Practical Identity-Based Encryption (IBE) in Multiple-PKG Environments and Its Applications

Shengbao Wang, Zhenfu Cao

Department of Computer Science and Engineering,
Shanghai Jiao Tong University
800 Dongchuan Road, Shanghai 200240, China
{shengbao-wang,cao-zf}@cs.sjtu.edu.cn

November 25, 2007

Abstract. Identity-based encryption (IBE) schemes are usually used in *multiple-PKG environments* — on the one hand, each administrative domain (e.g., a relatively small and close organization) maintains its own private key generator (PKG); on the other hand, encryption across domains becomes a prevalent requirement. In this paper, we present a new IBE scheme using bilinear pairings. Compared with the famous IBE scheme of Boneh and Franklin, we show that ours is more practical in the multiple-PKG environment. We prove that our scheme meets chosen ciphertext security in the random oracle model, assuming the intractability of the standard Bilinear Diffie-Hellman (BDH) problem. As an application of our IBE scheme, we also propose an escrowed ElGamal scheme which possesses certain good properties in practice.

Keywords: identity-based encryption (IBE), multiple-PKG environments; public key encryption (PKE), escrowed ElGamal, bilinear pairings

Revision Notes:

1. A global setup algorithm (G-Setup) is explicitly introduced to formalize IBE schemes used in multiple-PKG environments.
2. Thanks to a novel technique for simulating the H_1 oracle (due to Lal and Sharma, on page 6 of [15], which is adapted in our proof for Lemma 2), the IND-ID-CCA security of the full M-IBE scheme is now reduced to the standard BDH problem.

1 Introduction

The concept of *identity(ID)-based cryptography* (IBC) was first introduced by Shamir in 1984 [17]. The basic idea behind an ID-based cryptosystem is that end users can choose arbitrary strings, for example their email addresses or other online identifiers, as their public keys. The corresponding private keys are created by binding the identity with a *master private key* of a trusted authority (called private key generation, or PKG for short). This eliminates much of the overhead associated with key management.

In 2001, Boneh and Franklin [3] gave the first fully functional solution for ID-based encryption (IBE) using the bilinear pairing over elliptic curves. Based on pairings, Sakai and Kasahara presented another IBE (SK-IBE for short) scheme in 2003 [18]. However, its applicabilities to some circumstances (e.g., hierarchical IBE [14] and threshold decryption [2]) are not comparable to the Boneh–Franklin scheme (BF-IBE for short). Therefore, the BF-IBE scheme has received much more attention in recent years.

IBE in Multiple-PKG Environments. Although IBE eliminates much of the overhead associated with key management in conventional public-key infrastructure (PKI) [17], to deploy an IBE scheme in practice, it is unrealistic to setup a single global private key generator (PKG) mainly because of the inherent *key escrow* problem, i.e., the PKG knows all its users' private keys. Another difficulty in applying an IBE scheme is that when distributing private keys, secure channels between the PKG and its users are required. Therefore, in order to apply IBE schemes and at the same time to mitigate the aforementioned two problems, each administrator domain (e.g., a relatively small and close organization like a university) will set up its own *domain PKG*, which is only responsible for generating and distributing private keys for the users within the domain/organization. On the other hand, with the development of the Internet and e-business, there are many requirements that users in a domain would like to securely communicate with users in other domains. We name this real-world application setting of IBE schemes as *multiple-PKG environment*.

For an IBE scheme to be applicable in the multiple-PKG environment, all that is needed is the availability of a global setup procedure G-Setup run by a globally trusted third-party, which generates and publishs the *standard* global parameters `params` that consists of pairing-friendly curves, the admissible bilinear pairings, a common group generator point P, and the other common cryptographic tools such as hash functions and encoding algorithms. We note that this is quite a reasonable requirement. In fact, elliptic curves, suitable group generator points and other cryptographic tools have been standardized for non-IBE applications, for example in the NIST FIPS standards [16]. Once these global parameters `params` have been agreed upon, each domain PKG only needs to generate its own master private key and compute the corresponding master public key using the global parameters `params`. For example, the Ministry of Education can serve as the trusted third-party for all the universities in the nation, while each university will setup its own domain PKG to generate and distribute private keys for its teachers and students.

In this paper, we present a new IBE scheme using bilinear pairings. We show that ours is more practical in the multiple-PKG environment. The new IBE scheme (hereafter referred to as M-IBE) enjoys the same Setup and Decrypt algorithms with the BF-IBE scheme, while differs from the latter in that it has different Key-Extraction and Encrypt algorithms. The M-IBE scheme is provably secure in the random oracle model, assuming the hardness of the standard Bilinear Diffie-Hellman (BDH) problem [3]. Parallel to [3], we also derive an escrowed ElGamal [9] encryption scheme from the M-IBE scheme. Moreover, we show how the derived ElGamal encryption enables a dual-decryptor public key encryption (PKE) scheme.

Paper Organization. The rest of this paper is organized as follows. In the next section, we give the necessary definitions for bilinear pairings, the related complexity assumptions and IBE schemes in multiple-PKG environment, together with the related security definitions for IBE schemes. In Section 3, we describes our new IBE scheme — the M-IBE scheme and compares it with the BF-IBE scheme [3, 4] in multiple-PKG environment. In Section 4, we prove that the chosen ciphertext security of the new M-IBE scheme is reducible to the standard BDH assumption. Next, we propose a new escrowed ElGamal encryption scheme in Section 5 and finally Section 6 contains a brief conclusion.

2 Preliminaries

2.1 Pairings and MBDH Assumption

In this section, we describe in a more general format the basic definition and properties of the pairing: more details can be found in [3].

Let $\mathbb{G}_1$ be a cyclic additive group generated by an element P, whose order is a prime p, and $\mathbb{G}_2$ be a cyclic multiplicative group of the same prime order p. We assume that the discrete logarithm problem (DLP) in both $\mathbb{G}_1$ and $\mathbb{G}_2$ are hard.

