AN EFFICIENT ID-KEM BASED ON THE SAKAI-KASAHARA KEY CONSTRUCTION

L. CHEN, Z. CHENG, J. MALONE-LEE, AND N.P. SMART

ABSTRACT. We describe an identity based key encapsulation mechanism (ID-KEM). It is possible to use this ID-KEM to build a secure identity based encryption scheme using the techniques of Bentahar et al. The resulting encryption scheme has a number of performance advantages over existing methods.

1. INTRODUCTION

To simplify the management of public keys in public key based cryptosystems, Shamir [14] proposed identity-based cryptography in which the public key of each party may be derived from the party's identity. For a long while it was an open problem to obtain a secure and efficient identity based encryption (IBE) scheme. In 2000 and 2001, Sakai et al. presented an elegant identity-based key construction and other applications [16, 17]. Also in 2001 Boneh and Franklin [3], and Cocks [7] presented another two IBE solutions. Among these three schemes, the Sakai et al. scheme and the Boneh-Franklin scheme use bilinear pairings on elliptic curves.

In [3], Boneh and Franklin defined a security model for IBE. The Boneh-Franklin scheme (which we shall denote by BF-IBE) has received much attention owing to the fact that it was the first IBE scheme to have a proof of security in an appropriate model.

In 2003 Sakai and Kasahara proposed a new IBE system using elliptic curve pairings [18]. This system constructs keys using a different technique from previous schemes. In particular the key construction has the potential to improve performance over existing schemes. After employing the Fujisaki-Okamoto transformation [9], as in the BF-IBE construction, Chen and Cheng [6] proved that the security of the strengthened variant of Sakai-Kasahara scheme (which we shall denote by SK-IBE) can be reduced to a well-exploited hard problem: the q-bilinear Diffie-Hellman inversion problem (q-BDHI) [2].

A natural way to process arbitrarily long messages is to use *hybrid encryption*, unlike BF-IBE and SK-IBE which are stand alone constructions making use of the Fujisaki-Okamoto transformation. A hybrid encryption scheme consists of two basic operations. One operation uses a public-key encryption technique (a so called *key encapsulation mechanism* or KEM) to derive and encrypt a shared key; the other operation uses the shared key in a symmetric-key algorithm (a so called *data encapsulation mechanism* or DEM) to encrypt the actual message. Cramer and Shoup [8] formalized the notion of hybrid encryption and presented sufficient conditions for a KEM and a DEM to construct IND-CCA2 secure public key encryption. Recently, Bentahar et al. [4] extended the KEM concept to the identity based setting and gave three constructions of such an ID-KEM which when combined with a standard DEM provides a hybrid identity based encryption scheme which is ID-IND-CCA2, as defined by Boneh and Franklin [3].

One of the constructions of Bentahar et al. is a generic construction. It takes any identity based encryption scheme that is one-way under chosen plaintext attack (ID-OW-CPA) and transforms it into an ID-KEM that can easily be used to construct encryption schemes that are semantically secure against adaptive chosen ciphertext attack (ID-IND-CCA2). We shall present an ID-OW-CPA encryption scheme based on the Sakai-Kasahara method of constructing keys, and then via the generic construction of Bentahar et al. we shall produce an ID-IND-CCA2 secure ID-KEM and hence an ID-IND-CCA2 hybrid encryption scheme.

The scheme that we describe in this paper is more efficient than all previous schemes, and avoids many of the potential pitfalls related to the exact choice of groups which are used to instantiate the pairing. For more on these pitfalls consult [19].

The paper proceeds as follows. In the next section we set up notation and explain the concepts from related work on which we build. In particular we review the security definitions we require. In Section 3 we present an ID-KEM following the SK-IBE construction (which we call SK-ID-KEM) and prove its security. Then in Section 4 we compare our SK-ID-KEM's security and performance with some other ID based encryption schemes and ID-KEMs.

2. Preliminaries

We first present details of the bilinear groups we require and some associated hard problems. Having done this, in Section 2.2, we formally describe ID-based encryption and cover the basic security definitions. In Section 2.3 we present the extension of these ideas to the hybrid setting by recapping on ID-KEMs and how one constructs a full IBE scheme by combining an ID-KEM with a DEM.

2.1. **Bilinear Groups.**

Our schemes will require groups equipped with a bilinear map. Here we review the necessary facts about bilinear maps and the associated groups using the notation of [5].

- $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are (multiplicative) cyclic groups of prime order p.
- g_1 is a generator of $\mathbb{G}_1$ and g_2 is a generator of $\mathbb{G}_2$.
- ψ is an isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$ with $\psi(g_2) = g_1$.
- $\hat{e}$ is a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.

The map $\hat{e}$ must have the following properties.

Bilinear: For all $u \in \mathbb{G}_1$, all $v \in \mathbb{G}_2$ and all $a, b \in \mathbb{Z}$ we have $\hat{e}(u^a, v^b) = \hat{e}(u, v)^{ab}$.

Non-degenerate: $\hat{e}(g_1, g_2) \neq 1$.

Computable: There is an efficient algorithm to compute $\hat{e}(u, v)$ for all $u \in \mathbb{G}_1$ and $v \in \mathbb{G}_2$.

Note that the map ψ always exists, the issue is whether it can be efficiently computed. For the purposes of defining our schemes we do not assume that ψ is efficiently computable, however our security proofs require the simulator to be able to compute ψ. Hence, following [19], we can either assume that ψ is efficiently computable or make our security proofs relative to some oracle which computes ψ. This property occurs for a number of pairing based cryptographic schemes, but is very rarely pointed out by the authors.

Since the publication of [10] many hard problems pertaining to bilinear groups have been suggested for use in the design of cryptosystems. We describe two of these here.

Definition 1 (Bilinear Diffie-Hellman (BDH) [3]).
Given group elements $(g_1, g_2, g_2^x, g_2^y, g_2^z)$ *for* $x, y, z \in_R \mathbb{Z}_p$, *compute* $\hat{e}(g_1, g_2)^{xyz}$.

Definition 2 (q-Bilinear Diffie-Hellman Inverse (q-BDHI) [2]).
Given group elements $(g_1, g_2, g_2^x, g_2^{x2}, \ldots, g_2^{x^q})$ *with* $x \in_R \mathbb{Z}_p$, *compute* $\hat{e}(g_1, g_2)^{1/x}$.

It is the last of these problems on which our scheme's security is based, however we present the BDH problem for the purpose of subsequent comparisons between various schemes.

