

# Boneh-Franklin Identity Based Encryption Revisited

David Galindo

Institute for Computing and Information Sciences  
Radboud University Nijmegen  
P.O.Box 9010 6500 GL, Nijmegen, The Netherlands. [d.galindo@cs.ru.nl](mailto:d.galindo@cs.ru.nl)

**Abstract.** The first practical identity based encryption (IBE) scheme was proposed by Boneh and Franklin in [BF03]. In this work we point out that there is a flawed step in the security reduction exhibited by the authors. Fortunately, it is possible to fix it without changing the scheme or the underlying assumption.

In the second place, we introduce a variant of the seminal IBE scheme which allows a more efficient security reduction. The new scheme is simpler, and has more compact ciphertexts than Boneh-Franklin's proposal, while keeping the computational cost.

Finally, we observe that the flawed step pointed out here is present in several works, and that our techniques can be applied to obtain tighter reductions for previous relevant schemes.

**Keywords:** provable security, identity-based encryption, exact security, bilinear maps.

## 1 Introduction

The concept of Identity Based Encryption (IBE) was proposed by Shamir in [Sha85], aimed at simplifying certificate management in e-mail related systems. The idea is that an arbitrary string such as an e-mail address or a telephone number could serve as a public key for an encryption scheme. Once a user  $U$  receives a communication encrypted using its identity  $ID_U$ , the user authenticates itself to a Private Key Generation Center (KGC) from which it obtains the corresponding private key  $d_{ID_U}$ .

The problem was not satisfactorily solved until the work by Boneh and Franklin [BF03]. They proposed formal security notions for IBE systems and designed a fully functional secure IBE scheme using bilinear maps. The security is based on a variant of the Computational Diffie-Hellman assumption, called Bilinear Diffie-Hellman assumption. This scheme and the tools developed in its design have been successfully applied in numerous cryptographic settings, transcending by far the identity based cryptography framework.

On the other hand, an important concern when exhibiting a security reductionist proof is that of the efficiency of the reduction. One of the goals pursued

is to preserve as much as possible the strength of the underlying hard problem which is used in the protocol's design. An inefficient security reduction would imply the use of larger key sizes to attain a given security level.

**Our contributions.** In the first place, we show there is a flawed step in the security reduction exhibited in [BF03] for the scheme proposed for chosen ciphertext security. Fortunately, the reduction can be changed without modifying the original scheme or the underlying hard problem used to state the security. The efficiency of the new security reduction is a bit worse than the previous one. This is just another example in which a well-known and widely used construction turns out to have an unnoticed flawed security reduction.

In the second place, we modify the scheme by Boneh and Franklin towards obtaining a more efficient security reduction. Indeed, it is possible to show a tighter security reduction for a modified scheme which uses one less random oracle. The new proposal also presents more compact ciphertexts than the original scheme.

Finally, since Boneh-Franklin IBE scheme has been used as a building block for numerous protocols, the corrections and improvements we present here are likely to be applied to further schemes. For instance, this is the case for the schemes in [GS02,HL02,Gen03,AP03,YFDL04,CC05].

## 2 Preliminaries

We begin by fixing some notation. If  $A$  is a non-empty set, then  $x \leftarrow A$  denotes that  $x$  has been uniformly chosen in  $A$ . If  $A$  is a finite set, then  $|A|$  denotes its cardinality.

### 2.1 Definitions for IBE schemes

**Identity based encryption (IBE).** An IBE is specified by four probabilistic polynomial time (PPT) algorithms:

**Setup** takes a security parameter  $1^\ell$  and returns the system parameters **params** and **master-key**. The system parameters include the description of sets  $\mathcal{M}, \mathcal{C}$ , which denote the set of messages and ciphertexts respectively. **params** is publicly available, while the **master-key** is kept secret by the KGC.

**Extract** takes as inputs **params**, **master-key** and an arbitrary string  $ID \in \{0, 1\}^*$  and returns a private key  $d_{ID}$  to the user with identity  $ID$ . This must be done over a secure channel, since  $d_{ID}$  enables to decrypt ciphertexts under the identity  $ID$ .

**Encrypt** takes as inputs **params**,  $ID \in \{0, 1\}^*$  and  $M \in \mathcal{M}$ . It returns a ciphertext  $C \in \mathcal{C}$ .

**Decrypt** takes as inputs **params**,  $C \in \mathcal{C}$  and a private key  $d_{ID}$ , and it returns  $M \in \mathcal{M}$  or rejects.

**Chosen ciphertext security.** An IBE scheme is said to have indistinguishability against an adaptive chosen ciphertext attack (IND-ID-CCA) if any PPT algorithm  $\mathcal{A}$  has a negligible advantage in the following game:

**Setup** The challenger takes a security parameter  $1^\ell$  and runs the **Setup** algorithm. It gives **params** to the adversary. It keeps the **master-key** to itself.

**Phase 1** The adversary issues queries of the form

- Extraction query  $\langle \text{ID}_i \rangle$ . The challenger runs algorithm **Extract** to generate the private key  $d_i$  corresponding to  $\text{ID}_i$ . It sends  $d_i$  to the adversary.
- Decryption query  $\langle \text{ID}_i, C_i \rangle$ . The challenger generates the private key  $d_i$ . It then runs **Decrypt** to decrypt  $C_i$  under  $\text{ID}_i$ .

These queries may be asked adaptively, that is, each query may depend on the answers obtained to the previous queries.

**Challenge** The adversary outputs equal length plaintexts  $M_0, M_1 \in \mathcal{M}$  and an identity  $\text{ID}_{\text{ch}}$ . The only constraint is that the private key for  $\text{ID}_{\text{ch}}$  was not requested in Phase 1. The challenger picks  $\beta \leftarrow \{0, 1\}$  and sets  $C = \text{Encrypt}(\text{params}, \text{ID}_{\text{ch}}, M_\beta)$ . It sends  $C$  to the adversary.

**Phase 2** The adversary issues extraction and decryption queries as in Phase 1, with the restriction  $\langle \text{ID}_i \rangle \neq \langle \text{ID}_{\text{ch}} \rangle$  and  $\langle \text{ID}_i, C_i \rangle \neq \langle \text{ID}_{\text{ch}}, C \rangle$ .