Definition 1 (Pairing). *An admissible pairing e is a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, which satisfies the following three properties:*

1. Bilinear*: If $P, Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p^*$, then $e(aP, bQ) = e(P, Q)^{ab}$;*
2. Non-degenerate*: $e(P, P) \neq 1$;*
3. Computable*: If $P, Q \in \mathbb{G}_1$, one can compute $e(P, Q) \in \mathbb{G}_2$ in polynomial time.*

Typically, the map e will be derived from either the Weil or Tate pairing on an elliptic curve over a finite field. We refer to [5, 3, 4, 13] for a more comprehensive description of how these groups, pairings and other parameters should be selected in practice for efficiency and security.

Definition 2 (Bilinear Diffie-Hellman (BDH) Parameter Generator). *As in [3], we say that a randomized algorithm $\mathcal{IG}$ is a* BDH *parameter generator if $\mathcal{IG}$ takes a security parameter $k > 0$, runs in time polynomial in k, and outputs the description of two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of the same prime order q and the description of an admissible pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$.*

Definition 3 (Bilinear Diffie-Hellman (BDH) Problem). *Let $\mathbb{G}_1$, $\mathbb{G}_2$, P and e be as above. The* BDH *problem in $\langle \mathbb{G}_1, \mathbb{G}_2, e\rangle$ is as follows: Given $\langle P, aP, bP, cP\rangle$ with uniformly random choices of a, b, $c \in \mathbb{Z}_q^*$, compute $e(P, P)^{abc} \in \mathbb{G}_2$.*

The security of our new pairing-based IBE scheme is based on the difficulty of the above BDH problem. However, it will simplify the presentation of our proofs to use the following equivalent formulation of the BDH problem, known as *modified* BDH problem [19]. The modified BDH problem is identical to the BDH problem, except that the output is $e(P, P)^{a^{-1}bc}$ (instead of $e(P, P)^{abc}$) [1].

Definition 4 (MBDH Problem [19]). *Let $\mathbb{G}_1$, $\mathbb{G}_2$, P and e be as above. The* MBDH *problem in $\langle \mathbb{G}_1, \mathbb{G}_2, e\rangle$ is as follows: Given $\langle P, aP, bP, cP\rangle$ with uniformly random choices of a, b, $c \in \mathbb{Z}_q^*$, compute $e(P, P)^{a^{-1}bc} \in \mathbb{G}_2$.*

[1] Note that in an earlier version of the paper, the weaker so-called mBDH problems is that, given $\langle P, aP, a^{-1}P, bP, cP\rangle$, to compute $e(P, P)^{abc}$.

A real-valued function $f(l)$ is *negligible* if for any integer k, $|f(l)| < l^{-k}$ for sufficiently large l. The following MBDH assumption states that, roughly, this problem is computational infeasible.

Definition 5 (Modified Bilinear Diffie-Hellman (MBDH) Assumption). *As in [3], if $\mathcal{IG}$ is a* BDH *parameter generator, the advantage $Adv_{\mathcal{IG}}(\mathcal{B})$ that an algorithm $\mathcal{B}$ has in solving the* MBDH *problem is defined to be the probability that the algorithm $\mathcal{B}$ outputs $e(P,P)^{a^{-1}bc} \in \mathbb{G}_2$ when the inputs to the algorithm are $\mathbb{G}_1, \mathbb{G}_2, e, P, aP, bP, cP$ where $\langle \mathbb{G}_1, \mathbb{G}_2, e\rangle$ is $\mathcal{IG}'s$ output for large enough security parameter k, P is a random generator of $\mathbb{G}_1$, and $a, b, c \in \mathbb{Z}_q^*$. The* MBDH *assumption is that $Adv_{\mathcal{IG}}(\mathcal{B})$ is negligible for all efficient algorithms $\mathcal{B}$.*

Here the probability is measured over the random choices of $a, b, c \in \mathbb{Z}_q^*$ and the random bits of $\mathcal{B}$.

The BDH assumption can be defined similarly. In [8], Canetti and Hohenberger proved that the decisional variants of the two assumptions are equivalent. Using the same reduction technique, we show that the MBDH and BDH assumptions are equivalent.

Lemma 1. *If the MBDH problem is solvable in in $\langle \mathbb{G}_1, \mathbb{G}_2, e\rangle$ with probability ϵ, then the BDH problem is also solvable in $\langle \mathbb{G}_1, \mathbb{G}_2, e\rangle$ with probability with the same probability; and vice versa.*

Proof. (BDH Assumption ⇒ MBDH Assumption.) On BDH input $\langle P, aP, bP, cP\rangle$, query the MBDH problem solver on input $\langle bP, P, cP, aP\rangle = \langle Q, xQ, yQ, zQ\rangle$ and output its response. Observe that the MBDH solver will output $e(Q,Q)^{x^{-1}yz}$, by substitution, we have $e(Q,Q)^{x^{-1}yz} = e(bP, bP)^{b(c/b)(a/b)} = e(P,P)^{abc}$ for the BDH solver.

(MBDH Assumption ⇒ BDH Assumption.) Omitted. □

2.2 Definitions for IBE in Multiple-PKG Environments

Our definition for IBE schemes in multiple-PKG environments is sightly different from that of Boneh and Franklin [3, 4] in that there is an extra distilled global setup algorithm G-Setup, which is responsible for generating the global agreed parameters `params`.

Definition 6 (IBE in Multiple-PKG Environments). *An IBE scheme in the multiple-PKG environment handling identities of length (where is a polynomially-bounded function) is specified by five polynomial time algorithms:*

G-Setup: *is a probabilistic algorithm run by a globally trusted third-party that takes as input a security parameter to output the global public parameters* `params`.

Setup: *is a probabilistic algorithm run by a domain PKG that outputs a master public/private key pair (P_{Pub}, msk) for the domain (P_{Pub} is its* master public key *and msk is its master private key).*

Key-Extraction: *is a key generation algorithm run by the domain PKG on input of a master private key msk and a user's identity ID to return the user's private key d_{ID}.*

Encrypt: *is a probabilistic algorithm which takes as input a plaintext M, the global public parameters* `params`*, a recipient's identity ID and the designated domain PKG's master public key P_{Pub} to output a ciphertext C.*

Decrypt: *is a deterministic decryption algorithm which takes as input a ciphertext C and the private key d_{ID} to return a plaintext M or a distinguished symbol $\perp$ if C is not a valid ciphertext.*

The security of an IBE scheme in multiple-PKG environment is defined by the following game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ which is almost identical to that formalized in [3].

