Direct Chosen-Ciphertext Secure Identity-Based Encryption in the Standard Model with short Ciphertexts

Eike Kiltz

CWI Amsterdam
kiltz@cwi.nl

Abstract

We describe a practical identity-based encryption scheme that is secure in the standard model against chosen-ciphertext (IND-CCA2) attacks. Security is based on an assumption comparable to (but slightly stronger than) Bilinear Decisonal Diffie-Hellman (BDDH). A comparison shows that our construction outperforms all known identity-based encryption schemes in the standard model and its performance is even comparable with the one from the random-oracle based Boneh/Franklin IBE scheme. Our proposed IBE scheme has furthermore the property that it fulfills some notion of "redundancy-freeness", i.e. the encryption algorithm is not only a probabilistic injection but also a surjection. As a consequence the ciphertext overhead is nearly optimal: to encrypt k bit messages for k bit identities and with k bit randomness we get $3k$ bit ciphertexts to guarantee (roughly) k bits of security.

Keywords: Chosen-ciphertext security, Identity-Based Encryption, Bilinear Maps.

1 Introduction

Identity-Based Encryption. An Identity-Based Encryption (IBE) scheme is a public-key (asymmetric) encryption scheme where any string such as email addresses, server names or phone numbers, can be used as public keys. The ability to use identities as public keys largely reduces the need for public key certificates and certificate authorities to distribute public key certificates.

After Shamir proposed the concept of IBE in 1984 [42] it remained an open problem for almost two decades to come up with a satisfying construction for it. In 2001, Boneh and Franklin [10] proposed formal security notions for IBE systems and designed a fully functional secure IBE scheme using bilinear maps. This scheme and the tools developed in its design have been successfully applied in numerous cryptographic settings, transcending by far the identity based cryptography framework. Though relatively recent invented IBE is already intensively applied in practice (see, e.g., `http://www.voltage.com`). Furtheremore, IBE is currently in the process of getting standardized — from February 2006 on the new IEEE P1363.3 standard for "Identity-Based Cryptographic Techniques using Pairings" [28] accepts submissions.

An alternative but less efficient IBE construction was proposed by Cocks [19] based on quadratic residues. Both IBE schemes provide security against *chosen-ciphertext attacks* (through Fujisaki-Okamoto [22]). In a chosen ciphertext attack, the adversary is given access to a decryption oracle that allows him to obtain the decryptions of ciphertexts of his choosing. Intuitively, security in this setting means that an adversary obtains (effectively) no information about encrypted messages, provided the corresponding ciphertexts are never submitted to the decryption oracle. For different reasons, the notion of chosen-ciphertext security has emerged as the "right" notion of security for encryption schemes. We stress that, in general, chosen-ciphertext security is a much stronger security requirement than chosen-plaintext attacks [3], where in the latter an attacker is not given access to the decryption oracle.

The drawback of the IBE scheme from Boneh-Franklin and Cocks is that security can only be guaranteed in the *random oracle* model [4], i.e. in an idealized world where all parties magically get black-box access to a truly random function. Unfortunately a proof in the random oracle model can only serve as a heuristic argument and has proved to possibly lead to insecure schemes when the random oracles are implemented in the standard model (see, e.g., [13]).

WATERS' IBE. To fill this gap Waters [45] presents the first efficient Identity-Based Encryption scheme that is chosen-plaintext secure without random oracles. The proof of his scheme makes use of an algebraic method first used by Boneh and Boyen [7] and security of the scheme is based on the Bilinear Decisional Diffie-Hellman (BDDH) assumption. However, Waters' plain IBE scheme only guarantees security against passive adversaries (chosen-plaintext security).

FROM 2-LEVEL HIERARCHICAL IBE TO CHOSEN-CHIPERTEXT SECURE IBE. Hierarchical identity-based encryption (HIBE) [27, 24] is a generalization of IBE allowing for hierarchical delegation of decryption keys. Recent results from Canetti, Halevi, and Katz [14], further improved upon by Boneh and Katz [11] show a generic and practical transformation from any chosen-plaintext secure 2-level HIBE scheme to a chosen-ciphertext secure IBE scheme. Since Waters' IBE scheme can naturally be extended to a 2-level HIBE this implies the first efficient chosen-ciphertext secure IBE in the standard model. Key size, as well as the security reduction of the resulting scheme are comparable to the ones from Waters' IBE. However, the transformation involves some symmetric overhead to the ciphertext in form of a one-time signature or a MAC with their respective keys.

The first "direct" (non 2-level HIBE based) chosen-ciphertext IBE construction in the standard model was mentioned by Boyen, Mei, and Waters [12] and later improved by Galindo and Kiltz [29]. Both constructions are based on Waters' IBE and add one additional element to the ciphertext that is used for a consistency check in the decryption algorithm. However, in terms of ciphertext size and performance it did not introduce a dramatic improvement over the generic 2-level HIBE based constructions.

IDENTITY-BASED KEY ENCAPSULATION. Instead of providing the full functionality of an IBE scheme, in many applications it is sufficient to let sender and receiver agree on a common random session key. This can be accomplished with an *identity-based key encapsulation mechanism* (IB-KEM) as formalized in [21, 6]. Any IB-KEM can be updated to a full IBE scheme by adding a symmetric encryption scheme. The latter one is also called a data encapsulation scheme (DEM) and the resulting identity-based encryption scheme the resulting hybrid IBE scheme. If both the IB-KEM and the DEM are chosen-ciphertext secure, then the hybrid IBE scheme is also chosen-ciphertext secure. We note that chosen-ciphertext secure DEMs can be created from relatively weak primitives such as a one-time symmetric encryption scheme (e.g., a one-time pad) plus a message authentication code (MAC). In the public-key setting most standards are given in terms of KEM primitives and we find it very likely that the upcoming IEEE P1363.3 standard [28] will also follow this principle. We therefore decided to focus in this paper on IB-KEM's only.

1.1 Our Contributions

A NEW CHOSEN-CIPHERTEXT SECURE IB-KEM/IBE SCHEME. Based on Waters' chosen-plaintext secure IBE scheme we present a new and direct identity-based key encapsulation mechanism with short ciphertexts and very efficient encapsulation/decapsulation algorithms. Chosen-ciphertext security is obtained at sheer optimal cost. Compared to Waters' raw chosen-plaintext secure IBE scheme (viewed as an IB-KEM) our scheme comes with the same ciphertext overhead whereas computational overhead is one more exponentiation for encapsulation and two more exponentiations for decapsulation. We give a rigorous game-based proof reducing chosen-ciphertext security of our scheme to breaking the *modified Bilinear Decisional Diffie-Hellman assumption* (mBDDH), an assumption closely related to BDDH. By adding a one-time secure symmetric encryption scheme and a MAC we obtain a new hybrid IBE scheme with short ciphertexts using the IB-KEM/DEM methodology [6].

An identity-preserving redundancy-free IBE scheme in the standard model. It is furthermore possible to obtain a full IBE scheme with shorter ciphertexts by using the DEMs based super stranding permutations [38] that avoid the usual overhead due to the MAC. Then ciphertexts of our IBE come with minimal overhead, i.e they are *identity-preserving redundancy-free.* Following Phan and Pointcheval [37] this property means that the IBE encryption algorithm (viewed as a mapping from randomness space, identity space, and message space into the ciphertext space) is a bijection. Consequently all possible ciphertexts in the ciphertext space are reachable by the encryption algorithm — shrinking the ciphertext any further is not possible. Our construction is the first identity-preserving redundancy-free IBE scheme in the standard model. The existence of redundancy DEMs (super pseudorandom permutations) is quite strong and the resulting schemes are computationally very inefficient. To this end we propose a direct construction of a identity-preserving redundancy-free IBE scheme only based on the mBDDH assumption.

A (stronger) notion of redundancy-free IBE schemes further requires that even *for any possible identity* from the identity-space the encryption algorithm (now viewed as a mapping from randomness space and message space into the ciphertext space) is a bijection. Obtaining such strongly redundancy-free IBE schemes is possible but they are only known to exist in the random oracle model and under the highly non-standard "gap-BDDH" assumption [31].

We find even the existence of identity-preserving redundancy-free IBE schemes in the standard model particularly remarkable since in the standard public-key encryption setting redundancy-free schemes (in the sense of [37]) are not known to exist. We further remark that the ciphertexts of our IBE scheme have the same message expansion as the most efficient standard public-key encryption schemes (like Kurosawa/Desmedt [30] and BMW [12]), i.e. compared to standard PKE we obtain identity-based encryption with no overhead.

Extensions. Furthermore, we present a couple of extensions of our IB-KEM including a chosen-ciphertext secure hierarchical identity-based KEM with short ciphertexts.

The mBDDH assumption and its relation to known assumptions. As a by-product we formalize and study our new mBDDH assumption and relate its hardness to well-known pairing-based "standard assumptions". In particular we show that "2-BDDHI is at least as strong as mBDDH is at least as strong as BDDH". The 2-BDDHI (2 Biliear Decisional Diffie-Hellman Inversion) assumption was introduced by Boneh and Boyen [7] and its stronger variants (q-BDDHI for some polynomial q) already found numerous applications in [7, 8, 9, 35].

1.2 Related Work and Comparison

In [12] it was shown how to use "identity-based techniques" from [14] to obtain direct chosen-ciphertext secure public-key encryption schemes. The techniques from [12] basically rely on combining [14] with a trick already appearing in a paper by Cramer and Shoup [20] to use a (target collision resistant) hash function to "tie" some elements in the ciphertext together. As we already pointed out, chosen-ciphertext secure IBE schemes were known to exist using generic reductions [14] based on Waters' 2-level HIBE [45]. The first direct chosen-ciphertext secure IBE scheme was mentioned in [12]. Improving on the results from [12] the first concrete full construction with a formal security proof was provided in [29]. The latter scheme can be seen as combining the 2-level HIBE scheme obtained from Waters' IBE at the first level and Boneh-Boyen [7] at the second level with the "direct chosen-ciphertext secure techniques" from [12] to obtain a direct chosen-ciphertext secure IBE scheme. Compared to Waters' chosen-plaintext secure IBE scheme, the latter direct construction adds one additional redundant element to the ciphertext. Like in the construction from [12] this element is used as a "validity check" to defend against invalid ciphertexts, where the check had to be carried out using bilinear pairings. A similar validity check is implicitly contained in the generic constructions based on 2-level HIBEs [14].

The main idea of our new scheme is to encode the information necessary for the validity check into Waters' original ciphertext. More precisely, we were able to encode the consistency information in ciphertext element containing the receiver's identity. This more efficient encoding also enables us to perform a more efficient decryption. In a broader view our new scheme can also be seen as combining the 2-level HIBE scheme obtained using the construction from Boneh-Boyen-Goh [9] with Waters' IBE at the first level and Boneh-Boyen [7] at the second level, with some variant of the techniques from [12] to obtain a direct chosen-ciphertext secure IBE scheme. However, we want to stress that it is not obvious if the Boneh-Boyen-Go [9] HIBE can be instantiated with Waters' technique to get a fully secure HIBE scheme, similar to the one described above. Nor if the technique of [20, 12] can be applied to the latter construction to obtain a direct chosen-ciphertext secure IBE scheme. In some sense our results answer the two above questions to the positive. However, we consider our specific scheme and its proof of security as novel contributions that are not self-evident given the state of our knowledge in this area. In this context we want to repeat again that unlike the construction given in [12] our direct chosen-ciphertext technique does not expand the ciphertext by one element. Unfortunately it does not seem to be applicable to the original public-key setting to obtain shorter ciphertexts in [12].

A comparison with chosen-ciphertext secure IBE schemes in the standard model. We will (in Section 6) carefully review all known chosen-ciphertext secure IBE constructions, including the above proposals, and make an extensive comparison with our scheme. In terms of ciphertext expansion our IBE scheme saves (at least) one group element compared to all so far known constructions, which makes a relative saving of 33% (i.e., two instead of three elements). The relative savings for encryption/decryption are (at least) one exponentiation and one pairing plus one exponentiation, respectively which again sums up to a relative saving of (roughly) 33%. We conclude that, to the best of our knowledge, the proposed IBE scheme is the most efficient chosen-ciphertext secure IBE scheme in the standard model.

A comparison with the Boneh/Franklin random oracle IBE scheme. Using recent experimental data for atomic primitives (such as exponentiations and pairings) from Granger, Page, and Smart [26] we estimate the efficiency of a possible implementation of our scheme using asymmetric pairings over non-singular elliptic curves. We make a careful comparison at various practical security levels with the only IBE scheme that is currently employed in practice: the IBE scheme from Boneh and Franklin [10], which is only known to be secure in the random oracle model. In turns out that the efficiency of our scheme is comparable to the one from Boneh and Franklin — ciphertext expansion is more or less the same and encryption is a factor of 3 to 10 faster (depending on the chosen security parameter), whereas decryption is about 1.5 to 3 times slower. We conclude that our scheme has ciphertext size and efficiency comparable to the random oracle based Boneh/Franklin IBE scheme.

Recent Results. Very recently (and independent from our work) Gentry presents another practical and direct chosen-ciphertext secure IBE scheme without random oracles [23], based on different techniques. We remark that our scheme is still more efficient than Gentry's IBE scheme, in particular our ciphertexts are shorter by one group element. Furthermore, the security of Gentry's scheme is based on a much stronger and new assumption (related to the q-BDDHI assumption).

2 Definitions

2.1 Notation

If x is a string, then $|x|$ denotes its length, while if S is a set then $|S|$ denotes its size. If $k \in \mathbb{N}$ then 1^k denotes the string of k ones. If S is a set then $s \xleftarrow{\$} S$ denotes the operation of picking an element s of S uniformly at random. We write $\mathcal{A}(x, y, \ldots)$ to indicate that $\mathcal{A}$ is an algorithm with inputs $x, y, \ldots$ and by $z \xleftarrow{\$} \mathcal{A}(x, y, \ldots)$ we denote the operation of running $\mathcal{A}$ with inputs $(x, y, \ldots)$ and letting z be the output. We write $\mathcal{A}^{\mathcal{O}_1, \mathcal{O}_2, \cdots}(x, y, \ldots)$ to indicate that $\mathcal{A}$ is an algorithm with inputs $x, y, \ldots$ and access to oracles $\mathcal{O}_1, \mathcal{O}_2, \ldots$ and by $z \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_1, \mathcal{O}_2, \cdots}(x, y, \ldots)$ we denote the operation of running $\mathcal{A}$ with inputs $(x, y, \ldots)$ and access to oracles $\mathcal{O}_1, \mathcal{O}_2, \ldots$, and letting z be the output.