2.2. **ID-Based Encryption Schemes.** For an IBE scheme we define the message, ciphertext and randomness spaces by $\mathbb{M}_{\mathtt{ID}}(\cdot)$, $\mathbb{C}_{\mathtt{ID}}(\cdot)$, $\mathbb{R}_{\mathtt{ID}}(\cdot)$. These spaces are parametrised by the master public key $M_{\mathfrak{pk}}$, and hence by the security parameter t. The scheme itself is specified by four polynomial time algorithms:

- $\mathbb{G}_{\mathtt{ID}}(1^t)$: A probabilistic, polynomial-time (PPT henceforth) algorithm which takes as input 1^t and returns the master public key $M_{\mathfrak{pk}}$ and the master secret key $M_{\mathfrak{sk}}$.
- $\mathbb{X}_{\mathtt{ID}}(M_{\mathfrak{pk}}, M_{\mathfrak{sk}}, \mathtt{ID}_A)$: A PPT private key extraction algorithm which takes as input $M_{\mathfrak{pk}}, M_{\mathfrak{sk}}$ and $\mathtt{ID}_A \in \{0,1\}^*$, an identifier string for A, and returns the associated private key $D_{\mathtt{ID}_A}$.
- $\mathbb{E}_{\mathtt{ID}}(M_{\mathfrak{pk}}, \mathtt{ID}_A, m; r)$: This is the PPT encryption algorithm. On input of an identifier $\mathtt{ID}_A$, the master public key $M_{\mathfrak{pk}}$, a message $m \in \mathbb{M}_{\mathtt{ID}}(M_{\mathfrak{pk}})$ and possibly some randomness $r \in \mathbb{R}_{\mathtt{ID}}(M_{\mathfrak{pk}})$ this algorithm outputs $c \in \mathbb{C}_{\mathtt{ID}}(M_{\mathfrak{pk}})$.
- $\mathbb{D}_{\mathtt{ID}}(M_{\mathfrak{pk}}, \mathtt{ID}_A, D_{\mathtt{ID}_A}, c)$: This is the deterministic decryption algorithm. On input of the master public key $M_{\mathfrak{pk}}$, the identifier $\mathtt{ID}_A$, the private key $D_{\mathtt{ID}_A}$ and a ciphertext c this outputs the corresponding value of the plaintext m or a failure symbol $\perp$.

Following Boneh and Franklin [3] we can define various security notions for an IBE scheme. All are based on one of the following two-stage games between an adversary $A = (A_1, A_2)$ of the encryption algorithm and a challenger.

ID-OW Adversarial Game	ID-IND Adversarial Game
(1) $(M_{\mathfrak{pk}}, M_{\mathfrak{sk}}) \leftarrow \mathbb{G}_{\mathtt{ID}}(1^t)$.	(1) $(M_{\mathfrak{pk}}, M_{\mathfrak{sk}}) \leftarrow \mathbb{G}_{\mathtt{ID}}(1^t)$.
(2) $(s, \mathtt{ID}^*) \leftarrow A_1^{\mathcal{O}_{\mathtt{ID}}}(M_{\mathfrak{pk}})$.	(2) $(s, \mathtt{ID}^*, m_0, m_1) \leftarrow A_1^{\mathcal{O}_{\mathtt{ID}}}(M_{\mathfrak{pk}})$.
(3) $m \leftarrow \mathbb{M}_{\mathtt{ID}}(M_{\mathfrak{pk}})$.	(3) $b \leftarrow \{0,1\}$.
(4) $c^* \leftarrow \mathbb{E}_{\mathtt{ID}}(M_{\mathfrak{pk}}, \mathtt{ID}^*, m; r)$.	(4) $c^* \leftarrow \mathbb{E}_{\mathtt{ID}}(M_{\mathfrak{pk}}, \mathtt{ID}^*, m_b; r)$.
(5) $m' \leftarrow A_2^{\mathcal{O}_{\mathtt{ID}}}(M_{\mathfrak{pk}}, c^*, s, \mathtt{ID}^*)$.	(5) $b' \leftarrow A_2^{\mathcal{O}_{\mathtt{ID}}}(M_{\mathfrak{pk}}, c^*, s, \mathtt{ID}^*, m_0, m_1)$.

In the above, s is some state information and $\mathcal{O}_{\mathtt{ID}}$ are oracles to which the adversary has access. There are various possibilities for these oracles depending on the attack model for our game:

- CPA Model: In this model the adversary only has access to a private key extraction oracle which on input of $\mathtt{ID} \neq \mathtt{ID}^*$ will output the corresponding value of $D_{\mathtt{ID}}$.
- CCA2 Model: In this model the adversary has access to the private key extraction oracle as above and it also has access to a decryption oracle with respect to any identity $\mathtt{ID}$ of its choice. There is one restriction on how the adversary uses this oracle: in the second phase A is not allowed to call the decryption oracle with the pair $(c^*, \mathtt{ID}^*)$.

If we let MOD denote the mode of attack, either CPA or CCA2, the adversary's advantage in the first game is defined to be

$$\mathrm{Adv}_{\mathtt{ID}}^{\mathtt{ID-OW-MOD}}(A) = \Pr[m' = m],$$

while the advantage in the second game is given by

$$\mathrm{Adv}_{\mathtt{ID}}^{\mathtt{ID-IND-MOD}}(A) = |2\Pr[b' = b] - 1|.$$

An IBE algorithm is considered to be secure, in the sense of a given goal and attack model (ID-IND-CCA2 for example) if, for all PPT adversaries, the advantage in the relevant game is a negligible function of the security parameter t.

To cope with probabilistic ciphers, we will require that not too many choices for r encrypt a given message to a given ciphertext. To formalize this concept we let $\gamma(M_{\mathfrak{pk}})$ be the least upper bound such that

$$|\{r \in \mathbb{R}_{\mathtt{ID}}(M_{\mathfrak{pk}}) : \mathbb{E}_{\mathtt{ID}}(M_{\mathfrak{pk}}, \mathtt{ID}, m; r) = c\}| \leq \gamma(M_{\mathfrak{pk}}) \quad (1)$$

for every $\mathtt{ID}$, $m \in \mathbb{M}_{\mathtt{PK}}(M_{\mathfrak{pk}})$ and $c \in \mathbb{C}_{\mathtt{PK}}(M_{\mathfrak{pk}})$. Our requirement is that the quantity $\gamma(M_{\mathfrak{pk}})/|\mathbb{R}_{\mathtt{PK}}(M_{\mathfrak{pk}})|$ is a negligible function of the security parameter.

2.3. **ID-Based Key Encapsulation Mechanisms.** Following Cramer and Shoup's formalisation of hybrid encryption [8], Bentahar et al. extended the hybrid encryption concept to identity-based schemes [4]. The idea is to construct an ID-IND-CCA2 secure IBE scheme from an ID-IND-CCA2 secure ID-KEM and a secure DEM.

An ID-KEM scheme is specified by four polynomial time algorithms:

- $\mathbb{G}_{\mathtt{ID-KEM}}(1^t)$: The PPT *master key generation algorithm* which takes as input 1^t. It outputs the master public key $M_{\mathfrak{pk}}$ and the master secret key $M_{\mathfrak{sk}}$.
- $\mathbb{X}_{\mathtt{ID-KEM}}(M_{\mathfrak{pk}}, M_{\mathfrak{sk}}, \mathtt{ID}_A)$: The PPT *private key extraction algorithm* which takes as input $M_{\mathfrak{pk}}, M_{\mathfrak{sk}}$ and $\mathtt{ID}_A \in \{0,1\}^*$, an identifier string for A. It outputs the associated private key $D_{\mathtt{ID}_A}$.
- $\mathbb{E}_{\mathtt{ID-KEM}}(M_{\mathfrak{pk}}, \mathtt{ID}_A)$: The PPT *encapsulation algorithm* which takes as input $\mathtt{ID}_A$ and $M_{\mathfrak{pk}}$. It outputs a pair (k, c) where $k \in \mathbb{K}_{\mathtt{ID-KEM}}(M_{\mathfrak{pk}})$ is a key and $c \in \mathbb{C}_{\mathtt{ID-KEM}}(M_{\mathfrak{pk}})$ is the encapsulation of that key.
- $\mathbb{D}_{\mathtt{ID-KEM}}(M_{\mathfrak{pk}}, \mathtt{ID}_A, D_{\mathtt{ID}_A}, c)$: The deterministic *decapsulation algorithm* which takes as input $M_{\mathfrak{pk}}$, $\mathtt{ID}_A$, c and $D_{\mathtt{ID}_A}$. It outputs k or a failure symbol $\perp$.