Setup. $\mathcal{C}$ takes a security parameter k and runs the G-Setup and Setup algorithms. It gives $\mathcal{A}$ the global public parameters `params` *and* the domain master public key P_{Pub}, while keeps msk to itself.

Find Stage. $\mathcal{A}$ issues queries as one of follows:

- Extraction query on ID_i. $\mathcal{C}$ runs the Key-Extraction algorithm to generate d_{ID_i} and passes it to $\mathcal{A}$.
- Decryption query on (ID_i, C_i). $\mathcal{C}$ decrypts the ciphertext by finding d_{ID_i} first (through running Key-Extraction algorithm if necessary), and then running the Decrypt algorithm. It responds with the resulting plaintext M_i.

Challenge. Once $\mathcal{A}$ decides that Phase 1 is over, it outputs two equal length plaintexts M_0, M_1, and an identity ID^* (called the challenge identity) on which it wishes to be challenged. The only constraint is that $\mathcal{A}$ must not have queried the extraction query on ID^* in Phase 1. $\mathcal{C}$ picks a random bit $t \in \{0, 1\}$ and sets $C^* = Encrypt(ID^*, M_t)$. It sends C^* as the challenge to $\mathcal{A}$.

Guess Stage. $\mathcal{A}$ issues more queries as in Phase 1 but with two restrictions: (1) Extraction queries cannot be issued on ID^*; (2) Decryption queries cannot be issued on (ID^*, C^*).

Output. Finally, $\mathcal{A}$ outputs a guess $t' \in \{0, 1\}$ and wins the game if $t' = t$.

We refer to this type of adversary as an IND-ID-CCA adversary [3, 4]. If $\mathcal{A}$ cannot ask decryption queries, we call it an IND-ID-CPA adversary. The advantage of an adversary $\mathcal{A}$ against an IBE scheme is the function of security parameter k defined as:

$$Adv_{\mathcal{A}}(k) = |\Pr[t' = t] - 1/2|.$$

Definition 7 (IBE Security). *An identity-based encryption (IBE) scheme is IND-ID-CCA secure (resp. IND-ID-CPA) if for any IND-ID-CCA (resp. IND-ID-CPA) adversary, $Adv_{\mathcal{A}}(k)$ is negligible.*

3 Proposed IBE Scheme

Now we describe our M-IBE scheme — a practical IBE scheme in the multiple-PKG environment. Following the same exploration as in [4], we first give a basic version of our scheme which is only chosen plaintext attack (CPA) secure. We then extend the basic scheme to get security against adaptive chosen ciphertext attack (CCA) in the random oracle model [6], using the second Fujisaki-Okamoto transformation [10].

3.1 Basic Scheme with CPA Security

The basic M-IBE scheme works as follows.

G-Setup. Given a security parameter k, the globally trusted third-party does the following:

1. Outputs two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order p, a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, a generator points P of $\mathbb{G}_1$.
2. Picks a cryptographic hash functions $H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1^*$, a cryptographic hash function $H_2 : \mathbb{G}_2 \rightarrow \{0, 1\}^n$ for some n.

The message space is $\mathcal{M} = \{0,1\}^n$. The ciphertext space is $C = \mathbb{G}_1^* \times \{0,1\}^n$. The global public parameters `params` are $\langle p, \mathbb{G}_1, \mathbb{G}_2, e, P, n, H_1, H_2 \rangle$.

Setup. Each domain PKG does the following:

1. Chooses a random $s \in \mathbb{Z}_p$.
2. Calculates $P_{Pub} = sP \in \mathbb{G}_1$

The domain master public key is P_{Pub}, and the master private key msk is s.

Key-Extraction. To generate a private key for identity $ID \in \{0,1\}^*$, the domain PKG first computes $Q_{ID} = H_1(ID) \in \mathbb{G}_1^*$, and then sets the private key d_{ID} to be $d_{ID} = s^{-1}Q_{ID}$ where s is the master private key [2].

Encrypt. To encrypt message $m \in \mathcal{M}$, the sender picks randomly a $r \in \mathbb{Z}_p$, using the recipient's identity ID to compute $Q_{ID} = H_1(ID) \in \mathbb{G}_1^*$, sets the ciphertext to be

$$C = \langle rP_{Pub},\ m \oplus H_2(g_{ID}^r) \rangle, \text{ where } g_{ID} = e(P, Q_{ID}) \in \mathbb{G}_2^*.$$

Decrypt. This algorithm is identical to that of BF-IBE. To decrypt a ciphertext $C = \langle U,\ V \rangle \in \mathcal{C}$, using the private key d_{ID} of the identity ID computes

$$m = V \oplus H_2(e(U,\ d_{ID})).$$

Consistence: The recipient can correctly decrypt C to get m since

$$\begin{aligned} & e(U,\ d_{ID}) \\ = & e(rsP,\ s^{-1}Q_{ID}) \\ = & e(P, Q_{ID})^r. \end{aligned}$$

3.2 Full Scheme with CCA Security

In this subsection we extend the above basic scheme to a full IBE scheme with adaptive chosen ciphertext security using the general transformation due to Fujisaki and Okamoto [10].

We borrow the description of the transformation from [11]. This conversion starts from an IND-CPA encryption scheme and builds an IND-CCA scheme in the random oracle model. If we denote by $E_{pk}(M, r)$ the encryption of M using the random bits r under the public key pk, with set of messages $M = \{0,1\}^n$, set of coins R and set of ciphertexts C, the new transformation is the scheme

$$E_{pk}^{hy}(M) = E_{pk}(M||r, H(M||r)),$$

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

Now we describe the full M-IBE scheme thereby obtained.

G-Setup. Given a security parameter k, the global trusted third-party does the following:

1. Outputs two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order p, a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, a generator points P of $\mathbb{G}_1$.

[2] Note that in BF-IBE [3, 4], the private key of a user is computed as $d_{ID} = sQ_{ID}$ instead.

2. Picks three cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$, $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n$ for some n and $H_3 : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$.