2.2 Secure Identity Based Key Encapsulation

An *identity-based key-encapsulation mechanism* (IB-KEM) scheme [42, 10] $\mathcal{IBKEM} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encaps}, \mathsf{Decaps})$ consists of four polynomial-time algorithms. Via $(pk, msk) \xleftarrow{\$} \mathsf{Setup}(1^k)$ the randomized key-generation algorithm produces master keys for security parameter $k \in \mathbb{N}$; via $usk[id] \xleftarrow{\$} \mathsf{Extract}(msk, id)$ the master computes the secret key for identity id; via $(C, K) \xleftarrow{\$} \mathsf{Encaps}(pk, id)$ a sender creates a random session key K and a corresponding ciphertext C with respect to identity id; via $K \leftarrow \mathsf{Decaps}(usk, C)$ the possessor of secret key usk decapsulates ciphertext C to get back a session key K. Associated to the scheme is a key space KeySp. For consistency, we require that for all $k \in \mathbb{N}$, all identities id, and all $(C, K) \xleftarrow{\$} \mathsf{Encaps}(pk, id)$, we have $\Pr[\mathsf{Decaps}(\mathsf{Extract}(msk, id), C) = K] = 1$, where the probability is taken over the choice of $(pk, msk) \xleftarrow{\$} \mathsf{Setup}(1^k)$, and the coins of all the algorithms in the expression above.

The strongest and commonly accepted notion of security for an indentity-based key encapsulation scheme is that of *indistinguishability against an adaptive chosen ciphertext attack*. This notion, denoted IND-CCA, is defined using the following game between a challenger and an adversary $\mathcal{A}$. Let $\mathcal{IBKEM} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encaps}, \mathsf{Decaps})$ be an IB-KEM with associated key space KeySp. To an adversary $\mathcal{A}$ we associate the following experiment:

Experiment $\mathbf{Exp}^{\text{ind-cca}}_{\mathcal{IBKEM},\mathcal{A}}(k)$

$(pk, msk) \xleftarrow{\$} \mathsf{Setup}(1^k)$
$(id^*, st) \xleftarrow{\$} \mathcal{A}^{\textsc{Extract}(\cdot), \textsc{Decaps}(\cdot,\cdot)}(\texttt{find}, pk)$
$K_0^* \xleftarrow{\$} \mathsf{KeySp}\ ;\ (C^*, K_1^*) \xleftarrow{\$} \mathsf{Encaps}(pk, id^*)$
$\gamma \xleftarrow{\$} \{0,1\}\ ;\ K^* \leftarrow K_\gamma^*$
$\gamma' \xleftarrow{\$} \mathcal{A}^{\textsc{Extract}(\cdot), \textsc{Decaps}(\cdot,\cdot)}(\texttt{guess}, K^*, C^*, st)$
If $\gamma \neq \gamma'$ then return 0 else return 1

The oracle $\textsc{Extract}(id)$ returns $sk[id] \xleftarrow{\$} \textsc{Extract}(sk, id)$ with the restriction that $\mathcal{A}$ is not allowed to query oracle $\textsc{Extract}(\cdot)$ for the target identity id^*. The oracle $\textsc{Decaps}(id, C)$ first computes $sk[id] \xleftarrow{\$} \textsc{Extract}(sk, id)$ and then returns $K \leftarrow \mathsf{Decaps}(sk[id], id, C)$ with the restriction that in the guess stage $\mathcal{A}$ is not allowed to query oracle $\textsc{Decaps}(\cdot, \cdot)$ for the tuple (id^*, C^*). Here (and in contrast to the weaker original security definition [10] [1]) the output of $\textsc{Extract}(id)$ is stored internally by the experiment and multiple queries to $\textsc{Decaps}(id, \cdot)$ are answered with respect to the *same* user secret key $sk[id]$. The variable st represents some internal state information of adversary $\mathcal{A}$ and can be any

[1] We note that the IBE schemes presented in [10] have deterministic key derivation and hence remain secure under this stronger definition

(polynomially bounded) string. We define the advantage of $\mathcal{A}$ in the chosen-ciphertext experiment as

$$\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{IBKEM},\mathcal{A}}(k) \;=\; \left|\Pr\left[\,\mathbf{Exp}^{\text{ind-cca}}_{\mathcal{IBKEM},\mathcal{A}}(k)=1\,\right]-\frac{1}{2}\right| \;.$$

An IB-KEM $\mathcal{IBKEM}$ is said to be *secure against chosen-ciphertext attacks* (CCA secure) if the advantage functions $\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{IBKEM},\mathcal{A}}(k)$ is a negligible function in k for all polynomial-time adversaries $\mathcal{A}$.

We stress that storing $sk[id]$ when anwering decapsulation queries is an important detail when modeling the idea of a chosen-ciphertext attack for IB-KEMs. In practise [43], decapsulation queries for some identity will be answered by one single user holding one fixed user secret key. We also mention that by using the original definition from [10], i.e. relaxing this to answering decapsulation queries using a freshly generated user secret key, one may obtain more efficient schemes since now decapsulation may take benefit of the randomness used in key derivation. In fact, it is easy to verify that our IB-KEM without *(implicit) consistency check* already satisfies this weaker variant of CCA security.

We remark that our security definition is given with respect to "full-identity" attacks, as opposed to the much weaker variant of "selective-identity" attacks where the adversary has to commit to its target identity id^* in advance, even before seeing the public key.

3 Assumptions

3.1 Parameter generation algorithms for Bilinear Groups.

All pairing based schemes will be parameterized by a *pairing parameter generator*. This is a PTA $\mathcal{G}$ that on input 1^k returns the description of an multiplicative cyclic group $\mathbb{G}$ of prime order p, where $2^k < p < 2^{k+1}$, the description of a multiplicative cyclic group $\mathbb{G}_T$ of the same order, and a non-degenerate bilinear pairing $\hat{e}$: $\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$. See [10] for a description of the properties of such pairings. We use $\mathbb{G}^*$ to denote $\mathbb{G}\setminus\{1\}$, i.e. the set of all group elements except the neutral element. Throughout the paper we use $\mathcal{PG}=(\mathbb{G},\mathbb{G}_T,p,\hat{e},g)$ as shorthand for the description of bilinear groups, where g is a generator of $\mathbb{G}$.

3.2 The modified BDDH assumption

Let $\mathcal{PG}$ be the description of pairing groups. Consider the following problem: Given $(g,g^a,g^b,g^{(b^2)},g^c,W)\in\mathbb{G}^5\times\mathbb{G}_T$ as input, output yes if $W=\hat{e}(g,g)^{abc}$ and no otherwise. The mBDDH assumption states that, roughly, this problem is computational infeasible. Note that this is nearly the standard BDDH assumption (see Appendix D for a formal definition) with the only difference that with mBDDH a distinguisher is additionally provided with the element $g^{(b^2)}$ (which is hard to compute from g^b).

More formally, to a parameter generation algorithm for pairing-groups $\mathcal{G}$ and an adversary $\mathcal{B}$ we assotiate the following experiment.

Experiment $\mathbf{Exp}^{\text{mbddh}}_{\mathcal{G},\mathcal{B}}(1^k)$
$\mathcal{PG}\xleftarrow{\$}\mathcal{G}(1^k)$
$a,b,c,w\xleftarrow{\$}\mathbb{Z}_p^*$
$\beta\xleftarrow{\$}\{0,1\}$. If $\beta=1$ then $W\leftarrow\hat{e}(g,g)^{abc}$ else $W\leftarrow\hat{e}(g,g)^{w}$
$\beta'\xleftarrow{\$}\mathcal{B}(1^k,\mathcal{PG},g,g^a,g^b,g^{b^2},g^c,W)$
If $\beta\neq\beta'$ then return 0 else return 1

We define the advantage of $\mathcal{B}$ in the above experiment as

$$\mathbf{Adv}^{\text{mbddh}}_{\mathcal{G},\mathcal{B}}(k) \;=\; \left|\Pr\left[\,\mathbf{Exp}^{\text{mbddh}}_{\mathcal{G},\mathcal{B}}(1^k)=1\,\right]-\frac{1}{2}\right| \;.$$

We say that the *modified Bilinear Decision Diffie-Hellman (mBDDH) assumption relative to generator* $\mathcal{G}$ holds if $\mathbf{Adv}^{\mathrm{mbddh}}_{\mathcal{G},\mathcal{B}}$ is a negligible function in k for all PTAs $\mathcal{B}$.

3.3 Relation to BDDH and q-BDDHI

The next lemma classifies the strength of the modified BDDH assumption we introduced between the well known *standard pairing-based assumptions* BDDH and 2-BDDHI (see Appendix D for definitions). Here "A $\leq$ B" means that assumption B implies assumption A (in a black-box sense), i.e. assumption B is a stronger assumption than A.

Lemma 3.1 BDDH $\leq$ mBDDH $\leq$ 2-BDDHI $\leq$ 3-BDDHI $\leq \ldots$

The simple proof is postponed until Appendix D.3. Since 2-BDDHI is known to hold in the generic-group model [8] this in particular implies correctness of the mBDDH assumption in generic-groups.

4 A chosen-ciphertext secure IB-KEM based on mBDDH

In this section we present our new chosen-ciphertext secure IB-KEM. Let $\mathcal{PG} = (\mathbb{G}, \mathbb{G}_T, p, \hat{e}, g)$ be public system parameters obtained by running the group parameter algorithm $\mathcal{G}(1^k)$.

4.1 Waters' Hash

We review the hash function $\mathsf{H} : \{0,1\}^n \to \mathbb{G}$ used in Waters' identity based encryption schemes [45]. On input of $\mathbb{G}$ and an integer n, the randomized hash key generator $\mathsf{HGen}(\mathbb{G}; n)$ chooses $n+1$ random group elements $h_0, \ldots, h_n \in \mathbb{G}$ and returns $h = (h_0, h_1, \ldots, h_n) \in \mathbb{G}^{n+1}$ as the public description of the hash function. The hash function $\mathsf{H} : \{0,1\}^n \to \mathbb{G}^*$ is evaluated on a string $id = (id_1, \ldots, id_n) \in \{0,1\}^n$ as the product

$$\mathsf{H}(id) = h_0 \prod_{i=1}^{n} h_i^{id_i} \in \mathbb{G}.$$

In Appendix E.1 we remind the reader of Water's original chosen-plaintext secure IBE scheme.

4.2 The IB-KEM Construction

Let $\mathsf{TCR} : \mathbb{G} \to \mathbb{Z}_p$ be a target collision-resistant hash function (i.e. given $t = \mathsf{TCR}(a)$ for a random $a \in \mathbb{G}$ it should be hard to find $b \in \mathbb{G} \setminus \{a\}$ such that $\mathsf{TCR}(b) = a$; a formal definition can be looked up in Appendix B). Our IB-KEM with identity space $\mathsf{IDSp} = \{0,1\}^n$ ($n = n(k)$) and key space $\mathsf{KeySp} = \mathbb{G}_T$ is depicted in Figure 1.

We call a (possibly malformed) ciphertext $C = (c_1, c_2) \in \mathbb{G}^2$ *consistent* (w.r.t identity id and public key pk) if $(g, c_1, \mathsf{H}(id) \cdot u^t, c_2)$ is a Diffie-Hellman tuple[2], where $t = \mathsf{TCR}(c_1)$. A correctly generated ciphertext for identity id has the form $C = (c_1, c_2) = (g^r, (\mathsf{H}(id) \cdot u^t)^r)$ and therefore $(g, c_1, \mathsf{H}(id) \cdot u^t, c_2) = (g, c_1, \mathsf{H}(id) \cdot u^t, (\mathsf{H}(id) \cdot u^t)^r)$ is always a DH tuple and consequently C is consistent. Testing for a DH-tuple is equivalent to checking if $\hat{e}(g, c_2) = \hat{e}(\mathsf{H}(id) \cdot u^t, c_1)$ and therefore consistency of C can be implemented by evaluating the bilinear map twice. Note that this consistency test can be performed by anybody knowing the public-key only. This property is called "public verification" of the ciphertext.

[2]A tuple $(g, g^a, g^b, g^c) \in \mathbb{G}^4$ is said to be a *Diffie-Hellman tuple* (DH tuple) if $ab = c \bmod p$.

$\mathsf{Setup}(1^k)$	$\mathsf{Extract}(sk, id)$
$\quad \alpha, u \xleftarrow{\$} \mathbb{G}^*$; $z \leftarrow \hat{e}(g, \alpha)$; $\mathsf{H} \xleftarrow{\$} \mathsf{HGen}(\mathbb{G}; n)$	$\quad s \xleftarrow{\$} \mathbb{Z}_p$
$\quad pk \leftarrow (\mathsf{H}, u, z) \in \mathbb{G}^{n+1} \times \mathbb{G} \times \mathbb{G}_T$	$\quad sk[id] \leftarrow (\alpha \cdot \mathsf{H}(id)^s, g^s, u^s) \in \mathbb{G}^3$
$\quad sk \leftarrow \alpha \in \mathbb{G}$	$\quad$ Return $sk[id]$
$\quad$ Return (pk, sk)	
$\mathsf{Encaps}(pk, id)$	$\mathsf{Decaps}(pk, id, sk[id], C)$
$\quad r \xleftarrow{\$} \mathbb{Z}_p^*$	$\quad$ Parse C as (c_1, c_2)
$\quad c_1 \leftarrow g^r$; $t \leftarrow \mathsf{TCR}(c_1)$	$\quad$ Parse $sk[id]$ as (d_1, d_2, d_3)
$\quad c_2 \leftarrow (\mathsf{H}(id) \cdot u^t)^r$	$\quad t \leftarrow \mathsf{TCR}(c_1)$
$\quad K \leftarrow z^r \in \mathbb{G}_T$	$\quad v \xleftarrow{\$} \mathbb{Z}_p^*$
$\quad C \leftarrow (c_1, c_2) \in \mathbb{G}^2$	$\quad$ Return $K \leftarrow \dfrac{\hat{e}(d_1 \cdot d_3^t \cdot (\mathsf{H}(id) \cdot u^t)^v, c_1)}{\hat{e}(g^v \cdot d_2, c_2)}$
$\quad$ Return (C, K)	

Figure 1: Our chosen-ciphertext secure identity-based key encapsulation.