We shall only require one security definition for our ID-KEMs, although other weaker definitions can be defined in the standard way. Consider the following two-stage game between an adversary $A = (A_1, A_2)$ of the ID-KEM and a challenger.

ID-IND Adversarial Game

(1) $(M_{\mathfrak{pk}}, M_{\mathfrak{sk}}) \leftarrow \mathbb{G}_{\mathtt{ID-KEM}}(1^t)$.
(2) $(s, \mathtt{ID}^*) \leftarrow A_1^{\mathcal{O}_{\mathtt{ID}}}(M_{\mathfrak{pk}})$.
(3) $(k_0, c^*) \leftarrow \mathbb{E}_{\mathtt{ID-KEM}}(M_{\mathfrak{pk}}, \mathtt{ID}^*)$.
(4) $k_1 \leftarrow \mathbb{K}_{\mathtt{ID-KEM}}(M_{\mathfrak{pk}})$.
(5) $b \leftarrow \{0, 1\}$.
(6) $b' \leftarrow A_2^{\mathcal{O}_{\mathtt{ID}}}(M_{\mathfrak{pk}}, c^*, s, \mathtt{ID}^*, k_b)$.

In the above s is some state information and $\mathcal{O}_{\mathtt{ID}}$ denotes oracles to which the adversary has access. We shall be interested in the CCA2 attack model where the adversary has access to two oracles:

(1) A private key extraction oracle which, on input of $\mathtt{ID} \neq \mathtt{ID}^*$, will output the corresponding value of $D_{\mathtt{ID}}$.
(2) A decapsulation oracle which, on input an identity $\mathtt{ID}$ and encapsulation of its choice, will return the encapsulated key. This is subject to the restriction that in the second phase A is not allowed to call this oracle with the pair $(c^*, \mathtt{ID}^*)$.

The adversary's advantage is defined to be

$$\mathrm{Adv}_{\mathtt{ID-KEM}}^{\mathtt{ID-IND-CCA2}}(A) = |2\Pr[b' = b] - 1|.$$

An ID-KEM is considered to be secure, if for all PPT adversaries A, the advantage in the game above is a negligible function of the security parameter t.

2.4. **Hybrid IBE.** A hybrid IBE $\mathcal{E} = (\mathbb{G}_{\mathtt{ID}}, \mathbb{X}_{\mathtt{ID}}, \mathbb{E}_{\mathtt{ID}}, \mathbb{D}_{\mathtt{ID}})$ construction consists of combining an ID-KEM $\mathcal{E}_1 = (\mathbb{G}_{\mathtt{ID-KEM}}, \mathbb{X}_{\mathtt{ID-KEM}}, \mathbb{E}_{\mathtt{ID-KEM}}, \mathbb{D}_{\mathtt{ID-KEM}})$ with a standard DEM $\mathcal{E}_2 = (\mathbb{E}_{\mathtt{SK}}, \mathbb{D}_{\mathtt{SK}})$ as described below. For the formal definition of a DEM and its security definition that we use in Theorem 1, refer to [8] and [4].

We assume that the key-space of the KEM is the same as the key-space of the associated DEM. To generate $M_{\mathfrak{pk}}$, for the hybrid IBE scheme, the algorithm $\mathbb{G}_{\mathtt{ID-KEM}}(1^t)$ is run. We denote the resulting full key $M_{\mathfrak{pk}}$ below. Key extraction for $\mathcal{E}$ is simply the key extraction of $\mathcal{E}_1$.

$\mathbb{E}_{\mathtt{ID}}(M_{\mathfrak{pk}}, \mathtt{ID}, m)$

- $(k, c_1) \leftarrow \mathbb{E}_{\mathtt{ID-KEM}}(M_{\mathfrak{pk}}, \mathtt{ID})$
- $c_2 \leftarrow \mathbb{E}_{\mathtt{SK}}(k, m)$
- Return $c = (c_1, c_2)$

$\mathbb{D}_{\mathtt{ID}}(M_{\mathfrak{pk}}, \mathtt{ID}, D_{\mathtt{ID}}, c)$

- Parse c as (c_1, c_2)
- $k \leftarrow \mathbb{D}_{\mathtt{ID-KEM}}(M_{\mathfrak{pk}}, \mathtt{ID}, D_{\mathtt{ID}}, c)$
- If $k = \perp$, return $\perp$
- $m \leftarrow \mathbb{D}_{\mathtt{SK}}(k, c_2)$
- Return m

Similar to the result of hybrid encryption in [8], Bentahar et al. obtained the following theorem concerning the security of hybrid IBE.

Theorem 1. *[Bentahar et al. [4]] Let A be a PPT ID-IND-CCA2 adversary of the IBE scheme $\mathcal{E}$ above. There exists PPT adversaries B_1 and B_2, whose running time is essentially that of A, such that*

$$\mathrm{Adv}_{\mathtt{ID}}^{\mathtt{ID-IND-CCA2}}(A) \leq 2\mathrm{Adv}_{\mathtt{ID-KEM}}^{\mathtt{ID-IND-CCA2}}(B_1) + \mathrm{Adv}_{\mathtt{DEM}}^{\mathtt{FG-CCA}}(B_2).$$

Some IND-CCA secure DEMs are readily available, see [15] and [1]. Bentahar et al. presented two secure ID-KEMs using the same key format as that used in the BF-IBE scheme [3]. In the following section, we introduce another ID-KEM based on Sakai and Kasahara's IBE proposal which has the potential to achieve even better performance.

3. An SK-ID-KEM Construction

Before discussing our construction we briefly summarise the primitives proposed by Sakai et al. [16, 17, 18], on which our own contribution is based. In what follows we will assume that we have bilinear groups as defined in Section 2.1. We make the assumption that $\mathbb{G}_1 = \mathbb{G}_2$ while we are describing the schemes of Sakai et al.; elsewhere in the paper we do not make this assumption.

In the first two of the papers by Sakai et al. [16, 17], the schemes work using a function

$$\zeta : \{0,1\}^* \to \mathbb{G}_1$$

which is used to map identities $\mathtt{ID} \in \{0,1\}^*$ to elements of $\mathbb{G}_1$. There is a TA that chooses a master secret key s from $\mathbb{Z}_p$. One application discussed by Sakai et al. is key agreement. Suppose that $\mathtt{ID}_a$ wishes to encrypt a message and send it to $\mathtt{ID}_b$. It first obtains its secret key $D_{\mathtt{ID}_a} = \zeta(\mathtt{ID}_a)^s$ from the TA. It then computes a key $K_{ab} \leftarrow \hat{e}(D_{\mathtt{ID}_a}, \zeta(\mathtt{ID}_b))$. If $\mathtt{ID}_b$ obtains its own secret key $D_{\mathtt{ID}_b} = \zeta(\mathtt{ID}_b)^s$, it can also compute $K_{ab} \leftarrow \hat{e}(\zeta(\mathtt{ID}_a), D_{\mathtt{ID}_b})$. This construction of keys is exactly the same as that used in the Boneh–Franklin scheme [3], where ζ is instantiated using a cryptographic hash function.