The message space is $\mathcal{M} = \{0,1\}^{n-k_0}$. The ciphertext space is $C = \mathbb{G}_1^* \times \{0,1\}^n$. The global public parameters `params` are $\langle p, \mathbb{G}_1, \mathbb{G}_2, e, P, n, k_0, H_1, H_2, H_3 \rangle$.

Setup. This algorithm is identical to that of the basic M-IBE scheme.

Key-Extraction. This algorithm is also identical to that of the basic M-IBE scheme.

Encrypt. To encrypt message $m \in \mathcal{M}$, the sender picks randomly a $\sigma \in \{0,1\}^{k_0}$, using the recipient's identity ID to compute $Q_{ID} = H_1(ID) \in \mathbb{G}_1^*$, sets $r = H_3(m, \sigma) \in \mathbb{Z}_p^*$ and finally sets the ciphertext to be

$$C = \langle rP_{Pub},\ (m||\sigma) \oplus H_2(g_{ID}^r) \rangle, \text{ where } g_{ID} = e(P, Q_{ID}) \in \mathbb{G}_2^*.$$

Decrypt. This algorithm is identical to that of Galindo's BF-IBE variant [11]. To decrypt a ciphertext $C = \langle U,\ V \rangle \in \mathcal{C}$, using the private key d_{ID} of the identity ID do

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

Consistence: The consistence of this scheme directly follows that of the basic scheme.

3.3 Comparison of IBE Schemes in Multiple-PKG Environments

Now we compare our new M-IBE scheme with the BF-IBE scheme [3] to show that ours is more practical in multiple-PKG environments.

As we pointed out in Section 1, in the real-world application setting of IBE schemes (i.e. the multiple-PKG environment), it is quite normal for a user to encrypt messages to users from different administrative domains. For example, for a student Alice of university A, she may need to encrypt messages to Bob from university B, Carol from university C, or Emmy whose university is unknown to Alice by now (note that, however, Alice already knows Emmy's identity information).

Now we compare our new IBE scheme with the BF-IBE scheme [3] in such a multiple-PKG environment. Firstly, the Setup algorithms run by each domain PKG in the two schemes are the same, resulting in master public keys of the same length. Secondly, the Decrypt algorithms in the two IBE schemes are also the same, requiring identical computational overhead. In the following, we discuss what significance our different Encrypt and Key-Extraction algorithms could bring in practice, particularly in the multiple-PKG environment.

In BF-IBE [3], the session secret, i.e. the term g_{ID}^r is computed as $g_{ID}^r = e(P_{Pub},\ Q_{ID})^r$, in which P_{Pub} is the master public key of the intended recipient's PKG. Therefore, in a multiple-PKG environment, before computing the term g_{ID}^r (which requires a relatively expensive pairing evaluation that is the main operations of the overall encryption), the BF-IBE scheme requires the encryptor to first get to know the following two things:

- which domain/organization the recipient is from, *and*
- the master public key associated with the domain PKG of the recipient.

Compared with the BF-IBE scheme, the biggest difference of our M-IBE scheme is that the computation of the term $g_{ID}^r = e(P,\ Q_{ID})^r$ in its Encrypt algorithm is independent of *any* PKG's master public key P_{Pub}. Consequently, in the M-IBE scheme, the encryptor can compute the pairing *before* getting the master public key of the recipient's PKG. Interestingly, the encryptor can even pre-compute g_{ID} *before* she knows which domain/organization the recipient is from.

Therefore, our scheme enables a type of efficient "on the move" IBE scheme in the multiple-PKG environment, which requires very small on-online work for the sender (i.e. encryptor). We note that this feature is particularly useful in (ID-based) broadcasting (or *multiple-recipient*) encryption scenario [7], namely with most of the expensive computation pre-computed, the overall performance will be upgraded to a large extent.

Table 1. Comparison between M-IBE and BF-IBE

↓Schemes / Items→	Private key	g_{ID}	Assumption
BF-IBE [3]	sQ_{ID}	$e(P_{Pub}, Q_{ID})$	BDH
M-IBE	$s^{-1}Q_{ID}$	$e(P, Q_{ID})$	BDH

Table 1 summarizes the above comparison between our M-IBE scheme and the BF-IBE scheme. The two IBE schemes have the same performance features (e.g., the same overall computational overhead, master public key length and the ciphertext length) and security strength (as we will prove in the following section that the IND-ID-CCA security of our M-IBE scheme can also be reduced to the BDH assumption). However, our M-IBE scheme has a distinct advantage over the BF-IBE scheme in that it can have the dominating operation in the Encrypt algorithm (i.e. the pairing evaluation) pre-computed (i.e., computed off-line, or in other words, computed before querying the master public key of the designated domain). In a word, our M-IBE is more practical than the BF-IBE scheme in the multiple-PKG environment.

4 Security Proof for M-IBE

Now we evaluate the security of our M-IBE scheme. We prove that, same as the original BF-IBE scheme [3, 4] and the improved BF-IBE scheme [11], the security of the M-IBE scheme can also be reduced to the hardness of the BDH problem. The reduction is similar to the proof of BF-IBE [4]. However, we will take into account the reduction error found by Galindo [11].

We prove the security of our IBE scheme along the similar lines to that in [4, 11]. The proof is completed in three steps that can be sketched as follow. 1) First we prove that if there exists an IND-ID-CCA adversary, who is able to break the IBE by launching the adaptive chosen ciphertext attacks as defined in the security model, then there exists an IND-CCA adversary to break the **BasicPub**^{hy} scheme defined in **Lemma 2** with the adaptive chosen ciphertext attacks. 2) Second, if such IND-CCA adversary exists, then we show (in **Lemma 3**) that there must be an IND-CPA adversary that breaks the corresponding **BasicPub** scheme. 3) Finally, in **Lemma 4** we prove that if the **BasicPub** scheme is not secure against an IND-CPA adversary, then the corresponding MBDH assumption is flawed.

We first define the related non-ID-based public key encryption scheme **BasicPub**. It is described by three algorithms: keygen, encrypt, decrypt.

keygen: Given a security parameter k, the user does the following:

1. Chooses a random $s \in \mathbb{Z}_p$, calculates $P_0 = sP \in \mathbb{G}_1$.

2. Picks a random $Q_0 \in \mathbb{G}_2^*$.
3. Picks a cryptographic hash functions $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n$ for some n.