4.3 Alternative Decapsulation

We now describe an alternative deterministic decapsulation algorithm which is more intuitive but less efficient. We claim that the decapsulation algorithm from Figure 1 is equivalent to

1. Compute $t = \mathsf{TCR}(c_1)$ and check if $(g, c_1, \mathsf{H}(id) \cdot u^t, c_2)$ is a DH tuple. If not, a random session key K is returned (or the ciphertext gets rejected).
2. Otherwise return $K \leftarrow \hat{e}(c_1, d_1 \cdot d_3^t)/\hat{e}(c_2, d_2)$

To prove this claim we define the function $\Delta(C) = \hat{e}(c_1, \mathsf{H}(id)u^t)/\hat{e}(g, c_2)$. Then $\Delta(C) = 1$ if and only if C is consistent. Consequently, for a random $v \in \mathbb{Z}_p^*$, $K = \hat{e}(d_1 \cdot d_3^t, c_1)/\hat{e}(d_2, c_2) \cdot (\Delta(C))^v \in \mathbb{G}_T^*$ evaluates to $\hat{e}(d_1 \cdot d_3^t, c_1)/\hat{e}(d_2, c_2)$ if C is consistent and to a random group element otherwise. The claim then follows by

$$\begin{aligned} K &= \hat{e}(c_1, d_1 \cdot d_3^t)/\hat{e}(c_2, d_2) \cdot (\Delta(C))^v \\ &= \hat{e}(c_1, d_1 \cdot d_3^t)/\hat{e}(c_2, d_2) \cdot (\hat{e}(c_1, \mathsf{H}(id)u^t)/\hat{e}(g, c_2))^v \\ &= \frac{\hat{e}(c_1, d_1 \cdot d_3^t \cdot (\mathsf{H}(id)u^t)^v)}{\hat{e}(c_2, g^v \cdot d_2)} \,. \end{aligned}$$

We remark that the original decapsulation algorithm roughly saves two pairing operations.

We now show correctness of the scheme, i.e. that the K computed in the encapsulation algorithm matches the key K computed in the alternative decapsulation algorithm. We already showed that a correctly generated ciphertext is always consistent. A correctly generated secret key for identity id has the form $sk[id] = (d_1, d_2, d_3) = (\alpha \cdot \mathsf{H}(id)^s, g^s, u^s)$. Therefore the key decryption algorithm computes the key K as

$$\begin{aligned} K &= \hat{e}(c_1, d_1 \cdot d_3^t)/\hat{e}(c_2, d_2) \\ &= \hat{e}(g^r, \alpha\mathsf{H}(id)^s \cdot (u^s)^t)/\hat{e}((\mathsf{H}(id) \cdot u^t)^r, g^s) \\ &= \hat{e}(g^r, \alpha) \cdot \hat{e}(g^r, \mathsf{H}(id)^s \cdot (u^s)^t)/\hat{e}((\mathsf{H}(id) \cdot u^t)^r, g^s) \\ &= z^r \cdot \hat{e}(g^r, (\mathsf{H}(id) \cdot u^t)^s)/\hat{e}((\mathsf{H}(id) \cdot u^t)^s, g^r) \\ &= z^r, \end{aligned}$$

as the key computed in the encryption algorithm. This shows correctness.

4.4 Security

Theorem 4.1 Assume TCR is a target collision resistant hash function. Under the modified Bilinear Decisional Diffie-Hellman (mBDDH) assumption relative to generator $\mathcal{G}$, the IB-KEM from Section 4.2 is secure against chosen-ciphertext attacks.

In particular, given an adversary $\mathcal{A}$ attacking the chosen-ciphertext security of the IB-KEM with advantage $\varepsilon_{\mathcal{A}} = \mathbf{Adv}^{\text{ind-cca}}_{\mathcal{IBKEM},\mathcal{A}}$ and running time $\mathbf{Time}_{\mathcal{A}}(k)$ we construct an adversary $\mathcal{B}$ breaking the mBDDH assumption with advantage $\varepsilon_{\mathcal{B}} = \mathbf{Adv}^{\text{mbddh}}_{\mathcal{G},\mathcal{B}}(k)$ and running time $\mathbf{Time}_{\mathcal{B}}(k)$ with

$$\begin{aligned}\varepsilon_{\mathcal{B}}(k) &\geq \frac{\varepsilon_{\mathcal{A}}(k) - \mathbf{Adv}^{\text{hash-tcr}}_{\mathsf{TCR},\mathcal{H}}(k)}{8(n+1)q} - q/p;\\ \mathbf{Time}_{\mathcal{B}}(k) &\leq \mathbf{Time}_{\mathcal{A}} + \tilde{\mathcal{O}}(nq\cdot\varepsilon_{\mathcal{A}}^{-2}(k)),\end{aligned}$$

where q is an upper bound on the number of key derivation/decryption queries made by adversary $\mathcal{A}$.

The proof of Theorem 4.1 will be given in Appendix C. It uses ideas from Waters [45] and will and is given in a game-based version from [29].

5 Extensions

5.1 (Redundancy-free) Identity-Based Encryption

In this section we present various (known) extensions of our IBE construction, some of them are critical for its application. Given an IB-KEM and a symmetric encryption scheme, a hybrid identity-based encryption scheme can be obtained by using the IB-KEM to securely transport a random session key that is fed into the symmetric encryption scheme (also called data encapsulation mechanism — DEM) to encrypt the plaintext message. It was recently shown in [6] that if both the IB-KEM and the DEM are chosen-ciphertext secure, then the resulting hybrid encryption is also chosen-ciphertext secure. The security reduction is tight.

A DEM secure against chosen-ciphertext attacks can be built from relatively weak primitives, i.e. from any one-time symmetric encryption scheme by essentially adding a MAC. For concreteness we mention that a chosen-ciphertext secure IBE scheme can be built from our IB-KEM construction with an additional overhead of a DEM which consists of a (one-time secure) symmetric encryption plus additional 128 bits for the MAC. Furthermore, Phan and Pointcheval [38] showed that *super pseudo-randomn permutations* directly imply redundancy-free chosen-ciphertext secure DEMs that avoid the usual overhead due to the MAC.

At an abstract level, for each identity *id* from identity space IDSp, an IBE encryption algorithm IBEenc_{id} can be viewed as a mapping

$$\mathsf{IBEenc}_{id} : \mathsf{RandSp} \times \mathsf{MsgSp} \rightarrow \mathsf{CipherSp},$$

where RandSp is the randomness space, MsgSp is the message space, and $\mathsf{CipherSp}$ is the ciphertext space. That also implies that decrypting a fixed ciphertext with respect to different identities must consequently lead to distinct plaintexts. By our security definition we need a sufficiently large randomness space since otherwise the IBE scheme is not even indistinguishable against chosen-plaintext attacks [25]. Following Phan and Pointcheval [37] we say that an IBE scheme is *redundancy-free* if for any possible identity *id* the above encryption mapping IBEenc_{id} is a bijection, i.e. if all elements from the ciphertext space are "reachable". This redundancy-free property means that in some sense the ciphertexts are minimal and can't be further shrunk.

Our scheme only satisfies redundancy-freeness with respect to a different (weaker) notion, it is *identity-preserving redundancy-free* in the sense that the mapping

$$\mathsf{IBEenc} : \mathsf{RandSp} \times \mathsf{IDSp} \times \mathsf{MsgSp} \to \mathsf{CipherSp}$$

is a bijection, i.e. information about the identity id is absorbed by the ciphertext $\mathsf{CipherSp}$. This relaxation is useful if an IBE ciphertext is needed to be verifiable, i.e. if one can (publicly) verify if an IBE ciphertext was indeed encrypted with some given identity. Applications of this property can be found, e.g., in threshold IBE schemes [29]. It is easy to argue that IBE schemes that are identity-preserving redundancy-free are optimal among all schemes that are (publicly) verifiable (this is since the identity has to be somehow encoded in the ciphertext).

Using the IB-KEM/DEM paradigm with our IB-KEM constriction and one of the DEMs based on super pseudorandom permutations [38] we get an identity-preserving redundancy-free chosen-ciphertext secure IBE scheme that is publicly verifiable. However, the existence of super pseudorandom permutations is a very strong assumptions and their construction is computationally inefficient. Section 5.2 gives an alternative identity-preserving redundancy-free chosen-ciphertext secure IBE scheme without any additional assumption.

As a consequence the ciphertext overhead of our IBE scheme is optimal with respect to the verifiability property. Suppose an adversary attacking the IBE scheme makes at most q decryption/key derivation queries. A common estimate used here is $q = 2^{30}$ (suggested by Bellare and Rogaway [5]). According to Theorem 4.1, to encrypt k bit messages for $n = k$ bit identities and with k bit randomness we get $3k$ bit ciphertexts to guarantee $\approx k - 30$ bits of security. The $30 = \log(q)$ bits of loss in the security stems from the fact that the security reduction in Theorem 4.1 is not optimal (and depends multiplicatively on q). We remark that the ciphertext size of our IBE scheme is about the same as the one of the most efficient (standard) public-key encryption schemes in the standard model [30, 12].

We mention that there exists a redundancy-free IBE scheme [31] (in the sense of the first definition) in the random oracle model but its security proof depends on a highly non-standard assumption.[3]

5.2 Direct chosen-ciphertext secure IBE scheme

The IB-KEM from Section 4.2 combined with a chosen-ciphertext secure DEM yields a chosen-ciphertext secure IBE scheme. We now show a direct construction of a chosen-ciphertext secure IBE scheme that is identity-preserving redundancy free only based on the mBDDH assumption.

The idea is to modify the IB-KEM scheme from Section 4.2 to work with a one-time secure DEM (instead of a chosen-ciphertext secure DEM). To this end the DEM ciphertext e is also included in the TCR hash function. However, this bring some difficulties in the security proof since now the value c_1^* from the challenge ciphertext is not independent of the challenge message anymore (as needed in the proof of Theorem 4.1). Similar to [12] we use a different consitency check based on an independent instantiation of Water's H' hash to account for this. Therefore, compared to the IBE scheme from Section 4.2, the resulting IBE scheme has the disadvantage of having twice as large public-keys and having a greater loss in the security reduction.

Let $\mathsf{CR} : \mathbb{G}_T \to \{0,1\}^n$ be a collision-resistant hash function. Our construction of a direct IBE scheme with identity space $\mathsf{IDSp} = \{0,1\}^n$ $(n = n(k))$ and message space $\mathsf{MsgSp} = \mathbb{G}_T$ is depicted in Figure 2. Here the multiplicative one-time pad may be replace with any (one-time secure) symmetric encryption scheme. We note that the two elements h_0 and h_0' can be merged into one single element.

[3] The scheme in [31] is secure under the "gap-BDDH assumption" which is same as the BDDH assumption but it additionally assumes the existence of an efficient DDH algorithm in the target group $\mathbb{G}_T$ which is not known to exist.

$\mathsf{Setup}(1^k)$	$\mathsf{Extract}(sk, id)$
$\alpha, u \xleftarrow{\$} \mathbb{G}^*$; $z \leftarrow \hat{e}(g, \alpha)$	$s \xleftarrow{\$} \mathbb{Z}_p$
$\mathsf{H} \xleftarrow{\$} \mathsf{HGen}(\mathbb{G}; n)$; $\mathsf{H}' \xleftarrow{\$} \mathsf{HGen}(\mathbb{G}; n)$	$sk[id] \leftarrow (\alpha \cdot \mathsf{H}(id)^s, g^s, h_0'^s, \ldots, h_n'^s) \in \mathbb{G}^{n+3}$
$pk \leftarrow (\mathsf{H}, \mathsf{H}', z) \in \mathbb{G}^{n+m+2} \times \mathbb{G}_T$; $sk \leftarrow \alpha \in \mathbb{G}$	Return $sk[id]$
Return (pk, sk)	
$\mathsf{Encrypt}(pk, id, M)$	$\mathsf{Decrypt}(pk, id, sk[id], C)$
$r \xleftarrow{\$} \mathbb{Z}_p^*$	Parse C as (c_1, c_2, e)
$c_1 \leftarrow g^r$;	Parse $sk[id]$ as $(d_1, d_2, e_0, \ldots, e_n)$
$K \leftarrow z^r$; $e \leftarrow M \cdot K$	$t \leftarrow \mathsf{CR}(e)$; $v \xleftarrow{\$} \mathbb{Z}_p^*$
$t \leftarrow \mathsf{CR}(e)$; $c_2 \leftarrow (\mathsf{H}(id) \cdot \mathsf{H}'(t))^r$	$K \leftarrow \dfrac{\hat{e}(d_1 \cdot (\mathsf{H}(id) \prod_{i=0}^{n} e_i^{t_i})^v, c_1)}{\hat{e}(g^v \cdot d_2, c_2)}$
$C \leftarrow (c_1, c_2, e) \in \mathbb{G}^2 \times \mathbb{G}_T$	
Return C	Return $M = e/K$.

Figure 2: Direct chosen-ciphertext secure identity-based encryption.

Theorem 5.1 Assume CR is a collision resistant hash function. Under the mBDDH assumption relative to generator $\mathcal{G}$, the IBE scheme from Section 5.2 is secure against chosen-ciphertext attacks.

In particular, given an adversary $\mathcal{A}$ attacking the chosen-ciphertext security of the IBE scheme with advantage $\varepsilon_{\mathcal{A}} = \mathbf{Adv}^{\text{ind-cca}}_{\mathcal{IBE},\mathcal{A}}$ and running time $\mathbf{Time}_{\mathcal{A}}(k)$ we construct an adversary $\mathcal{B}$ breaking the mBDDH assumption with advantage $\varepsilon_{\mathcal{B}} = \mathbf{Adv}^{\text{mbddh}}_{\mathcal{G},\mathcal{B}}(k)$ and running time $\mathbf{Time}_{\mathcal{B}}(k)$ with

$$\begin{aligned} \varepsilon_{\mathcal{B}}(k) &\geq \frac{\varepsilon_{\mathcal{A}}(k) - \mathbf{Adv}^{\text{hash-cr}}_{\mathsf{CR},\mathcal{H}}(k)}{(8(n+1)q)^2} - q/p; \\ \mathbf{Time}_{\mathcal{B}}(k) &\leq \mathbf{Time}_{\mathcal{A}} + \tilde{\mathcal{O}}(n^2 q^2 \cdot \varepsilon_{\mathcal{A}}^{-2}(k)), \end{aligned}$$

where q is an upper bound on the number of key derivation/decryption queries made by adversary $\mathcal{A}$.

The proof of this theorem is similar to the one of Theorem 4.1 and will be given in the full version of this paper. The quadratic security loss comes from the fact that we use two independent instances of Water's hash function. Furthermore it is possible to trade this quadratic loss for a MAC tag or an aditional element from $\mathbb{Z}_p^*$.

5.3 A Tradeoff between Public Key Size and Security Reduction and Arbitrary identities

As independently discovered in [15, 33], there exists an interesting trade-off between key-size of Waters' hash H and the security reduction of the IBE scheme.

The construction modifies Waters hash H as follows: Let the integer $l = l(k)$ be a new parameter of the scheme. In particular, we represent an identity $id \in \{0,1\}^n$ as an n/l-dimensional vector $id = (id_1, \ldots, id_{n/l})$, where each id_i is an l bit string. Waters hash is then redefined to $\mathsf{H} : \{0,1\}^n \to \mathbb{G}$, with $\mathsf{H}(id) = h_0 \prod_{i=1}^{n/l} h_i^{id_i}$ for random public elements $h_0, h_1, \ldots, h_{n/l} \in \mathbb{G}$. Waters' original hash function is obtained as the special case $l = 1$. It is easy to see that using this modification in our IBE scheme (i) reduces the size of the public key from $n+3$ to $n/l+3$ group elements, whereas (ii) it adds another multiplicative factor of 2^l to the security reduction of the IBE scheme (Theorem 4.1).[4]

Furthermore we want to remark that using a simple and well-known trick we can allow the identity-space to contain arbitrary bitstrings by applying a collision resistant hash function $\mathsf{CR} : \{0,1\}^* \to \{0,1\}^n$ to the identities before applying Waters' hash.