The method used to construct keys given in the paper by Sakai and Kasahara [18] is slightly different. The TA chooses two generators u_α and u_β of $\mathbb{G}_1$. It also chooses a polynomial η of degree d (where d is a parameter of the scheme)

$$\eta(x) = a_d x^d + a_{d-1}x^{d-1} + \ldots + a_1 x + a_0$$

where the coefficients are randomly chosen from $\mathbb{Z}_p$. The TA's master secret key is made up of the coefficients of the polynomial. Its corresponding master public key is $\hat{e}(u_\alpha, u_\beta)$ together with $u_\beta^{a_d}, u_\beta^{a_{d-1}}, \ldots, u_\beta^{a_1}, u_\beta^{a_0}$. Now, to extract a secret key for identity $\mathtt{ID}$, the TA computes

$$D_{\mathtt{ID}} \leftarrow u_\alpha^{1/\eta(\mathtt{ID})}.$$

It is this method of constructing keys that our scheme uses to produce our KEM, however we are able to use the simplification of setting $d = 1$.

We are now ready to describe our construction. Two stages are required. In the first stage, Section 3.1, we present a concrete instantiation of a new ID-OW-CPA secure IBE scheme. One should think of this construction as analogous to the BasicIdent scheme in [3]. In the second stage, Section 3.1, we use a generic construction from [4] which turns an ID-OW-CPA secure IBE scheme into an ID-IND-CCA2 secure ID-KEM. Such an ID-KEM can then be used to build an ID-IND-CCA2 secure encryption scheme using the construction of Theorem 1 [4]. We denote the resulting encryption SK-C2 henceforth.

3.1. An ID-OW-CPA IBE scheme based on Sakai-Kasahara keys.

Let t be the security parameter. The system parameters consist of groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$, as defined in Section 2.1, with order $p \approx 2^t$ and a bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. In addition we require a generator u_1 for $\mathbb{G}_1$ and a generator u_2 for $\mathbb{G}_2$ such that $u_1 = \psi(u_2)$. The scheme also uses two hash functions:

$$H_1 : \{0,1\} \to \mathbb{Z}_p \text{ and } H_2 : \mathbb{G}_T \to \{0,1\}^n$$

where $\{0,1\}^n$ is the message space. It works as follows.

- $\mathbb{G}_{\mathtt{ID}}(1^t)$: Select $s \in \mathbb{Z}_p$ at random and set $R = u_1^s$. The value s is the master secret key $M_{\mathfrak{sk}}$ of the TA (a trusted authority), while R along with the other system parameters is the master public key $M_{\mathfrak{pk}}$.
- $\mathbb{X}_{\mathtt{ID}}(M_{\mathfrak{pk}}, \mathtt{ID}, s)$: This outputs the identity-based secret key
$$D_{\mathtt{ID}} = u_2^{1/(s+H_1(\mathtt{ID}))}.$$
Note this will fail and, moreover, $M_{\mathfrak{sk}}$ will be revealed if $H_1(\mathtt{ID}) = -s$; however, this happens with negligible probability.
- $\mathbb{E}_{\mathtt{ID}}(M_{\mathfrak{pk}}, \mathtt{ID}, m; r)$:
  - $Q \leftarrow R \cdot u_1^{H_1(\mathtt{ID})}$
  - $U \leftarrow Q^r$
  - $V \leftarrow m \oplus H_2(\hat{e}(u_1, u_2)^r)$
  - Return (U, V)
- $\mathbb{D}_{\mathtt{ID}}(M_{\mathfrak{pk}}, \mathtt{ID}, D_{\mathtt{ID}}, (U, V))$: This outputs
$$V \oplus H_2(\hat{e}(U, D_{\mathtt{ID}}))$$

We now present the security result for the IBE scheme above.

Theorem 2. *Suppose that there is algorithm A which breaks the above scheme in terms of ID-OW-CPA. If we model H_1 and H_2 as random oracles, and we let q_1, q_2 and q_X be the number of queries that A makes to H_1, H_2 and its key extraction oracle respectively. Then there is an algorithm B to solve the q-BDHI problem in groups of order p with $q = q_1 + q_X + 1$ such that*
$$\mathrm{Adv}_{\mathtt{ID}}^{\mathtt{ID-OW-CPA}}(A) \leq (q \cdot q_2 + 1) \cdot \mathrm{Adv}^{q\mathtt{-BDHI}}(B) + \frac{1}{2^n} + \frac{q+1}{p}.$$

The proof of this theorem is given in the appendix.

3.2. **Generic Reduction.** Here we take a generic, probabilistic ID-based encryption scheme, which is ID-OW-CPA secure. Let the encryption algorithm be denoted $\mathbb{E}_{\mathtt{ID}}(M_{\mathfrak{pk}}, \mathtt{ID}, m; r)$ and the decryption algorithm be denoted $\mathbb{D}_{\mathtt{ID}}(M_{\mathfrak{pk}}, \mathtt{ID}, D_{\mathtt{ID}}, c)$, where $D_{\mathtt{ID}}$ is the output from the extraction algorithm $\mathbb{X}_{\mathtt{ID-KEM}}(M_{\mathfrak{pk}}, M_{\mathfrak{sk}}, \mathtt{ID})$. We assume the message space of $\mathbb{E}_{\mathtt{ID}}$ is given by $\mathbb{M}_{\mathtt{ID}}(M_{\mathfrak{pk}})$ and the space of randomness is given by $\mathbb{R}_{\mathtt{ID}}(M_{\mathfrak{pk}})$. The construction uses two cryptographic hash functions:
$$H_3 : \{0,1\}^* \to \mathbb{R}_{\mathtt{ID}}(M_{\mathfrak{pk}}) \text{ and } H_4 : \{0,1\}^* \to \{0,1\}^\kappa$$
for $\kappa \in \mathbb{Z}$: the length of the resulting keys. Using this we construct an ID-KEM as follows.

$\mathbb{E}_{\mathtt{ID-KEM}}(M_{\mathfrak{pk}}, \mathtt{ID})$:
- $m \leftarrow \mathbb{M}_{\mathtt{ID}}(M_{\mathfrak{pk}})$
- $r \leftarrow H_3(m)$
- $c \leftarrow \mathbb{E}_{\mathtt{ID}}(M_{\mathfrak{pk}}, \mathtt{ID}, m; r)$
- $k \leftarrow H_4(m)$
- Return (k, c)

$\mathbb{D}_{\mathtt{ID-KEM}}(M_{\mathfrak{pk}}, \mathtt{ID}, D_{\mathtt{ID}}, c)$:
- $m \leftarrow \mathbb{D}_{\mathtt{ID}}(M_{\mathfrak{pk}}, \mathtt{ID}, D_{\mathtt{ID}}, c)$
- If $m = \perp$, return $\perp$
- $r \leftarrow H_3(m)$
- If $c \neq \mathbb{E}_{\mathtt{ID}}(M_{\mathfrak{pk}}, \mathtt{ID}, m; r)$, return $\perp$
- $k \leftarrow H_4(m)$
- Return k

From [4] we have the following theorem concerning the security of the construction above.