The message space is $\mathcal{M} = \{0,1\}^n$. The ciphertext space is $C = \mathbb{G}_1^* \times \{0,1\}^n$. The public key is $\langle p, \mathbb{G}_1, \mathbb{G}_2, e, P, P_0, n, Q_0, H_2 \rangle$ and the private key is $d_0 = s^{-1}Q_0 \in \mathbb{G}_1^*$.

encrypt: To encrypt message $m \in \mathcal{M}$, the sender picks randomly a $r \in \mathbb{Z}_p$ and sets the ciphertext to be

$$C = \langle rP_0,\ m \oplus H_2(g_{ID}^r) \rangle,\ \text{where } g_{ID} = e(P, Q_0) \in \mathbb{G}_2^*.$$

decrypt: To decrypt a ciphertext $C = \langle U,\ V \rangle \in \mathcal{C}$, using the private key d_0 computes

$$m = V \oplus H_2(e(U,\ d_0)).$$

The correctness of the above public key encryption scheme can be easily verified. We refer to the full scheme of applying the Fujisaki-Okamoto transformation to **BasicPub** as **BasicPub**^{hy}.

Lemma 2. *Let $\mathcal{A}$ be an IND-ID-CCA adversary with advantage ϵ against the full IBE scheme making at most q_E private key extraction queries, q_D decryption queries and q_1 hash queries. Then there is an IND-CCA adversary $\mathcal{B}$ that has advantage at least $\frac{\epsilon}{q_1}(1 - \frac{q_E}{q_1}) \approx \frac{\epsilon}{q_1}$ against* ***BasicPub***^{hy}*. Its running time is $t_{\mathcal{B}} \leq t_{\mathcal{A}} + c_{\mathbb{G}_1}(q_D + q_E + q_1)$, where $c_{\mathbb{G}_1}$ denotes the time of computing a random multiplication in $\mathbb{G}_1$.*

Proof. Using similar reduction to Result 5 from [11] as well as a new technique for simulating the H_1 random oracle due to Lal and Sharma [15], the proof is given as follows.

We show how to construct an IND-CCA adversary $\mathcal{B}$ against **BasicPub**^{hy} by using an IND-ID-CCA adversary $\mathcal{A}$ against the full M-IBE scheme. Let ϵ denotes the advantage of $\mathcal{A}$. Algorithm $\mathcal{B}$ receives a public key $\langle p, \mathbb{G}_1, \mathbb{G}_2, e, P, P_0,\ Q_0, n, k_0, H_1,\ H_2, H_3 \rangle$ from its challenger. Then $\mathcal{B}$ simulates the challenger for $\mathcal{A}$ as follows.

Setup. $\mathcal{B}$ gives $\mathcal{A}$ the parameters $\langle p, \mathbb{G}_1, \mathbb{G}_2, e, P, n, k_0, H_1,\ H_2, H_3 \rangle$ as the global public parameters `params` and sets the *master public key* P_{Pub} to be P_0, where H_1 is an oracle controlled by $\mathcal{B}$ as indicated in the following:

H_1**-queries**. To respond to the queries, algorithm $\mathcal{B}$ maintains a list H_1^{list} of tuples $\langle ID_i, Q_i, b_i \rangle$. Before initializing the list, $\mathcal{B}$ picks a random $j \in \{1, ..., q_1\}$. When $\mathcal{A}$ queries H_1 at ID_i, $\mathcal{B}$ proceeds as follows:

- If $i \neq j$, it picks at random a $b_i \in \mathbb{Z}_p^*$, sets $Q_i = b_i P_0$, adds $\langle ID_i, Q_i, b_i \rangle$ to the list and returns Q_i to $\mathcal{A}$.
- Otherwise (i.e., $i = j$), it sets $Q_j = Q_0$, adds $\langle ID_i, Q_i, \spadesuit \rangle$ to the list and gives Q_j to $\mathcal{A}$. Here $\spadesuit$ denotes a special symbol. As noted in [11], Q_0 is unknown to $\mathcal{A}$ and is uniformly distributed in $\mathbb{G}_1$, hence the outputs of H_1 are uniformly distributed in $\mathbb{G}_1$ and independent of $\mathcal{A}$'s current view.

Find Stage. $\mathcal{A}$ issues queries as one of the following:

- **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_l^{list}.
  - If $i = j$, then $\mathcal{B}$ aborts the game and the attack against **BasicPub**^{hy} failed. (**Event E1**)

  - Otherwise, $\mathcal{B}$ sets $d_i = b_i P$ and gives d_i to $\mathcal{A}$.
- **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 $\langle ID_i, Q_i, b_i\rangle$ be the corresponding entry in H_l^{list}.
  - If $i \neq j$, then $\mathcal{B}$ retrieves the private key d_i and decrypts C_i using the Decrypt algorithm. $\mathcal{B}$ gives the decrypted message back to $\mathcal{A}$.
  - Otherwise, $\mathcal{B}$ asks its challenger to decrypt C_j and relays the answer to A. (Note that if $i = j$, then $Q_i = Q_0$, and the decryption of $\langle ID_j, C_j\rangle$ is identical to the decryption of C_j by **BasicPub**^{hy}.)

Challenge. Once $\mathcal{A}$ decides that Phase 1 (i.e. the Find Stage) is over, it outputs two equal length plaintexts M_0, M_1, and an challenge identity ID^* on which it wishes to be challenged. Algorithm $\mathcal{B}$ proceeds as follows.
- If $i \neq j$, then $\mathcal{B}$ aborts the game and the attack against **BasicPub**^{hy} failed. (**Event E2**)
- Otherwise, $\mathcal{B}$ sends M_0, M_1 to its own challenger and gets back a ciphertext C^* — the encryption of M_t for a random bit $t \in \{0,1\}$ under **BasicPub**^{hy}. $\mathcal{B}$ relays C to $\mathcal{A}$, which is also an encryption of M_t under ID^* for the full M-IBE scheme.