**Guess** The adversary outputs a guess  $\beta' \in \{0, 1\}$ .

Such an adversary is called an IND-ID-CCA adversary  $\mathcal{A}$ , and its advantage is defined as  $\text{Adv}_{\mathcal{E}, \mathcal{A}}^{\text{CCA}}(1^\ell) = |\Pr[\beta = \beta'] - 1/2|$ .

Similarly, indistinguishability against passive adversaries (IND-ID-CPA) can also be defined. In this case, the game between the challenger and the adversary is similar to the IND-ID-CCA case, but disallowing decryption queries. The advantage of an adversary in this game is defined as  $\text{Adv}_{\mathcal{E}, \mathcal{A}}^{\text{CPA}}(1^\ell) = |\Pr[\beta = \beta'] - 1/2|$ .

**Definition 1.** An IBE system  $\mathcal{E}$  is secure under chosen ciphertext attacks (*resp.* chosen plaintext attacks) if for any probabilistic polynomial time IND-ID-CCA (*resp.* IND-ID-CPA) adversary  $\mathcal{A}$  the function  $\text{Adv}_{\mathcal{E}, \mathcal{A}}^{\text{CCA}}(1^\ell)$  (*resp.*  $\text{Adv}_{\mathcal{E}, \mathcal{A}}^{\text{CPA}}(1^\ell)$ ) is negligible.

## 2.2 Bilinear maps and bilinear groups

Let  $\mathbb{G}_1, \mathbb{G}_2$  and  $\mathbb{G}_T$  be finite abelian groups in which the discrete logarithm is believed to be hard. We use additive notation for  $\mathbb{G}_1, \mathbb{G}_2$  whereas multiplicative notation is used for  $\mathbb{G}_T$ . Thus,  $\mathbb{G}_1^* = \mathbb{G}_1 \setminus \{O_1\}$  and  $\mathbb{G}_T^* = \mathbb{G}_T \setminus \{1_T\}$ , where  $O_1$  and  $1_T$  are the identity elements in  $\mathbb{G}_1$  and  $\mathbb{G}_T$  respectively. By a *pairing* or *bilinear map* we will refer to a non-degenerate bilinear function  $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ . In some protocols the existence of a computable isomorphism  $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$  is assumed. In particular, this implies that  $\psi(aP_2) = a\psi(P_2)$ . By a *bilinear group* we refer to a tuple  $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, \psi)$  with the properties described above.

Bilinear maps are usually implemented using the Weil or modified Tate pairings on an elliptic curve. In general, the elements in  $\mathbb{G}_1$  allow more compact representation than those in  $\mathbb{G}_2$ . In the following it is assumed that  $|\mathbb{G}_1| =$

$|\mathbb{G}_2| = |\mathbb{G}_T| = p$ , where  $p$  is prime;  $\mathbb{G}_1, \mathbb{G}_2$  are cyclic groups generated by  $P_1, P_2$  respectively and  $\psi(P_2) = P_1$ . In this context, the map  $\hat{e}$  is non-degenerate if and only if  $\hat{e}(P_1, P_2) \neq 1_{\mathbb{G}_T}$ . We refer to [BF03] for further details.

**Bilinear Diffie-Hellman (BDH) Problem** on  $(\mathbb{G}_1, \mathbb{G}_2)$ . Given  $aP_2, bP_2 \in \mathbb{G}_2^*$  and  $cP_1 \in \mathbb{G}_1^*$ , where  $P_2 \leftarrow \mathbb{G}_2^*, P_1 = \psi(P_2), a, b, c \leftarrow \mathbb{Z}_p^*$ ; compute  $W = \hat{e}(P_1, P_2)^{abc} \in \mathbb{G}_T$ .

We say that an algorithm  $\mathcal{B}(t, \varepsilon)$  breaks BDH on  $(\mathbb{G}_1, \mathbb{G}_2)$  if it runs in time at most  $t$  and has advantage at least  $\varepsilon$ , that is,

$$\Pr[\mathcal{B}(P_2, aP_2, bP_2, cP_1) = \hat{e}(P_1, P_2)^{abc}] \geq \varepsilon,$$

where the probability is taken over the random choices of the parameters, and the random bits of  $\mathcal{B}$ .

**Bilinear Decision Diffie-Hellman (BDDH) Problem** on  $(\mathbb{G}_1, \mathbb{G}_2)$ . Let  $aP_2, bP_2 \in \mathbb{G}_2^*, cP_1 \in \mathbb{G}_1^*$ , and  $T \leftarrow \mathbb{G}_T$ , where  $P_2 \leftarrow \mathbb{G}_2^*, P_1 = \psi(P_2), a, b, c \leftarrow \mathbb{Z}_p^*$ . We say that an algorithm  $\mathcal{B}(t, \varepsilon)$  breaks BDDH on  $(\mathbb{G}_1, \mathbb{G}_2)$  if it runs in time at most  $t$  and

$$|\Pr[\mathcal{B}(P_2, aP_2, bP_2, cP_1, \hat{e}(P_1, P_2)^{abc}) = 1] - \Pr[\mathcal{B}(P_2, aP_2, bP_2, cP_1, T) = 1]| \geq \varepsilon,$$

where the probability is computed over the random choices of the parameters, and the random bits of  $\mathcal{B}$ . Hereafter, the distribution on the left side is called *BDH distribution* and is denoted by  $\mathcal{P}_{BDH}$ , while the distribution on the right is called *random BDH distribution* and is denoted by  $\mathcal{R}_{BDH}$ .

### 3 Security proof of Boneh-Franklin identity based encryption scheme revisited

In this section we consider the identity based encryption (IBE) scheme by Boneh and Franklin [BF03]. In the first place, we point out and fix a flaw in the security reduction given by the authors. In repairing the proof, we do not need to change the security assumption neither the specification of the scheme. However, the security reduction is a bit worse than the original one.

#### 3.1 Boneh-Franklin IBE scheme

We will not directly use the original description of the BF scheme, because it is phrased with bilinear group pairs where  $\mathbb{G}_1 = \mathbb{G}_2$ , so we must adapt their scheme to the more general case  $\mathbb{G}_1 \neq \mathbb{G}_2$ . In choosing how to use  $\mathbb{G}_1$  and  $\mathbb{G}_2$ , we preferred to minimize the length of the ciphertexts. This means we use  $\mathbb{G}_2$  as the set of private keys and then ciphertexts are elements in  $\mathbb{G}_1^* \times \{0, 1\}^n$ . Here follows the description of the BF scheme, which is called *Full-Ident* in [BF03].

