Efficient Selective Identity-Based Encryption Without Random Oracles *

Dan Boneh † Xavier Boyen ‡

March 21, 2011

Abstract

We construct two efficient Identity-Based Encryption (IBE) systems that admit selective-identity security reductions without random oracles in groups equipped with a bilinear map. Selective-identity secure IBE is a slightly weaker security model than the standard security model for IBE. In this model the adversary must commit ahead of time to the identity that it intends to attack, whereas in an adaptive-identity attack the adversary is allowed to choose this identity adaptively. Our first system—BB_1—is based on the well studied decisional bilinear Diffie-Hellman assumption, and extends naturally to systems with hierarchical identities, or HIBE. Our second system—BB_2—is based on a stronger assumption which we call the Bilinear Diffie-Hellman Inversion assumption and provides another approach to building IBE systems.

Our first system, BB_1, is very versatile and well suited for practical applications: the basic hierarchical construction can be efficiently secured against chosen-ciphertext attacks, and further extended to support efficient non-interactive threshold decryption, among others, all without using random oracles. Both systems, BB_1 and BB_2, can be modified generically to provide "full" IBE security (i.e., against adaptive-identity attacks), either using random oracles, or in the standard model at the expense of a non-polynomial but easy-to-compensate security reduction.

Keywords: identity-based encryption, selective-ID security, adaptive-ID security, pairing-based cryptography, asymmetric bilinear maps, BDH assumption, BDHI assumption, security proofs.

1 Introduction

Identity-Based Encryption (IBE) provides a public-key encryption mechanism where a public key is an arbitrary string such as an email address or a telephone number. The corresponding private key can only be generated by a Private-Key Generator (PKG) who has knowledge of a master secret. In an IBE system, users authenticate themselves to the PKG and obtain private keys corresponding to their identities. The identity-based encryption concept was first proposed two decades ago [61] and several approaches were subsequently suggested in a few precursor papers [67, 68, 49]. It is only a few years ago, however, that a formal security model and a practical implementation were proposed. Boneh and Franklin [13, 14] define a security model for identity-based encryption and give a construction based on the Bilinear Diffie-Hellman (BDH) problem. Cocks [28] describes another

*An extended abstract entitled "Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles" appears in Eurocrypt 2004 [6].

†Stanford University, USA — *dabo@cs.stanford.edu*

‡Université de Liège, Belgium — *xb@boyen.org*

construction using quadratic residues modulo a composite (see also [15, 30]). Gentry et al. [39] give a construction using lattices. The security of all these systems requires cryptographic hash functions that are modeled as random oracles; i.e., these systems are only proven secure (under non-interactive assumptions) in the random-oracle model [3]. A natural question is to devise a secure IBE system without random oracles.

In the Boneh-Franklin security model the adversary can issue both adaptive chosen-ciphertext queries and adaptive chosen-identity queries (i.e., the adversary can request the private key for identities of its choice). Eventually, the adversary adaptively chooses the identity it wishes to attack and asks for a semantic security challenge for this identity. Canetti et al. [22, 23] proposed a slightly weaker security model, called selective-identity IBE. In this model the adversary must commit ahead of time (non-adaptively) to the identity it intends to attack. The adversary can still issue adaptive chosen-ciphertext and adaptive chosen-identity queries. Canetti et al. constructed a provably secure IBE in this weaker model without the random-oracle model. However, their construction views identities as bit strings, causing their system to require a bilinear map computation for every bit in the identity.