Guess Stage. $\mathcal{A}$ issues more queries and $\mathcal{B}$ proceeds as in Phase 1. Recall that by the rules of the game, $\mathcal{A}$ has two restrictions: (1) Extraction queries cannot be issued on ID^*; (2) Decryption queries cannot be issued on (ID^*, C^*).

Output. Finally, $\mathcal{A}$ outputs a guess $t' \in \{0,1\}$ for t. $\mathcal{B}$ relays t' as the guess to its challenger.

In the above simulation, H_1 behaves as a random oracle, and the extraction as well as decryption queries are valid, hence if $\mathcal{B}$ does not abort during the simulation (i.e., both **Event E1** and **E2** do not happen), then $\mathcal{A}$'s view is identical to its view in a real attack. Therefore, we have $|\Pr[t' = t] - 1/2| \geq \epsilon$, where this probability is over the random bits of $\mathcal{A}$, $\mathcal{B}$ and the challenger for the IND-ID-CCA game.

Now we evaluate the probability that the simulation does not abort. We have

$$\Pr[\mathcal{B} \text{ does not abort}] = \Pr[\neg\text{E1} \wedge \neg\text{E2}] = \Pr[\neg\text{E2}|\neg\text{E1}]\Pr[\neg\text{E1}].$$

As in [11], we can upper bound for $\Pr[E1] \leq q_E/q_1$, which is the probability that $\mathcal{A}$ makes an extraction query at ID_j in the Find Stage, since the maximum number of such queries is q_E. On the other hand, a lower bound for $\Pr[\neg\text{E2}|\neg\text{E1}]$, that is the probability that $\mathcal{A}$ chooses ID_j as the challenge identity, is $1/q_1$. Therefore,

$$\Pr[\mathcal{B} \text{ does not abort}] \geq \frac{1}{q_1}(1 - q_E/q_1).$$

This means that $\mathbb{B}$'s advantage is as least $\frac{\epsilon}{q_1}(1 - q_E/q_1)$. □

Lemma 3. *Let $\mathcal{A}$ be an IND-CCA adversary against **BasicPub**^{hy} whose advantage is ϵ, making at most q_D decryption queries and q_2 hash queries. Then there is an IND-CPA adversary $\mathcal{B}$ that has advantage at least $(\epsilon - q_2 2^{-(k_0-1)})(1-1/p)^{q_D} \approx \epsilon$ against **BasicPub**^{hy}. Its running time is $t_{\mathcal{B}} \leq t_{\mathcal{A}} + q_2(T_{\boldsymbol{BasicPub}} + \log p)$, where $T_{\boldsymbol{BasicPub}}$ is the running time of Encrypt algorithm in **BasicPub**.*

Proof. This result is obtained applying the Fujisaki-Okamoto transformation, and the proof can be found in [10]. □

Lemma 4. *Let $\mathcal{A}$ be an IND-CPA adversary with advantage ϵ against* ***BasicPub*** *making at most q_2 queries to H_2. Then there is an algorithm $\mathcal{B}$ that has advantage at least $2\epsilon/q_2$ in solving the BDH problem. Its running time is $t_{\mathcal{B}} = O(t_{\mathcal{A}})$.*

Proof. See Appendix A. □

We are now ready to state the security of our full IBE scheme.

Theorem 1. *The proposed full M-IBE scheme is (t, q_H, q_D, ϵ)-secure if the BDH problem on $(\mathbb{G}_1, \mathbb{G}_2, e)$ is*

$$(t + c_{\mathbb{G}_1}(2q_D + q_H) + q_H O(\log^3 p + \log p), \epsilon/q_H^2)) - secure.$$

Proof. This follows directly from Lemma 2, 3 and 4. □

5 Applications of Our IBE Scheme

Now we investigate the applicabilities of our newly proposed M-IBE scheme in some other real-world scenarios. We note that, using similar ideas to the ones presented in [14, 2], our M-IBE scheme can also be extended to work in hierarchical as well as threshold decryption contexts. We leave the details to the interested reader.

5.1 Escrowed ElGamal Encryption

Parallel to [3], we introduce a new ElGamal encryption system in which a single escrow key enables the decryption of ciphertexts encrypted under any public key.

Our escrowed ElGamal encryption scheme works as follows:

Setup. Given a security parameter k, the *escrow authority* (EA) does the following:

1. Chooses a random $s \in \mathbb{Z}_p$, calculates two points $Q_1 = sP$ and $Q_2 = s^{-1}P \in \mathbb{G}_1$ [3].
2. Chooses a cryptographic hash functions $H : \mathbb{G}_2 \rightarrow \{0,1\}^n$ for some n.

The message space is $\mathcal{M} = \{0,1\}^n$. The ciphertext space is $C = \mathbb{G}_1^* \times \{0,1\}^n$. The public *params* are $\langle p, \mathbb{G}_1, \mathbb{G}_2, e, n, P, Q_1, Q_2, H\rangle$ and the *escrow key* is s.

Key Generation. Same as in [3], a user generates a public/private key pair for herself by picking a random $x \in \mathbb{Z}_q$ and computing $P_{Pub} = xP \in \mathbb{G}_1$. Her private key is x, her public key is P_{Pub}.

Encrypt. To encrypt message $m \in \mathcal{M}$, the sender picks randomly a $r \in \mathbb{Z}_p$, sets the ciphertext to be

$$C = \langle rQ_2,\ m \oplus H_2(g^r)\rangle,\ \text{where } g = e(P, P_{Pub}) \in \mathbb{G}_2^*.$$

Decrypt. To decrypt a ciphertext $C = \langle U,\ V\rangle \in \mathcal{C}$, using the private key x of the identity ID computes

$$m = V \oplus H_2(e(U,\ xQ_1)).$$

Escrow Decrypt. To decrypt a ciphertext $C = \langle U,\ V\rangle$, using the escrow key s of the EA computes

$$m = V \oplus H_2(e(U,\ P_{Pub})^s).$$

[3] Note that in [4], the public key of the EA is one point $Q = sP \in \mathbb{G}_1$ instead.