For concreteness and for a scheme implemented in groups offering ≈ 80 bits of security we have $n = 160$ bits and therefore propose to use $l = 16$ or $l = 32$. This shrinks the public-key size to reasonable 10 or 5 group elements, respectively.

5.4 Chosen-ciphertext secure Hierarchical Identity-Based key encapsulation

Hierarchical identity-based encryption is a generalization of IBE to identities supporting hierarchical structures [27, 24]. By the relation to Waters IBE scheme it is easy to see that our technique can also be used to obtain a chosen-ciphertext secure HIBE. Using a technique from [9] it is furthermore possible to reduce the HIBE ciphertext size to three elements, i.e. it is independent of the hierarchy's depth. To be more precise, the IB-KEM from Section 4.2 is modified to an HIB-KEM supporting maximal d hierarchies as follows. The setup algorithm chooses d different and independent hash functions $H_i \stackrel{\$}{\leftarrow} \mathsf{HGen}(\mathbb{G}; n)$, for $1 \leq i \leq d$. The user secret key for the hierarchical identity $\vec{id} = (\vec{id}^{(1)}, \ldots, \vec{id}^{(\mu)})$ of depth $\mu \leq d$ is defined as $sk[\vec{id}] = (d_1, d_2, d_3, (d_{ij})_{\mu+i \leq j \leq d, 0 \leq i \leq n}) \in \mathbb{G}^{3+(n+1)\cdot(d-\mu-1)}$, where $d_1 = \alpha \cdot (\prod_{j=1}^{\mu} H_i(\vec{id}^{(j)}))^r$, $d_2 = g^r$, $d_3 = u^r$, and $d_{ij} = ((h_i^{(j)})^r)$. We remark that the latter $(n+1)\cdot(d-\mu-1)$ elements d_{ij} are only needed for hierarchical key delegation (and may be not included in $sk[\vec{id}]$ if such a feature is not wanted). Encapsulation with respect to $\vec{id}$ computes the two ciphertext elements $c_1 = g^r$ and $c_2 = (u^t \prod_{j=1}^{\mu} H_i(\vec{id}^{(j)}))^r$ and the key $K = z^r$. Decapsulation reconstructs K from (c_1, c_2) as $K \leftarrow \frac{\hat{e}(d_1 \cdot d_3^t \cdot (u^t \prod_{j=1}^{\mu} \mathsf{H}_j(\vec{id}_h))^v, c_1)}{\hat{e}(g^v \cdot d_2, c_2)}$ for a random v. Note that this only needs two pairing operations, independent of the depth of the hierarchy d. (In contrast the HIB-KEM from [29] needs $d+1$ pairings.)

The keysize of the HIB-KEM is roughly nd, whereas the same tradeoff between public-key size and security reduction mentioned in the last subsection is possible to achieve public-key sizes of nd/l. Security can be proved with respect to the *d-modified BDDH assumption*, where compared to the mBDDH assumption the adversary gets the values $g^b, g^{b^2}, \ldots, g^{b^{d+1}}$ (instead of just g^b, g^{b^2}). As in [24, 45] the security reduction is exponential in the depth d of the hierarchy, i.e. it introduces, roughly, a multiplicative factor of $(nq)^d$.

5.5 Selective-Identity Chosen-Ciphertext Secure IB-KEM

For the definition of a selective-identity chosen-ciphertext secure IB-KEM we change the security experiment such that the adversary has to commit to the target identity id^* before seeing the public key. Clearly, this is a weaker security requirement. We quickly note that (using an algebraic technique from [7]) by replacing Waters' hash H with $\mathsf{H}(id) = h_0 \cdot h_1^{id}$ (for $id \in \mathbb{Z}_p$) we get a selective-id chosen-ciphertext secure IB-KEM. Note that the size of the public-key of this scheme drops to 3 elements.

5.6 Implementing the Target Collision Resistant Hash Function TCR

In practice, to build a target collision resistant hash function, one can use a dedicated cryptographic hash function, like SHA-1 [41]. Every injective function $\mathsf{TCR} : \mathbb{G} \to \mathbb{Z}_p$ trivially also is (target) collision resistant (with zero advantage). Boyen, Mei and Waters [12] note that for bilinear maps defined on

[4] On the technical side our proof basically stays the same, only the bound from Lemma C.2 needs to be adapted to take the modified Waters' hash into account.

Scheme	Size		Encrypt	Decrypt	Key Der.
	$\lvert C \rvert$ (overhead)	pk	#pairings + #[multi,reg]-exp		
Ours+DEM	$2\lvert\mathbb{G}\rvert$	$n+3$	$0+[1,2]$	$2+[1,1]$	$0+[0,3]$
Kiltz/Galindo+DEM	$3\lvert\mathbb{G}\rvert$	$n+4$	$0+[1,3]$	$3+[1,3]$	$0+[0,2]$
Waters/BB+CHK	$3\lvert\mathbb{G}\rvert+704$	$n+4$	$0+[1,3]$	$3+[1,2]$	$0+[0,2]$
Gentry+DEM	$1\lvert\mathbb{G}\rvert+2\lvert\mathbb{G}_T\rvert$	4	$0+[2,2]$	$2+[0,3]$	$0+[1,0]$
(Waters)	$2\lvert\mathbb{G}\rvert$	$n+2$	$0+[0,3]$	$2+[0,0]$	$0+[0,2]$
(Boneh/Franklin)	$1\lvert\mathbb{G}\rvert+256$	1	$1+[0,2]$	$1+[0,1]$	$0+[0,1]$
(SK-KEM+DEM)	$1\lvert\mathbb{G}\rvert+256$	1	$0+[0,2]$	$1+[0,2]$	$0+[0,1]$

Table 1: Efficiency comparison for chosen-ciphertext secure IBE schemes. Ciphertext overhead represents the difference (in bits) between the ciphertext length and the message length. The additional bits account for the necessary symmetric overhead for 128 bits security. The keysize of the public key is measured in terms of the number of group elements. The size of the secret key sk is the same for all three schemes (a single element in $\mathbb{G}$). For computational efficiency we neglect all symmetric operations (like symmetric encryption, random oracle hashes, and MACs). For comparison we mention that relative timings for the various operations are as follows: regular pairing $\approx 3-5$ [36], multi-exponentiation ≈ 1.5, regular exponentiation $= 1$.

elliptic curves there exists a very efficient way to implement such injective mappings. We refer to [12] for more details.

6 Comparison

In this section we compare our scheme with the known IBE schemes from the literature. For a uniform treatment we do all comparisons in terms of the respective IBE schemes. The previously most efficient CCA-secure IBE scheme is the one from Kiltz and Galindo [29]. We also compare our scheme with the generic construction [14] obtained from a 2-level HIBE [45, 7] and with the original (only chosen-plaintext secure) IBE scheme from Waters. Furthermore we compare or scheme with the reference random-oracle IBE scheme from Boneh and Franklin [10].

In pairing based cryptography efficiency depends on the chosen curve and how well the scheme can be adapted to it. Usually [14, 12] a comparison is done by taking the pairing as a black-box and under the simplified assumption that all exponentiations carried out in different groups have about the same running time. We will follow this approach in Section 6.1. Then, in Section 6.2 we will discuss how our scheme can possibly be instantiated in non-supersingular asymmetric pairing groups. A carefull implementation-based comparison in the asymmetric setting with the Boneh/Franklin IBE scheme will then be done in Section 6.3.

There is an independent comparison of the efficiency aspects for IBE schemes [1] that mostly support our conclusions except that they do not take the tightness of the security reduction into account.

6.1 Comparison in the symmetric setting

We will consider the following IBE schemes:

Ours+DEM: Our construction from Section 4 updated with a (redundancy-free) DEM to get a full IBE scheme.

Kiltz/Galindo+DEM: The IB-KEM from [29] updated with a DEM to get a full IBE scheme.

Scheme	CCA?	Standard Model?	Assumption	Security Reduction Normal	Security Reduction Normalized
Ours+DEM	$\surd$	$\surd$	mBDDH	nq	$2nq \approx 2^{48}$
Kiltz/Galindo+DEM	$\surd$	$\surd$	BDDH	nq	$nq \approx 2^{47}$
Waters/BB+CHK	$\surd$	$\surd$	BDDH	nq	$nq \approx 2^{47}$
Gentry	$\surd$	$\surd$	q-ABDHE	1	$\sqrt{q} \approx 2^{20}$
(Waters)	—	$\surd$	BDDH	nq	$nq \approx 2^{47}$
(Boneh/Franklin)	$\surd$	—	BDDH	q_H	$q_H \approx 2^{80}$
(SK-KEM)	$\surd$	—	q_H-BDDHI	q_H^3	$q_H^{3.5} \approx 2^{280}$

Table 2: Security assumptions and reduction factors for IBE schemes. For 128 bits security we estimate the number of key derivation queries as $q \approx 2^{40}$, and the number of hash queries as $q_H \approx 2^{80}$ (following [5]). The bitsize of the identity space is $n = 2^7$. Normalized reduction refers to the estimated reduction factor with respect to the BDDH assumption.

Hybrid Waters/BB+CHK: The IBE scheme obtained by the generic transformation [14, 11] applied to the 2-level hybrid HIBE consisting of Waters' IBE scheme [45] at the first level and the Boneh/Boyen IBE scheme [7] at the second level (as proposed in [45]).

Gentry+DEM: The recent IBE scheme from Gentry [23] (in it's more efficient IB-KEM/DEM variant).

Waters: Waters' plain chosen-plaintext secure IBE scheme [45].

Boneh/Franklin: The random-oracle "fullident" chosen-ciphertext secure IBE scheme [10].

SK-KEM+DEM: Then random-oracle IB-KEM from Sakai-Kasahara [40, 17, 16, 1].

Since evaluating Waters' hash H requires computing $n/2$ products in $\mathbb{G}$ on the average, where $n \leq \log_2 p$, it can be seen as a single exponentiation. Therefore we count computing $\mathsf{H}(id)^r$ for random r as two exponentiations in $\mathbb{G}$. For decryption the value $\mathsf{H}(id)$ can be precomputed (and assumed to be contained in $sk[id]$).

Comparison. An efficiency comparison is done in Table 1. We conclude that our scheme is the most efficient chosen-ciphertext secure IBE scheme in the standard model. Furthermore its performance and ciphertext expansion seems comparable to the random-oracle based reference scheme from Boneh/Franklin.

Security Assumptions and Tightness of Security Reductions. Concerning tightness of the security reductions we make the (quite conservative) assumption that the family of ℓ-BDDHI assumptions (including ℓ-ABDHE) is by a factor of $\sqrt{\ell}$-times less secure than the family of BDDH assumptions (including mBDDH). This is supported by the results in the generic-group mode. The best known generic attack on BDDH has running time $\sqrt{p}$ which matches the generic lower bound. On the other hand, the best known generic attack on ℓ-BDDHI has running time roughly $\sqrt{p/\ell}$ [18] that again matches the generic lower bound. Security assumptions and reduction factors are depicted in Figure 2. For simplicity we may assume that all considered schemes (except the SK-KEM) have more or less the same security reduction with respect to the BDDH assumption.

6.2 Our IBE scheme in asymmetric pairing groups

Our definition of the bilinear groups assumed a symmetric pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. However, there is a large class of admissible bilinear groups which have an asymmetric pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, i.e.

Variant	Element ... in group					key decapsulation	Encryption	Decryption
	c_1	c_2	d_1	d_2	d_3		#pairings + #exp in $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$	
V1	$\mathbb{G}_2$	$\mathbb{G}_2$	$\mathbb{G}_1$	$\mathbb{G}_1$	$\mathbb{G}_1$	$\frac{\hat{e}(d_1 d_3^t (\mathsf{H}(id)u^t)^v, c_1)}{\hat{e}(g^v d_2, c_2)}$	$0 + (0, 3.5, 1)$	$1.5 + (2.5, 0, 0)$
V2	$\mathbb{G}_1$	$\mathbb{G}_1$	$\mathbb{G}_2$	$\mathbb{G}_2$	$\mathbb{G}_2$	$\frac{\hat{e}(c_1, d_1 d_3^t (\mathsf{H}(id)u^t)^v)}{\hat{e}(c_2, g^v d_2)}$	$0 + (3.5, 0, 1)$	$1.5 + (0, 2.5, 0)$
V3	$\mathbb{G}_2$	$\mathbb{G}_1$	$\mathbb{G}_1$	$\mathbb{G}_2$	$\mathbb{G}_1$	$\frac{\hat{e}(d_1 d_3^t (\mathsf{H}(id)u^t)^v, c_1)}{\hat{e}(c_2 g^v, d_2)}$	$0 + (2.5, 1, 1)$	$1.5 + (2.5, 0, 0)$

Table 3: Different asymmetric variants of our IBE scheme.

Variant	Ciphertext space	Ciphertext size	Encryption	Decryption
V1	$\mathbb{G}_2 \times \mathbb{G}_2$	big	slow	fast
V2	$\mathbb{G}_1 \times \mathbb{G}_1$	small	fast	slow
V3	$\mathbb{G}_2 \times \mathbb{G}_1$	big	medium	fast

Table 4: Tradeoff between ciphertext size and efficiency for our IBE variants.

$\mathbb{G}_1 \neq \mathbb{G}_2$. Such asymmetric bilinear groups have the advantage of being less special than symmetric ones — and consequently have better security properties since their greater generality makes it harder to design tailor-made attacks. Furthermore, as we will sketch below, they can lead to considerably shorter ciphertexts than symmetric pairings.