We construct two efficient IBE systems that are provably secure in the selective-identity sense without the random oracle model. In both systems, encryption requires no bilinear map computation and decryption requires at most two. Our first construction, which has become known as BB_1, is based on the Decision Bilinear Diffie-Hellman (Decision-BDH) assumption. This construction extends to give an efficient selective-identity secure Hierarchical IBE (HIBE) without random oracles. Hierarchical IBE was defined in [42] and the first construction in the random oracle model was given by Gentry and Silverberg [40]. Our efficient HIBE construction is related to the Gentry-Silverberg system, but we are able to prove security without using random oracles. Our second IBE construction, referred to as BB_2, has comparable efficiency, but is based on a non-standard assumption we call Decision Bilinear Diffie-Hellman Inversion (Decision-BDHI). Roughly speaking, the assumption says that no efficient algorithm can distinguish $e(g,g)^{1/x}$ from random, given $g, g^x, g^{(x^2)}, \ldots, g^{(x^q)}$ for some specified value of the parameter q.

Canetti et al. [23, 12] showed that any selective-identity, chosen-plaintext IBE gives a chosen-ciphertext secure (CCA2) public key system. Consequently, both our IBE systems give efficient CCA2-secure public key systems without random oracles. Performance of both CCA2-secure systems is close to the performance of the Cramer-Shoup system [29] which is based on Decision Diffie-Hellman. Boneh and Katz [16, 12] gave a more efficient transformation from IBE to CCA2-security, which when applied to our systems, gives CCA2-secure systems where encryption time is as efficient as the Kurosawa-Desmedt [46] fast variant of Cramer-Shoup. Boyen, Mei, and Waters [20] constructed an even more efficient CCA2-secure system that non-generically exploits the structure of our IBE constructions.

In subsequent work [7] we extended our first IBE system (BB_1) to obtain the first "full" or adaptive-ID secure IBE construction with a polynomial-time security reduction in the standard model, albeit with a construction that was not practical. An important result of Waters [69] enhances the BB_1 framework into a practical fully secure IBE construction. The BB_2 system can be made fully secure in a similar way, as noticed independently by Shen and Kiltz [62, 45]. In Section 7 we observe that any selective-ID secure IBE can be turned into an adaptive-ID secure IBE in the standard model, generically, but at the cost of a non-polynomial-time security reduction. One can however compensate for the inefficient reduction by increasing the size of the groups used by the system by a constant ratio. The transformation applied to BB_1 and BB_2 results in two simple and fully secure IBE systems without random oracles.

For completeness, we also discuss a few simple ways in which random oracles can be used to

strengthen our constructions with respect to both adaptive-identity and chosen-ciphertext security. When implemented intelligently using (asymmetric) pairings on curves, both our constructions result in efficient adaptive-ID CCA2-secure IBE systems in the random-oracle model. The BB_1 scheme in particular (being based on the decision-BDH assumption) leads to a very practical IBE system.

2 Preliminaries

Before presenting our results we briefly review the definition of security for an IBE system. We also review the definition of groups equipped with a bilinear map.

2.1 Selective Identity Secure IBE and HIBE Systems

Recall that an Identity-Based Encryption system (IBE) consists of four algorithms [61, 13]: *Setup*, *Extract*, *Encrypt*, *Decrypt*. The *Setup* algorithm generates system parameters, denoted by *params*, and a master key *mk*. The *Extract* algorithm uses the master key to extract a private key corresponding to a given identity. The encryption algorithm encrypts messages for a given identity (using the system parameters) and the decryption algorithm decrypts ciphertexts using the private key.

In a Hierarchical IBE [42, 40], identities are vectors, and there is a fifth algorithm called *Derive*. A vector of dimension ℓ represents an identity at depth ℓ and a private key for it can be generated using algorithm *Extract*, which requires the master key. Algorithm *Derive* is used to delegate keys along the hierarchy. The algorithm takes as input an identity $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_\ell)$ at depth ℓ and the private key $d_{\mathsf{ID}|\ell-1}$ of the parent identity $\mathsf{ID}_{|\ell-1} = (\mathrm{I}_1, \ldots, \mathrm{I}_{\ell-1})$ at depth $\ell - 1 > 0$. It outputs the private key d_{ID} for identity ID. For convenience, we sometimes refer to the master key as the private key at depth 0. We note that an IBE system is an HIBE where all identities are at depth 1. The *Setup* algorithm in an HIBE scheme takes the maximum depth of the hierarchy as input.