Consistence: The two recipients can correctly decrypt C to get m since

$$\begin{aligned} & e(U,\ xQ_1) \\ &= e(rQ_2,\ xQ_1) \\ &= e(rs^{-1}P,\ xsP) \\ &= e(rP,\ xP) \\ &= e(P,\ P_{Pub})^r \\ &= g^r \end{aligned}$$

and

$$\begin{aligned} & e(U,\ P_{Pub})^s \\ &= e(rs^{-1}P,\ P_{Pub})^s \\ &= e(rP,\ P_{Pub}) \\ &= e(P,\ P_{Pub})^r \\ &= g^r. \end{aligned}$$

Compared with the scheme in [3], our escrowed ElGamal requires the EA to publish one more point as its public key. An advantage of our scheme is that the encryptor can choose a designated EA (from multiple EAs) after she finished most of the operations of encrypting a message. This provides the encryptor with more flexibility in practice.

A Variant. If we look the escrow authority (EA) in the above escrowed ElGamal scheme as an ordinary principal (who has her own private and public key pair), it can be then used as a *dual-decryptor PKE scheme*, i.e., a single ciphertext can be decrypted *independently* by two different principals. However, unlike in conventional setting, we require that at least one of the recipients to publish two points (e.g. $Y_1,\ Y_2$) as her public key, in the form of $Y_1 = \alpha P$ and $Y_2 = \alpha^{-1}P$ (assuming α is the private key of the recipient).

A good property of this scheme is that the encryptor can encrypt the message before she picks up the second recipient. In other words, after the encryption has been down, the encryptor can change her mind on who the second recipient will be. More interestingly, the encryptor can efficiently add more such "second recipient", each time she adds one, only one scalar multiplication of computation is needed, without any expensive pairing evaluation. However, we note that the size of the ciphertext will grow linearly.

5.2 Efficient Multi-Recipient IBE

We now look at the multi-recipient IBE setting, whereby a sender wants to send an encrypted message to n recipients. In 2004, Baek *et al.* [7] proposed the first construction based on the BF-IBE scheme, which reduces the number of pairing evaluations to 1. We remark that their scheme only works well within an administrative domain, namely with all the n recipients getting their private keys from the same one domain PKG. However, in the multiple-PKG context, the sender still has to compute q pairings if the n designated recipients are from q different domains.

As mentioned in Section 3, our M-IBE scheme is even more attractive when used in the multi-recipient context. In [20], using a similar idea to Baek *et al.*'s [7] and based on the new M-IBE scheme, we propose an efficient multi-recipient IBE scheme which works efficiently across domains. Notably, the new multi-recipient IBE scheme requires only 1 pairing evaluation for the sender, no matter how many domains the n recipients are from.

6 Conclusions

In this paper, we gave a new IBE scheme that is provably secure in the random oracle model. The security is based on a the standard Bilinear Diffie-Hellman assumption. We showed that the new scheme is more practical than the famous IBE scheme due to Boneh and Franklin in multiple-PKG environment. As applications, we also proposed a related escrowed ElGamal encryption scheme which has its distinct advantages over that in [3, 4]. Compared with the Boneh–Franklin scheme, our IBE scheme is even more practical in the multiple-recipient setting.

Future work includes exploring the merits of our new IBE scheme in constructing Certificate-Based Encryption (CBE) [12] and Certificateless Public Key Encryption (CL-PKE) schemes [1].

References

1. S.S. Al-Riyami and K.G. Paterson. Certificateless public key cryptography. In *Proc. of ASIACRYPT'03*, LNCS vol. 2894, pp. 452-473, 2003. [13]
2. J. Baek. Identity-based threshold decryption. In *Proc. PKC'04*, LNCS vol. 2947, pp. 262-276, 2004. [2, 11]
3. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In *Proc. of CRYPTO'01*, LNCS vol. 2139, pp. 213-229, 2001. [2, 3, 4, 5, 6, 7, 8, 11, 12, 13]
4. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. SIAM J. Computing, 32(3):586-615, 2003. [3, 4, 5, 6, 8, 11, 13, 14]
5. P.S.L.M. Barreto, H.Y. Kim, B. Lynn, and M. Scott. Efficient algorithms for pairing-based cryptosystems. In *Proc. CRYPTO'02*, LNCS vol. 2442, pp. 354-368, 2002. [3]
6. M. Bellare and P. Rogaway, Random oracles are practical: a paradigm for designing effiient protocols, In *Proc. of 1st ACM Conference on Computer and Communications Security*, pp.62-73, 1993. [5]
7. J. Baek, R. Safavi-Naini, and W. Susilo. Efficient multi-receiver identity-based encryption and its application to broadcast encryption. In *Proc. of PKC'05*, LNCS vol. 3386, pp. 380-397, 2005. [8, 12]
8. R. Canetti and S. Hohenberger. Chosen-Ciphertext Secure Proxy Re-Encryption. In *Proc. of ACM-CCS'007*, pp. 185-194, 2007. [4]
9. T. ElGamal. A public key cryptosystem and signature scheme based on discrete logarithms. *IEEE Trans. Info. Theory*, 31(4), pp. 469-472, 1985. [2]
10. 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. [5, 6, 10]
11. D. Galindo. Boneh-Franklin identity based encryption revisited. In *Proc. of ICALP'05*, LNCS vol. 3580, pp. 791-802, 2003. [6, 7, 8, 9, 10]
12. C. Gentry. Certificate-based encryption and the certificate revocation problem. In *Proc. of Eurorypt'03*, volume 2656 of LNCS, pp. 272-293, 2003. [13]
13. S.D. Galbraith, K. Harrison, and D. Soldera. Implementing the Tate pairing. In *Proc. of ANTS-V*, LNCS vol. 2369, pp. 324-337, 2002. [3]
14. C. Gentry and A. Silverberg. Hierarchical identity-based cryptography. In *Proc. of Asiacrypt'02*, LNCS vol. 2501, pp. 548C566, 2002. [2, 11]
15. S. Lal and P. Sharma. Security proof for Shengbao Wang's identity-based encryption scheme. Cryptology ePrint Archive, Report 2007/316. http://eprint.iarc.org. [1, 9]
16. N. McCullagh and P.S.L.M. Barreto. A new two-party identity-based authenticated key agreement. In *Proc. of CT-RSA'05*, LNCS vol. 3376, pp. 262-274, 2005. [2]
17. A. Shamir. Identity-based cryptosystems and signature schemes. In *Proc. of CRYPTO'84*, LNCS vol. 196, pp. 47-53, 1984. [2]
18. R. Sakai and M. Kasahara. ID based cryptosystems with pairing on elliptic curve. Cryptology ePrint Archive, Report 2003/054. http://eprint.iarc.org. [2]