Theorem 3. *If $\mathbb{E}_{\mathtt{ID}}$ is an ID-OW-CPA secure ID-based encryption scheme and H_3 and H_4 are modelled as random oracles then the construction above is secure against adaptive chosen ciphertext attack.*

Specifically, if A is a PPT algorithm that breaks the ID-KEM construction above using a chosen ciphertext attack, then there exists a PPT algorithm B, with

$$\mathrm{Adv}_{\mathtt{ID-KEM}}^{\mathtt{ID-IND-CCA2}}(A) \leq 2(q_3+q_4+q_D)\cdot \mathrm{Adv}_{\mathtt{ID}}^{\mathtt{ID-OW-CPA}}(B) + \frac{2q_D\gamma(M_{\mathfrak{pk}})}{|\mathbb{R}_{\mathtt{ID}}(M_{\mathfrak{pk}})|},$$

where q_3, q_4 and q_D are the number of queries made by A to H_3, H_4 and the decryption oracle respectively, and $\gamma(M_{\mathfrak{pk}})$ is as in (1).

When we instantiate this generic construction with our ID-OW-CPA scheme from Stage 1, we have

$$\frac{\gamma(M_{\mathfrak{pk}})}{|\mathbb{R}_{\mathtt{ID}}(M_{\mathfrak{pk}})|} \approx \frac{1}{p}.$$

3.3. **Full Scheme.** The full ID-KEM scheme works as follows. The algorithms $\mathbb{G}_{\mathtt{ID-KEM}}$ and $\mathbb{X}_{\mathtt{ID-KEM}}$ are simply $\mathbb{G}_{\mathtt{ID}}$ and $\mathbb{X}_{\mathtt{ID}}$ for the IBE scheme above.

$\mathbb{E}_{\mathtt{ID-KEM}}(M_{\mathfrak{pk}}, \mathtt{ID})$
- $m \leftarrow \{0,1\}^n$
- $r \leftarrow H_3(m)$
- $Q \leftarrow R \cdot u_1^{H_1(\mathtt{ID})}$
- $U \leftarrow Q^r$
- $V \leftarrow m \oplus H_2(\hat{e}(u_1,u_2)^r)$
- $k \leftarrow H_4(m)$
- $c \leftarrow (U,V)$
- Return (k,c)

$\mathbb{D}_{\mathtt{ID-KEM}}(M_{\mathfrak{pk}}, \mathtt{ID}, D_{\mathtt{ID}}, c)$
- Parse c as (U,V)
- $\alpha \leftarrow \hat{e}(U, D_{\mathtt{ID}})$
- $m \leftarrow H_2(\alpha) \oplus V$
- $r \leftarrow H_3(m)$
- If $(U,V) \neq \mathbb{E}_{\mathtt{ID}}(M_{\mathfrak{pk}}, \mathtt{ID}, m; r)$, return $\perp$
- $k \leftarrow H_4(m)$
- Return k

Note that $\hat{e}(u_1,u_2)$ can be included in the master public key to minimise the number of pairing computations necessary.

We now look at the validity check in more detail. We need to ensure that the following holds

$$\begin{aligned} U &= Q^r \\ V &= m \oplus H_2(\hat{e}(u_1,u_2)^r), \end{aligned}$$

where

$$\begin{aligned} Q &= R \cdot u_1^{H_1(\mathtt{ID})} \\ m &= V \oplus H_2(\hat{e}(u_1,u_2)^r). \end{aligned}$$

However, if $U = Q^r$ then α is always equal to $\hat{e}(u_1,u_2)^r$. In this case V always equals $m \oplus H_2(\alpha)$ and m is defined to be $V \oplus H_2(\alpha)$. This means that checking whether or not V is correct is redundant. Hence, we only need to check whether $U = Q^r$. Since the decryptor knows its own identity, it can be assumed to have precomputed the value of Q, therefore the validity check involves only one exponentiation in $\mathbb{G}_1$.

4. Comparison with Other Schemes

In this section we compare the SK-C2 scheme from Section 3 with the other efficient ID-based encryption schemes in the literature.

Scheme	pairings		exponentiations		hashes	
	$\mathbb{E}_{\mathtt{ID}}$	$\mathbb{D}_{\mathtt{ID}}$	$\mathbb{E}_{\mathtt{ID}}$	$\mathbb{D}_{\mathtt{ID}}$	$\mathbb{E}_{\mathtt{ID}}$	$\mathbb{D}_{\mathtt{ID}}$
BF-IBEa	1	1	2	1	4	3
SK-IBEa	0	1	3	1	4	3
BF-C1	1	1	2	0	2	1
BF-C2	1	1	2	1	4	3
SK-C2	0	1	3	1	4	3

TABLE 1. The computations necessary for various IBE schemes

- **BF-IBE**: The original Boneh-Franklin scheme which is secure assuming the BDH problem is hard. The ID-based keys are constructed in the standard way by hashing to a point in either $\mathbb{G}_1$ or $\mathbb{G}_2$. The associated secret key is obtained by multiplying this point by the master secret. We use **BF-IBEa** to denote the extension of the Boneh-Franklin in which an arbitrary block cipher is used instead of **xor**. In [4] this latter version is referred to as **FullIdent-2**. Note, BF-IBEa does not need to be used with a full DEM; a standard block cipher secure against passive attacks is sufficient.
- **SK-IBE**: The scheme described in [6]. This uses the keys construction of Sakai and Kasahara as in the current paper. The scheme is secure assuming the q-BDHI problem is hard. Similar to BF-IBEa, we can define an SK-IBEa by replacing **xor** with a block cipher.
- **BF-C1**: Construction C-1 from [4]. This is a hybrid KEM based construction, originally mentioned in a paper by Lynn [11]. It is secure assuming a suitable gap problem is hard. The keys are of the same form as those in the Boneh-Franklin scheme.
- **BF-C2**: Construction C-2 from [4]. This uses the generic construction used in this paper applied to the BasicIdent scheme of [3].

Note, all of the above scheme are secure in the random oracle model. We have not considered comparisons with schemes secure in the standard model as they are very inefficient.

To compare efficiency we first look at the computations necessary to implement the various schemes in Table 1. The first two rows of the table correspond to IBE schemes, while the last three refer to ID-KEM/DEM hybrid constructions. We assume that the obvious precomputations have been performed in all cases.

We see that the schemes based on the Sakai-Kasahara key construction do not have to perform a pairing in their encryption routine. This comes at the expense of an extra group exponentiation, however these are usually much cheaper than a pairing computation. In addition we note that using the Sakai-Kasahara method of constructing keys, as opposed to the method of Boneh and Franklin, avoids the need to hash into an elliptic curve group. As pointed out in [19], hashing into the group can cause problems if the groups are not chosen in a suitable way. In addition, hashing into an elliptic curve is in general more expensive both in terms of CPU time and code footprint size than hashing into the integers.