### Full-Ident

**Setup.** Let  $(\mathbb{G}_1, \mathbb{G}_2, G_T, \hat{e}, \psi)$  a bilinear group. Choose a generator  $P_2 \leftarrow \mathbb{G}_2$  and set  $P_1 = \psi(P_2)$ . Next pick  $s \leftarrow \mathbb{Z}_p^*$  and set  $Q_{pub} = sP_2 \in \mathbb{G}_2^*$ ,  $P_{pub} = sP_1 \in \mathbb{G}_1^*$ . Choose cryptographic hash functions  $H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_2^*$ ,  $H_2 : \mathbb{G}_T \rightarrow \{0, 1\}^n$ ,  $H_3 : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \mathbb{Z}_p^*$ ,  $H_4 : \{0, 1\}^n \rightarrow \{0, 1\}^n$ . The message space is  $\mathcal{M} = \{0, 1\}^n$  and the ciphertext space is  $\mathcal{C} = \mathbb{G}_1^* \times \{0, 1\}^n \times \{0, 1\}^n$ .  
**Extract.** For a given string  $ID \in \{0, 1\}^*$ , compute  $Q_{ID} = H_1(ID) \in \mathbb{G}_2^*$  and set the private key  $d_{ID}$  to be  $d_{ID} = sQ_{ID} \in \mathbb{G}_2^*$ .

**Encrypt.** To encrypt  $M \in \{0, 1\}^n$  under identity  $ID$ , compute  $Q_{ID} = H_1(ID) \in \mathbb{G}_2^*$ , choose  $\sigma \leftarrow \{0, 1\}^n$ , set  $r = H_3(\sigma, M) \in \mathbb{Z}_p^*$  and finally

$$C = \langle rP_1, \sigma \oplus H_2(g_{ID}^r), M \oplus H_4(\sigma) \rangle \quad \text{where} \quad g_{ID} = \hat{e}(P_{pub}, Q_{ID}) \in \mathbb{G}_T.$$

**Decrypt.** Let  $C = \langle U, V, W \rangle \in \mathcal{C}$  be a ciphertext under the identity  $ID$ . To decrypt  $C$  using the private key  $d_{ID} \in \mathbb{G}_2^*$  do:

1. Compute  $V \oplus H_2(\hat{e}(U, d_{ID})) = \sigma$ .
2. Compute  $W \oplus H_4(\sigma) = M$ .
3. Set  $r = H_3(\sigma, M)$ . Check that  $U = rP$ . If not, reject the ciphertext.
4. Output  $M$ .

This completes the description of Full-Ident. This IBE scheme is sound since

$$\hat{e}(U, d_{ID}) = \hat{e}(rP_1, sQ_{ID}) = \hat{e}(P_1, Q_{ID})^{sr} = \hat{e}(P_{pub}, Q_{ID})^r = g_{ID}^r.$$

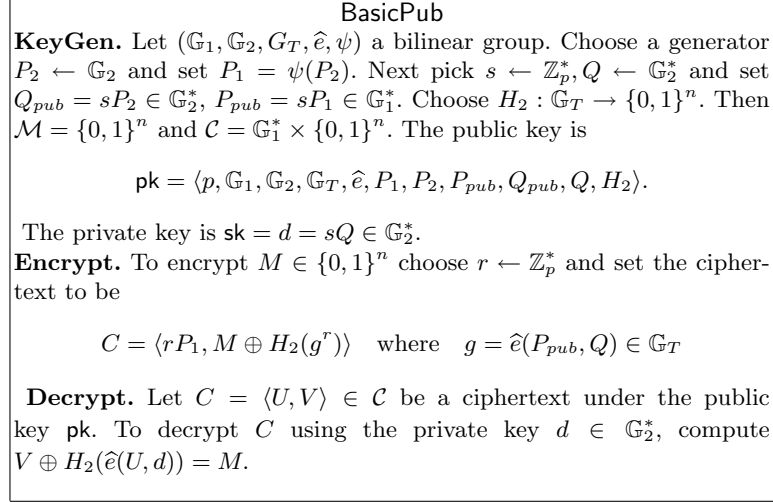
In [BF03] it is proven that the above scheme is IND-ID-CCA secure under the BDH assumption in the Random Oracle model. That scheme uses Fujisaki and Okamoto transformation [FO99] from a one-way encryption scheme into an IND-CCA encryption scheme in the ROM (we refer to [BDPR98] for public key encryption security notions). If we denote by  $E_{pk}(M, r)$  the encryption of  $M$  using the random bits  $r$  under the public key  $pk$ , the transformation by Fujisaki and Okamoto is the hybrid scheme<sup>1</sup>

$$E_{pk}^{hy}(M) = \langle E_{pk}(\sigma, H_3(\sigma, M)), H_4(\sigma) \oplus M \rangle \quad (1)$$

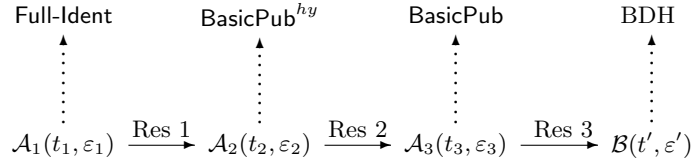
where  $\sigma$  is random and  $H_3, H_4$  are random oracles. To decrypt  $(C_1, C_2)$ , one first obtains  $\sigma'$  decrypting  $C_1$  using the original scheme, next computes  $M'$  and finally checks if  $E_{pk}(\sigma', H_3(\sigma', M)) = C_1$ . If this is so, outputs  $M$ ; otherwise outputs reject.

Two additional schemes are needed in order to exhibit the security proof in [BF03]. These schemes are not IBE schemes but merely public key encryption schemes. They are called BasicPub and BasicPub<sup>hy</sup>. Here follows the description of

<sup>1</sup> In the case where the symmetric encryption scheme is the one-time pad.



Finally, the scheme  $\text{BasicPub}^{hy}$  is the result of applying Fujisaki-Okamoto transformation (1) to the above scheme. The security reduction for Full-Ident scheme under the BDH assumption follows the diagram below



The following results are shown in [BF03]. Hereafter,  $q_E, q_D, q_{H_i}$  denote the number of extraction, decryption and random oracle  $H_i$  queries respectively.