In this setting we have to allocate the various group elements appearing in our IBE scheme to the two groups $\mathbb{G}_1$ and $\mathbb{G}_2$. Depending on how this is done we can give different trade-offs between computational efficiency for encryption/decryption and ciphertext size. To this end we will use the following conventions [26, 2]: (i) For general curves an element in $\mathbb{G}_2$ takes about $\alpha/2$ times as much space to represent as one in $\mathbb{G}_1$, where α (usually called k) is the embedding degree (typical values for α are $\alpha = 6, 12, 24$). We note that for some "special" curves (with $D = -3$) one can reduce this to $\alpha/6$.
Representing an element in $\mathbb{G}_T$ in general takes as least as much space as one in $\mathbb{G}_2$. However, in practice that means that elements in $\mathbb{G}_1$ have a small representation whereas elements in $\mathbb{G}_2$ and $\mathbb{G}_T$ not. (ii) An exponentiation in $\mathbb{G}_2$ takes about α as much time as an exponentiation in $\mathbb{G}_1$. We adapt the convention to count one multi-exponentiation as 1.5 exponentiations [14] and the ratio of two pairings as 1.5 pairings [12].[5] Based on those assumptions in Table 3 we give three variants of our IBE scheme with different tradeoffs between ciphertext size and encryption/decryption efficiency. The relative advantages are summarized in Table 4. We note that in case of asymmetric pairing groups the public key pk consists of $\mathbb{G}_c^{n+1} \times \mathbb{G}_T$, where $c = 2$ for variant 1 and $c = 1$ for variants 2 and 3 (i.e. the elements h_i and u have to be in the same group as the ciphertext element c_2). Therefore for variants 2 and 3 we can take benefit of the small representation in group G_1. We remark that some further care should be taken when instantiating pairing-based schemes in the asymmetric setting in a black-box way since due to the different premises the proof of security may not longer be valid. Indeed it is easy to verify that in our case the proofs are still valid for the three proposed variants.

6.3 A comparison in the asymmetric setting

In this section we demonstrate the practicability of our IBE scheme by comparing it with the one from Boneh/Franklin. We remark that the latter scheme is intensively used in practice (see, e.g., `http://www.voltage.com`). We aim to compare the schemes for fixed security parameters $k =$

[5]Actually [12] mentions in Section 5.1 that "computation of a ratio of two pairings [...] can be done almost as efficiently as a single pairing, by modifying Miller's algorithm in a manner akin to multi-exponentiation [32]". We think that a factor of 1.5 is more realistic.

k	Curve ($\log_2 p, \alpha$)	Our IBE (Variant 2)			Our IBE (Variant 3)			Boneh/Franklin			Boneh/Franklin2		
		Enc	Dec	$\|C\|$	Enc	Dec	$\|C\|$	Enc	Dec	$\|C\|$	Enc	Dec	$\|C\|$
80	A (160,6)	4	20	320	8	10	640	16	6	320	11	10	640
128	B (512,6)	60	325	1024	115	187	2048	458	115	768	196	170	1792
128	C (256,12)	18	220	512	64	103	1792	314	66	512	118	113	1792
192	D (1365,6)	611	3687	2730	1170	2286	5460	7970	1431	1749	2472	1991	4479
192	E (683,12)	174	2405	1366	632	1259	4781	5472	815	1067	1350	1273	4482
256	F (2560,6)	2808	19074	5120	5386	12629	10240	51256	7989	3072	13613	10567	8192
256	G (1280,12)	788	11920	2560	2877	6697	8960	34950	4355	1792	6904	6445	8192
256	H (640,24)	262	7627	1280	1602	4276	8320	21418	2822	1152	4289	4163	8192

Table 5: Number of estimated 32 bit multiplications needed to perform Encryption/Decryption (scaled by 10^5) and ciphertext overhead in bits. The column $|C|$ gives the ciphertext overhead in bits.

$80, 128, 192, 256$. We denote the size of the message space by m. We further comment on the schemes by Gentry and the SK-KEM.

Boneh/Franklin. We consider the fullident chosen-ciphertext secure Boneh/Franklin IBE scheme (which for completeness can be looked up in Appendix E.2). For encryption it performs one exponentiation in $\mathbb{G}_1$, one exponentiation in $\mathbb{G}_T$, one pairing, and one call a "hash-to-point" hash function $H_1 : \{0,1\}^n \to \mathbb{G}_2^*$, modeled as a random oracle. The latter one was already identified in [36, 17, 26] to be problematic to implement since one some curves it is not known to be efficiently implementable at all or it needs one "cofactor" exponentiation in $\mathbb{G}_2$. For decryption it needs one exponentiation in $\mathbb{G}_1$ and one pairing. The ciphertext space is $\mathbb{G}_1 \times \{0,1\}^{2k} \times \{0,1\}^m$, the $\{0,1\}^{2k}$ stems from the output of a hash function H_2 (due to the birthday attack a domain of $2k$ is needed to guarantee security of k bits).

Boneh/Franklin2. We denote by Boneh/Franklin2 the above scheme with switched roles of $\mathbb{G}_1$ and $\mathbb{G}_2$. In variant the expensive "hash-to-point" hash function maps into the group $\mathbb{G}_1$ but on the other hand we all exponentiations have to be carried out in $\mathbb{G}_2$ and furthermore the ciphertext lies in $\mathbb{G}_2$.

Our Scheme. We consider variants two and three of our IBE scheme from Table 3. Since we consider full IBE schemes ciphertexts consist of the IB-KEM ciphertext plus a summetric one-time encryption and a MAC. More precisely, the ciphertext space of our IBE scheme is $\mathbb{G}_a \times \mathbb{G}_b \times \{0,1\}^k \times \{0,1\}^m$, where the $\{0,1\}^k$ stands for the tag of the MAC (k bits are sufficient to guarantee security of k bits).

We estimate the cost of encryption and decryption using the timings for each atomic primitive (exponentiations/hashes in $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ and pairings) calculated in [26], where we used the timings for the "pairing friendly curves" and the Tate pairing. Here we counted one hash-into-curve operation used in the Boneh/Franklin scheme (the random oracle H_2) as one co-factor exponentiation [26]. For completeness all used timing data for the atomic primitives is given in Table 6 of Appendix A. The comparison is done with respect to the different curves A-H considered in [26], the names correspond the ones given therein. Important parameters for the curves are the estimated bits of security they provide, the embedding degree $\alpha = 6, 12, 24$, and the field size of the underlying finite field. We assume that one element in $\mathbb{G}_1$ can be represented using $\log_2 p$ bits. We furthermore assume that one element in $\mathbb{G}_2$ needs a factor of α as much space to represent as one element in $\mathbb{G}_1$, i.e. $\alpha \log_2 p$ bits.

Comparison. The extensive comparison matrix for the different curves A-H is given in Table 5. We conclude that efficiency of our scheme is comparable to the one from Boneh and Franklin — ciphertext sizes of our scheme are more or less the same and encryption is a factor of 3 to 10 faster (depending on the chosen security parameter), whereas decryption is about 1.5 to 3 times slower (depending on the variant). One disadvantage of our scheme seems to be the relatively large public-key ($n+3$ group elements for n bit identities). We stress that with the techniques from Section 5.3 the public-key size can easily be shrunk to n/l group elements with losing only l bits of security.

$\mathsf{Setup}(1^k)$

$u, \alpha \xleftarrow{\$} \mathbb{G}_1^* \; ; \; z \leftarrow \hat{e}(\alpha, g_2) \in \mathbb{G}_T$

$\mathsf{H} \xleftarrow{\$} \mathsf{HGen}(\mathbb{G}_1) \; // \mathsf{H} = (h_0, \ldots, h_{10}) \in \mathbb{G}_1^{11}$

$pk \leftarrow (\mathsf{H}, u, z) \in \mathbb{G}_1^{12} \times \mathbb{G}_T \; ; \; sk \leftarrow \alpha \in \mathbb{G}_1$

Return (pk, sk)

$\mathsf{Extract}(sk, id)$

$s \xleftarrow{\$} \mathbb{Z}_p$

$sk[id] \leftarrow (\alpha \cdot \mathsf{H}(id)^s, g_2^s, u^s) \in \mathbb{G}_1 \times \mathbb{G}_2 \times \mathbb{G}_1$

Return $sk[id]$

$\mathsf{Encaps}(pk, id)$

$r \xleftarrow{\$} \mathbb{Z}_p^*$

$c_1 \leftarrow g_2^r \in \mathbb{G}_2 \; ; \; t \leftarrow \mathsf{TCR}(c_1)$

$c_2 \leftarrow (\mathsf{H}(id) \cdot u^t)^r \in \mathbb{G}_1$

$K \leftarrow z^r \in \mathbb{G}_T$

$C \leftarrow (c_1, c_2) \in \mathbb{G}_1^2$

Return (C, K)

$\mathsf{Decaps}(pk, id, sk[id], C)$

Parse C as (c_1, c_2)

Parse $sk[id]$ as (d_1, d_2, d_3)

$t \leftarrow \mathsf{TCR}(c_1)$

$v \xleftarrow{\$} \mathbb{Z}_p^*$

Return $K \leftarrow \dfrac{\hat{e}(d_1 \cdot d_3^t \cdot (\mathsf{H}(id)u^t)^v, c_1)}{\hat{e}(c_2, g_2^v \cdot d_2)}$

Figure 3: A concrete instantiation of Variant 3 on curves with 80 bits security and recommended $l = 16$. We use an asymmetric pairing where $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ are chosen according to the parameters of curve A from [26].

The IBE scheme from Gentry [23]. We want to remark that the recent IBE scheme from Gentry [23] becomes quite impractical when implemented in the asymmetric setting since its ciphertexts contain at least two elements from $\mathbb{G}_T$ (for concreteness [1] predicts 9472 bits ciphertexts for 128 bits of security).

The IBE scheme from Sakai and Kasahara [40]. The encryption speed is roughly 1.5 times faster than ours (and therefore 6 to 30 times faster than the one from Boneh/Franklin), and decryption speed is 2 to 3 times as fast (and therefore comparable to the one from Boneh/Franklin). Therefore it outperforms our IBE construction as well as the Boneh/Franklin IBE scheme. As discussed before, the drawback is the strong security assumption (q_H-BDDHI assumption) and the non-tight reduction involving a factor of $\approx 2^{280}$ in the security reduction to BDDH.

6.4 A concrete instantiation of Variant 3

We conclude out paper by presenting details of a concrete instantiation of Variant 3 from Table 3. Let $\mathsf{TCR} : \mathbb{G} \rightarrow \mathbb{Z}_p$ be a target collision-resistant hash function, and let $\mathsf{CR} : \{0,1\}^* \rightarrow \{0,1\}^{80}$. All hash functions may be implemented in practice using a suitable variant of SHA-1. Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ be an asymmetric pairing where $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ are chosen according to the parameters of curve A in [26], and let g_2 be a fixed generator of G_2. Waters's hash function is defined as $\mathsf{H}(id) = h_0 \prod_{i=1}^{10} h_i^{id'_i} \in \mathbb{G}_1$, where $id' = (id'_1, \ldots, id'_{10}) \in (\{0,1\}^{16})^{10}$ and $id' = \mathsf{CR}(id)$ is the output of the collision resistant hash function CR (c.f. Section 5.3). All the above information is considered as public system parameters. Our IB-KEM with identity space $\mathsf{IDSp} = \{0,1\}^*$ and key space $\mathsf{KeySp} = \mathbb{Z}_p$ is depicted in Figure 3.

Acknowledgements

We thank Eugene Vahlis for pointing out the limitations of the original IB-KEM security definition from [10] used in an earlier version of this paper and Nigel Smart for answering various questions on bilinear maps. We further thank Alex Dent, Javier Herranz, Dennis Hofheinz, and John Malone-Lee for their comments and suggestions.

The author was supported in part by the research program Sentinels (`http://www.sentinels.nl`). Sentinels is being financed by Technology Foundation STW, the Netherlands Organization for Scientific

Research (NWO), and the Dutch Ministry of Economic Affairs.

References

[1] M. Barbosa, L. Chen, Z. Cheng, M. Chimley, A. Dent, P. Farshim, K. Harrison, J. Malone-Lee, N. P. Smart, and F. Vercauteren. SK-KEM : AN IDENTITY-BASED KEM. Submission to IEEE P1363.3, August 2006. Avaiable from `http://grouper.ieee.org/groups/1363/IBC/index.html`.

[2] P. S. L. M. Barreto and M. Naehrig. Pairing-friendly elliptic curves of prime order. In *Selected Areas in Cryptography – SAC'2005*, pages 319–331, 2006.

[3] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In H. Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 26–45. Springer-Verlag, Aug. 1998.

[4] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS 93*, pages 62–73. ACM Press, Nov. 1993.

[5] M. Bellare and P. Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In U. M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*. Springer-Verlag, May 1996.

[6] K. Bentahar, P. Farshim, J. Malone-Lee, and N. Smart. Generic constructions of identity-based and certificateless KEMs. Cryptology ePrint Archive, Report 2005/058, 2005. `http://eprint.iacr.org/`.

[7] D. Boneh and X. Boyen. Efficient selective-id secure identity based encryption without random oracles. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer-Verlag, May 2004.

[8] D. Boneh and X. Boyen. Short signatures without random oracles. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73. Springer-Verlag, May 2004.

[9] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 440–456. Springer-Verlag, May 2005.

[10] D. Boneh and M. K. Franklin. Identity based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.

[11] D. Boneh and J. Katz. Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In A. Menezes, editor, *CT-RSA 2005*, volume 3376 of *LNCS*, pages 87–103. Springer-Verlag, Feb. 2005.

[12] X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based techniques. In *ACM Conference on Computer and Communications Security—CCS 2005*, pages 320–329. New-York: ACM Press, 2005.

[13] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. In *30th ACM STOC*, pages 209–218. ACM Press, May 1998.

[14] R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222. Springer-Verlag, May 2004.

[15] S. Chatterjee and P. Sarkar. Trading time for space: Towards an efficient ibe scheme with short(er) public parameters in the standard model. Proceedings of ICISC, to appear, 2005.

[16] L. Chen and Z. Cheng. Security proof of sakai-kasahara's identity-based encryption scheme. In *Proceedings of Cryptography and Coding 2005*, volume 2796 of *LNCS*, pages 442–459. Berlin: Springer-Verlag, 2005.

[17] L. Chen, Z. Cheng, J. Malone-Lee, and N. Smart. An efficient ID-KEM based on the sakai-kasahara key construction. Cryptology ePrint Archive, Report 2005/224, 2005. `http://eprint.iacr.org/`.

[18] J. H. Cheon. Security analysis of the strong diffie-hellman problem. In *EUROCRYPT 2006*, volume ???? of *LNCS*, pages ???–??? Berlin: Springer-Verlag, 2006.

[19] C. Cocks. An identity based encryption scheme based on quadratic residues. In B. Honary, editor, *Cryptography and Coding, 8th IMA International Conference*, volume 2260 of *LNCS*, pages 360–363, Cirencester, UK, Dec. 17–19, 2001. Springer-Verlag.

[20] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In H. Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 13–25. Springer-Verlag, Aug. 1998.

[21] 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(1):167–226, 2003.

[22] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In M. J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 537–554. Springer-Verlag, Aug. 1999.

[23] C. Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT 2006*, volume ???? of *LNCS*, pages ???–??? Berlin: Springer-Verlag, 2006.

[24] C. Gentry and A. Silverberg. Hierarchical ID-based cryptography. In Y. Zheng, editor, *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 548–566. Springer-Verlag, Dec. 2002.

[25] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.

[26] R. Granger, D. Page, and N. P. Smart. High security pairing-based cryptography revisited. Cryptology ePrint Archive, Report 2006/059, 2006. `http://eprint.iacr.org/`.

