Des. Codes Cryptogr. (2009) 50:115–133
DOI 10.1007/s10623-008-9218-2

A variant of Boneh-Franklin IBE with a tight reduction in the random oracle model

Jean-Sébastien Coron

Received: 13 July 2007 / Revised: 19 May 2008 / Accepted: 22 May 2008 /
Published online: 21 June 2008

Abstract The first practical identity based encryption (IBE) scheme was published by Boneh and Franklin at Crypto 2001, based on the elliptic curve pairing. Since that time, many other IBE schemes have been published. In this paper, we describe a variant of Boneh-Franklin with a tight reduction in the random oracle model. Our new scheme is quite efficient compared to existing schemes; moreover, upgrading from Boneh-Franklin to our new scheme is straightforward.

Keywords Identity-based encryption · Tight security · Pairing

Mathematics Subject Classification (2000) 94A60

1 Introduction

The concept of Identity-Based Encryption (IBE) was invented in 1984 by Adi Shamir [13]. It allows for a party to encrypt a message using the recipient's identity as the public key. The corresponding private-key is provided by a central authority. The first efficient and secure IBE scheme was proposed by Boneh and Franklin at Crypto 2001 [3]. It is based on a bilinear map between two groups, that is usually implemented using a pairing operation on some well chosen elliptic-curve. The Boneh-Franklin scheme is provably secure in the random oracle model, assuming that the Computational Bilinear Diffie-Hellman problem is hard to solve on the underlying elliptic curve.

Since that time, many other IBE schemes have been proposed. An important research direction consists in achieving security without the random oracle model, because the random oracle model provides only heuristic security (see [6]). The two first IBE schemes secure

Communicated by S. Galbraith.

J.-S. Coron (✉)
University of Luxembourg, Luxembourg City, Luxembourg
e-mail: jean-sebastien.coron@uni.lu

without random oracles were proposed by Boneh and Boyen in [4], but only in a restricted model of security in which the attacker must tell in advance which identity he is going to attack. Boneh and Boyen later proposed in [5] the first IBE scheme secure without random oracles in the full model, but the scheme was impractical. The first practical and fully secure IBE without random oracles was proposed at Eurocrypt 2005 by Waters [15], but with a relatively long public parameter size. Eventually Gentry published at Eurocrypt 2006 [9] an IBE scheme with short public parameters that is as of 2008 the most efficient IBE scheme secure without random oracles.

An interesting property for a cryptographic scheme is to have a tight security reduction. A security reduction is said to be "tight" when breaking the scheme is exactly as hard as solving the underlying problem. This enables to avoid any security loss for the resulting scheme. The Boneh-Franklin scheme does not have a tight reduction; namely, for standard parameters, the Boneh-Franklin scheme looses roughly 151 bits of security compared to the underlying elliptic-curve. Therefore if for some given parameter size the elliptic-curve is assumed to provide 128 bits of security, then for this parameter size the Boneh-Franklin scheme does not provide any security guarantee. To obtain a meaningful security guarantee, one must therefore select an elliptic-curve with larger parameters, but this will make the scheme less efficient.

Galindo [8] and Libert and Quisquater [12] proposed two IBE schemes with a better reduction than Boneh-Franklin, but still not tight. The first IBE scheme with a tight security reduction in the random oracle model was proposed by Katz and Wang in [10]; the scheme was later described in more details and refined by Attrapadung et al. in [1]. The Gentry scheme also achieves a tight security reduction (moreover, without random oracles), but with an underlying assumption that depends on the number of private-key queries.

In this paper, we describe a variant of Boneh-Franklin with a tight reduction in the random oracle model. Our scheme is quite efficient; we provide in Table 1 a comparison with other IBE schemes. Compared to Boneh-Franklin, our scheme provides a tight security reduction;

Table 1 Comparison between IBE schemes

Scheme	Assumption	IND-ID-X	Reduction	ROM	Ciphertext size	Enc	Dec
Boneh-Franklin [3]	BDH	CCA	$\mathcal{O}\left(\frac{1}{q_h^2 q_e}\right)$	yes	$2k+2n$	1P+2E	1P+1E
Galindo [8]	BDH	CCA	$\mathcal{O}\left(\frac{1}{q_h^2}\right)$	yes	$k+n+80$	1P+2E	1P+1E
Libert-Quisquater [12]	GBDH	CCA	$\mathcal{O}\left(\frac{1}{q_e}\right)$	yes	$k+n$	1P+2E	1P+0E
Katz-Wang [10]	BDH	CPA	$\mathcal{O}(1)$	yes	$2k+2n$	2P+4E	1P+1E
ACF+ [1]	DBDH	CCA	$\mathcal{O}(1)$	yes	$k+n+420$	2P+2E	1P+1E
Waters [15]	DBDH	CPA	$\mathcal{O}\left(\frac{1}{q_e}\right)$	no	$2k+n$	0P+4E	2P+0E
Gentry [9]	q_e-ABDHE	CCA	$\mathcal{O}(1)$	no	$5k+n$	0P+6E	2P+3E
Our scheme	D-Square-BDH	CCA	$\mathcal{O}(1)$	yes	$3k+2n$	1P+3E	1P+3E

P denotes a pairing operation, and E a group exponentiation in $\mathbb{G}$ or $\mathbb{G}_1$. Integer k denotes the bit-size of p, and integer n denotes the bit-size of the message space. Here p is the prime modulus used to define the underlying elliptic-curve. An element of $\mathbb{G}$ requires k bits (using point compression), whereas an element of $\mathbb{G}_1$ requires $2k$ bits

however, Boneh-Franklin is slightly more efficient (for the same security parameters), with a shorter ciphertext; moreover we use a stronger security assumption. Compared to the Katz and Wang scheme and the Attrapadung et al. scheme (ACF+), our scheme is slightly more efficient since only one pairing computation is required for encryption, instead of two; however, our scheme has a longer ciphertext; our scheme also uses a stronger assumption compared to Katz and Wang. Compared to Gentry, our scheme is more efficient for decryption (one pairing instead of two), but less efficient for encryption (one pairing instead of zero). However, our scheme is only proved secure in the random oracle model; but it has a more natural assumption, because our assumption is static and does not depend on the number of private-key queries.

In summary, our new IBE scheme compares favourably with other existing IBE schemes. Clearly, the main drawback of our scheme is that it is only provably secure in the random oracle model. The main advantage is that our scheme provides an easy upgrade from Boneh-Franklin to a tightly secure scheme; namely, we show in Sect. 6 that one can keep the same system parameters and that Boneh-Franklin private-keys can be upgraded offline by existing users.

We note that our security reduction is based on the Decisional Square Bilinear Diffie-Hellman (D-Square-BDH) assumption, which is a stronger assumption than the Computational Bilinear Diffie-Hellman (BDH) assumption used in Boneh-Franklin. Therefore, it could be misleading to claim that our scheme's tight reduction from D-Square-BDH is necessarily better than Boneh-Franklin's loose reduction from BDH; in other words, it is unclear whether or not a tighter reduction under a stronger assumption improves security.

However, nothing is known about the relative hardness of the D-Square-BDH problem and BDH problems, except that D-Square-BDH is not harder than BDH. Both problems could be equally hard, or D-Square-BDH could be much easier than BDH. Since the best known algorithm to solve both problems is to compute the discrete log, we assume in this paper that both problems are equally hard.

