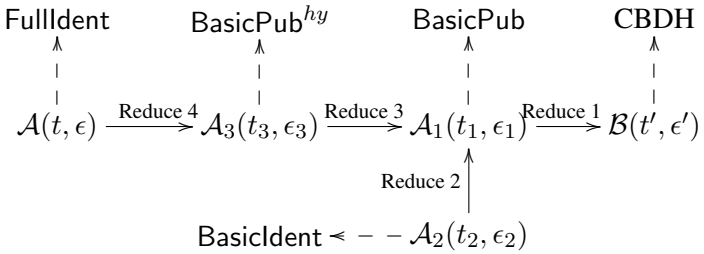
C Boneh-Franklin's IBE Scheme

In this section, we briefly describe BF-IBE scheme [1] 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 BasicPub^{hy}. BasicIdent which has only CPA security, is a simplified version of FullIdent, BasicPub is a PKE scheme derived from BasicIdent, and BasicPub^{hy} is a PKE scheme obtained by applying the Fujisaki-Okamoto conversion [21] to BasicPub. We first review the FullIdent in Figure 1.

BF-IBE(FullIdent)	
Setup(1^k): $s \leftarrow \mathbb{Z}_q^*$; $P_{pub} = sP$ 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, $d_{\mathsf{ID}})$ 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

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



The diagram of security reductions

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 $O(time(\mathcal{A}_1))$.

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 $\left[\frac{\epsilon(k)}{e(1+q_E+q_D)+1}(1-2/q)^{q_D}-1\right]/q_{H_2}(q_{H_3}+q_{H_4})$.

D Analysis of Nishioka's proof

In this section, we review simulation algorithm for challenge phase proposed by Nishioka [3].

Challenge. Once algorithm $\mathcal{A}$ decides that Phase 1 is over, it outputs a public key ID_{ch} and two messages M_0, M_1 on which it wishes to be challenged. Algorithm B gives the challenger M_0 and M_1 as the messages that it wishes to be challenged on. The challenger responds with a BasicPub^{hy} ciphertext $C = \langle U, V, W\rangle$ such that C is the encryption of M_β for a random $\beta \in \{0,1\}$.

- Suppose that there is a tuple $\langle i, \mathsf{ID}_i, Q_i, b_i\rangle$ on the H_1^{list1} such that $\mathsf{ID}_{ch} = \mathsf{ID}_i$. Let the probability of this case be ε.
  1. If $i = j$ then $H_1(\mathsf{ID}_{ch}) = Q$. Algorithm $\mathcal{B}$ responds to $\mathcal{A}$ with the challenge C. It is easy to see the probability of this subcase is $1/(1+q_{H_1}+q_D)$.
  2. Otherwise, algorithm $\mathcal{B}$ picks a random $\beta' \in \{0,1\}$ as its guess for β. Algorithm $\mathcal{B}$ then halts.

– Suppose that there is no tuple $\langle i, \mathsf{ID}_i, Q_i, b_i \rangle$ on the H_1^{list1} such that $\mathsf{ID}_{ch} = \mathsf{ID}_i$. Accordingly, the probability of this case is $(1 - \varepsilon)$.
  1. If $l_1 \neq j - 1$, algorithm $\mathcal{B}$ picks a random $\beta' \in \{0, 1\}$ as its guess for β. Algorithm $\mathcal{B}$ then halts.
  2. If $l_1 = j - 1$, algorithm sets $Q = Q_{\mathsf{ID}}$, and adds $\langle j, \mathsf{ID}_{ch}, Q, * \rangle$ to the H_1^{list1}, such that $H_1(\mathsf{ID}_{ch}) = Q$. Algorithm $\mathcal{B}$ responds to $\mathcal{A}$ with the challenge C. It is easy to verify that the probability of this subcase is also $1/(1 + q_{H_1} + q_D)$.

  Combine the above analysis, we have

$$\Pr[\neg\mathcal{E}_2|\neg\mathcal{E}_1] = \frac{1}{1 + q_{H_1} + q_D}\varepsilon + \frac{1}{1 + q_{H_1} + q_D}(1 - \varepsilon) = \frac{1}{1 + q_{H_1} + q_D}$$