[27] J. Horwitz and B. Lynn. Toward hierarchical identity-based encryption. In L. R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 466–481. Springer-Verlag, Apr. 2002.

[28] IEEE P1363.3 Committee. IEEE 1363.3 / CfS — standard for identity-based cryptographic techniques using pairings. `http://grouper.ieee.org/groups/1363/index.html/`, Feb. 2006. Call for submissions.

[29] E. Kiltz and D. Galindo. Direct chosen-ciphertext secure identity-based encryption without random oracles. Cryptology ePrint Archive, Report 2006/034, 2006. `http://eprint.iacr.org/`.

[30] K. Kurosawa and Y. Desmedt. A new paradigm of hybrid encryption scheme. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 426–442. Springer-Verlag, Aug. 2004.

[31] B. Libert and J. Quisquater. Identity based encryption without redundancy. In *ACNS'05*, pages 285–300, 2005.

[32] V. Miller. The Weil pairing, and its efficient calculation. *Journal of Cryptology*, 17(4), 2004.

[33] D. Naccache. Secure and *practical* identity-based encryption. Cryptology ePrint Archive, Report 2005/369, 2005. `http://eprint.iacr.org/`.

[34] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *21st ACM STOC*, pages 33–43. ACM Press, May 1989.

[35] T. Okamoto. Efficient blind and partially blind signatures without random oracles. In S. Halevi and T. Rabin, editors, *TCC 2006*, LNCS, pages ???–??? Springer-Verlag, Mar. 2006.

[36] D. Page, N. Smart, and F. Vercauteren. A comparison of MNT curves and supersingular curves. Cryptology ePrint Archive, Report 2004/165, 2004. `http://eprint.iacr.org/`.

[37] D. H. Phan and D. Pointcheval. Chosen-ciphertext security without redundancy. In C.-S. Laih, editor, *ASIACRYPT 2003*, volume 2894 of *LNCS*, pages 1–18. Springer-Verlag, Nov. 2003.

[38] D. H. Phan and D. Pointcheval. About the security of ciphers (semantic security and pseudo-random permutations). In H. Handschuh and A. Hasan, editors, *SAC 2004*, volume 3357 of *LNCS*. Springer-Verlag, Aug. 2004.

[39] J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd ACM STOC*, pages 387–394. ACM Press, May 1990.

[40] R. Sakai and M. Kasahara. ID based cryptosystems with pairing on elliptic curve. Cryptology ePrint Archive, Report 2003/054, 2003. `http://eprint.iacr.org/`.

[41] Secure hash standard. National Institute of Standards and Technology, NIST FIPS PUB 180-1, U.S. Department of Commerce, Apr. 1995.

[42] A. Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and D. Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53. Springer-Verlag, Aug. 1985.

[43] V. Shoup. Why chosen ciphertext security matters. IBM Research Report RZ 3076, November 1998. Available from `http://shoup.net/papers/`.

[44] V. Shoup. OAEP reconsidered. In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 239–259. Springer-Verlag, Aug. 2001.

[45] B. R. Waters. Efficient identity-based encryption without random oracles. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer-Verlag, May 2005.

A Timing data from [26].

The timing data used in our comparison in Section 6.3 is given in Table 6.

k	Curve ($\log_2 p, \alpha$)	exp $\mathbb{G}_1$	hash $\mathbb{G}_1$	exp $\mathbb{G}_2$	hash $\mathbb{G}_2$	exp $\mathbb{G}_T$	pairing	$\lvert\mathbb{G}_1\rvert$	$\lvert\mathbb{G}_2\rvert(\lvert G_T\rvert)$
80	A (160,6)	0.9	0	4.8	9.4	0.8	5.4	160	480
128	B (512,6)	14	14	69	330	11	101	512	1536
128	C (256,12)	3.6	0	50	243	5	63	256	1536
192	D (1365,6)	140	361	700	6419	120	1291	1365	4095
192	E (683,12)	36	28	494	4608	49	780	683	4098
256	F (2560,6)	644	2493	3223	42714	552	7345	2560	7680
256	G (1280,12)	163	241	2252	30377	217	4192	1280	7680
256	H (640,24)	42	10	1382	18480	115	2781	640	7680

Table 6: Timings in terms of estimated 32 bit multiplications needed to perform atomic primitives (scaled by 10^5) and representation of group elements in bits. Hashing into the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ is dominated by a cofactor exponentiation which in case of hashing into $\mathbb{G}_1$ for curves A and C (nearly) for free due to the small cofactor. All data is taken from Section 5 of [26].

B Target Collision Resistant Hash Functions

Let $\mathcal{F} = (\mathsf{CR}_s)_{s\in S}$ be a family of hash functions for security parameter k and with seed $s \in S = S(k)$. $\mathcal{F}$ is said to be *collision resistant* if, for a hash function $\mathsf{CR} = \mathsf{CR}_s$ (where the seed is chosen at random from S), it is infeasible for an efficient adversary to find two distinct values $x \neq y$ such that $\mathsf{CR}(x) = \mathsf{CR}(y)$.

A weaker notion is that of *target collision resistant hash functions*. Here it should be infeasible for an efficient adversary to find, given a randomly chosen element x and a randomly drawn hash function $\mathsf{TCR} = \mathsf{TCR}_s$, a distinct element $y \neq x$ such that $\mathsf{TCR}(x) = \mathsf{TCR}(y)$. (In collision resistant hash functions the value x may be chosen by the adversary.) Such hash functions are also called *universal one-way hash functions* [34] and can be built from arbitrary one-way functions [34, 39]. We define (slightly informal)

$$\mathbf{Adv}^{\text{hash-tcr}}_{\mathsf{TCR},\mathcal{H}}(k) = \Pr[\mathcal{H} \text{ finds a collision in } \mathsf{TCR}].$$

Hash function family is said to be a *target collision resistant* if the advantage function $\mathbf{Adv}^{\text{hash-tcr}}_{\mathsf{TCR},\mathcal{H}}$ is a negligible function in k for all polynomial-time adversaries $\mathcal{H}$.

In practice, to build a target collision resistant hash function TCR, one can use a dedicated cryptographic hash function, like SHA-1 [41]. For that reason and to simplify our presentation, in what follows we will consider the hash function TCR to be a fixed function.

C Proof of Theorem 4.1

In this section we provide a game-based proof of Theorem 4.1. In fact in our proof some games could be absorbed nearly verbatim from [29], i.e. Games 1-4 are the nearly same as Games 1-4 in [29]. The rest of the games are similar but due to the short ciphertexts and the different security assumption in our construction important and non-trivial changes had to be made to the respective games.

We will make use of the following simple "Difference Lemma" [44].

Lemma C.1 Let X_1,X_2, B be events defined in some probability distribution, and suppose that $X_1 \wedge \neg B \Leftrightarrow X_2 \wedge \neg B$. Then $|\Pr[X_1] - \Pr[X_2]| \leq \Pr[B]$.

We assume modified BDDH is hard, i.e. for any adversary $\mathcal{B}$ running for polynomial time $\mathbf{Time}_{\mathcal{B}}(k)$ we have $\mathbf{Adv}^{\text{mbddh}}_{\mathcal{G},\mathcal{B}}(k) = \epsilon(k)$, for a negligible function $\epsilon = \epsilon(k)$. We will show that for any adversary $\mathcal{A}$ against the chosen-ciphertext security of the IBE scheme running for time

$$\mathbf{Time}_{\mathcal{A}}(k) = \mathbf{Time}_{\mathcal{B}}(k) - \Omega(\epsilon^{-2} \cdot \ln(\epsilon^{-1}) + q)$$

and making a maximum of $q = q(k)$ key-derivation/decapsulation queries, we have

$$\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{IBKEM},\mathcal{A}}(k) \quad = \quad \mathcal{O}(nq \cdot (\epsilon + q/p) + \mathbf{Adv}^{\text{hash-tcr}}_{\mathsf{TCR},\mathcal{H}}(k)) \ .$$

Game 0. Fix an efficient adversary $\mathcal{A}$. We now define a game, Game 0, an interactive game between adversary $\mathcal{A}$ and a simulator. Game 0 is simply the same game as the IBE security experiment of Section 2.2 in which the simulator provides adversary $\mathcal{A}$'s environment. While describing the experiment we wll make a couple of conventions on how the simulator chooses the values appearing in its simulation. These conventions will be purely conceptual and, compared to the original experiment, do not change the distribution of any value appearing during the experiment. We will also make a couple of definitions of values appearing during the experiments.

We assume that in the beginning the simulator chooses some values a, b, and c, uniformly distributed from $\mathbb{Z}_p$. The whole simulation will depend on these values (i.e., the key generation will depend on a, b, where the challenge ciphertext will depend on c). In sequel games the simulator will "forget" the values a, b, and c and instead only use the values g^a, g^b, g^{b^2}, and g^c.

Key Generation. Initially the simulator runs the IBE key generation algorithm $\mathsf{Setup}(1^k)$ and obtains the public key $pk = (u, z, \mathsf{H})$ and secret key $sk = \alpha$. We make the convention that the keys are generated as

$$u \xleftarrow{\$} g^b \ ; \quad z \leftarrow \hat{e}(g^a, g^b) \ ; \quad \mathsf{H} \xleftarrow{\$} \mathsf{HGen}(\mathbb{G}) \tag{1}$$

depending on the element g^a, g^b chosen by the simulator in advance. Note that the way the value $z = \hat{e}(g^a, g^b) = \hat{e}(g, g^{ab})$ from the public key is generated implies $\alpha = g^{ab} = u^a$. Note that α can be computed by the simulator since a is still known in this game. The public key is given to the adversary to start its `find` phase.

Find Phase. During its execution adversary $\mathcal{A}$ makes a number of key derivation and decapsulation requests. If the adversary makes a key derivation query $\mathsf{IBKeyDer}(id)$ then (using its secret key α) the simulator computes the secret key $sk[id]$ and returns it to the adversary. If the adversary makes a decapsulation query $\textsc{Decaps}(id, C)$ the simulator (using α) decapsulates the ciphertext and returns the session key to the adversary.

Eventually, the adversary returns a target identity id^*. The simulator chosen a random key K_0^* and run the encapsulation algorithm to create a key K_1^* together with the the challenge ciphertext $C^* = (c_1^*, c_2^*)$. We make the convention that the challenge ciphertext $C^* = (c_1^*, c_2^*)$ is computed as

$$c_1^* \leftarrow g^c, \quad t^* \leftarrow \mathsf{TCR}(g^c), \quad c_2^* \leftarrow \mathsf{H}(id^*)^c u^{ct^*}, \tag{2}$$

depending on the random value c chosen by the simulator in advance, and the key $K_1^* = z^c$. Then the simulator chooses a random bit γ and the challenge ciphertext C^* together with the key $K^* = K_\gamma^*$ is returned to the adversary.

Guess Phase. The adversary continues to make its oracle queries, subsequent key derivation requests must be different from the target identity id^* and decapsulation requests must be different from (id^*, C^*). Finally, adversary $\mathcal{A}$ returns a bit $\gamma' \in \{0, 1\}$. If $\gamma \neq \gamma'$ the simulator returns $\beta' = 0$, else it returns $\beta' = 1$. This completes the description of the simulator. Note that the simulator behaves exactly as in the original IBE security experiment.

Now a few important definitions are in place. During its execution $\mathcal{A}$ may query the key derivation oracle for some identity id or the decapsulation oracle for the identity/ciphertext pair (id, C). We collect all those identities used to make queries to the key derivation and decapsulation oracle in the set $\widetilde{ID}$. Note that $\widetilde{ID}$ may contain the target identity id^* or one identity more than once. Let ID be the subset of queried identities obtained by removing from $\widetilde{ID}$ all multiples and the target identity. We write $ID = \{id^{(1)}, \dots, id^{(q_0)}\}$ (without any particular order) for some $q_0 \le q$ such that $id^{(i)} \ne id^{(j)}$ for each $1 \le i \ne j \le q_0$ and $id^* \notin ID$. Furthermore, we define $ID^* = ID \cup \{id^*\} = \{id^{(1)}, \dots, id^{(q_0)}, id^*\}$.

The proof of the theorem is obtained by considering subsequent games, Game 1, Game 2, ..., These games will be quite similar to Game 0. In every game the simulators' output bit β' will be well-defined. For each i we define the event

$$X_i : \text{The simulator outputs } \beta' = 1 \text{ in Game } i.$$

Then, since in Game 0 the simulator exactly plays the IBE security experiment with adversary $\mathcal{A}$, we have

$$|\Pr[X_0] - 1/2| = \mathbf{Adv}^{\text{ind-cca}}_{\mathcal{IBKEM},\mathcal{A}} .$$

Game 1. (Eliminate hash collisions) Note that the values $c_1^* = g^c$ and $t^* = \mathsf{TCR}(g^c)$ from the challenge ciphertext Equation (2) are completely independent of the view of adversary $\mathcal{A}$ until $\mathcal{A}$'s guess phase (since c is simply not touched by the simulator before generating the challenge ciphertext). Therefore we may assume that the value c_1^* and t^* are already generated by the simulator before the key generation.

In this game the simulator changes its answers to all decapsulation queries $\textsc{Decaps}(id, C)$ made by $\mathcal{A}$ as follows: Let $C = (c_1, c_2)$ and $t = \mathsf{TCR}(c_1)$. If $t = t^*$ and $c_1 \ne c_1^*$, the simulator aborts. Otherwise it continues as in the last game. Let $\textsc{HashAbort}$ be the event that this new abortion rule applies. Until $\textsc{HashAbort}$ happens Game 0 and Game 1 are identical. Therefore by Lemma C.1 we have

$$|\Pr[X_1] - \Pr[X_0]| \le \Pr[\textsc{HashAbort}] .$$

Furthermore,

$$\Pr[\textsc{HashAbort}] \le \mathbf{Adv}^{\text{hash-tcr}}_{\mathsf{TCR},\mathcal{H}}(k) ,$$

i.e. there exists an adversary $\mathcal{H}$ against the target collision resistence of TCR (note that $c_1^* = g^c$ is a random element coming from outside $\mathcal{H}$'s view) running in time $\mathbf{Time}_{\mathcal{H}}(k) = \mathbf{Time}_{\mathcal{A}}(k) + \mathcal{O}(1)$ that succeeds with probability at least $\Pr[\textsc{HashAbort}]$.

Game 2. (Change of the hash keys) This is the same as Game 1 except that the simulator changes the generation of the hash keys $h = (h_0, h_1, \dots, h_n)$ as follows.