2 Definitions

First, we recall the formal definition of Identity-Based Encryption, following [3]; it consists of four algorithms: Setup, Keygen, Encrypt and Decrypt:

- Setup: the Setup algorithm takes as input a security parameter k and generates the system public parameters, denoted by *params*, and a private master-key denoted *master-key*.
- Keygen: the Keygen algorithm takes as input *params*, *master-key* and an identity $v \in \{0, 1\}^*$ and outputs a private key d_v for identity v. The Keygen algorithm may be probabilistic.
- Encrypt: the encryption algorithm takes as input a message m, an identity v and the system parameters *params* and returns a ciphertext c. The Encrypt algorithm may be probabilistic.
- Decrypt: the decryption algorithm takes as input *params*, a ciphertext c and a private-key d_v, and returns a plaintext m. The Decrypt algorithm must be deterministic.

These algorithms are required to satisfy a straightforward consistency constraint, namely for any identity v, if $d_v \leftarrow \mathsf{Extract}(params, v)$, then for any message m:

$$\mathsf{Decrypt}(\mathsf{Encrypt}(m, v, params), d_v, params) = m$$

2.1 Security of IBE

The IND-ID-CPA security model [3] for IBE ensures that even if the adversary can learn private-keys for identities of his choice, this does not help him in obtaining information about a plaintext encrypted for a different identity v^*; we recall the formal definition in Appendix A.

Similarly, the IND-ID-CCA security model [3] for IBE ensures that even if the adversary can additionally obtain the decryption of ciphertexts of his choice, this does not help him in obtaining information about a plaintext encrypted for a different identity v^*. This is the strongest notion of security for IBE, whose formal definition is recalled in Appendix A.

Our new construction will be proved secure in the random oracle model, as for the Boneh-Franklin scheme. The random oracle model, introduced by Bellare and Rogaway in [2], is a theoretical framework in which a hash function is seen as an oracle that outputs a random value for each new query. Actually, a security proof in the random oracle model does not necessarily imply that a scheme is secure in the real world (see [6]); this means that the random oracle model only provides heuristic security. Nevertheless, it seems to be a good engineering principle to design a scheme so that it is at least provably secure in the random oracle model; of course, it is always better to have a proof without random oracles.

3 Bilinear map and complexity assumptions

3.1 Bilinear map

The new construction proposed in this report is based on bilinear maps, defined as follows. Let $\mathbb{G}$ and $\mathbb{G}_1$ be two groups of order q for some large prime q. Throughout the paper we view both groups $\mathbb{G}$ and $\mathbb{G}_1$ a multiplicative groups. We say that $\mathbb{G}$ has an admissible bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$ if the following conditions hold:

1. Bilinear: $e(g^a, h^b) = e(g, h)^{ab}$ for all $g, h \in \mathbb{G}$ and all $a, b \in \mathbb{Z}$.
2. Non-degenerate: $e(g, g) \neq 1$ for some $g \in \mathbb{G}$.
3. Computable: there exists an efficient algorithm to compute $e(g, h)$ for any $g, h \in \mathbb{G}$.

The Weil pairing or the Tate pairing over certain class of elliptic curves are examples of admissible bilinear map. In this paper we only consider symmetric pairings; our constructions could also be used with more general types of pairings.

3.2 Computational assumptions

Let $\mathbb{G}$ and $\mathbb{G}_1$ be groups of prime order q and let e be an admissible bilinear map for $\mathbb{G}$ into $\mathbb{G}_1$. Let g be a generator of $\mathbb{G}$. The Bilinear Diffie-Hellman problem is defined as follows [3]:

Definition 1 (*Bilinear Diffie-Hellman Problem* (*BDH*)) Given the 4-uple (g, g^a, g^b, g^c) where $a, b, c \leftarrow \mathbb{Z}_q$, output $e(g, g)^{abc}$.

Definition 2 (*BDH assumption*) We say that the BDH problem is (t, ε)-hard in $\mathbb{G}$ if no t-time algorithm can solve the BDH problem with probability at least ε.

The decisional version is defined in the usual manner:

Definition 3 (*Decisional Bilinear Diffie-Hellman Problem* (*DBDH*)) Let g, g^a, g^b, g^c defined as previously. Let β be a random binary coin. Let $z = e(g, g)^{abc}$ if $\beta = 1$, and let z be a random element in $\mathbb{G}_1$ otherwise. Given (g, g^a, g^b, g^c, z), output a guess β' of β.

The Decisional Square Bilinear Diffie-Hellman Problem (D-Square-BDH) is a variant of DBDH, introduced by Kiltz in [11]:

Definition 4 (*Decisional Square Bilinear Diffie-Hellman* (*D-Square-BDH*)) Let g, g^a, g^b defined as previously. Let β be a random binary coin. Let $z = e(g, g)^{a^2 b}$ if $\beta = 1$, and let z be a random element in $\mathbb{G}_1$ otherwise. Given (g, g^a, g^b, z), output a guess β' of β.

We say that an algorithm as an advantage ε in solving DBDH or D-Square-BDH if

$$\left| \Pr[\beta' = \beta] - \frac{1}{2} \right| \geq \varepsilon$$

Definition 5 (*D-Square-BDH assumption*) We say that the D-Square-BDH problem is (t, ε)-hard in $\mathbb{G}$ if no t-time algorithm has an advantage at least ε in solving the D-Square-BDH problem in $\mathbb{G}$.

It is shown in [11] that the D-Square-BDH assumption is a stronger assumption than the DBDH assumption.

When e is the Weil pairing or the Tate pairing over an elliptic-curve, no efficient algorithms are known for solving the BDH, DBDH and D-Square-BDH problems.

4 The Boneh-Franklin scheme

In the following, we recall the Boneh-Franklin IBE scheme [3]. We first recall the basic Boneh-Franklin scheme, referred to as BasicIdent. The basic scheme achieves security against passive adversaries (IND-ID-CPA) but *not* against chosen-ciphertext attacks (IND-ID-CCA). Then we recall the FullIdent Boneh-Franklin scheme that achieves IND-ID-CCA security.

4.1 BasicIdent

Let $\mathbb{G}$ be a group of prime order q, let g be a generator of $\mathbb{G}$, and let e be an admissible bilinear map into $\mathbb{G}_1$. Let $H_1 : \{0, 1\}^* \to \mathbb{G}$ and $H_2 : \mathbb{G}_1 \to \{0, 1\}^n$ be hash-functions, for some parameter n. The message space is $\mathcal{M} = \{0, 1\}^n$. The hash functions H_1 and H_2 are viewed as random oracles in the security proof.

Setup: A secret $a \in \mathbb{Z}_q$ is chosen at random. One sets $h = g^a$. The public parameters are g, h. The master secret is a.

Keygen: Let $v \in \{0, 1\}^*$ be an identity. Given the master-key a, the private-key d_v for identity v is computed as:

$$d_v = H_1(v)^a$$

Encryption: A message $m \in \{0, 1\}^n$ is encrypted for identity v as follows. A random $r \in \mathbb{Z}_q$ is generated; the ciphertext is then:

$$C = \left(g^r,\ m \oplus H_2\left(e(H_1(v), h)^r \right) \right)$$

Decryption: To decrypt a ciphertext $C = (c_1, c_2)$ using private-key $d_v = H_1(v)^a$, compute:

$$m = H_2\left(e(d_v, c_1) \right) \oplus c_2$$

This completes the description of the BasicIdent scheme. We note that decryption works because

$$e(H_1(v), h)^r = e(H_1(v), g^a)^r = e(H_1(v)^a, g^r) = e(d_v, c_1)$$

Theorem 1 (Boneh-Franklin) *The* BasicIdent *scheme is a* $(t, q_h, q_e, \varepsilon)$ *semantically-secure IBE scheme (*IND-ID-CPA*) if the BDH problem is* (t', ε')*-hard on* $\mathbb{G}$, $\mathbb{G}_1$, *where:*

$$t = \mathcal{O}(t')$$
$$\varepsilon = \frac{e}{2} \cdot (1 + q_e) \cdot q_h \cdot \varepsilon'$$

where $e \simeq 2.71$ *is the base of the natural logarithm,* q_h *the number of hash queries and* q_e *the number of private-key queries.*