Delegation history independence. In all our HIBE constructions, algorithm *Derive* outputs the same distribution of private keys as algorithm *Extract*. This property, called *delegation history independence*, ensures that the private key of an identity ID at a certain depth u reveals no information about the delegation process used to derive that key. For example, for an identity ID at level $u > 3$, a private key derived for ID from a level-1 key is sampled from the same distribution as a private key for ID derived from a level-3 key. This property greatly simplifies the definition of HIBE security. We define this property more precisely and discuss this issue further in Section 2.1.1.

Selective and Adaptive ID Security. The standard IBE security model of [13, 14] defines the indistinguishability of ciphertexts under an adaptive chosen-ciphertext and chosen-identity attack (IND-ID-CCA2). An adaptive chosen-identity attack means that the adversary is allowed to narrow in adaptively to the identity it wishes to target (i.e., the public key on which it will be challenged). A weaker notion of IBE security given by Canetti, Halevi, and Katz [22, 23] forces the adversary to announce ahead of time the public key it will target, which is known as a selective-identity attack (IND-sID-CCA2). We refer to such a system as a selective identity, chosen ciphertext secure IBE.

Security Game. We define IBE and HIBE security under a selective-identity attack (for a hierarchy of maximum depth ℓ) using the following game between a challenger and an adversary:

Init: The adversary outputs an identity $\mathsf{ID}^* = (\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*)$ where it wishes to be challenged.

Setup: The challenger runs the *Setup* algorithm giving it the maximum depth ℓ as input (where $\ell = 1$ for IBE). It gives the adversary the resulting system parameters *params*. It keeps the master key *mk* to itself.

Phase 1: The adversary issues queries $q_1, \ldots, q_m$ where the i-th query q_i is one of:

- Private-key extraction query on ID_i, where $\mathsf{ID}_i = (\mathrm{I}_1, \ldots, \mathrm{I}_u)$ for some $1 \leq u \leq \ell$. We require that ID_i is not a prefix of ID^*, (i.e., it is not the case that $u \leq k$ and $\mathrm{I}_i = \mathrm{I}_i^*$ for all $i = 1, \ldots, u$). The challenger responds by running algorithm *Extract* to obtain a private key d_i corresponding to the public key ID_i. It sends d_i to the adversary.
- Decryption query on a ciphertext C_i for an identity ID_i (which may be equal to ID^* or a prefix of ID^*). To respond, the challenger runs algorithm *Extract* to extract a private key d corresponding to ID_i, and then runs algorithm *Decrypt* to decrypt the ciphertext C_i using the private key d. It sends the result to the adversary.

All queries may be made adaptively, that is, the adversary may ask q_i with knowledge of the challenger's responses to $q_1, \ldots, q_{i-1}$.

Challenge: Once the adversary decides that Phase 1 is over it outputs two equal length plaintexts $M_0, M_1 \in \mathcal{M}$ on which it wishes to be challenged. The challenger picks a random bit $b \in \{0, 1\}$ and sets the challenge ciphertext to $C = \mathit{Encrypt}(\mathit{params}, \mathsf{ID}^*, M_b)$. It sends C as the challenge to the adversary.

Phase 2: The adversary issues additional adaptive queries $q_{m+1}, \ldots, q_n$ where q_i is one of:

- Private-key extraction query on ID_i, where ID_i is not a prefix of ID^*. The challenger responds as in Phase 1.
- Decryption query on C_i for ID_i, where $C_i \neq C$ when $\mathsf{ID}_i = \mathsf{ID}^*$ or some prefix of ID^*. The challenger responds as in Phase 1.

Guess: Finally, the adversary outputs a guess $b' \in \{0, 1\}$. The adversary wins if $b = b'$.