**Result 1** *Let  $\mathcal{A}_1$  an IND-ID-CCA adversary that has advantage  $\varepsilon_2$  against Full-Ident making at most  $q_E, q_D$  and  $q_{H_1}$  queries. Then there is an IND-CCA adversary  $\mathcal{A}_2$  that has advantage at least  $\frac{\varepsilon_2}{e(1+q_E+q_D)}$  against  $\text{BasicPub}^{hy}$ . Its running time is  $t_2 \leq t_1 + c_{\mathbb{G}_2}(q_D + q_{H_1} + q_E)$ , where  $c_{\mathbb{G}_2}$  denotes the time of computing a random multiple in  $\mathbb{G}_2$ .*

**Result 2** *Let  $\mathcal{A}_2$  an IND-CCA adversary that has advantage  $\varepsilon_2$  against  $\text{BasicPub}^{hy}$  making at most  $q_D, q_{H_3}$  and  $q_{H_4}$  queries. Then there is an IND-CPA adversary  $\mathcal{A}_3$  that has advantage at least  $\frac{1}{2(q_{H_3}+q_{H_4})}[(\varepsilon_2 + 1)(1 - 2/p)^{q_D} - 1]$  against BasicPub. Its running time is  $t_3 \leq t_2 + \mathcal{O}((q_{H_3} + q_{H_4}) \cdot (n + \log p))$ .*

**Result 3** *Let  $\mathcal{A}_3$  an IND-CPA adversary that has advantage  $\varepsilon_3$  against BasicPub making at most  $q_{H_2}$  queries. Then there is an algorithm  $\mathcal{B}$  breaking the BDH problem on  $(\mathbb{G}_1, \mathbb{G}_2)$  with advantage at least  $\frac{2\varepsilon_3}{q_{H_2}}$  and running time  $t' \approx t_3$ .*

In order to come up with the total concrete security, we can bound any  $q_{H_i}$  with a single  $q_H$ , and assume that  $q_E = q_D$ , since extraction and decryption operations have roughly the same computational complexity. Then, taking the above reductions, we obtain that the BF scheme is  $(t_1, q_H, q_D, \varepsilon_1)$  IND-ID-CCA secure if the BDH problem on  $(\mathbb{G}_1, \mathbb{G}_2)$  is

$$\left( t_1 + c_{\mathbb{G}_1}(2q_D + q_H) + 2q_H(n + \log p), \frac{\varepsilon_1}{8eq_H^2q_D} \right)\text{-secure.} \quad (2)$$

Therefore, the security reduction is far from tight, mainly because of the  $q_H^2q_D$  factor relating the advantages against the scheme and the underlying problem.

### 3.2 A flaw in the security reduction

In this section we point out a flaw in the reduction used to state Result 1, which is Lemma 4.6 in [BF03].

The goal of that reduction is to construct an IND-CCA adversary  $\mathcal{B}$  with advantage  $\varepsilon/e(1 + q_E + q_D)$  against  $\text{BasicPub}^{hy}$  by using an IND-ID-CCA adversary  $\mathcal{A}$  with advantage  $\varepsilon$  against  $\text{Full-Ident}$ .  $\mathcal{B}$  receives a public key

$$K_{pub} = \langle p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, P_1, P_2, P_{pub}, Q, Q_{pub}, H_2, H_3, H_4 \rangle$$

from its challenger. Then  $\mathcal{B}$  simulates the challenger for  $\mathcal{A}$  as follows:

**Setup**  $\mathcal{B}$  gives  $\mathcal{A}$  the parameters  $\langle p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, P_1, P_2, P_{pub}, Q_{pub}, H_1, H_2, H_3, H_4 \rangle$ , where  $H_1$  is an oracle controlled by  $\mathcal{B}$  as indicated in the following:

**$H_1$ -queries** To respond to  $\mathcal{A}$  queries, algorithm  $\mathcal{B}$  maintains a list  $H_1^{\text{list}}$  of tuples  $\langle \text{ID}_i, Q_i, b_i, c_i \rangle$  as explained below. When  $\mathcal{A}$  queries  $H_1$  at an unrepeated  $\text{ID}_i$ ,  $\mathcal{B}$  generates a random coin  $c_i \in \{0, 1\}$  such that  $\Pr[c_i = 0] = \delta$ , and a random  $b_i \leftarrow \mathbb{Z}_p^*$ . If  $c_i = 0$  it computes  $Q_i = b_i P_2 \in \mathbb{G}_2^*$ , and if  $c_i = 1$  it computes  $Q_i = b_i Q \in \mathbb{G}_2^*$ . Finally,  $\mathcal{B}$  adds the tuple  $\langle \text{ID}_i, Q_i, b_i, c_i \rangle$  to the  $H_1^{\text{list}}$  and sends  $H_1(\text{ID}_i) = Q_i$  to  $\mathcal{A}$ .

The idea is that tuples with  $c_i = 0$  enable  $\mathcal{B}$  to answer private key queries for identity  $\text{ID}_i$ , while  $\mathcal{B}$  can only take profit of  $\mathcal{A}$ 's advantage when  $\mathcal{A}$  chooses a challenge identity  $\text{ID}_{\text{ch}}$  such that  $c_{\text{ch}} = 1$ .

**Phase 1 - Extraction queries** When  $\mathcal{A}$  asks for the private key associated to  $\text{ID}_i$ ,  $\mathcal{B}$  runs the algorithm for responding  $H_1$ -queries and gets  $H_1(\text{ID}_i) = Q_i$ , where  $\langle \text{ID}_i, Q_i, b_i, c_i \rangle$  is the corresponding entry in  $H_1^{\text{list}}$ . If  $c_i = 1$  then  $\mathcal{B}$  aborts the game and the attack against  $\text{BasicPub}^{hy}$  failed. Otherwise,  $c_i = 0$  and therefore  $Q_i = b_i P_2$ . It turns out that  $d_i$  can be computed as  $d_i := b_i Q_{pub}$ , since by definition  $d_i = sQ_i$ . Finally,  $\mathcal{B}$  gives  $d_i$  to algorithm  $\mathcal{A}$ .