We observe that the security of the basic Boneh-Franklin scheme is *not* tightly related to the security of the underlying BDH problem. Namely, if we assume that no attacker can solve the BDH problem with probability at least ε' in a given amount of time, then the probability to break BasicIdent in roughly the same amount of time is only upper-bounded by $\varepsilon \simeq q_e \cdot q_h \cdot \varepsilon'$. This implies that the probability of breaking Boneh-Franklin can be much higher than the probability of breaking the elliptic-curve. For example, if we assume that the adversary makes at most $q_e = 2^{30}$ private-key queries and $q_h = 2^{60}$ hash-queries, the probability to break BasicIdent is upper-bounded by $\varepsilon \simeq 2^{90} \cdot \varepsilon'$. Therefore, even if no adversary can solve BDH with probability greater than $\varepsilon' = 2^{-91}$, we only obtain that the probability to break BasicIdent is bounded by $\varepsilon = 1/2$, which is clearly insufficient. In other words, $30 + 60 = 90$ bits of security are lost compared to the security provided by the underlying elliptic-curve. To obtain a meaningful security guarantee, one must therefore increase the parameter size so that a smaller probability ε' of solving BDH can be assumed. However, this decreases the scheme's efficiency.

4.2 FullIdent

The BasicIdent scheme is converted into a chosen-ciphertext secure scheme FullIdent using a technique due to Fujisaki and Okamoto [7]; one obtains the following scheme:
Setup: Identical to the BasicIdent scheme. A secret $a \in \mathbb{Z}_q$ is chosen at random. One sets $h = g^a$. The public parameters are g, h. The master secret is a. Additionally, hash functions $H_3 : \{0,1\}^n \times \{0,1\}^n \to \mathbb{Z}_q^*$ and $H_4 : \{0,1\}^n \to \{0,1\}^n$ are used.
Keygen: Identical to the BasicIdent scheme. Let v be an identity. Given the master-key a, the private-key d_v for identity v is computed as:

$$d_v = H_1(v)^a$$

Encryption: A message m is encrypted for identity v as follows.

1. Generate a random $\sigma \in \{0,1\}^n$
2. Set $r = H_3(\sigma, m)$
3. The ciphertext is then:

$$C = \left(g^r,\ \sigma \oplus H_2\left(e(H_1(v), h)^r\right),\ m \oplus H_4(\sigma)\right)$$

Decryption: To decrypt a ciphertext $C = (c_1, c_2, c_3)$ using private-key d_v

1. Compute $c_2 \oplus H_2\left(e(d_v, c_1)\right) = \sigma$
2. Compute $c_3 \oplus H_4(\sigma) = m$

3. Set $r = H_3(\sigma, m)$. Test if $c_1 = g^r$. If not, reject the ciphertext.
4. Output m as the decryption of C.

Theorem 2 (Boneh-Franklin) *The* FullIdent *scheme is a* $(t, q_h, q_e, q_d, \varepsilon)$-IND-ID-CCA *secure IBE scheme in the random oracle model, assuming that the BDH problem is* (t', ε')-*hard in* $\mathbb{G}$*, where:*

$$\varepsilon = e \cdot (1 + q_e + q_d) \cdot \left(\frac{4q_d}{q} + (q_h)^2 \cdot \varepsilon' \right)$$
$$t = \mathcal{O}(t')$$

where q *is the group order,* q_h *is the number of hash queries,* q_e *is the number of private-key queries, and* q_d *is the number of decryption queries.*

As previously, we note that the FullIdent Boneh-Franklin scheme does not achieve a tight security, as there is a security loss factor of roughly $(q_e + q_d) \cdot q_h^2$. In particular, the security loss is a function of the number of issued private-keys. This means that if the adversary performs at most $q_e = 2^{30}$ private-key queries, $q_d = 2^{30}$ decryption queries and $q_h = 2^{60}$ hash queries, then $1 + 30 + 2 \cdot 60 = 151$ bits of security are lost compared to the underlying elliptic-curve. As previously, this security loss must be compensated by selecting larger security parameters. The goal of the new construction in the next section is to eliminate this security loss, so as to obtain a variant of Boneh-Franklin that is as secure as the underlying elliptic-curve.

5 The new construction

In the following, we describe our new IBE scheme. As for the Boneh-Franklin scheme, for clarity of exposition, we first describe a basic version that achieves only IND-ID-CPA security; then we describe the fully secure IND-ID-CCA construction.

5.1 Basic construction

The parameters are the same as in the basic Boneh-Franklin IBE scheme: let $\mathbb{G}$ be a group of prime order q and g be a generator of $\mathbb{G}$, and let e be an admissible bilinear map into $\mathbb{G}_1$. Let $H_1 : \{0,1\}^* \to \mathbb{G}$ and $H_2 : \mathbb{G}_1 \to \{0,1\}^n$ for some n, be hash-functions. The message space is $\mathcal{M} = \{0,1\}^n$. The hash functions H_1 and H_2 are viewed as random oracles in the security proof.

Setup: Identical to the basic Boneh-Franklin scheme. A secret $a \in \mathbb{Z}_q$ is chosen at random. One sets $h = g^a$. The public parameters are g, h. The master secret is a.

Keygen: Let v be an identity. Let y be random in $\mathbb{Z}_q$. Given the master-key a, the private key d_v for identity v is computed as:

$$d_v = \left(\left(H_1(v) \cdot h^{-y} \right)^a , \ y \right)$$

If the same query for identity v is repeated twice, then the same private-key d_v is provided, *i.e.* the same value for y is used. This can be done by storing (v, y) in a database, or by using a keyed hash-function for generating y.

Encryption: A message m is encrypted for identity v as follows. A random $r \in \mathbb{Z}_q$ is generated; the ciphertext is then:

$$C = \left(g^r , \ e(h,h)^r , \ m \oplus H_2 \left(e(H_1(v), h)^r \right) \right)$$

Decryption: To decrypt a ciphertext $C = (c_1, c_2, c_3)$ using private-key $d_v = (\delta, y)$, compute:

$$m = c_3 \oplus H_2\left(e(\delta, c_1) \cdot (c_2)^y\right)$$

This completes the description of our new scheme. We note that decryption works because:

$$\begin{aligned} e(H_1(v), h)^r &= e(H_1(v), g^a)^r = e(H_1(v)^a, g^r) = e(\delta \cdot h^{y \cdot a}, g^r) = e(\delta, g^r) \cdot e(h^{ya}, g^r) \\ &= e(\delta, c_1) \cdot e(h, g^a)^{ry} = e(\delta, c_1) \cdot e(h, h)^{ry} = e(\delta, c_1) \cdot (c_2)^y \end{aligned}$$

The difference with the BasicIdent Boneh-Franklin scheme is that we add a "randomisation" of the private-key d_v by multiplying the $H_1(v)^a$ element by h^{-ya}, for a random $y \in \mathbb{Z}_q$. The Boneh-Franklin scheme can then be seen as a particular case with $y = 0$. The following theorem shows that our new construction achieves IND-ID-CPA security in the random oracle model, with a tight security.

Theorem 3 *The previous scheme is* $(t,\ q_h,\ q_e,\ \varepsilon)$-IND-ID-CPA *secure in the random oracle model, assuming that the D-Square-BDH problem is* $(t',\ \varepsilon')$*-hard in the group* $\mathbb{G}$*, where:*

$$\varepsilon = 2 \cdot \varepsilon' + \frac{q_h}{q} \tag{1}$$

$$t = \mathcal{O}(t') \tag{2}$$

where q_h *is the number of hash-queries,* q_e *the number of private-key queries and* q *the group order.*

Proof Assume that there exists a $(t,\ q_h,\ q_e,\ \varepsilon)$ adversary $\mathcal{A}$ against the basic scheme. We construct a simulator $\mathcal{B}$ that solves the D-Square-BDH problem with advantage at least ε' while interacting with $\mathcal{A}$.