We refer to such an adversary $\mathcal{A}$ as an IND-sID-CCA2 adversary. We define the advantage of the adversary $\mathcal{A}$ in attacking an HIBE scheme $\mathcal{E} = (\mathit{Setup}, \mathit{Extract}, \mathit{Derive}, \mathit{Encrypt}, \mathit{Decrypt})$, or an IBE scheme $\mathcal{E} = (\mathit{Setup}, \mathit{Extract}, \mathit{Encrypt}, \mathit{Decrypt})$, as

$$\mathrm{Adv}_{\mathcal{E},\mathcal{A}} = \left|\Pr[b = b'] - 1/2\right|$$

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

Definition 2.1. We say that an IBE or HIBE system $\mathcal{E}$ is $(t, q_{\mathsf{ID}}, q_{\mathrm{C}}, \epsilon)$-selective-identity, adaptive chosen-ciphertext secure if for any IND-sID-CCA2 adversary $\mathcal{A}$ that runs in time t, makes at most q_{ID} chosen private-key queries, and at most q_{C} chosen decryption queries, we have that $\mathrm{Adv}_{\mathcal{E},\mathcal{A}} < \epsilon$. We abbreviate this by saying that $\mathcal{E}$ is $(t, q_{\mathsf{ID}}, q_{\mathrm{C}}, \epsilon)$-IND-sID-CCA2 secure.

It is also customary to define the weaker notion of semantic security, or security under chosen plaintext attack (CPA), where the adversary is forbidden from making decryption queries. The adversary is still allowed to issue adaptive private-key queries.

Definition 2.2. We say that an IBE or HIBE system $\mathcal{E}$ is $(t, q_{\mathsf{ID}}, \epsilon)$-selective identity, chosen-plaintext secure if $\mathcal{E}$ is $(t, q_{\mathsf{ID}}, 0, \epsilon)$-selective-identity, adaptive chosen-ciphertext secure. For conciseness, we say that $\mathcal{E}$ is $(t, q_{\mathsf{ID}}, \epsilon)$-IND-sID-CPA secure.

Finally, we define the adaptive-identity counterparts to the above notions by removing the Init phase from the attack game, and allowing the adversary to wait until the Challenge phase to announce the identity ID^* it wishes to attack. The adversary is allowed to make arbitrary private-key queries in Phase 1 and then choose an arbitrary target ID^*. The only restriction is that he did not issue a private-key query for ID^* or a prefix of ID^* during phase 1. The resulting security notions are defined using the modified game as in Definitions 2.1 and 2.2, and are denoted IND-ID-CCA2 and IND-ID-CPA respectively.

In the sequel, our main focus will be to construct (H)IBE systems in the selective security model. We briefly discuss adaptive-ID security in Section 7. We also come back to chosen-ciphertext security in Section 8 where we strengthen our systems against active attacks.

2.1.1 Security with respect to HIBE Key Delegation

Our description of HIBE security follows the definitions in [42, 40] and the selective variants in [22, 23]. In this attack game, all private-key queries are answered using the root master secret mk by running $\mathit{Extract}(mk, \cdot)$. In reality, however, HIBE supports delegation where the private-key for identity $\mathsf{ID} = (I_1, \ldots, I_u)$ can be derived from the private key of any parent identity such as $(I_1, \ldots, I_{u-1})$. Note that a key derived for ID from the root key mk may be sampled from a different distribution than a key derived for ID from a non-root key.

The security model presented in the previous section assumes that the system has *delegation history independence* which means that all private keys, whether derived from the root or derived by delegation, are sampled from the same distribution. Hence, in the security game it suffices to respond to all private-key queries using the root master secret. All the HIBE constructions in this paper have this property. More precisely, delegation history independence is defined as follows.

Definition 2.3. We say that an HIBE system has *delegation history independence* if for all outputs of the *Setup* algorithm and all tuples of identities $(\mathsf{ID}_1, \ldots, \mathsf{ID}_q)$ the distribution of private keys generated by *Extract* for these identities is the same as the distribution produced by *Derive* for the same identities, no matter what valid parent keys are given to *Derive* as input.