**Phase 1 - Decryption queries**  $\mathcal{B}$  answers to a decryption query  $\langle \text{ID}_i, C_i \rangle$  as follows. It runs  $H_1$ -queries algorithm and let  $\langle \text{ID}_i, Q_i, b_i, c_i \rangle \in H_1^{\text{list}}$ . If  $c_i = 0$ , then  $\mathcal{B}$  retrieves the private key  $d_i$  and decrypts  $C_i$  using the decryption algorithm. If  $c_i = 1$ , then  $Q_i = b_i Q$ . Recall that the unknown private key is  $d_i = sQ_i = sb_i Q$ . Set  $C'_i = \langle b_i U_i, V_i, W_i \rangle$ , where  $C_i = \langle U_i, V_i, W_i \rangle$ . Then, the

authors claim that the Full-Ident decryption of  $C_i$  is *equal* to the BasicPub<sup>hy</sup> decryption of  $C'_i$ . The reason given is that

$$\widehat{e}(b_i U_i, d) = \widehat{e}(b_i U_i, sQ) = \widehat{e}(U_i, s b_i Q) = \widehat{e}(U_i, s Q_i) = \widehat{e}(U_i, d_i),$$

which implies that the values  $\sigma$  and  $M$  obtained by decrypting  $C_i$  with Full-Ident and by decrypting  $C'_i$  with BasicPub<sup>hy</sup> are equal. However, BasicPub<sup>hy</sup> will output the reject symbol when decrypting  $C'_i$  with overwhelming probability. To see this, remember that  $b_i U_i = b_i r_i P_1$ , and at least  $b_i \leftarrow \mathbb{Z}_p^*$ , which implies that  $b_i r$  is uniformly random in  $\mathbb{Z}_p^*$ . On the other hand, we have that  $H_3$  is a random oracle not controlled by  $\mathcal{B}$ . These facts imply that  $H_3(\sigma, M) \neq b_i r$  with probability  $1 - 1/p$ , and therefore the decryption algorithm of BasicPub<sup>hy</sup> will reject the ciphertext. Thereby, we can not use the decryption oracle for BasicPub<sup>hy</sup> to decrypt ciphertexts under any  $ID_i$  such that  $H_1(ID_i) \neq Q$ . Therefore, the reduction in [BF03] is not valid.

### 3.3 Fixing the security reduction

Due to the ciphertext integrity checking in FO transformation [FO99], we can only answer decryption queries  $\langle ID_i, C_i \rangle$  such that:

- $H_1(ID_i) = b_i P_2$ , since we can use the private key  $d_i$ , or
- $H_1(ID_i) = Q$ , since in this case, the decryption of  $C_i$  under such  $ID_i$  is equal to the decryption of  $C_i$  by BasicPub<sup>hy</sup>.

This remark enables us to fix the flawed reduction shown above. In the following we describe the new answers delivered by  $\mathcal{B}$ .

**Setup** As in Section 3.2.

**$H_1$ -queries** Before initializing  $H_1^{\text{list}}$ ,  $\mathcal{B}$  selects at random  $j \leftarrow \{1, \dots, q_{H_1}\}$ . When  $\mathcal{A}$  queries  $H_1$  at  $ID_i$ , algorithm  $\mathcal{B}$  proceeds as follows: if  $i \neq j$ , it picks  $b_i \leftarrow \mathbb{Z}_p^*$ , sets  $Q_i = b_i P_2$ , adds  $\langle ID_i, Q_i, b_i \rangle$  to the list and gives back  $Q_i$  to  $\mathcal{A}$ . If  $i = j$ , it sets  $Q_j = Q$ , adds  $\langle ID_i, Q_i, * \rangle$  to the list and sends  $Q_j$  to  $\mathcal{A}$ . Here  $*$  denotes a special symbol. Note that the outputs of  $H_1$  are uniformly distributed in  $\mathbb{G}_2^*$  and independent of  $\mathcal{A}$ 's current view, since  $Q$  is unknown to  $\mathcal{A}$  and is uniformly distributed in  $\mathbb{G}_2^*$ .

**Phase 1 - Extraction queries** When  $\mathcal{A}$  asks for the private key for  $ID_i$ ,  $\mathcal{B}$  runs the algorithm for responding  $H_1$ -queries and gets  $H_1(ID_i) = Q_i$ , where  $\langle ID_i, Q_i, b_i \rangle$  is the corresponding entry in  $H_1^{\text{list}}$ . If  $i = j$ , then  $\mathcal{B}$  aborts the game and the attack against BasicPub<sup>hy</sup> failed. Otherwise, it sets  $d_i := b_i Q_{pub}$ . Finally,  $\mathcal{B}$  gives  $d_i$  to algorithm  $\mathcal{A}$ .

**Phase 1 - Decryption queries**  $\mathcal{B}$  answers to a decryption query  $\langle ID_i, C_i \rangle$  as follows. It runs  $H_1$ -queries algorithm and let  $\langle ID_i, Q_i, b_i \rangle \in H_1^{\text{list}}$ . 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  $Q_i = Q$ , and the decryption of  $\langle ID_j, C_j \rangle$  is the same as the decryption of  $C_j$  under BasicPub<sup>hy</sup>. Then,  $\mathcal{B}$  asks its challenger to decrypt  $C_j$  and relays the answer to  $\mathcal{A}$ .



**Challenge**  $\mathcal{A}$  outputs a public key  $\text{ID}_{\text{ch}}$  and two equal length plaintexts  $M_0, M_1$ . Algorithm  $\mathcal{B}$  proceeds as follows. If  $\text{ID}_{\text{ch}} \neq \text{ID}_j$ , it aborts the game and the attack against  $\text{BasicPub}^{\text{hy}}$  failed. Otherwise, it sends  $M_0, M_1$  to its own challenger and gets back  $C$ , the encryption of  $M_\beta$  for a random bit  $\beta$  under  $\text{BasicPub}^{\text{hy}}$ . Finally,  $\mathcal{B}$  relays  $C$  to  $\mathcal{A}$ , which is an also encryption of  $M_\beta$  under  $\text{ID}_{\text{ch}}$  for Full-Ident.

**Phase 2 - Extraction queries** Algorithm  $\mathcal{B}$  proceeds as in Phase 1, except for the extraction query for  $\text{ID}_{\text{ch}}$ , which is rejected.