Setup: The simulator $\mathcal{B}$ receives the D-Square-BDH challenge $(g, A = g^a, B = g^b, T)$ and must output a guess β' as to whether $T = e(g, g)^{a^2 b}$ (when $\beta = 1$) or T is uniformly distributed in $\mathbb{G}_1$ (when $\beta = 0$). The simulator first sets $h = A$ and sends (g, h) to the adversary as the system public parameters. Note that $h = g^a$ but the corresponding master-key a is unknown to $\mathcal{B}$.

Hash-queries: When $\mathcal{A}$ submits a fresh hash-query for $H_1(v)$, the simulator $\mathcal{B}$ generates two randoms x and y in $\mathbb{Z}_q$; it stores (v, x, y) in a table and returns:

$$H_1(v) = g^x \cdot h^y \tag{3}$$

When $\mathcal{A}$ submits a fresh hash-query for H_2, the simulator $\mathcal{B}$ returns a random element in $\{0, 1\}^n$.

Phase 1: We assume that when $\mathcal{A}$ submits a private-key query for identity v, it has already made a query for $H_1(v)$. If this is not the case, $\mathcal{B}$ can simulate this hash-query before answering the corresponding private-key query. When $\mathcal{A}$ submits a private-key query for identity v, the simulator $\mathcal{B}$ can therefore recover (x, y) such that $H_1(v) = g^x \cdot h^y$. It lets $\delta = h^x$ and returns:

$$d_v = (\delta,\ y)$$

as a private-key for identity v. Note that this is a valid private-key for v because using (3) we have

$$\delta = h^x = g^{ax} = \left(H_1(v) \cdot h^{-y}\right)^a$$

as required and y is uniformly distributed in $\mathbb{Z}_q$. Observe also that our simulator $\mathcal{B}$ can always answer the private-key queries made by the adversary.

Challenge: the adversary submits an identity v^* and two messages m_0 and m_1. Here we also assume that $\mathcal{A}$ has already done a hash-query for v^*; in this is not the case, $\mathcal{B}$ can simulate this hash-query by himself. Therefore, the simulator knows (x^*, y^*) such that

$$H_1(v^*) = g^{x^*} \cdot h^{y^*}$$

and can compute a private key $d_{v^*} = (\delta^*, y^*)$ where $\delta^* = h^{x^*}$. The simulator then flips a fair binary coin γ and returns the following ciphertext:

$$C = \left(B,\ T,\ m_\gamma \oplus H_2(\alpha^*)\right) \tag{4}$$

where

$$\alpha^* = e(\delta^*, B) \cdot T^{y^*} \tag{5}$$

Phase 2: the simulator answers private-key queries as in phase 2.

Guess: the adversary outputs a guess γ' for γ. If $\gamma = \gamma'$, the simulator $\mathcal{B}$ answer $\beta' = 1$. Otherwise, it answers $\beta' = 0$.

This terminates the description of the simulator. In the following, we show that when (g, A, B, T) is a legitimate D-Square-BDH tuple (*i.e.*, $\beta = 1$), the adversary's view has exactly the same distribution as in the original attack scenario. Since the adversary is assumed to $(t, q_h, q_e, \varepsilon)$-break the scheme, we have:

$$\left|\Pr[\gamma' = \gamma | \beta = 1] - \frac{1}{2}\right| \geq \varepsilon$$

When $\beta = 1$, we have that $\gamma' = \gamma \Leftrightarrow \beta' = \beta$; this implies:

$$\left|\Pr[\beta' = \beta | \beta = 1] - \frac{1}{2}\right| \geq \varepsilon \tag{6}$$

On the other hand, we show that when $T \neq e(g, g)^{a^2 b}$, then conditioned on the adversary's view except $c_3 = m_\gamma \oplus H_2(\alpha^*)$, the random variable:

$$\alpha^* = e(\delta^*, B) \cdot T^{y^*}$$

has the uniform distribution in $\mathbb{G}_1$. Therefore, the adversary makes a H_2-query for α^* with probability at most q_h/q (where q is the group order). If the adversary has not made a H_2 query for α^*, then $H_2(\alpha^*)$ has the uniform distribution in $\{0, 1\}^n$; the message m_γ in $c_3 = m_\gamma \oplus H_2(\alpha^*)$ is then perfectly masked and the adversary obtains no information about γ. Therefore, we obtain:

$$\left|\Pr[\gamma' \neq \gamma | \beta = 0] - \frac{1}{2}\right| \leq \frac{q_h}{q}$$

which gives:

$$\left|\Pr[\beta' = \beta | \beta = 0] - \frac{1}{2}\right| \leq \frac{q_h}{q} \tag{7}$$

Combining (6) and (7), we obtain:

$$\left|\Pr[\beta' = \beta] - \frac{1}{2}\right| \geq \frac{\varepsilon}{2} - \frac{q_h}{2q}$$

This shows that from an adversary that ε-breaks the scheme, one can build an algorithm that ε'-solves D-Square-BDH, with $\varepsilon' = \varepsilon/2 - q_h/(2q)$, which gives (1).

We now proceed to prove the two previous claims. First, we write $T = e(h,h)^s$ for some $s \in \mathbb{Z}_q$; writing $B = g^r$, we obtain from (5):

$$\alpha^* = e(\delta^*, B) \cdot T^{y^*} = e(h^{x^*}, g^r) \cdot e(h,h)^{sy^*} = e(g^{x^*}, h)^r \cdot e(h,h)^{sy^*}$$
$$\alpha^* = e(H_1(v^*) \cdot h^{-y^*}, h)^r \cdot e(h,h)^{sy^*} = e(H_1(v^*), h)^r \cdot e(h,h)^{y^* \cdot (s-r)}$$

First, we show that that when (g, A, B, T) is a legitimate D-Square-BDH tuple, the adversary's view has exactly the same distribution as in the original attack scenario. Namely in this case we have that $T = e(g,g)^{a^2 b} = e(g,g)^{a^2 r} = e(h,h)^r$ which gives $s = r$; the challenge ciphertext is then equal to:

$$C = \left(g^r,\ e(h,h)^r,\ m_b \oplus H_2\left(e(H_1(v^*), h)^r\right)\right)$$

which shows that C is a legitimate encryption of m_b under identity v^*. This proves the first claim.

Now we consider the case when (g, A, B, T) is not a legitimate D-Square-BDH tuple, that is $s \neq r$. We consider that all the random variables that appear in the adversary's view are fixed, except c_3. Note that since T is part of the ciphertext, we have that s is fixed. Moreover we have that h, r and $H_1(v^*)$ are fixed. We claim the variable y^* still has the uniform distribution in $\mathbb{Z}_q$. Namely, since the adversary is not allowed to make a private-key query for v^*, the variable y^* only appears in the adversary's view with $H_1(v^*) = g^{x^*} \cdot h^{y^*}$ where x^* and y^* are randomly generated in $\mathbb{Z}_q$. This implies that for a fixed $H_1(v^*)$, the variable:

$$y^* = \log_h(H_1(v^*) \cdot g^{-x^*}) = \log_h H_1(v^*) - x^*/a$$

has the uniform distribution in $\mathbb{Z}_q$. This in turn implies that with $s \neq r$,

$$\alpha^* = e(H_1(v^*), h)^r \cdot e(h,h)^{y^* \cdot (s-r)}$$

has the uniform distribution in $\mathbb{G}_1$, conditioned on the adversary's view except c_3. This proves the second claim and terminates the proof. □

5.2 Discussion

First, we note that hash function H_2 can be eliminated by having the message m lie in the group $\mathbb{G}_1$ and computing $m \cdot e(H_1(v), h)^r$ instead.