Shi and Waters [63] define a relaxed HIBE security model which does not require delegation history independence. The security model is more complex since the adversary gets to specify the exact delegation path used in answering every private key query, as well as dependencies between private keys. Removing the requirement for delegation history independence enables us to slightly simplify algorithm *Derive* in our HIBE system. We revisit this issue in Section 4.4.

2.2 Bilinear Groups and Maps

We briefly review the necessary facts about bilinear maps or pairings and the groups over which they are defined [44, 50]. For a prime p, we denote the finite field of order p by $\mathbb{Z}_p$. Wet let $\mathbb{Z}_p^*$ denote the multiplicative group of order $p-1$ consisting of the elements in $\mathbb{Z}_p \setminus \{0\}$.

Let $\mathbb{G}$ and $\hat{\mathbb{G}}$ be two (possibly distinct, but isomorphic) cyclic groups of large prime order p. Let $g \in \mathbb{G}$ and $\hat{g} \in \hat{\mathbb{G}}$ be respective generators of $\mathbb{G}$ and $\hat{\mathbb{G}}$. Let $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_t$ be a function that maps pairs of elements in $(\mathbb{G}, \hat{\mathbb{G}})$ to elements of a group $\mathbb{G}_t$, where $\mathbb{G}_t$ has order p. The group operations in $\mathbb{G}$, $\hat{\mathbb{G}}$, and $\mathbb{G}_t$ are written multiplicatively, with identity elements denoted by 1. Further assume that:

- the map e is efficiently computable, and so are the group operations in $\mathbb{G}$, $\hat{\mathbb{G}}$, and $\mathbb{G}_t$;
- the map e is non-degenerate, in the sense that $e(g, \hat{g}) \neq 1$;

– the map e is bilinear, meaning that $\forall u \in \mathbb{G}, \forall v \in \hat{\mathbb{G}}, \forall a, b \in \mathbb{Z}, e(u^a, v^b) = e(u, v)^{ab}$.

We say that $(\mathbb{G}, \hat{\mathbb{G}})$ forms a bilinear group pair, and that e is a bilinear map from $(\mathbb{G}, \hat{\mathbb{G}})$ into $\mathbb{G}_t$. Such bilinear maps are often called bilinear pairings, or pairings for short.

Symmetric versus Asymmetric Pairings. The above definition is general in the sense that no special constraint is placed on $\mathbb{G}$ and $\hat{\mathbb{G}}$ (other than having prime order p). In a number of applications, however, it is important [59] to require that $\mathbb{G}$ and $\hat{\mathbb{G}}$ be the same group (i.e. $\mathbb{G} = \hat{\mathbb{G}}$); this leads to the notion of "symmetric" bilinear pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$ where both left and right inputs to the pairing live in the same group. In the general case, $\mathbb{G}$ and $\hat{\mathbb{G}}$ are distinct groups and the bilinear group pair $(\mathbb{G}, \hat{\mathbb{G}})$ is called "asymmetric".

If a cryptographic scheme does not explicitly require the symmetry, it is usually advantageous to use asymmetric bilinear groups. The advantages include more compact representations of group elements, a far broader choice of elliptic curve implementations [32], and possibly even additional security properties [35]. Asymmetric pairings have been used in signature schemes [17, 8, 11] to minimize the signature size, but, by and large, the designers of encryption schemes have tended to prefer symmetric pairings, favoring simplicity over generality. In this paper, we opt for the more general asymmetric formulation. (To emulate the symmetric notation, the reader may disregard all hats $\hat{}$ from the symbols.)

For concreteness, we shall suppose that the elements of $\mathbb{G}$ can be represented more compactly than those of $\hat{\mathbb{G}}$, as happens frequently with asymmetric pairings.