19. A. Sahai and Brent Waters. Fuzzy identity-based encryption. In *Proc. of EUROCRYPT'05*, LNCS vol. 3494, pp. 457-473, 2005. [3]

20. H. Wang, S. Wang and Z. Cao. Efficient multi-receiver ID-based encryption scheme from pairings. Preprint, 2007. [12]

A Proof of Lemma 4

The proof idea is largely based on that of Lemma 4.3 in [4]. Let $\mathcal{A}$ be an IND-CPA adversary against **BasicPub** who makes at most q_2 queries to random oracle H_2 and who has advantage ϵ. We show how to construct an algorithm $\mathcal{B}$ which interacts with $\mathcal{A}$ to solve the MBDH problem.

Suppose $\mathcal{B}$ has an input $\langle \mathbb{G}_1, \mathbb{G}_2, e\rangle$ and $\langle P, aP, bP, cP\rangle$ (where $a, b, c \in \mathbb{Z}_q^*$ are unknown to $\mathcal{B}$). Let $D = e(P,P)^{a^{-1}bc} \in \mathbb{G}_2$ denote the solution to the MBDH problem on these inputs.

Setup: Algorithm $\mathcal{B}$ creates the public key of **BasicPub** $\langle p, \mathbb{G}_1, \mathbb{G}_2, e, P, P_{Pub}, n, Q_{ID}, H_2\rangle$ by setting $P_{Pub} = aP$ and $Q_{ID} = bP$. Here H_2 is a random oracle controlled by $\mathcal{B}$ as described bellow. $\mathcal{A}$ is given the public key. Observe that the unknown private key associated to the public key is $d_{ID} = a^{-1}Q_{ID} = a^{-1}bP$.

H_2**-queries:** To simulate H_2-queries by $\mathcal{A}$, $\mathcal{B}$ maintains a list (H_2-list) of pairs $\langle X_j, H_j\rangle$. To respond to an H_2 query X, $\mathcal{B}$ checks first if $X = X_j$ for some X_j already on the list. If it is, then $\mathcal{B}$ responds with H_j. Otherwise, $\mathcal{B}$ chooses H_2 uniformly at random from $\{0,1\}^m$ and places $\langle X, H\rangle$ on the H_2 list.

Challenge: $\mathcal{A}$ outputs two messages M_0, M_1 on which it wishes to be challenged. $\mathcal{B}$ picks randomly a bit $t \in \{0,1\}$, a string $S \in \{0,1\}^m$ and defines C to be the ciphertext of M_t, where $C = \langle U, V\rangle$, with $U = cP$ and $V = M_t \oplus S$. It then gives C to $\mathcal{A}$ as the challenge.

Notice that, by definition, the decryption of C is $V \oplus H(e(cP, a^{-1}bP)) = V \oplus H(D)$. (Recall that $a^{-1}bP$ is unknown and D is the solution to the above MBDH problem.)

Guess: $\mathcal{A}$ outputs its guess $t' \in \{0,1\}$.

Output: At this point, $\mathcal{B}$ picks a random tuple $\langle X_j, H_j\rangle$ from the H_2-list and outputs X_j as the solution to the given instance of MBDH problem.

It is easy to see that $\mathcal{A}$'s view in $\mathcal{B}$'s simulation is the same as in a real attack, in other words, the simulation is perfect. So $\mathcal{A}$'s advantage in this simulation will be ϵ. We let $\mathcal{H}$ be the event that D is queried to H_2 oracle during $\mathcal{B}$'s simulation.

Notice that $H_2(D)$ is independent of $\mathcal{A}$'s view, so if $\mathcal{A}$ never queries D to the H_2 oracle in the above simulation, then the decryption of C is also independent of its view. Therefore, in the simulation we have $\Pr[t = t' | \neg\mathcal{H}] = 1/2$. By the definition of $\mathcal{A}$, we know that in the real attack (and also in the simulation) $\mid \Pr[t = t'] - 1/2 \mid \geq \epsilon$. We have the following bounds on $\Pr[t = t']$:

$$\begin{aligned}
\Pr[t = t'] &= \Pr[t = t' | \neg\mathcal{H}]\Pr[\neg\mathcal{H}] + \Pr[t = t' | \mathcal{H}]\Pr[\mathcal{H}] \\
&\leq \Pr[t = t' | \neg\mathcal{H}]\Pr[\neg\mathcal{H}] + \Pr[\mathcal{H}] \\
&= \frac{1}{2}\Pr[\neg\mathcal{H}] + \Pr[\mathcal{H}] \\
&= \frac{1}{2} + \frac{1}{2}\Pr[\mathcal{H}],
\end{aligned}$$

$$\begin{aligned}
\Pr[t = t'] &\geq \Pr[t = t' | \neg\mathcal{H}]\Pr[\neg\mathcal{H}] \\
&= \frac{1}{2}\Pr[\neg\mathcal{H}] \\
&= \frac{1}{2}(1 - \Pr[\mathcal{H}]) \\
&= \frac{1}{2} - \frac{1}{2}\Pr[\mathcal{H}]).
\end{aligned}$$

Hence we have $\mid \Pr[t = t'] - 1/2 \mid \leq \frac{1}{2}\Pr[\mathcal{H}]$. By $\mid \Pr[t = t'] - 1/2 \mid \geq \epsilon$ we know that $\Pr[\mathcal{H}] \geq 2\epsilon$. Furthermore, by the definition of the event $\mathcal{H}$, we know that D appears in some tuple on the H_2-list with probability at least 2ϵ. It follows that $\mathcal{B}$ outputs the correct answer to the MBDH problem instance with probability at least $2\epsilon/q_2$ as required. Recall that in Lemma 1 we proved that the MBDH problem is as hard as the BDH problem, this ends the proof of Lemma 4. □