Theorem 3 shows that if no algorithm can solve the D-Square-BDH problem with probability at least ε', then the probability to break the new construction is upper-bounded by $\varepsilon \simeq 2 \cdot \varepsilon'$. This shows that as opposed to the Boneh-Franklin scheme, there is no security loss: if the probability to solve D-Square-BDH is assumed to be at most 2^{-91}, then the probability of breaking the new IBE scheme is upper-bounded by 2^{-90}. For the same level of security, one can now use smaller parameters than for Boneh-Franklin, which improves efficiency.

5.3 A variant with tight security under DBDH assumption

The following variant was suggested by one of the referees:
Setup: Let $a \leftarrow \mathbb{Z}_q$ and $b \leftarrow \mathbb{Z}_q$. Let $h = g^a$ and $t = g^b$. Public parameters are (g, h, t). Master secret key is: a
Keygen: Let $y \leftarrow \mathbb{Z}_q$, unique for a given identity v.

$$d_v = \left(\left(H_1(v) \cdot t^{-y}\right)^a,\ y\right)$$

Encryption: Let $r \leftarrow \mathbb{Z}_q$. The encryption of m is:

$$C = \left(g^r,\ e(t,h)^r,\ m \oplus H_2\left(e(H_1(v),h)^r\right)\right)$$

Decryption: Given $C = (c_1, c_2, c_3)$ and $d_v = (\delta, y)$, let:

$$m = c_3 \oplus H_2\left(e(\delta, c_1) \cdot (c_2)^y\right)$$

Using the same approach as in Theorem 3, one can show that this variant as a tight security proof under the DBDH assumption instead of the D-Square-BDH assumption. Therefore, the advantage of this variant is that it is based on a weaker assumption; a (minor) drawback is that it has a longer public parameters.

5.4 Chosen-ciphertext secure construction

In this section, we construct an IBE scheme that achieves CCA security, in the random oracle model. As for the Boneh-Franklin IBE scheme, we use the Fujisaki-Okamoto [7] generic conversion. Note that we cannot use the generic approach in [16], because it does not provide a tight security reduction.

Setup: Identical to our basic scheme. A secret $a \in \mathbb{Z}_q$ is chosen at random. One sets $h = g^a$. The public parameters are g, h. The master secret is a. Additionally, hash functions $H_3 : \{0,1\}^n \times \{0,1\}^n \rightarrow \mathbb{Z}_q^*$ and $H_4 : \{0,1\}^n \rightarrow \{0,1\}^n$ are used.

Keygen: Identical to our basic scheme. Let v be an identity. Let y be random in $\mathbb{Z}_q$. Given the master-key a, the private key d_v for identity v is computed as:

$$d_v = \left(\left(H_1(v) \cdot h^{-y}\right)^a,\ y\right)$$

As previously, when the same identity v is queried twice, the same private-key d_v is provided.

Encryption: A message m is encrypted for identity v as follows.

1. Generate a random $\sigma \in \{0,1\}^n$
2. Set $r = H_3(\sigma, m)$
3. The ciphertext is then:

$$C = \left(g^r,\ e(h,h)^r,\ \sigma \oplus H_2\left(e(H_1(v),h)^r\right),\ m \oplus H_4(\sigma)\right)$$

Decryption: To decrypt a ciphertext $C = (c_1, c_2, c_3, c_4)$ using private-key $d_v = (d_1, d_2)$, one computes:

$$\alpha = e(d_1, c_1) \cdot (c_2)^{d_2} \tag{8}$$

$$\sigma = c_3 \oplus H_2(\alpha) \tag{9}$$

$$m = c_4 \oplus H_4(\sigma) \tag{10}$$

$$r = H_3(\sigma, m) \tag{11}$$

Then it is checked that $c_1 = g^r$ and $c_2 = e(h,h)^r$. If not, the ciphertext gets rejected. Otherwise, m is returned as the decryption of C.

This completes the description of the new IBE scheme. The following theorem shows that this new construction achieves IND-ID-CCA security in the random oracle model, with security tightly related to the D-Square-BDH problem. The proof is given in Appendix B.

Theorem 4 *The previous scheme is* $(t, q_h, q_e, q_d, \varepsilon)$-IND-ID-CCA *secure in the random oracle model, assuming that the D-Square-BDH problem is* (t', ε')*-hard in the*

group $\mathbb{G}$*, where:*

$$\varepsilon = 2 \cdot \varepsilon' + \frac{4q_h + 4q_d}{q}$$
$$t = \mathcal{O}(t')$$

6 Upgrading from Boneh-Franklin to the new construction

As mentioned previously, the Boneh-Franklin private-keys $H_1(v)^a$ are identical to the new construction private-keys with $y = 0$. This implies that given a Boneh-Franklin private-key $H_1(v)^a$, one can generate the private-key:

$$d_v = (H_1(v)^a, 0)$$

which a valid private-key for the new construction. This enables an easy upgrade from Boneh-Franklin to our new scheme. Namely, the system public parameter need not be changed and the PKG does not need to distribute new private-keys to already registered users, since those users can upgrade their private-keys by themselves.

More precisely, assume that an IBE system based on Boneh-Franklin has been implemented. The PKG has already published the system parameters (g, h), and has already sent to registered users their Boneh-Franklin private-keys d_v. The IBE system can then be upgraded to our new construction as follows:

1. Users convert their existing private-key d_v into $d'_v = (d_v, 0)$, which is a valid private-key for the new construction.
2. Users are instructed to send ciphertexts using the new construction instead of Boneh-Franklin.
3. The PKG now distributes private-keys according to the new construction.

The advantage of the upgrade is that the security level is now independent of the number of newly issued private-keys.

7 Implementation and comparison with Boneh-Franklin

We have implemented the Boneh-Franklin scheme and our new construction, using the Weil pairing on the elliptic-curve $y^2 = x^3 + 1$ over $\mathbb{F}_p$ as in [3]. We summarise in Table 2 the observed timings with a prime p of size 512 bits and a subgroup size of 164 bits.

From Table 2 we observe that the difference in timing between Boneh-Franklin and our new scheme is quite negligible: our new scheme is only 2% slower than Boneh-Franklin. This is due to the fact that the additional operation required in our scheme—exponentiation in $\mathbb{G}_1$—takes a negligible amount of time compared to the main pairing operation. We note that this holds for the particular curve that we have chosen and for our particular implementation; a more efficient implementation over a possibly different curve could exhibit a less favourable ratio between exponentiation cost and pairing cost. With $k = 512$ and $n = 160$, ciphertext size in Boneh-Franklin is 1344 bits while ciphertext size in our construction is 1856 bits.

Table 2 Timings observed for a C++ implementation of Boneh-Franklin and the new scheme, on a 1.55 GHz laptop

Operation	Time
Pairing (P)	1.05 s
Exponentiation in $\mathbb{G}$ (G)	0.05 s
Exponentiation in $\mathbb{G}_1$ (E)	0.01 s
Hash into $\mathbb{G}$ (H)	0.08 s
Encryption (Boneh-Franklin)	1.18 s
Decryption (Boneh-Franklin)	1.09 s
Encryption (new scheme)	1.19 s
Decryption (new scheme)	1.11 s

8 Conclusion

We have described a variant of Boneh-Franklin IBE with a tight security reduction in the random oracle model. This enables to select smaller security parameters, which in turn improves the scheme's efficiency. Our new scheme is quite efficient compared to existing IBE schemes. The main advantage is that our scheme provides an easy upgrade from Boneh-Franklin to a tightly secure IBE scheme; namely, we have shown that one can keep the same system parameters and that Boneh-Franklin private-keys can be upgraded offline by existing users.