**Phase 2 - Decryption queries** Algorithm  $\mathcal{B}$  proceeds as in Phase 1, except for the decryption query  $\langle \text{ID}_{\text{ch}}, C_\beta \rangle$ , which is rejected.

**Guess** Algorithm  $\mathcal{A}$  outputs a guess  $\beta'$  for  $\beta$ .  $\mathcal{B}$  outputs  $\beta'$  as its guess.

Using this algorithm  $\mathcal{B}$ , we are able to state the following:

**Result 4** *Let  $\mathcal{A}$  an IND-ID-CCA adversary that has advantage  $\varepsilon$  against Full-Ident making at most  $q_E, q_D$  and  $q_{H_1}$  queries. Then there is an IND-CCA adversary  $\mathcal{B}$  that has advantage at least  $\frac{\varepsilon}{q_{H_1}} \left(1 - \frac{q_E}{q_{H_1}}\right) \approx \frac{\varepsilon}{q_{H_1}}$  against  $\text{BasicPub}^{\text{hy}}$ . Its running time is  $t_2 \leq t_1 + c_{\mathbb{G}_2}(q_D + q_{H_1} + q_E)$ , where  $c_{\mathbb{G}_2}$  denotes the time of computing a random multiple in  $\mathbb{G}_2$ .*

*Proof:* See Appendix A. □

Therefore, joining Results 2, 3 and 4, an IND-ID-CCA advantage  $\varepsilon_1$  against Full-Ident is turned into an algorithm with advantage roughly  $\varepsilon_1/(q_H^3)$  in solving the BDH problem. Compared to the original flawed reduction, where the advantage obtained against BDH was roughly  $\varepsilon_1/(q_H^2 q_D)$ , the new reduction is a bit worse, since in general  $q_D \ll q_H$ . In the next section we show a modification of Full-Ident which allows a tighter security reduction.

## 4 A new identity based encryption scheme with improved tightness

In this section we design a new IBE scheme using the scheme Basic-Ident from the previous section and a second general transformation also due to Fujisaki and Okamoto [FO00]. This conversion starts from an IND-CPA encryption scheme and builds an IND-CCA scheme in the ROM. If we denote by  $E_{\text{pk}}(M, r)$  the encryption of  $M$  using the random bits  $r$  under the public key  $\text{pk}$ , with set of messages  $\mathcal{M} = \{0, 1\}^n$ , set of coins  $\mathcal{R}$  and set of ciphertexts  $\mathcal{C}$ , the new transformation is the scheme

$$E_{\text{pk}}^{\text{hyNew}}(M) = E_{\text{pk}}(M||r, H(M||r)) \quad (3)$$

where  $M||r \in \{0, 1\}^{n-k_0} \times \{0, 1\}^{k_0}$  and  $H : \{0, 1\}^* \rightarrow \mathcal{R}$  is a hash function. Then,  $\mathcal{M}^{\text{hyNew}} = \{0, 1\}^{n-k_0}$ ,  $\mathcal{R}^{\text{hyNew}} = \{0, 1\}^{k_0}$  and  $\mathcal{C}^{\text{hyNew}} = \mathcal{C}$ . To decrypt  $C$ , one first obtains  $M'||r'$  using the original decryption algorithm, and next checks if  $E_{\text{pk}}(M'||r', H(M'||r')) = C$ . If this is so, outputs  $M$ ; otherwise outputs

reject.

Let us describe the new IBE scheme thereby obtained.

**NewFull-Ident**

**Setup.** Let  $(\mathbb{G}_1, \mathbb{G}_2, G_T, \hat{e}, \psi)$  a bilinear group. Choose a generator  $P_2 \leftarrow \mathbb{G}_2$  and set  $P_1 = \psi(P_2)$ . Next pick  $s \leftarrow \mathbb{Z}_p^*$  and set  $Q_{pub} = sP_2 \in \mathbb{G}_2^*, P_{pub} = sP_1 \in \mathbb{G}_1^*$ . Choose hash functions  $H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_2^*$ ,  $H_2 : \mathbb{G}_T \rightarrow \{0, 1\}^n$  and  $H_3 : \{0, 1\}^* \rightarrow \mathbb{Z}_p^*$ . Now  $\mathcal{M} = \{0, 1\}^{n-k_0}$ ,  $\mathcal{C} = \mathbb{G}_1^* \times \{0, 1\}^n$  and  $\text{params} = \langle p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, P_1, P_2, P_{pub}, Q_{pub}, H_1, H_2, H_3 \rangle$ . The master-key is  $s \in \mathbb{Z}_p^*$ .

**Extract.** For a given ID  $\in \{0, 1\}^*$ , compute  $Q_{ID} = H_1(\text{ID}) \in \mathbb{G}_2^*$  and set  $d_{ID} = sQ_{ID} \in \mathbb{G}_2^*$  where  $s$  is the master key.

**Encrypt.** To encrypt  $M \in \{0, 1\}^{n-k_0}$  under ID, compute  $Q_{ID} = H_1(\text{ID}) \in \mathbb{G}_2^*$ , choose  $\sigma \leftarrow \{0, 1\}^{k_0}$ , set  $r = H_3(M, \sigma) \in \mathbb{Z}_p^*$  and finally

$$C = \langle rP_1, (M||\sigma) \oplus H_2(g_{ID}^r) \rangle \quad \text{where} \quad g_{ID} = \hat{e}(P_{pub}, Q_{ID}) \in \mathbb{G}_T$$

**Decrypt.** Let  $C = \langle U, V \rangle \in \mathcal{C}$  be a ciphertext under the public key ID. To decrypt  $C$  using the private key  $d_{ID} \in \mathbb{G}_2^*$  do:

1. Compute  $V \oplus H_2(\hat{e}(U, d_{ID})) = M||\sigma$ .
2. Parse  $M||\sigma$  and compute  $r = H_3(M, \sigma)$ . Check that  $U = rP$ . If not, reject the ciphertext.
4. Output  $M$ .