Computable and Invertible Isomorphisms. In some examples of bilinear group pairs $(\mathbb{G}, \hat{\mathbb{G}})$ there is an isomorphism $\phi : \hat{\mathbb{G}} \to \mathbb{G}$ that is easy to compute and easy to invert. For example, when $\mathbb{G} = \hat{\mathbb{G}}$ the isomorphism ϕ is simply the identity function. In other bilinear group pairs the isomorphism may be easy to compute, but hard to invert. In other cases still, both ϕ and ϕ^{-1} may be hard to compute.

All the constructions in this paper are indifferent to this issue; it makes no difference whether the isomorphism ϕ is easily computable or not. For this reason we will not refer to ϕ in our algorithms and proofs, thus enabling us to make use of all known bilinear group constructions.

Implementation Details. We refer the reader to [35] for a more detailed description of the various types of pairings, and to [32, 14, 52, 2, 31, 56] for the construction of suitable curves for their implementations. General algorithms for curve arithmetic including the pairing may be found in [50, 5].

3 Complexity Assumptions

We review the standard Bilinear Diffie-Hellman (BDH) assumption, and define the Bilinear Diffie-Hellman Inversion (BDHI) assumption. Since these assumptions were originally proposed in the symmetric setting, we restate them here under a natural generalization for asymmetric pairings.

3.1 Bilinear Diffie-Hellman Assumption

The BDH problem for a *symmetric* pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$ is stated as follows [44, 59, 13]:

> Given a tuple $(g, g^a, g^b, g^c) \in \mathbb{G}^4$ as input, output $e(g, g)^{abc} \in \mathbb{G}_t$.

We generalize the BDH problem to asymmetric bilinear groups, where $\mathbb{G}$ need not be the same as $\hat{\mathbb{G}}$, by giving the adversary $(g, g^a, g^c, \hat{g}, \hat{g}^a, \hat{g}^b) \in \mathbb{G}^3 \times \hat{\mathbb{G}}^3$.

Computational BDH. We say that an algorithm $\mathcal{A}$ has advantage $\mathsf{Adv}_{\mathcal{A}}^{\mathrm{BDH}} = \epsilon$ in solving the computational BDH problem in $(\mathbb{G}, \hat{\mathbb{G}})$ if

$$\Pr\left[\mathcal{A}\big(g\,, g^a, g^c, \hat{g}\,, \hat{g}^a, \hat{g}^b\big) = e(g, \hat{g})^{abc}\right] \geq \epsilon$$

where the probability is over the random choice of exponents a, b, c in $\mathbb{Z}_p$, the random choice of generators g of $\mathbb{G}$ and $\hat{g}$ of $\hat{\mathbb{G}}$, and the random bits used by $\mathcal{A}$. For symmetric pairings (i.e. when $\mathbb{G} = \hat{\mathbb{G}}$) we can simplify the assumption by choosing g at random in $\mathbb{G}$ and setting $\hat{g} = g$. Then $\mathcal{A}$ is only given (g, g^a, g^b, g^c) as input and is asked to compute $e(g, g)^{abc}$. As we shall see, our security proofs are oblivious to how the generators g and $\hat{g}$ are chosen in the assumption.

Decisional BDH. We similarly say that an algorithm $\mathcal{B}$ that outputs a bit $\gamma \in \{0, 1\}$ has advantage $\mathsf{Adv}_{\mathcal{A}}^{\text{D-BDH}} = \epsilon$ in solving the *Decision*-BDH problem in $(\mathbb{G}, \hat{\mathbb{G}})$ if

$$\left|\Pr\left[\mathcal{B}\big(g\,, g^a, g^c, \hat{g}\,, \hat{g}^a, \hat{g}^b, e(g, \hat{g})^{abc}\big) = 0\right] - \Pr\left[\mathcal{B}\big(g\,, g^a, g^c, \hat{g}\,, \hat{g}^a, \hat{g}^b, T\big) = 0\right]\right| \geq \epsilon$$

where the probability is over the random choice of generators g of $\mathbb{G}$ and $\hat{g}$ of $\hat{\mathbb{G}}$, the random choice of exponents a, b, c in $\mathbb{Z}_p$, the random choice of $T \in \mathbb{G}_t$, and the random bits used by $\mathcal{B}$.