	BF-IBEa	SK-C2
ID-Public Key Gen	18	4
ID-Private Key Gen	113	88
ID-Encrypt	75	30
ID-Decrypt	55	62

TABLE 2. Comparison of CPU time in milliseconds

In Table 2 we compare an implementation of our construction with that of BF-IBEa for a 160-bit MNT-type curve[1]. The improvement in performance comes from the lack of a pairing computation on encryption and the lack of a need to hash into an elliptic curve group.

We reiterate that using an ID-KEM/DEM construction is more flexible as it facilitates identity based encryption with any appropriate DEM to encrypt the actual data packet, or even the use of the KEM on its own to transmit a key for another application. This philosophy for designing public key encryption algorithms is well explained in [8] and [15], so we do not go into the benefits more here.

We now turn to the ciphertext sizes of the various schemes above. In Table 3 we let $|\mathbb{G}_1|$ denote the number of bits needed to represent an element in the group $\mathbb{G}_1$ and use analogous notation for other components. It is convention that when instantiated with elliptic curves, the group $\mathbb{G}_1$ refers to the subgroup of order p of an elliptic curve over the "small" finite field. Then for supersingular elliptic curves we have $\mathbb{G}_1 = \mathbb{G}_2$, however for so-called MNT curves we have that $\mathbb{G}_2$ is related to a subgroup of the twisted elliptic curve over a large finite field. Hence, representing elements of $\mathbb{G}_2$ can require more bits than are required to represent elements of $\mathbb{G}_1$.

In Table 3 we also mention whether the scheme requires hashing into either the group $\mathbb{G}_1$ or the group $\mathbb{G}_2$. One should note that hashing into $\mathbb{G}_2$ can be computationally expensive as pointed out in [19] for certain choices of groups, while hashing into $\mathbb{G}_1$ is usually very efficient. As in [13], we let BF-IBE$^\perp$ denote the protocol BF-IBE but with the roles of $\mathbb{G}_1$ and $\mathbb{G}_2$ reversed. We use analogous notation for other schemes. Note, reversing the roles of $\mathbb{G}_1$ and $\mathbb{G}_2$ can have effects on the security proof or on other aspects related to efficiency. See [19] for more details. Note that the only case in which reversing the roles of $\mathbb{G}_1$ and $\mathbb{G}_2$ makes no difference is the case of supersingular elliptic curves for which $\mathbb{G}_1 = \mathbb{G}_2$.

We do not give rows for the Sakai-Kasahara based schemes where the roles of $\mathbb{G}_1$ and $\mathbb{G}_2$ are reversed; reversing the roles of $\mathbb{G}_1$ and $\mathbb{G}_2$ only reduces bandwidth efficiency for no gain in performance, as for these schemes one never has to hash into $\mathbb{G}_1$ or $\mathbb{G}_2$.

In Table 3, n either refers to the key length of the DEM, or the size of σ in the standard Boneh-Franklin IBE schemes. We note that for the schemes with Boneh-Franklin style keys one either needs to choose, for MNT curves, between low bandwidth and hashing into $\mathbb{G}_2$, or high bandwidth and hashing into $\mathbb{G}_1$.

Bandwidth for ciphertexts can be further reduced as follows. In the ciphertext we transmit the element $U \in \mathbb{G}_1$, which is a point on an elliptic curve in practice. We could clearly compress the point U. However, compression usually entails sending

[1]The MNT curves are those non-supersingular elliptic curves that are suitable for pairing-based cryptography. The name comes from the initials of the authors who gave the first construction of such curves [12].

scheme	ciphertext size	hashing	
		$\mathbb{G}_1$	$\mathbb{G}_2$
BF-IBE	$\lvert\mathbb{G}_1\rvert + n + \lvert m\rvert$	N	Y
BF-IBEa	$\lvert\mathbb{G}_1\rvert + n + \lvert\mathbb{E}_{\mathtt{SK}}(m)\rvert$	N	Y
BF-IBE$^{\perp}$	$\lvert\mathbb{G}_2\rvert + n + \lvert m\rvert$	Y	N
BF-IBEa$^{\perp}$	$\lvert\mathbb{G}_2\rvert + n + \lvert\mathbb{E}_{\mathtt{SK}}(m)\rvert$	Y	N
SK-IBE	$\lvert\mathbb{G}_1\rvert + n + \lvert m\rvert$	N	N
SK-IBEa	$\lvert\mathbb{G}_1\rvert + n + \lvert\mathbb{E}_{\mathtt{SK}}(m)\rvert$	N	N
BF-C1	$\lvert\mathbb{G}_1\rvert + \lvert\mathbb{E}_{\mathtt{DEM}}(m)\rvert$	N	Y
BF-C2	$\lvert\mathbb{G}_1\rvert + n + \lvert\mathbb{E}_{\mathtt{DEM}}(m)\rvert$	N	Y
BF-C1$^{\perp}$	$\lvert\mathbb{G}_2\rvert + \lvert\mathbb{E}_{\mathtt{DEM}}(m)\rvert$	Y	N
BF-C2$^{\perp}$	$\lvert\mathbb{G}_2\rvert + n + \lvert\mathbb{E}_{\mathtt{DEM}}(m)\rvert$	Y	N
SK-C2	$\lvert\mathbb{G}_1\rvert + n + \lvert\mathbb{E}_{\mathtt{DEM}}(m)\rvert$	N	N

TABLE 3. The bandwidth requirements of various IBE schemes

an extra bit so as to uniquely decompress the point. This is unnecessary for the cost of one field inversion. Suppose we only transmit the x-coordinate of the point U, in which case the receiver only knows U up to sign. Hence, he can only compute

$$\alpha \leftarrow \hat{e}(\pm U, D_{\mathtt{ID}})^{\pm 1}.$$

But by computing

$$H_2(\alpha + \alpha^{-1})$$

instead of

$$H_2(\alpha),$$

a unique value will be produced. In particular this technique avoids the need to transmit an extra bit to uncompress the x-coordinate $x(U)$ to a unique point, and it does not affect the security proof. One obviously has to modify the validity check slightly.

We note that an analogous construction to C-1 from [4] can be applied to the Sakai-Kasahara method of constructing keys. This scheme is efficient and can be proved secure using a suitable, but slightly unnatural, gap problem using similar techniques to the proof of construction C-1 from [4].

REFERENCES

[1] ISO/IEC FDIS 18033-2. Information technology - Security techniques - Encryption algorithms - Part 2: Asymmetric ciphers. 2005.

[2] D. Boneh and X. Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In *Advances in Cryptology - Eurocrypt 2004*, volume 3027 of LNCS, pages 223-238. Springer-Verlag, 2004.

[3] D. Boneh and M. Franklin. Identity based encryption from the Weil pairing. In *Advances in Cryptology - Crypto 2001*, volume 2139 of LNCS, pages 213-229. Springer-Verlag, 2001.

[4] K. Bentahar, P. Farshim, J. Malone-Lee and N.P. Smart. Generic constructions of identity-based and certificateless KEMs. Cryptology ePrint Archive, Report 2005/058. 2005.

[5] D. Boneh, B. Lynn and H. Shacham. Short signatures from the Weil pairing. In *Advances in Cryptology - Asiacrypt 2001*, volume 2248 of LNCS, pages 514-532. Springer-Verlag, 2001.