Acknowledgments I wish to thank the anonymous referees for many useful comments. Work supported by Oberthur Card Systems contract ref. OCS-2005-607.

Appendix A: Security of IBE

A.1 IND-ID-CPA security

The IND-ID-CPA security model is formalised using the following game between an attacker and a challenger [3]:

Setup: the challenger generates the system public parameters and gives them to the adversary. The challenger keeps the corresponding master-key for himself.

Phase 1: the adversary can request the private-key corresponding to an identity v of his choice. The adversary can repeat this multiple times for different identities. The challenger answers the private-key queries using the master-key. These queries can be asked adaptively, that is, each query may depend on the previous replies.

Challenge: the adversary submits an identity v^*, different from the identities in phase 1, and two messages m_0 and m_1. The challenger flips a binary coin γ and returns the encryption c^* of m_γ under identity v^*.

Phase 2: phase 1 is repeated with the restriction that the adversary cannot request the private key for v^*. As in phase 1, the queries may be done adaptively.

Guess: eventually, the adversary submits a guess γ' of γ and wins the game if $\gamma' = \gamma$.

This completes the description of the scenario. We refer to such an adversary $\mathcal{A}$ as an IND-ID-CPA adversary (for Indistinguishability under a Chosen Plaintext Attack). The

adversary A's advantage in breaking the scheme is defined as:

$$\mathsf{Adv}^{\mathcal{A}} = \left| \Pr[\gamma' = \gamma] - \frac{1}{2} \right|$$

where the probability is taken over the random bits used by the adversary and the challenger.

Definition 6 (*IND-ID-CPA*) An IBE scheme is said to be (t, q, ε)-semantically secure against passive adversaries if all t-time adversaries making at most q private key queries have an advantage at most ε in breaking the scheme.

A.2 IND-ID-CCA security

The IND-ID-CCA security model is formalised using the following scenario between an attacker and a challenger [3]:
Setup: as previously, the challenger generates the system public parameters and gives them to the adversary. The challenger keeps the corresponding master-key for himself.
Phase 1: as previously, the adversary can request the private-key corresponding to an identity v of his choice. Additionally, the adversary may request the decryption of any ciphertext of his choice, for an identity v of his choice. Both types of queries can be done adaptively.
Challenge: the adversary submits an identity v^* for which no private-key query was done is Phase 1, and two messages m_0 and m_1. The challenger flips a binary coin γ and returns the encryption c^* of m_γ under identity v^*.
Phase 2: phase 1 is repeated with the restriction that the adversary cannot request a private key for v^*, and cannot request the decryption of ciphertext c^* for identity v^*. As in phase 1, the queries may be done adaptively.
Guess: eventually, the adversary submits a guess γ' of γ and wins the game if $\gamma' = \gamma$.

This completes the description of the scenario. We refer to such an adversary $\mathcal{A}$ as an IND-ID-CCA adversary (for Indistinguishability under a Chosen Ciphertext Attack). As previously, the adversary A's advantage in breaking the scheme is defined as:

$$\mathsf{Adv}^{\mathcal{A}} = \left| \Pr[\gamma' = \gamma] - \frac{1}{2} \right|$$

where the probability is taken over the random bits used by the adversary and the challenger.

Definition 7 (*IND-ID-CCA*) An IBE cryptosystem is said to be $(t, q_e, q_d, \varepsilon)$-semantically secure against passive adversaries if all t-time adversaries making at most q_e private key queries and q_d decryption queries have an advantage at most ε in breaking the scheme.

Appendix B: Proof of Theorem 4

As in [14], we describe a sequence of attacks games that each operate on the same underlying probability space. Game G_0 corresponds to the original IND-ID-CCA scenario between the attacker and the challenger, while Games $\mathsf{G}_1, \ldots, \mathsf{G}_7$ are modified games in which the adversary's view gets modified.

We use the following Difference lemma to bound the probability of certain events defined over the successive games:

Lemma 1 (Shoup) *Let E, E' and F be events defined on a probability space such that* $\Pr[E \wedge \neg F] = \Pr[E' \wedge \neg F]$. *Then we have*

$$|\Pr[E] - \Pr[E']| \leq \Pr[F].$$

Proof We have:

$$\begin{aligned} |\Pr[E] - \Pr[E']| &= |\Pr[E \wedge F] + \Pr[E \wedge \neg F] - \Pr[E' \wedge F] - \Pr[E' \wedge \neg F]| \\ &\leq |\Pr[E \wedge F] - \Pr[E' \wedge F]| \leq \Pr[F] \end{aligned}$$

□

We first begin with a few notations. Any ciphertext $C = (c_1, c_2, c_3, c_4)$ for identity v implicitly defines values α, σ, m, r via the decryption Eqs. 8–11. We let C^* be the challenge ciphertext for identity v^* and $\alpha^*, \sigma^*, m^*, r^*$ be the corresponding implicitly defined values, with $m^* = m_\gamma$.

We define the adversary's *view* as the sequence of random variables:

$$View = (\omega, params, X_1, \ldots X_{q_d+q_e+q_h})$$

where ω is the adversary's random tape, $params$ are the system public parameters and the X_i for $i \geq 1$ are a response to either a random oracle query, a decryption oracle query, or the challenge ciphertext itself. We also define

$$CurrentView = (\omega, params, X_1, \ldots, X_m)$$

at any fixed point of time when the adversary has made m queries.

Game G_0: the adversary and the challenger interact exactly as in the attack scenario. Let denote by S_0 the event that $\gamma' = \gamma$. Similarly we denote by $S_1, \ldots, S_7$ the event that $\gamma' = \gamma$ in games $\mathsf{G}_1, \ldots, \mathsf{G}_7$, respectively. We denote by S_{H_2} the list of queries made by the adversary to H_2.

Game G_1: one proceeds as in G_0, except that the decryption oracle now rejects a ciphertext C if $(c_1, c_2) = (c_1^*, c_2^*)$ and $\alpha = \alpha^*$. More precisely, given a ciphertext C, the decryption oracle computes α, σ, m and r as in G_0, and rejects the ciphertext if $c_1 \neq g^r$ or $c_2 \neq e(h, h)^r$ as in G_0; additionally, it rejects the ciphertext if $(c_1, c_2) = (c_1^*, c_2^*)$ and $\alpha = \alpha^*$. Otherwise, it outputs m as in G_0.

Let F_1 be the event that a ciphertext is rejected by the decryption oracle in G_1 that would not have been rejected under G_0. We have that G_0 and G_1 proceed identically unless event F_1 occurs, which implies $\Pr[S_0 \wedge \neg F_1] = \Pr[S_1 \wedge \neg F_1]$.