Hash BDH. We can also define a weaker version of the decisional assumption with the help of a hash function. Let $\mathcal{H}$ be a family of hash functions $\mathcal{H} = \{H : \mathbb{G}_t \to \{0,1\}^m\}$ for some $m \in \mathbb{Z}_{>0}$. We say that an algorithm $\mathcal{B}$ that outputs a bit $\gamma \in \{0, 1\}$ has advantage $\mathsf{Adv}_{\mathcal{A}}^{\mathrm{HashBDH}} = \epsilon$ in solving the *Hash*-BDH problem in $(\mathbb{G}, \hat{\mathbb{G}})$ with respect to $\mathcal{H}$ if

$$\left|\Pr\left[\mathcal{B}\big(H, g\,, g^a, g^c, \hat{g}\,, \hat{g}^a, \hat{g}^b, H(e(g, \hat{g})^{abc})\big) = 0\right] - \Pr\left[\mathcal{B}\big(H, g\,, g^a, g^c, \hat{g}\,, \hat{g}^a, \hat{g}^b, T\big) = 0\right]\right| \geq \epsilon$$

where the probability is over the random choice of generators g of $\mathbb{G}$ and $\hat{g}$ of $\hat{\mathbb{G}}$, the random choice of exponents a, b, c in $\mathbb{Z}_p$, the random selection of the function H from the family $\mathcal{H}$, the random choice of $T \in \{0,1\}^m$, and the random bits consumed by $\mathcal{B}$.

In the decisional BDH definition, we refer to the distribution (over $\mathbb{G}^3 \times \hat{\mathbb{G}}^3 \times \mathbb{G}_t$) of the 7-tuple in the *true* instance (on the left) as $\mathcal{P}_{BDH}$, and in the *false* instance (on the right) as $\mathcal{R}_{BDH}$.

Definition 3.1. We say that the (t, ϵ)-(Decision-)BDH assumption holds in $(\mathbb{G}, \hat{\mathbb{G}})$ if no t-time algorithm has advantage at least ϵ in solving the (Decision-)BDH problem in $(\mathbb{G}, \hat{\mathbb{G}})$.

Similarly, we say that the $(t, \epsilon, \mathcal{H})$-Hash-BDH assumption holds in $(\mathbb{G}, \hat{\mathbb{G}})$ if no t-time algorithm has advantage at least ϵ in solving the Hash-BDH problem in $(\mathbb{G}, \hat{\mathbb{G}})$ with respect to $\mathcal{H}$.

Occasionally we omit the parameters and talk of the BDH, Decision-BDH, and Hash-BDH assumptions. We remark that the Decision-BDH assumption in $(\mathbb{G}, \hat{\mathbb{G}})$ implies that the regular Decision Diffie-Hellman (DDH) assumption holds in $\mathbb{G}_t$.

For certain hash families $\mathcal{H}$, the Decision-BDH assumption implies the Hash-BDH assumption, and hence Hash-BDH is a weaker assumption. For example, a standard application of the left-over hash lemma [43] shows that with an appropriate choice of m and universal hash family $\mathcal{H}$, the Decision-BDH assumption implies the Hash-BDH assumption. For some groups $\mathbb{G}$ and $\hat{\mathbb{G}}$ one can prove this implication using a fixed hash function without resorting to the left-over hash lemma [27]. Either way we observe that Hash-BDH is a weaker (more precisely, no stronger) assumption than Decision-BDH.