[6] L. Chen and Z. Cheng. Security proof of Sakai-Kasahar's identity-based encryption scheme. In *Proceedings of Cryptography and Coding 2005*, volume 3796 of LNCS, pages 442-459. Springer-Verlag, 2005.

[7] C. Cocks. An identity-based encryption scheme based on quadratic residues. In *Proceedings of Cryptography and Coding 2001*, volume 2260 of LNCS, pages 360-363. Springer-Verlag, 2001.
[8] R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33, 167-226, 2003.
[9] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology - CRYPTO '99*, volume 1666 of LNCS, pages 535-554. Springer-Verlag, 1999.
[10] A. Joux. A one round protocol for tripartite Diffie-Hellman. In *Algorithmic Number Theory Symposium - ANTS IV*, volume 1838 of LNCS, pages 385-394. Springer-Verlag, 2000.
[11] B. Lynn. Authenticated identity-based encryption. Cryptology ePrint Archive, Report 2002/072. 2002.
[12] A. Miyaja and M. Nakabayashi and S. Takano. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E84-A(5):1234-1243, 2001.
[13] D. Page, N.P. Smart and F. Vercauteren. A comparison of MNT curves and supersingular curves. Cryptology ePrint Archive, Report 2004/165. 2004.
[14] A. Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology - CRYPTO '84*, volume 196 of LNCS, pages 47-53. Springer-Verlag, 1985.
[15] V. Shoup. A proposal for an ISO standard for sublic key encryption. 2001. Available from `www.shoup.net`.
[16] R. Sakai, K. Ohgishi and M. Kasahara. Cryptosystems based on pairing. In *The 2000 Symposium on Cryptography and Information Security*, Okinawa, Japan, January 2000.
[17] R. Sakai, K. Ohgishi and M. Kasahara. Cryptosystems based on pairing over elliptic curve (in Japanese). In *The 2001 Symposium on Cryptography and Information Security*, Oiso, Japan, January 2001.
[18] R. Sakai and M. Kasahara. ID based cryptosystems with pairing on elliptic curve. Cryptology ePrint Archive, Report 2003/054. 2003.
[19] N.P. Smart and F. Vercauteren. On computable isomorphisms in efficient pairing based systems. Cryptology ePrint Archive, Report 2005/116. 2005.

Appendix A. Proof of Theorem 2

To prove our theorem we will show how to use A to construct an algorithm B to solve the q-BDHI problem, where $q = q_1 + q_X + 1$. This construction will involve running A in a simulated environment. Henceforth all probabilities will be probabilities in our simulated environment.

Algorithm B proceeds as follows. It takes as input

$$(g_1, g_2, g_2^x, g_2^{x2}, g_2^{x3}, \ldots, g_2^{x^q}) \in \mathbb{G}_1 \times \mathbb{G}_2^{q+1}$$

with $g_1 = \psi(g_2)$ and then selects an integer $I \in \{1, \ldots, q\}$. It uses these to set up the domain parameters and keys for the ID-based encryption algorithm as described below.

Algorithm B selects $h_0, \ldots, h_{q-1}$ uniformly at random from $\mathbb{Z}_p$. We define the event `Guess` to be that in which $h_i = -x$ for some i in $\{1, \ldots, q-1\}$. (This event can be checked by computing $g_2^{-h_i}$ for i in $\{1, \ldots, q-1\}$ and comparing these values with g_2^x.)

We say that A *wins* if it outputs the correct value of the encrypted message in its attack. By definition have

$$\begin{aligned} \mathrm{Adv}_{\mathtt{ID}}^{\mathtt{ID-OW-CPA}}(A) &= \Pr[A \text{ wins} \wedge \mathtt{Guess}] + \Pr[A \text{ wins} \wedge \neg\mathtt{Guess}] \\ &\leq \Pr[\mathtt{Guess}] + \Pr[A \text{ wins} | \neg\mathtt{Guess}] \\ &\leq \mathrm{Adv}^{q-\mathtt{BDHI}}(B) + \Pr[A \text{ wins} | \neg\mathtt{Guess}]. \qquad (2) \end{aligned}$$

Equation (2) follows from the fact that, in the event $\mathtt{Guess}$, algorithm B finds x which it can then use to solve the q-BDHI problem directly by computing $\hat{e}(g_1, g_2)^{1/x}$.

We are now ready to describe the non-trivial part of the simulation. In the remainder of the proof we will assume that the event $\neg\mathtt{Guess}$ has occurred and so all probabilities are conditioned on this event.

Now, B defines the polynomial

$$f(z) = \prod_{i=1}^{q-1} (z + h_i) = \sum_{i=0}^{q-1} c_i z^i,$$

computes

$$u_2 = \prod_{i=0}^{q-1} (g_2^{x^i})^{c_i} = g_2^{f(x)}$$

and

$$u_2' = \prod_{i=0}^{q-1} (g_2^{x^{i+1}})^{c_i} = g_2^{x f(x)} = u_2^x.$$

Note that, in the event $\neg\mathtt{Guess}$, we have $u_2 \neq 1$ and so u_2 is a generator of $\mathbb{G}_2$.

Algorithm B then defines the polynomials

$$f_i(z) = f(z)/(z + h_i) = \sum_{j=0}^{q-2} d_{i,j} z^j, \text{ for } 1 \leq i < q.$$

Note that

$$u_2^{1/(x+h_i)} = g_2^{f_i(x)} = \prod_{j=0}^{q-2} (g_2^{x^j})^{d_{i,j}}.$$

Let PS denote the set

$$\left\{ \left(h_j + h_0, u_2^{1/(x+h_j)} \right) \right\}_{j=1}^{q-1}.$$

Algorithm B sets

$$t' = \prod_{i=1}^{q-1} (g_2^{x^{i-1}})^{c_i} = g_2^{(f(x) - c_0)/x}$$

and sets

$$\gamma_0 = \hat{e}(\psi(t'), u_2 \cdot g_2^{c_0}).$$

It defines $u_1 = \psi(u_2)$ and computes the public key of the TA as

$$R = \psi(u_2' \cdot u_2^{-h_0}) = \psi(u_2') \cdot u_1^{-h_0} = u_1^{x-h_0}.$$

We need to check that this has the correct distribution. Since we are conditioning on the event $\neg\mathtt{Guess}$ we know that u_2 is a generator of $\mathbb{G}_2$ which means that u_1 must be a generator of $\mathbb{G}_1$ as required for the scheme.

Consider the following distributions associated with a generator u_1 of $\mathbb{G}_1$. Note that in the description below $\mathcal{D}_x$ is one of a collection of distributions $\{\mathcal{D}_x\}_{x\in\mathbb{Z}_p}$ parameterised by $x \in \mathbb{Z}_p$.

$$\mathcal{D} = \{u_1^s \ : \ s \leftarrow \mathbb{Z}_p\} \text{ and } \mathcal{D}_x = \left\{u_1^{x-h_0} \ : \ h_0 \leftarrow \mathbb{Z}_p\right\}.$$

Clearly, for any $x \in \mathbb{Z}_p$, these distributions are identical and, moreover, R is chosen from $\mathcal{D}$ when the scheme is used in reality and R is chosen from $\mathcal{D}_x$ in our simulation (conditioned on the event $\neg\mathtt{Guess}$). We conclude that R has the correct distribution.