We now proceed to bound $\Pr[F_1]$. Let C be any ciphertext submitted to the decryption oracle. We first assume that the encryption oracle as already been queried before this decryption oracle query, so $C \neq C^*$; then if $(c_1, c_2) = (c_1^*, c_2^*)$, we must have $(\sigma, m) \neq (\sigma^*, m^*)$. If ciphertext C is not rejected under G_0, we must have $c_1 = g^r$ and $c_2 = e(h, h)^r$, which gives $r = r^*$. Then $H_3(\sigma, m) = H_3(\sigma^*, m^*)$ which happens with probability at most q_h/q over the course of the attack. If the encryption oracle has not been queried yet, then $(c_1, c_2) = (c_1^*, c_2^*)$ with probability at most $1/q$. We conclude that $\Pr[F_1] \leq q_h/q$ and then from the Difference lemma:

$$|\Pr[S_1] - \Pr[S_0]| \leq (q_h + q_d)/q \tag{12}$$

Game G_2: we proceed as in game G_1, except that instead of letting $c_3^* \leftarrow \sigma^* \oplus H_2(\alpha^*)$, the encryption oracle lets $c_3^* \leftarrow \sigma^* \oplus Y$ where $Y \leftarrow \{0, 1\}^n$. When the adversary or the decryption oracle makes a H_2-query for α^*, one returns Y instead of $H_2(\alpha^*)$. It is clear that games G_1 and G_2 are identical, so:

$$\Pr[S_2] = \Pr[S_1] \tag{13}$$

Game G_3: we proceed as in game G_2, except that if the adversary or the decryption oracle makes a H_2-query for α^*, we return $Y' \leftarrow \{0, 1\}^n$ instead of Y. Let F_3 the event that the

adversary or the decryption oracle makes a H_2-query to α^*; we have that games G_2 and G_3 proceed identically unless event F_3 occurs, which implies $\Pr[S_2 \wedge \neg F_3] = \Pr[S_3 \wedge \neg F_3]$, which gives:

$$|\Pr[S_3] - \Pr[S_2]| \leq \Pr[F_3] \tag{14}$$

The probability of event F_3 will be bounded in the next games.

Game G_4: we proceed as in game G_3, except that we let $c_4^* \leftarrow m^* \oplus Z$ where $Z \leftarrow \{0,1\}^n$ instead of $c_4^* \leftarrow m^* \oplus H_4(\sigma^*)$; moreover, we let $r^* \leftarrow \mathbb{Z}_q$ instead of $r^* \leftarrow H_3(\sigma^*, m^*)$. When the adversary or the decryption oracle make a H_4-query for σ^*, we return Z instead of $H_4(\sigma^*)$; similarly, when the adversary or the decryption oracle make a H_3-query for (σ^*, m^*), we return r^* instead of $H_3(\sigma^*, m^*)$. It is clear that games G_3 and G_4 are identical, so:

$$\Pr[S_4] = \Pr[S_3] \tag{15}$$

Moreover, letting F_4 be the event in game G_4 that the adversary or the decryption oracle make a H_2-query to α^*, we have that

$$\Pr[F_4] = \Pr[F_3] \tag{16}$$

Game G_5: we proceed as in game G_4, except that when the adversary or the decryption oracle makes a H_4-query for σ^*, we return Z' where $Z' \leftarrow \{0,1\}^n$ instead of Z. Similarly, when the adversary or the decryption oracle makes a H_3-query for (σ^*, m^*), we return r' where $r' \leftarrow \mathbb{Z}_q$ instead of r^*.

To summarise, the challenge ciphertext C^* for m^* is now:

$$C^* = \left(g^{r^*},\ e(h,h)^{r^*},\ \sigma^* \oplus Y,\ m^* \oplus Z\right)$$

where $\sigma^* \leftarrow \{0,1\}^n$, $r^* \leftarrow \mathbb{Z}_q$, $Y \leftarrow \{0,1\}^n$ and $Z \leftarrow \{0,1\}^n$. The value α^* is still computed as:

$$\alpha^* = e(H_1(v^*), h)^{r^*}$$

Finally, the decryption oracle Decryption' proceeds as follows:

Decryption':

Input: A ciphertext $C = (c_1, c_2, c_3, c_4)$ and a private-key $d_v = (d_1, d_2)$.

Output: A message m or $\perp$.

1. Compute $\alpha = e(d_1, c_1) \cdot (c_2)^{d_2}$.
2. If $(c_1, c_2) = (c_1^*, c_2^*)$ and $\alpha = \alpha^*$, return $\perp$.
3. If $\alpha \neq \alpha^*$, compute $\sigma = c_3 \oplus H_2(\alpha)$. Otherwise, compute $\sigma = c_3 \oplus Y'$.
4. If $\sigma \neq \sigma^*$, compute $m = c_4 \oplus H_4(\sigma)$. Otherwise, compute $m = c_4 \oplus Z'$.
5. If $(\sigma, m) \neq (\sigma^*, m^*)$, compute $r = H_3(\sigma, m)$. Otherwise, let $r = r'$.
6. If $c_1 = g^r$ and $c_2 = e(h,h)^r$, output m. Otherwise, output $\perp$.

Let denote by E_5 the event that the adversary or the decryption oracle makes a H_4-query for σ^* or a H_3-query for (σ^*, m^*). Since σ^* only appears in $c_3 = \sigma^* \oplus Y$ and Y is randomly generated in $\{0,1\}^n$, the random variable σ^* is independent of *CurrentView*. Therefore, $\Pr[E_5] \leq (q_h + q_d)/q$. Moreover, we have that games G_4 and G_5 proceed identically unless event E_5 occurs, which implies using the Difference lemma:

$$|\Pr[S_5] - \Pr[S_4]| \leq \frac{q_h + q_d}{q} \tag{17}$$

$$|\Pr[F_5] - \Pr[F_4]| \leq \frac{q_h + q_d}{q} \tag{18}$$

We also observe that in Game G_5, the message m^* is perfectly masked by Z, so

$$\Pr[S_5] = \frac{1}{2} \tag{19}$$

Game G_6: now we show that a game identical to G_5 can be obtained without knowing the master-key a. Namely, we assume that we are only given a legitimate D-Square-BDH 4-uple $(g, A = g^a, B = g^b, T = e(g,g)^{a^2 b})$ as input. Letting $h = A$, we proceed as in the proof of theorem 3: when the adversary makes a fresh hash-query for $H_1(v)$, we generate two randoms x and y in $\mathbb{Z}_q$ and return:

$$H_1(v) = g^x \cdot h^y$$

Then the corresponding private-key can be computed as

$$d_v = (h^x, y)$$

Moreover, the challenge ciphertext C^* is computed as follows:

$$C = \left(B,\ T,\ \sigma^* \oplus Y,\ m^* \oplus Z\right)$$

Therefore, the ciphertext is defined as in game G_5 but with $r^* = b$. Since b is unknown, the value α^* cannot be computed as $\alpha^* = e(H_1(v^*), h)^{r^*}$; instead, it is computed as:

$$\alpha^* = e(h^{x^*}, c_1^*) \cdot T^{y^*}$$

Using $H_1(v^*) = g^{x^*} \cdot h^{y^*}$ and $T = e(h,h)^r$, one obtains the same value for α^*.

It is clear that games G_5 and G_6 are identical. Therefore, we obtain:

$$\Pr[F_6] = \Pr[F_5] \tag{20}$$

Game G_7: we proceed as in game G_6, except that now T is uniformly distributed in $\mathbb{G}_1$; we write $T = e(h,h)^s$ for some $s \in \mathbb{Z}_q$. The variable α^* is computed as in game G_6. Writing $B = g^{r^*}$, we obtain as in the proof of Theorem 3:

$$\alpha^* = e(\delta^*, B) \cdot T^{y^*} = e(h^{x^*}, g^{r^*}) \cdot e(h,h)^{s y^*} = e(g^{x^*}, h)^{r^*} \cdot e(h,h)^{s y^*} \tag{21}$$

$$\alpha^* = e(H_1(v^*) \cdot h^{-y^*}, h)^{r^*} \cdot e(h,h)^{s y^*} = e(H_1(v^*), h)^{r^*} \cdot e(h,h)^{y^* \cdot (s - r^*)} \tag{22}$$

First, we argue that y^* is independent of *CurrentView*. The variable y^* appears both in $H_1(v^*) = g^{x^*} \cdot h^{y^*}$ and when computing decryption queries for identity v^*. First, we show that the output from the decryption queries for identity v^* is independent of y^*. Namely, the decryption oracle outputs a plaintext m only if $c_1 = g^r$ and $c_2 = e(h,h)^r$. Therefore, we have:

$$\alpha = e(d_1, c_1) \cdot c_2^{y^*} = e(h^{x^*}, g^r) \cdot e(h,h)^{r y^*} = e(g^{x^*}, h^r) \cdot e(h,h)^{r y^*} = e(g^{x^*} \cdot h^{y^*}, h)^r$$
$$= e(H_1(v^*), h)^r$$

Therefore, for a fixed $H_1(v^*)$, the value of α is independent of y^*. This implies that the message m output by the decryption oracle is independent of y^*.