On the basis of the proof sketched in the previous section, we define in a similar fashion a public key encryption scheme **NewBasicPub<sup>hy</sup>**, which is obtained applying the conversion from expression (3) to **Basic-Pub**. Then the following results hold:

**Result 5** *Let  $\mathcal{A}_1$  an IND-ID-CCA adversary with advantage  $\varepsilon_1$  against NewFull-Ident making at most  $q_E$  private key extraction queries,  $q_D$  decryption queries and  $q_{H_1}$  hash queries. Then there is an IND-CCA adversary  $\mathcal{A}_2$  that has advantage at least  $\frac{\varepsilon}{q_{H_1}} \left(1 - \frac{q_E}{q_{H_1}}\right) \approx \frac{\varepsilon}{q_{H_1}}$  against NewBasicPub<sup>Hy</sup>. Its running time is  $t_2 \leq t_1 + c_{\mathbb{G}_1}(q_D + q_{H_1} + q_E)$ .*

*Proof:* Use the same reduction as for Result 4 in Section 3.3.

**Result 6** *Let  $\mathcal{A}_2$  an IND-CCA adversary with advantage  $\varepsilon_2$  against NewBasicPub<sup>hy</sup> making at most  $q_D$  decryption queries and at most  $q_{H_2}$  hash queries. Then there is an IND-CPA adversary  $\mathcal{A}_3$  that has advantage at least*

$$\left(\varepsilon_2 - q_{H_2} \cdot 2^{-(k_0-1)}\right) \left(1 - \frac{1}{p}\right)^{q_D} \approx \varepsilon_2$$

*against BasicPub. Its running time is  $t_3 \leq t_2 + q_{H_2}(T_{\text{BasicPub}} + \log p)$ , where  $T_{\text{BasicPub}}$  is the running time of **Encrypt** algorithm in BasicPub.*

*Proof:* This result is obtained as a special case of Theorem 5.4 in [FO00].

Finally, taking into account these new reductions, we obtain that **NewFull-Ident** scheme is  $(t_1, q_H, q_D, \varepsilon_1)$  IND-ID-CCA secure if the BDH problem on  $(\mathbb{G}_1, \mathbb{G}_2)$  is

$$\left( t_1 + c_{\mathbb{G}_1}(2q_D + q_H) + q_H \mathcal{O}(\log^3 p + \log p), \frac{\varepsilon_1}{q_H} \right)\text{-secure}$$

The last expression has been simplified replacing any of the hash queries  $q_{H_i}$  by  $q_H$  and setting  $q_D = q_E$ . Then, we get rid of a  $q_H$  factor in the BDH advantage with respect to the reduction in expression (2).

Compared to **Full-Ident** scheme, which is the result of using FO transformation in expression (1), the **NewFull-Ident** scheme presents several advantages:

- It provides more compact ciphertexts. In fact, **Full-Ident** scheme adds a  $n$ -bits component to a **Basic-Ident** ciphertext to get chosen ciphertext security, while **NewFull-Ident** achieves this preserving **Basic-Ident** ciphertext's structure.
- It presents a tighter security reduction to the BDH problem.
- It uses one less hash function than **Full-Ident**.

We can obtain a second tightness improvement using a stronger assumption, namely, the BDDH assumption. In this case, we have the following result:

**Result 7** *Let  $\mathcal{A}_3$  an IND-CPA adversary that has advantage  $\varepsilon_3$  against **BasicPub** making at most  $q_{H_2}$  hash queries. Then there is an algorithm  $\mathcal{B}$  breaking the BDDH problem on  $(\mathbb{G}_1, \mathbb{G}_2)$  with advantage roughly  $\varepsilon_3$  and running time  $t' \approx t_3$ .*

*Proof:* See Appendix B. □

With this second tightness improvement, we obtain that **NewFull-Ident** scheme is  $(t_1, q_H, q_D, \varepsilon_1)$  IND-ID-CCA secure if the BDDH problem on  $(\mathbb{G}_1, \mathbb{G}_2)$  is

$$\left( t_1 + c_{\mathbb{G}_1}(2q_D + q_H) + q_H \mathcal{O}(\log^3 q + \log q), \frac{\varepsilon_1}{q_H} \right)\text{-secure}$$

Then, we get rid of a  $q_H$  factor in the security reduction at the cost of relying on a stronger assumption.

## 5 Conclusions

In this work, we have shown there is a flawed step in the security reduction exhibited in [BF03] for the so called Boneh-Franklin IBE scheme. We have provided a new reduction without modifying the original scheme neither the underlying hard problem used to state the security.

In the second place, we have proposed a new IBE scheme slightly changing the original scheme. The proposal presents a tighter reduction than BF scheme, uses one less random oracle and has more compact ciphertexts.

Finally, we point out that it is still an open problem to design an IND-ID-CCA IBE scheme with a tight security reduction under a reasonable assumption either in the standard or the random oracle models.

**Acknowledgements.** The author is grateful to Javier Herranz and Paz Morillo for useful comments on an early draft of this paper. The author also acknowledges the anonymous referees' remarks.

## References

- [AP03] S. AlRiyami and K.G. Paterson. Certificateless public key cryptography. In *Advances in Cryptology – ASIACRYPT 2003*, vol. 2894 of *LNCS*, pp. 452–473, 2003. Full version available at <http://eprint.iacr.org/>.
- [BDPR98] M. Bellare, A. Desai, D. Pointcheval and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in Cryptology – CRYPTO 1998*, vol. 1462 of *LNCS*, pp. 26–45, 1998.
- [BF03] D. Boneh and M. Franklin. Identity-Based encryption from the Weil pairing. *SIAM Journal of Computing*, 32(3):586–615, 2003. This is the full version of an extended abstract of the same title presented at *Crypto'01*.
- [CC05] Z. Cheng and R. Comley. Efficient certificateless public key encryption. Cryptology ePrint Archive, Report 2005/012, 2005. <http://eprint.iacr.org/>.
- [FO99] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology – CRYPTO 1999*, vol. 1666 of *LNCS*, pp. 537–554, 1999.
- [FO00] E. Fujisaki and T. Okamoto. How to enhance the security of public-key encryption at minimum cost. *IEICE Trans. Fundamentals*, E83-9(1):24–32, 2000. This is the full version of [?].
- [Gen03] C. Gentry. Certificate-based encryption and the certificate revocation problem. In *Advances in Cryptology – EUROCRYPT 2003*, vol. 2656 of *Lecture Notes in Computer Science*, pp. 272–293, 2003.
- [GS02] C. Gentry and A. Silverberg. Hierarchical ID-based cryptography. In *Advances in Cryptology – ASIACRYPT 2002*, vol. 2501 of *LNCS*, pp. 548–566, 2002.
- [HL02] J. Horwitz and B. Lynn. Toward hierarchical identity-based encryption. In *Advances in Cryptology – EUROCRYPT 2002*, vol. 2332 of *Lecture Notes in Computer Science*, pp. 466–481, 2002.
- [Sha85] A. Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology – CRYPTO 1984*, vol. 196 of *LNCS*, pp. 47–53, 1985.
- [YFDL04] D. Yao, N. Fazio, Y. Dodis and A. Lysyanskaya. Id-based encryption for complex hierarchies with applications to forward security and broadcast encryption. In *Proceedings of the 11th ACM CCS*, pp. 354–363. ACM Press, 2004.