Set $m = 2q$ (the choice of m will become clear later). Instead of generating the hash keys with the hash key-generation algorithm $\mathsf{HGen}(\mathbb{G})$ as in the last game the simulator chooses

$$\begin{aligned} x_0, x_1, \dots, x_n &\stackrel{\$}{\leftarrow} \{0, \dots, p-1\} \\ y_0', y_1, \dots, y_n &\stackrel{\$}{\leftarrow} \{0, \dots, m-1\} \\ k &\stackrel{\$}{\leftarrow} \{0, \dots, n\} \\ \text{and sets} & \\ y_0 &\leftarrow p - km + y_0' . \end{aligned} \tag{3}$$

The public keys $h = (h_0, \dots, h_n)$ of the hash function H are then defined as $h_0 = (g^a)^{y_0} \cdot (g^b)^{-t^*} \cdot g^{x_0}$ and $h_i = g^{x_i}(g^a)^{y_i}$, for $1 \le i \le n$. The public hash function is $\mathsf{H}(id) = h_0 \prod_{i=1}^n h_i^{id_i}$. From the simulator's point of view, the hash function evaluated in identity $id \in \{0,1\}^n$ is

$$\mathsf{H}(id) = g^{x(id)+y(id)a-t^*b}, \tag{4}$$

with $x(id) = x_0 + \sum_{i=1}^n id_i x_i$ and $y(id) = y_0 + \sum_{i=1}^n id_i y_i$ only known to the simulator. On the other hand note that this does not change the distribution of the hash keys $h = (h_0, h_1, \dots, h_n)$. Therefore we have

$$\Pr[X_1] = \Pr[X_2] \ .$$

Game 3. (Abort at the end of the game) Fix all the random variables adversary $\mathcal{A}$ gets to see during its execution, including its random coin tosses: fix pk, the challenge bit γ, and the randomness used in answering the key derivation and decapsulation queries. Now the adversary can be seen as a deterministic algorithm, in particular the set of all queried (distinct) identities $ID^* = \{id^{(1)}, \dots, id^{(q_0)}, id^*\}$ can be seen as fixed. By $view_{\mathcal{A}}$ we denote all these fixed variables.

Define $\mathbf{Y} = (y_0', y_1, \dots, y_n, k)$, where the random variables $(y_0', y_1, \dots, y_n, k)$ are distributed as in Equation (3). It is clear that once $view_{\mathcal{A}}$ is fixed, the random variable $\mathbf{Y}$ still has its original distribution. Define the event

$$\textsc{ForcedAbort} : \bigvee_{i=1}^{q_0} \left(y(id^{(i)}) = 0 \bmod p \right) \vee y(id^*) \neq 0 \bmod p \ .$$

We call this abort *forced* since in sequel games the simulator is modified such that it always *has to* abort once this event happens. For fixed $view_{\mathcal{A}}$ we define

$$\eta := \Pr_{\mathbf{Y}}[\neg\textsc{ForcedAbort}] \tag{5}$$

and let λ be a lower bound on η (that holds for every $view_{\mathcal{A}}$). The following lemma provides a lower bound on η.

Lemma C.2 For each possible choice of identities $ID^* = \{id^{(1)}, \dots, id^{(q_0)}, id^*\}$ we have $\eta \geq \lambda = \frac{1}{4(n+1)q}$.

The proof of the lemma is given in [29].

Compared to Game 2 we will make two modifications to the simulator in Game 3. The simulation is exactly the same as in Game 2 until adversary $\mathcal{A}$ outputs his guess bit γ'. Since adversary $\mathcal{A}$ already terminated we can assume $view_{\mathcal{A}}$ to be fixed from now on.

First modification: add forced abort. After adversary $\mathcal{A}$ outputs his guess bit γ', the simulator checks if the event ForcedAbort occurs. If yes, it aborts the game and returns a random bit as its output bit β'. If not the simulation is continued as before.

Let's first make an unsuccessful attempt to relate the two events X_3 and X_2. Clearly we have $\Pr[X_3] = \Pr[X_2 \wedge \neg\textsc{ForcedAbort}] + 1/2 \cdot \Pr[\textsc{ForcedAbort}]$. Now we would like to continue with $\Pr[X_2 \wedge \neg\textsc{ForcedAbort}] \geq \Pr[X_2] \cdot \Pr[\neg\textsc{ForcedAbort}]$. However, this is not correct since the simulator may aborts with a probability that is a function in the choices of the identities $ID^* = \{id^{(1)}, \dots, id^{(q_0)}, id^*\}$ queried by adversary $\mathcal{A}$ and hence the two events X_2 and $\neg\textsc{ForcedAbort}$ cannot be considered as independent.

To get rid of this unwanted dependence the simulator adds some *artificial abort* such that it always aborts with probability nearly λ (recall that λ was is upper bound on the abortion probability), independent of the choices of the identities $ID^* = \{id^{(1)}, \dots, id^{(q_0)}, id^*\}$. This way it will be possible to decorrelate the event X_2 with the abortion.

Second modification: add artificial abort. After the simulator has checked for the event ForcedAbort (and decided not to abort), it continues as follows: First it samples (using sufficiently many samples) an estimate η' of the probability η (over $\mathbf{Y}$) that the ForcedAbort happens (cf.

Eqn. (5)).[6] We want to stress that $view_{\mathcal{A}}$ is fixed at this point so sampling does not involve running adversary $\mathcal{A}$ again. This estimate η' is a function in $id^{(1)}, \ldots, id^{(q_0)}, id^*$.
Depending on the estimate η' the simulator distinguishes two cases:
Case $\eta' \le \lambda$: the simulator continues as before.
Case $\eta' > \lambda$: With probability $1 - \lambda/\eta'$ the simulator aborts and outputs a random bit β'. With probability λ/η' the simulator does not abort and continues as before.
This concludes the description of Game 3.

The following is formally proved in [29].

Lemma C.3 Let $0 < \rho \le 1$ be a function in k. If the simulator takes $\mathcal{O}(\rho^{-2}\ln(\rho^{-1}) \cdot \lambda^{-1}\ln(\lambda^{-1}))$ samples when computing the estimate η', then

$$\left|\Pr[X_2] - \frac{1}{2} - \frac{\Pr[X_3] - 1/2}{\lambda}\right| \le \frac{\rho}{2} .$$

The parameter ρ will be determined at the end of the proof.

Game 4. (Forced abort during the game I) Compared to the last game we make the following changes to the simulator: When identity $id \in ID$ is queried to the key derivation oracle, the simulator immediately aborts if $y(id) = 0 \bmod p$. When receiving the challenge identity id^*, the simulator immediately aborts if $y(id^*) \neq 0 \bmod p$. On abort the simulator returns a random bit β'. The artificial abort at the end of the simulation is the same as in the last game.

Clearly, this modification does not affect the adversary if there is no forced abort. In case there is a new forced abort the simulator outputs a random bit β' as in Game 3. Therefore we have

$$\Pr[\, X_4 \,] = \Pr[\, X_3 \,] .$$

Game 5. (Change key derivation oracle) The simulator changes its answers to all key derivation queries $\mathsf{IBKeyDer}(id)$ made by the adversary $\mathcal{A}$ as follows: By Eqn. (4) we have $\mathsf{H}(id) = g^{x(id)+y(id)a-t^*b}$, for some values $x(id)$ and $y(id)$ known to the simulator.
Case $y(id) = 0 \bmod p$: The simulator aborts (as in the last game).
Case $y(id) \neq 0 \bmod p$: The derived key $sk[id] = (d_1, d_2, d_3)$ is computed as follows:
For a random $r' \in \mathbb{Z}_p$, the simulator implicitly defines $r = -b/y(id) + r' \bmod p$ and computes

$$\begin{aligned}
d_1 &\leftarrow (g^a)^{y(id)r'} \cdot (g^b)^{-x(id)/y(id)-r't^*} \cdot (g^{b^2})^{t^*/y(id)} \cdot g^{x(id)r'} \\
d_2 &\leftarrow (g^b)^{-1/y(id)} \cdot g^{r'} \\
d_3 &\leftarrow (g^{b^2})^{-1/y(id)} \cdot (g^b)^{r'} .
\end{aligned}$$

Note that the randomness r is not known to the simulator and that the generation of the derived keys $sk[id]$ does not involve the knowledge of the secret key $\alpha = g^{ab}$ anymore.

Lemma C.4 $\Pr[X_4] = \Pr[X_5]$.

Proof: We have to verify that each derived key $sk[id] = (d_1, d_2, d_3)$ is identically distributed as in the last game. Let us abbreviate $x = x(id)$, and $y = y(id) \neq 0 \bmod p$. Clearly, if r' is uniform in $\mathbb{Z}_p$ then

[6] Unfortunately, there seems not to be an efficient way to compute the exact value η. If there was one we could greatly simplify our analysis.

so is r. Then

$$\begin{aligned} d_1 &= (g^a)^{yr'} \cdot (g^b)^{-x/y-r't^*} \cdot (g^{b^2})^{t^*/y} \cdot g^{xr'} \\ &= g^{ayr'-bx/y-br't^*+b^2t^*/y+xr'} \\ &= g^{ay(r+b/y)-bx/y-bt^*(r+b/y)^*+b^2t/y+x(r+b/y)} \\ &= g^{ayr+ab-bx/y-bt^*r-b^2t^*/y+b^2t^*/y+xr+xb/y} \\ &= g^{ayr+ab-bt^*r+xr} \\ &= \alpha \cdot (g^{ay-bt^*+x})^r \\ &= \alpha \cdot (\mathsf{H}(id))^r , \end{aligned}$$

and

$$\begin{aligned} d_2 &= (g^b)^{-1/y} \cdot g^{r'} \\ &= g^{-b/y} g^{r-b/y} \\ &= g^r , \end{aligned} \qquad \begin{aligned} d_3 &= (g^{b^2})^{-1/y} \cdot (g^b)^{r'} \\ &= u^{-b/y} u^{r-b/y} \\ &= u^r . \end{aligned}$$

▮

Game 6. (Forced abort during the game II) Compared to the last game we make the following changes to the simulator: When the tuple (id, C) is queried to the decapsulation oracle for $id \in ID \cup \{id^*\}$ and $C = (c_1, c_2)$ the simulator computes $t = \mathsf{TCR}(c_1)$ and immediately aborts if $y(id) = 0 \bmod p$, C is consistent, and $t = t^*$. In case of abort the simulator returns a random bit β'.

Lemma C.5 $|\Pr[X_5] - \Pr[X_6]| \leq \frac{2q_2}{p}$, where q_2 is an upper bound on the number of decapsulation queries an adversary makes.

Proof: Clearly, this modification does not affect the adversary if there is no new forced abort. Note that a new forced abort implies $c_1 = c_1^*$ since otherwise by $t = t^*$ the simulator already aborted in the last game and found a collision in the hash function TCR. If there is a new forced abort we distinguish between two cases:
Case 1: the new forced abort happens in the `guess` stage. Recall that we call a ciphertext $C = (c_1, c_2)$ consistent if $(g, c_1, \mathsf{H}(id) \cdot u^t, c_2)$ is a Diffie-Hellman tuple (where $t = \mathsf{TCR}(c_1)$), i.e. if $(g, c_1, \mathsf{H}(id) \cdot u^t, c_2) = (g, g^r, \mathsf{H}(id) \cdot u^t, (\mathsf{H}(id) \cdot u^t)^r)$ for some value $r \in \mathbb{Z}_p$.

Note that the way the public-key is generated by Eqn. (4) and since $y(id) = 0$ and $t = t^*$, for any consistent ciphertext C we have

$$c_2 = (\mathsf{H}(id) \cdot u^t)^r = g^{r(x(id)+b(t-t)^*)} = (c_1^b)^{t-t^*} \cdot c_1^{x(id)} = c_1^{x(id)} . \tag{6}$$

If $id = id^*$ (i.e., if $\mathcal{A}$ queries the decapsulation oracle with the target identity) then Equation (6) implies $c_2 = c_1^{x(id)} = (c_1^*)^{x(id^*)} = c_2^*$. Consequently $C = C^*$ and so the simulator rejects as in the original IBE security experiment. If $id \neq id^*$ then, by definition, $id \in ID$ and the simulator outputs a random bit β' as in Game 5 where the abort was still done at the end of the experiment. Therefore, conditioned on case 1 we have $\Pr[X_5] = \Pr[X_6]$.

Case 2: the new forced abort happens in the `find` stage. Since in the find stage the adversary has no information (in a statistical sense) about c_1^* from the challenge ciphertext C^*, and the adversary makes at most q_2 decapsulation queries in its find stage, this implies

$$|\Pr[X_5] - \Pr[X_6]| \leq \frac{1}{p} + \frac{1}{p-1} + \ldots + \frac{1}{p-q_2+1} \leq \frac{q_2}{p-q_2} \leq \frac{2q_2}{p} ,$$

as claimed. ▌

Game 7. (Change the answers to the decapsulation queries.) The simulator changes its answers to all decapsulation queries DECAPS(id, C) made by $\mathcal{A}$ as follows: By Eqn. (4) we have $\mathsf{H}(id) = g^{x(id)+y(id)a-t^*b}$ for some values $x(id)$ and $y(id)$ known to the simulator.
Case $y(id) \neq 0 \bmod p$: the query is answered using the key derivation oracle.
Case $y(id) = 0 \bmod p$: the simulator simulates the decapsulation queries as follows: Let $C = (c_1, c_2, E)$ be the queried ciphertext and let $t = \mathsf{TCR}(c_1)$.

If the ciphertext is not consistent then return a random session key K
If $t = t^*$ then the simulator aborts (as in the last game)
If $t \neq t^*$ then return $K \leftarrow \hat{e}(c_2/c_1^{x(id)}, g^a)^{(t-t^*)^{-1}}$

We claim that these changes do not affect the view of $\mathcal{A}$:

Lemma C.6 $\Pr[X_6] = \Pr[X_7]$.

Proof of Lemma C.6: Let $C = (c_1, c_2)$ be an arbitrary ciphertext submitted to the decapsulation oracle with respect to identity id. In case $y(id) \neq 0 \bmod p$ decapsulation will be done using the simulation of the key derivation oracle which we already showed to be correct so we may now assume $y(id) = 0 \bmod p$. Every inconsistent ciphertext leads to a random key K as in the original description of the scheme so in what follows we may also assume a consistent ciphertext.

We distinguish the following three cases: Case 1a: $t = t^*$ and $c_1 \neq c_1^*$. In this case the simulator has found a collision in the hash function TCR and aborts as in the last game.
Case 1b: $t = t^*$ and $c_1 = c_1^*$. In the case the simulator aborts as in forced abort introduced in the last game.