Algorithm B now invokes the first stage of algorithm A with the domain parameters that it has constructed. It responds to the oracle calls made by A as follows.

H_1-query on $\mathtt{ID}_i$: B maintains a list H_1 of tuples $(\mathtt{ID}_i, h_i, D_{\mathtt{ID}_i})$ indexed by $\mathtt{ID}_i$. On input of $\mathtt{ID}_i$, the ith distinct query, algorithm B responds as follows.

(1) If $i = I$ then B responds with h_0 and adds $(\mathtt{ID}_i, h_0, \perp)$ to the list H_1.
(2) Otherwise it selects a random element $(h_i+h_0, u_2^{1/(x+h_i)})$ from PS (without replacement). It adds $(\mathtt{ID}_i, h_i + h_0, u_2^{1/(x+h_i)})$ to the list H_1 and it returns $h_i + h_0$.

If the query is a repeat query then B responds with the response that it gave the first time by looking it up on the list.

H_2-query on α : B maintains a list H_2 of tuples (α, β). If α appears in the list H_2 then B responds with β. Otherwise it chooses β at random from $\{0,1\}^n$ and it adds (α, β) to the H_2 list before responding with β.

Extraction Query on $\mathtt{ID}_i$: If $\mathtt{ID}_i$ does not appear on the H_1 list then B first makes an H_1 query. Algorithm B then checks whether the corresponding value of $D_{\mathtt{ID}_i}$ is $\perp$. If so it terminates. (Note that this event corresponds to B failing to correctly guess at what point A queries H_1 with its chosen $\mathtt{ID}^*$.) Otherwise it responds with $D_{\mathtt{ID}_i}$ where $(\mathtt{ID}_i, h_i, D_{\mathtt{ID}_i})$ is the entry corresponding to $\mathtt{ID}_i$ in the H_1 list.

At some point A's first stage will terminate and it will return a challenge identity $\mathtt{ID}^*$. If A has not called H_1 with input $\mathtt{ID}^*$ then B does so for it. If the corresponding value of $D_{\mathtt{ID}^*}$ is not equal to $\perp$ then B will terminate.

Algorithm B chooses a random value of $r \in \mathbb{Z}_p$ and a random value V^* in $\{0,1\}^n$. It computes $U^* = u_1^r$ and sets the challenge ciphertext to be

$$c^* = (U^*, V^*).$$

This challenge ciphertext is now passed to algorithm A's second stage. Note, due to the rules of the game, B will not terminate unexpectedly when responding to extraction queries made once A has been given the challenge ciphertext.

At some point algorithm A responds with its guess as to the value of the underlying plaintext m^*. For a genuine challenge ciphertext we should have

$$m^* = V^* \oplus H_2(\hat{e}(U^*, D_{\mathtt{ID}^*})).$$

If H_2 is modelled as a random oracle we know that A only has any advantage if the list H_2 contains an input value

$$\alpha^* = \hat{e}(U^*, D_{ID^*}). \tag{3}$$

Algorithm B selects a value α at random from the list H_2. We assume that it correctly selects $\alpha = \alpha^*$ and add a factor $1/q_2$ to our subsequent analysis. It sets

$$\gamma = \alpha^{*1/r}.$$

We have that

$$D_{\mathtt{ID}^*} = u_2^{1/((x-h_0)+h_0)}$$

and so

$$\gamma = \hat{e}(u_1, u_2)^{1/x}.$$

Algorithm B's job is to compute $\hat{e}(g_1, g_2)^{1/x}$. It sets

$$\begin{aligned}\gamma/\gamma_0 &= \hat{e}(g_1, g_2)^{f(x)\cdot f(x)/x} / \hat{e}(g_1^{(f(x)-c_0)/x}, g_2^{f(x)+c_0}) \\ &= \hat{e}(g_1, g_2)^{f(x)\cdot f(x)/x - f(x)\cdot f(x)/x + c_0 2/x} \\ &= \hat{e}(g_1, g_2)^{c_0 2/x}\end{aligned}$$

and it solves the q-BDHI problem by outputting

$$\hat{e}(g_1, g_2)^{1/x} = (\gamma/\gamma_0)^{1/c_0 2}.$$

Note that the above procedure for calculating the solution can fail if (1) $r = 0$ or (2) $c_0 = 0$. However, this will not happen if $h_i \neq 0$ for $i = 0, \dots, q-1$ and $r \neq 0$. We say that the event `Fail` occurs if at least one of these conditions fails. We have

$$\begin{aligned}\Pr[A \text{ wins}|\neg\mathtt{Guess}] &= \Pr[A \text{ wins} \wedge \neg\mathtt{Fail}|\neg\mathtt{Guess}] + \Pr[A \text{ wins} \wedge \mathtt{Fail}|\neg\mathtt{Guess}] \\ &\leq \Pr[A \text{ wins} \wedge |\neg\mathtt{Guess} \wedge \neg\mathtt{Fail}] + \frac{q+1}{p} \end{aligned} \tag{4}$$

Let us denote the event that A makes the query α^*, as defined in (3), during its attack by `Ask`.

$$\begin{aligned}&\Pr[A \text{ wins}|\neg\mathtt{Guess} \wedge \neg\mathtt{Fail}] \\ &\quad = \Pr[A \text{ wins} \wedge \mathtt{Ask}|\neg\mathtt{Guess} \wedge \neg\mathtt{Fail}] + \Pr[A \text{ wins} \wedge \neg\mathtt{Ask}|\neg\mathtt{Guess} \wedge \neg\mathtt{Fail}] \\ &\leq \Pr[A \text{ wins} \wedge \mathtt{Ask}|\neg\mathtt{Guess} \wedge \neg\mathtt{Fail}] + \frac{1}{2^n}. \end{aligned} \tag{5}$$

The last inequality follows from the fact that, in the random oracle model, if the event `Ask` does not occur, then A has no information about the message encrypted in the challenge ciphertext.

To conclude the proof we note that, in event `Ask`, provided B (1) picks the correct index I, which happens with probability $1/(q_1 + q_X + 1)$, and (2) chooses the correct entry α^* from list H_2, which happens with probability $1/q_2$, then B succeeds in solving the q-BDHI problem. This means that

$$\Pr[A \text{ wins} \wedge \mathtt{Ask}|\neg\mathtt{Guess}] \leq ((q_1 + q_X + 1) \cdot q_2) \cdot \mathrm{Adv}^{q-\mathtt{BDHI}}(B). \tag{6}$$

The result now follows from (2), (4), (5) and (6).

Hewlett-Packard Laboratories, Filton Road, Stoke Gifford, Bristol, BS34 8QZ, UK
E-mail address: `liqun.chen@hp.com`

School of Computing Science, Middlesex University, White Hart Lane, London, N17 8HR, UK
E-mail address: `m.z.cheng@mdx.ac.uk`

Department of Computer Science, University of Bristol, Merchant Venturers Building, Woodland Road, Bristol, BS8 1UB, UK
E-mail address: `malone@cs.bris.ac.uk`

Department of Computer Science, University of Bristol, Merchant Venturers Building, Woodland Road, Bristol, BS8 1UB, UK
E-mail address: `nigel@cs.bris.ac.uk`