## A Proof of Result 4

If algorithm  $\mathcal{B}$  does not abort during the simulation,  $\mathcal{A}$ 's view is identical to its view in a real attack:  $H_1$  behaves as random oracle, and extraction as well as

decryption queries are valid. Therefore,  $|\Pr[\beta' = \beta] - 1/2| \geq \varepsilon$ , where this probability is over the random bits of  $\mathcal{A}, \mathcal{B}$  and the challenger for the IND-ID-CCA game.

It remains to bound the probability  $\Pr[\mathcal{B} \text{ does not abort}]$ . The algorithm can abort for two reasons: (1) it is asked in Phase 1 for the private key query corresponding to  $\text{ID}_j$ , or (2) the challenge identity  $\text{ID}_{\text{ch}} \neq \text{ID}_j$ . Note that  $\mathcal{B}$  can not abort in Phase 2, since in this case  $\mathcal{A}$  is not allowed to query the private key for  $\text{ID}_j = \text{ID}_{\text{ch}}$ . Let  $\mathcal{E}_1$  be the event that  $\mathcal{B}$  aborts due to (1), and define  $\mathcal{E}_2$  in the obvious way.

Then,  $\Pr[\mathcal{B} \text{ does not abort}] = \Pr[\neg\mathcal{E}_1 \wedge \neg\mathcal{E}_2] = \Pr[\neg\mathcal{E}_2 | \neg\mathcal{E}_1] \Pr[\neg\mathcal{E}_1]$ .

We can upper bound for  $\Pr[\mathcal{E}_1] \leq q_E/q_H$ , which is the probability that  $\mathcal{A}$  makes a extraction query for  $\text{ID}_j$  in Phase 1, since the maximum number of such queries is  $q_E$ .

On the other hand, a lower bound for  $\Pr[\neg\mathcal{E}_2 | \neg\mathcal{E}_1]$ , that is the probability that  $\mathcal{A}$  chooses  $\text{ID}_j$  as the challenge identity, is  $1/q_{H_1}$ . Therefore,

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

This shows that  $\mathcal{B}'$ 's advantage is at least

$$\frac{\varepsilon}{q_{H_1}} \left(1 - \frac{q_E}{q_{H_1}}\right).$$

□

## B Proof of Result 7

Algorithm  $\mathcal{B}$  receives as inputs the bilinear group  $(\mathbb{G}_1, \mathbb{G}_2)$ , and a random instance  $(P_2, aP_2, bP_2, cP_1, T)$  from either  $\mathcal{P}_{BDH}$  or  $\mathcal{R}_{BDH}$  distributions. Then  $\mathcal{B}$  uses  $\mathcal{A}$  IND-CPA advantage against BasicPub to distinguish  $\mathcal{P}_{BDH}$  from  $\mathcal{R}_{BDH}$ .

**Setup**  $\mathcal{B}$  provides  $\mathcal{A}$  with the public key  $\text{pk} = \langle p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, P_1, P_2, P_{\text{pub}}, Q_{\text{pub}}, Q, H_2 \rangle$ , where  $P_1 = \psi(P_2)$ ,  $Q_{\text{pub}} = aP_2$ ,  $P_{\text{pub}} = \psi(Q_{\text{pub}})$ ,  $Q = bP_2$  and  $H_2$  is a random oracle controlled by  $\mathcal{B}$  as explained below. Notice that the unknown private key of BasicPub is  $d = abP_2$ .

**$H_2$ -queries** To respond  $\mathcal{A}$  queries to  $H_2$ ,  $\mathcal{B}$  maintains a list  $H_2^{\text{list}}$  of tuples  $\langle X_i, H_i \rangle$ . When queried with  $X_i$ , algorithm  $\mathcal{B}$  does the following:

1. If  $\langle X_i, H_i \rangle \in H_2^{\text{list}}$ , it returns  $H_2(X_i) = H_i$ .
2. Otherwise,  $\mathcal{B}$  picks  $H_i \leftarrow \{0, 1\}^n$ , adds the tuple  $\langle X_i, H_i \rangle$  to the list and returns  $H_2(X_i) = H_i$ .

**Challenge**  $\mathcal{A}$  outputs two equal length plaintexts  $M_0, M_1$  in which it wishes to be challenged. Algorithm  $\mathcal{B}$  returns as the challenge ciphertext  $C = \langle cP_1, M_\beta \oplus H_2(T) \rangle$ , where  $\beta \leftarrow \{0, 1\}$ .

**Guess**  $\mathcal{A}$  eventually outputs a guess  $\beta'$  for  $\beta$ . Algorithm  $\mathcal{B}$  returns 1 if  $\beta' = \beta$  and 0 otherwise.

Algorithm  $\mathcal{B}$  is simulating a real attack environment for  $\mathcal{A}$ . If the random instance is from  $\mathcal{R}_{BDH}$ , then  $\Pr[\beta' = \beta] = 1/2$ , since in this case the distribution of the ciphertext  $C$  is independent of the bit  $\beta$ . Otherwise, the instance comes from  $\mathcal{P}_{BDH}$ ,  $C$  is a valid encryption of  $M_\beta$  and therefore  $\Pr[\beta' = \beta] = 1/2 + \varepsilon$  by definition of  $\mathcal{A}$ . Therefore,

$$|\Pr[\mathcal{B}(\mathcal{P}_{BDH}) = 1] - \Pr[\mathcal{B}(\mathcal{R}_{BDH}) = 1]| = |1/2 + \varepsilon - 1/2| = \varepsilon.$$

□