Case 2: $t \neq t^*$. Similar to Eqn. (6) consistency of C implies

$$c_2 = (\mathsf{H}(id) \cdot u^t)^r = g^{r(x(id)+b(t-t)^*)} = (c_1^b)^{t-t^*} \cdot c_1^{x(id)} ,$$

and we obtain

$$(c_2/c_1^{x(id)})^{(t-t^*)^{-1}} = ((c_1^b)^{t-t^*} \cdot c_1^{x(id)}/c_1^{x(id)})^{(t-t^*)^{-1}} = c_1^b . \tag{7}$$

In the original IBE decapsulation algorithm first the user secret key for identity id is computed as $sk[id] = (d_1, d_2, d_3) = (\alpha \cdot \mathsf{H}(id)^s, g^s, u^s)$ for random s, and then the session key K is reconstructed as

$$\begin{aligned} K = \hat{e}(c_1, d_1 \cdot d_3^t)/\hat{e}(c_2, d_2) &= \hat{e}(c_1, \alpha \cdot \mathsf{H}(id)^s \cdot (u^s)^t)/\hat{e}(c_2, g^s) \\ &= \hat{e}(c_1^b, g^a) \cdot \hat{e}(c_1, \mathsf{H}(id)^s \cdot (u^s)^t)/\hat{e}(c_2, g^s) \\ &\overset{(7)}{=} \hat{e}((c_2/c_1^{x(id)})^{(t-t^*)^{-1}}, g^a) \cdot (\hat{e}(c_1, \mathsf{H}(id) \cdot u^t)/\hat{e}(c_2, g))^s \\ &= \hat{e}(c_2/c_1^{x(id)}, g^a)^{(t-t^*)^{-1}} \cdot (\Delta(C))^s , \end{aligned}$$

with $\Delta(C) = \hat{e}(c_1, \mathsf{H}(id) \cdot u^t)/\hat{e}(c_2, g)$. Since $(\Delta(C))^s = 1$ if $\hat{e}(c_1, \mathsf{H}(id)u^t) = \hat{e}(g, c_2)$ and $(\Delta(C))^s$ is a random element in $\mathbb{G}_T$ otherwise, the decapsulated session key in the original scheme is distributed as in the simulation. ▌

Game 8. (Modify the challenge) After $\mathcal{A}$'s `find` stage the simulator inputs the target identity id^* from $\mathcal{A}$. The simulator modifies the computation of the challenge ciphertext C^* follows:
Case $y(id^*) \neq 0 \bmod p$: The simulator aborts (as in the last game).

Case $y(id^*) = 0 \bmod p$: The simulator chooses a random bit γ and creates the challenge ciphertext $C^* = (c_1^*, c_2^*)$ and key K_1^* as

$$c_1^* \leftarrow g^c, \quad c_2^* \leftarrow c_2^* \leftarrow (g^c)^{x(id^*)}, \quad K_1^* \leftarrow \hat{e}(g,g)^{abc} \ . \tag{8}$$

By virtue of Eqns. (4), (6), and since $\mathsf{TCR}(c_1^*) = t^*$ and $y(id^*) = 0 \bmod p$, C^* is a correctly distributed ciphertext of K_1^*. Clearly,

$$\Pr[X_8] = \Pr[X_7] \ .$$

Game 9. (Replace the Challenge) The simulator replaces the value K_1^* from the challenge C^* with a random element from $\mathbb{G}_T$. Since K_1^* is now completely independent of the challenge bit γ, we have

$$\Pr[X_9] = 1/2 \ .$$

Observe that Game 9 does not use the secret key anymore and that the whole simulation only depends on the values g^a, g^b, g^{b^2}, g^c (i.e., the simulator "forgot the values a, b, and c). Game 8 and Game 9 are equal unless adversary $\mathcal{A}$ can distinguish $\hat{e}(g,g)^{abc}$ (the value of K_1^* in Game 8) from a random element in $\mathbb{G}_T$ (the value of K_1^* in Game 9). Therefore we have

$$|\Pr[X_9] - \Pr[X_8]| \leq \mathbf{Adv}^{\mathrm{mbddh}}_{\mathcal{G},\mathcal{B}}(k),$$

for any adversary $\mathcal{B}$ against the hardness of mBDDH running in the same time as the simulator, i.e. $\mathbf{Time}_{\mathcal{B}} = \mathbf{Time}_{\mathcal{A}} + \mathcal{O}(\rho^{-2}\ln(\rho^{-1}) \cdot \lambda^{-1} \cdot \ln(\lambda^{-1}) + q)$.

Analysis. We collect the probabilities relating the different games as follows:

$$\begin{aligned}
\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{IBKEM},\mathcal{A}} &= |\Pr[X_0] - \frac{1}{2}| \\
&\leq |\Pr[X_1] + \mathbf{Adv}^{\text{hash-tcr}}_{\mathsf{TCR},\mathcal{H}}(k) - \frac{1}{2}| \\
&\leq |\Pr[X_2] - 1/2 + \mathbf{Adv}^{\text{hash-tcr}}_{\mathsf{TCR},\mathcal{H}}(k)| \\
&\leq |\frac{\Pr[X_3] - \frac{1}{2}}{\lambda} + \frac{\rho}{2} + \mathbf{Adv}^{\text{hash-tcr}}_{\mathsf{TCR},\mathcal{H}}(k)| \\
&\leq \frac{|\Pr[X_6] + \frac{2q_2}{p} - \frac{1}{2}|}{\lambda} + \frac{\rho}{2} + \mathbf{Adv}^{\text{hash-tcr}}_{\mathsf{TCR},\mathcal{H}}(k) \\
&\leq \frac{|\Pr[X_9] + \frac{2q_2}{p} - \frac{1}{2}|}{\lambda} + \frac{\rho}{2} + \mathbf{Adv}^{\text{hash-tcr}}_{\mathsf{TCR},\mathcal{H}}(k) \\
&\leq \frac{\mathbf{Adv}^{\mathrm{mbddh}}_{\mathcal{G},\mathcal{B}}(k) + \frac{2q_2}{p}}{\lambda} + \frac{\rho}{2} + \mathbf{Adv}^{\text{hash-tcr}}_{\mathsf{TCR},\mathcal{H}}(k) \ .
\end{aligned}$$

Using $\lambda = \frac{1}{4(n+1)q}$ (by Lemma C.2) and defining $\rho = \mathbf{Adv}^{\text{ind-cca}}_{\mathcal{IBKEM},\mathcal{A}}(k)$ we conclude the proof with

$$\begin{aligned}
\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{IBKEM},\mathcal{A}}(k) &\leq 8(n+1)q \cdot (\mathbf{Adv}^{\mathrm{mbddh}}_{\mathcal{G},\mathcal{B}}(k) + 2q_2/p) + \mathbf{Adv}^{\text{hash-tcr}}_{\mathsf{TCR},\mathcal{H}}(k) \\
&= \mathcal{O}\left(nq \cdot (\mathbf{Adv}^{\mathrm{mbddh}}_{\mathcal{G},\mathcal{B}}(k) + q/p) + \mathbf{Adv}^{\text{hash-tcr}}_{\mathsf{TCR},\mathcal{H}}(k)\right) \ ,
\end{aligned}$$

where q is an upper bound on all (derivation plus decapsulation) queries made by $\mathcal{A}$,

$$\begin{aligned}
\mathbf{Time}_{\mathcal{B}}(k) &= \mathbf{Time}_{\mathcal{A}}(k) + \mathcal{O}(\rho^{-2}\ln(\rho^{-1}) \cdot \lambda^{-1} \cdot \ln(\lambda^{-1})) \\
&= \mathbf{Time}_{\mathcal{A}}(k) + \mathcal{O}(\epsilon_{\mathcal{A}}^{-2}(k)\ln(\epsilon_{\mathcal{A}}^{-1}(k)) \cdot \lambda^{-1} \cdot \ln(\lambda^{-1})) \ ,
\end{aligned}$$

where $\epsilon_{\mathcal{A}}(k) = \mathbf{Adv}^{\text{ind-cca}}_{\mathcal{IBKEM},\mathcal{A}}(k)$, and

$$\mathbf{Time}_{\mathcal{H}}(k) \quad = \quad \mathbf{Time}_{\mathcal{A}}(k) + \mathcal{O}(1) \; .$$

D Relations between the Assumptions

D.1 The BDDH assumption

Let $\mathcal{PG}$ be the description of bilinear groups and let $g \in \mathbb{G}$ be a random element from group $\mathbb{G}$ of prime order p. Consider the following problem formalized by Boneh and Franklin [10]: Given $(g, g^a, g^b, g^c, W) \in \mathbb{G}^4 \times \mathbb{G}_2$ as input, output yes if $W = \hat{e}(g,g)^{abc}$ and no otherwise. The corresponding BDDH assumption can be formalized the same way as the modified BDDH assumption.

D.2 The q-BDDHI assumptions

Let $\mathcal{PG}$ as above and let $z \in \mathbb{G}$ be a random element from group $\mathbb{G}$. Let $q = q(k)$ be a function polynomial in the security parameter. Associated to q the following problem introduced by Boneh and Boyen [7]: Given $(h, h^a, h^{(a^2)}, \ldots, h^{(a^q)}, W) \in \mathbb{G}^{q+1} \times \mathbb{G}_2$ as input, output yes if $W = \hat{e}(h,h)^{1/a}$ and no otherwise.

D.3 Proof of Lemma 3.1

Proof: The implications BDDH $\leq$ mBDDH and 1-BDDHI $\leq$ 2-BDDHI $\leq$ 3-BDDHI $\leq \ldots$ are easy to show. To prove "modified BDDH assumption $\leq$ 2-BDDHI assumption", assume there exists a polynomial-time adversary $\mathcal{A}$ that breaks the modified BDDH assumption. We show that then there exists a polynomial-time adversary $\mathcal{B}$ with oracle access to $\mathcal{A}$ that breaks the 2-BDDHI assumption. Let (h, h^a, h^{a^2}, W) be an input instance of the 2-BDDHI problem given to $\mathcal{B}$. $\mathcal{B}$'s goal is to find out if $W = \hat{e}(h,h)^{1/a}$ or W is random. $\mathcal{B}$ picks two random values y_0, z_0 and defines its output bit as $\gamma := \gamma'$, where γ' is input from $\mathcal{A}$ as

$$\gamma' \leftarrow \mathcal{A}(h^{a^2}, h^a, h, h^{y_0}, h^{z_0}, W' = W^{y_0 z_0}).$$

We now show correctness. Defining $g := h^{a^2}$, $x = 1/a$, $y = y_0/a^2$, and $z = z_0/a^2$, we have $h^a = g^{1/a} = g^x$ and $h = g^{1/a^2} = g^{x^2}$. Consequently, $(h^{a^2}, h^a, h, h^{y_0}, h^{z_0}) = (g, g^x, g^{x^2}, g^y, g^z)$. If $W = \hat{e}(h,h)^{1/a}$, then

$$W' = W^{y_0 z_0} = \hat{e}(h,h)^{1/a \cdot y_0 \cdot z_0} = \hat{e}(g,g)^{1/a^5 \cdot y_0 z_0} = \hat{e}(g,g)^{1/a \cdot y_0/a^2 \cdot z_0/a^2} = \hat{e}(g,g)^{xyz}.$$

If W is a random element, so is W'. This proves the lemma. ▮

E Known IBE constructions

E.1 The IBE scheme from Waters [45]

Waters IBE scheme with identity space $\mathsf{IDSp} = \{0,1\}^n$ and message space $\mathsf{MsgSp} = \mathbb{G}_T$ is depicted in Figure 4.

E.2 The IBE scheme from Boneh/Franklin [10]

The Boneh/Franklin fullident IBE scheme with identity space $\mathsf{IDSp} = \{0,1\}^n$ and $\mathsf{MsgSp} = \{0,1\}^m$ is depicted in Figure 5. It needs four random oracles $\mathsf{H}_1 : \{0,1\}^n \to \mathbb{G}_2, \mathsf{H}_2 : \mathbb{G}_T \to \{0,1\}^{2k}, \mathsf{H}_3 : \{0,1\}^k \times \mathsf{MsgSp} \to \mathbb{Z}_p^*$, and $\mathsf{H}_4 : \{0,1\}^k \to \{0,1\}^m$.

$\mathsf{IBEkg}(1^k)$	$\mathsf{IBEkeyder}(sk, id)$
$\alpha \xleftarrow{\$} \mathbb{G}^*$; $z \leftarrow \hat{e}(g, \alpha)$	$s \xleftarrow{\$} \mathbb{Z}_p$
$\mathsf{H} \xleftarrow{\$} \mathsf{HGen}(m)$	$sk[id] \leftarrow (\alpha \cdot \mathsf{H}(id)^s, g^s)$
$pk \leftarrow (\mathsf{H}, z)$; $sk \leftarrow \alpha$	Return $sk[id]$
Return (pk, sk)	
$\mathsf{IBEenc}(pk, id, M)$	$\mathsf{Decaps}(sk[id], C)$
$r \xleftarrow{\$} \mathbb{Z}_p^*$; $c_1 \leftarrow g^r$; $c_2 \leftarrow \mathsf{H}(id)^r$	Parse C as (c_1, c_2, e)
$e \leftarrow M \cdot z^r$	Parse $sk[id]$ as (d_1, d_2)
$C \leftarrow (c_1, c_2, e) \in \mathbb{G}^2 \times \mathbb{G}_T$	Return $M \leftarrow e \cdot \hat{e}(c_2, d_2)/\hat{e}(c_1, d_1)$
Return C	

Figure 4: CPA-secure IBE from Waters.

$\mathsf{IBEkg}(1^k)$	
$\alpha \xleftarrow{\$} \mathbb{G}^*$	$\mathsf{IBEkeyder}(sk, id)$
Pick random oracles $\mathsf{H}_1, \mathsf{H}_2, \mathsf{H}_3, \mathsf{H}_4$	$sk[id] \leftarrow \mathsf{H}_1(id)^\alpha \in \mathbb{G}_2$
$pk \leftarrow (\mathsf{H}_1, \mathsf{H}_2, \mathsf{H}_3, \mathsf{H}_4)$; $sk \leftarrow \alpha$	Return $sk[id]$
Return (pk, sk)	
$\mathsf{IBEenc}(pk, id, M)$	$\mathsf{IBEdec}(sk[id], C)$
$\sigma \xleftarrow{\$} \{0,1\}^k$; $r \leftarrow \mathsf{H}_3(\sigma, M)$	Parse C as (c_1, c_2, e)
$c_1 \leftarrow g^r$; $c_2 \leftarrow \sigma \oplus \mathsf{H}_2(\hat{e}(g, \mathsf{H}_1(id))^r)$	$c_2 \leftarrow c_2 \oplus \mathsf{H}_2(\hat{e}(c_1, sk[id]))$
$e \leftarrow \mathsf{H}_4(\sigma) \oplus M$	$M \leftarrow e \oplus \mathsf{H}(\sigma)$
$C \leftarrow (c_1, c_2, e) \in \mathbb{G} \times \{0,1\}^{2k} \times \{0,1\}^m$	$r \leftarrow \mathsf{H}_3(\sigma, M)$; if $g^r \neq c_1$ then `reject`
Return C	Else return M

Figure 5: CCA-secure fullident IBE scheme from Boneh/Franklin.