Secondly, as in the proof of Theorem 3, for a fixed $H_1(v^*)$, the variable:

$$y^* = \log_h(H_1(v^*) \cdot g^{-x^*}) = \log_h H_1(v^*) - x^*/a$$

has the uniform distribution in $\mathbb{Z}_q$ and is therefore independent of *CurrentView*. This in turn implies that if $s \neq r$, then

$$\alpha^* = e(H_1(v^*), h)^r \cdot e(h,h)^{y^* \cdot (s - r)}$$

has the uniform distribution in $\mathbb{G}_1$ independently from *CurrentView*. Denoting by F_7' the event in game G_7 that the adversary makes a H_2-query to α^*, we obtain:

$$\Pr[F_7'] \leq \frac{q_h}{q} \tag{23}$$

It remains to consider the case when the decryption oracle makes a H_2-query to α^*. Let (c_1, c_2, c_3, c_4) be a ciphertext queried for decryption with identity v^*. We must have $(c_1, c_2) \neq (c_1^*, c_2^*)$ since when $(c_1, c_2) = (c_1^*, c_2^*)$ and $\alpha = \alpha^*$ the ciphertext gets immediately rejected. We write $c_1 = g^{\tilde{r}}$ and $c_2 = e(h, h)^{\tilde{s}}$. We have:

$$\alpha = e(d_1, c_1) \cdot c_2^{y^*} = e(H_1(v^*), h)^{\tilde{r}} \cdot e(h, h)^{y^* \cdot (\tilde{s} - \tilde{r})}$$

which gives:

$$\alpha/\alpha^* = e(H_1(v^*), h)^{\tilde{r} - r^*} \cdot e(h, h)^{y^* \cdot (\tilde{s} - \tilde{r} + r^* - s^*)}$$

Since y^* has the uniform distribution independently of *Currentview*, if $\tilde{s} - \tilde{r} + r^* - s^* \neq 0$ then $\alpha = \alpha^*$ with probability $1/q$. If $\tilde{s} - \tilde{r} + r^* - s^* = 0$ then $\alpha = \alpha^*$ implies $\tilde{r} = r^*$ and $\tilde{s} = s^*$, a contradiction since we must have $(c_1, c_2) \neq (c_1^*, c_2^*)$. This shows that the decryption oracle makes a H_2-query for α^* with probability at most q_d/q over the course of the game. Denoting F_7 the event that the adversary or the decryption oracle makes a H_2-query to α^*, we obtain using (23):

$$\Pr[F_7] \leq \frac{q_h + q_d}{q} \tag{24}$$

Observe that Games G_6 and G_7 only differ by the input 4-uple (g, A, B, T) which is a legitimate D-Square-BDH 4-uple in Game G_6 whereas T is random in $\mathbb{G}_1$ in Game G_7. One can therefore construct a distinguisher that performs the same operations as in Games G_6 and G_7 and outputs 1 if the adversary or the decryption oracle has made a H_2-query for α^* and 0 otherwise. Then in Game G_6, the distinguisher guesses correctly if event F_6 occurs, while in Game G_7 the distinguisher guesses correctly if event F_7 does not occur. Therefore, the distinguisher $\mathcal{D}$ advantage is:

$$\mathsf{Adv}^{\mathcal{D}} = \left| \frac{\Pr[F_6]}{2} + \frac{1 - \Pr[F_7]}{2} - \frac{1}{2} \right| = \frac{1}{2} \left| \Pr[F_6] - \Pr[F_7] \right|$$

Since the D-Square-BDH problem is assumed to be (t', ε')-hard, the distinguisher advantage must be bounded by ε' and therefore:

$$\left| \Pr[F_7] - \Pr[F_6] \right| \leq 2 \cdot \varepsilon' \tag{25}$$

Finally, combining inequalities (16), (18), (20), (24), (25), we obtain:

$$\Pr[F_3] \leq \frac{2q_h + 2q_d}{q} + 2 \cdot \varepsilon'$$

Combining inequalities (12), (13), (14), (15), (17) and (19), we obtain:

$$\left| \Pr[S_0] - \frac{1}{2} \right| \leq \frac{2q_h + 2q_d}{q} + \Pr[F_3] \leq \frac{4q_h + 4q_d}{q} + 2 \cdot \varepsilon'$$

which terminates the proof.

References

1. Attrapadung N., Chevallier-Mames B., Furukawa J., Gomi T., Hanaoka G., Imai H., Zhang R.: Efficient Identity-Based Encryption with Tight Security Reduction. Cryptology ePrint Archive, Report 2005/320 (2005).
2. Bellare M., Rogaway P.: Random oracles are practical: a paradigm for designing efficient protocols. In: Proceedings of the First Annual Conference on Computer and Communications Security, ACM (1993).
3. Boneh D., Franklin M.K.: Identity-based encryption from the Weil pairing. In: Proceedings of Crypto (2001).
4. Boneh D., Boyen X.: Efficient selective-ID secure identity based encryption without random oracles. In: Proceedings of Eurocrypt (2004).
5. Boneh D., Boyen X.: Secure Identity Based Encryption without random oracles. In: Proceedings of Crypto'04, LNCS, vol. 3152 (2004).
6. Canetti R., Goldreich O., Halevi S.: The random oracle methodology, revisited. In: STOC'98, ACM (1998).
7. Fujisaki E., Okamoto T.: Secure integration of asymmetric and symmetric encryption schemes. In: Crypto'99, LNCS, vol. 1666, Springer-Verlag, pp. 537–554 (1999).
8. Galindo D.: Boneh-Franklin identity based encryption revisited. In: Proceedings of ICALP 2005, LNCS, vol. 3580 (2005).
9. Gentry C.: Practical identity-based encryption without random oracles. In: Proceedings of Eurocrypt, LNCS, vol. 4004 (2006).
10. Katz J., Wang N.: Efficiency improvements for signature schemes with tight security reductions. In: ACM Conference on Computer and Communications Security, pp. 155–164 (2003).
11. Kiltz E.: On the Limitations of the spread of an IBE-to-PKE transformation. In: Proceedings of PKC, pp. 274–289 (2006).
12. Libert B., Quisquater J.J.: Identity based encryption without redundancy. In: Proceedings of ACNS, LNCS, vol. 3531 (2005).
13. Shamir A.: Identity-based cryptosystems and signature schemes. In: Proceedings of Crypto'84, Springer-Verlag (1985).
14. Shoup V.: Sequences of games: a tool for taming complexity in security proofs, manuscript, Nov. 30, 2004. Available at http://shoup.net/papers/.
15. Waters B.: Efficient Identity-Based Encryption without random oracles. In: Proceedings of Eurocrypt (2005).
16. Yang P., Kitagawa T., Hanaoka G., Zhang R., Matsuura K., Imai H.: Applying Fujisaki-Okamoto to identity-based encryption. In Fossorier M. et al. (eds.) AAECC 2006, LNCS, vol. 3857, pp. 183–192. Springer-Verlag (2006).