3.2 Bilinear Diffie-Hellman Inversion Assumption

The q-BDHI problem for a symmetric pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$ is stated as follows [6]:

> Given a $(q+1)$-tuple $(g, g^x, g^{(x^2)}, \ldots, g^{(x^q)}) \in (\mathbb{G})^{q+1}$ as input, output $e(g,g)^{1/x} \in \mathbb{G}_t$.

We generalize it to asymmetric bilinear groups and define the complexity assumption by providing a sequence of powers of the generator in one of the groups (arbitrarily taken to be $\hat{\mathbb{G}}$).

Computational q-BDHI. An algorithm $\mathcal{A}$ has advantage $\mathsf{Adv}^{\text{BDHI}}_{q,\mathcal{A}} = \epsilon$ in solving q-BDHI in $(\mathbb{G}, \hat{\mathbb{G}})$ if

$$\Pr\left[\mathcal{A}\big(g, g^x, \hat{g}, \hat{g}^x, \ldots, \hat{g}^{(x^q)}\big) = e(g,\hat{g})^{1/x}\right] \geq \epsilon$$

where the probability is over the random choice of generators g of $\mathbb{G}$ and $\hat{g}$ of $\hat{\mathbb{G}}$, the random choice of $x \in \mathbb{Z}_p^*$, and the random bits used by $\mathcal{A}$. For symmetric pairings (i.e. when $\mathbb{G} = \hat{\mathbb{G}}$) we can simplify the assumption by choosing g at random in $\mathbb{G}$ and setting $\hat{g} = g$. Then $\mathcal{A}$ is only given $g, g^x, g^{(x^2)}, \ldots, g^{(x^q)}$ as in [6].

Decisional q-BDHI. Similarly, we say that an algorithm $\mathcal{B}$ that outputs a bit $\gamma \in \{0,1\}$ has advantage $\mathsf{Adv}^{\text{D-BDHI}}_{q,\mathcal{A}} = \epsilon$ in solving the q-Decision-BDHI problem in $(\mathbb{G}, \hat{\mathbb{G}})$ if

$$\left|\Pr\left[\mathcal{B}\left(\begin{matrix} g, g^x, \\ \hat{g}, \hat{g}^x, \ldots, \hat{g}^{(x^q)}, e(g,\hat{g})^{1/x}\end{matrix}\right) = 0\right] - \Pr\left[\mathcal{B}\left(\begin{matrix} g, g^x, \\ \hat{g}, \hat{g}^x, \ldots, \hat{g}^{(x^q)}, T\end{matrix}\right) = 0\right]\right| \geq \epsilon$$

where the probability is over the random choice of generators g of $\mathbb{G}$ and $\hat{g}$ of $\hat{\mathbb{G}}$, the random choice of $x \in \mathbb{Z}_p^*$, the random choice of $T \in \mathbb{G}_t$, and the random bits used by $\mathcal{B}$.

Hash q-BDHI. As before, we can relax the decisional assumption with the help of a hash function family $\mathcal{H} = \{H : \mathbb{G}_t \to \{0,1\}^m\}$ for some $m \in \mathbb{Z}_{>0}$. We say that an algorithm $\mathcal{B}$ that outputs a bit $\gamma \in \{0,1\}$ has advantage $\mathsf{Adv}^{\text{HashBDHI}}_{q,\mathcal{A}} = \epsilon$ in solving the q-Hash-BDHI problem in $(\mathbb{G}, \hat{\mathbb{G}})$ with respect to $\mathcal{H}$ if

$$\left|\Pr\left[\mathcal{B}\left(\begin{matrix} H, g, g^x, \\ \hat{g}, \hat{g}^x, \ldots, \hat{g}^{(x^q)}, H(e(g,\hat{g})^{1/x})\end{matrix}\right) = 0\right] - \Pr\left[\mathcal{B}\left(\begin{matrix} H, g, g^x, \\ \hat{g}, \hat{g}^x, \ldots, \hat{g}^{(x^q)}, T\end{matrix}\right) = 0\right]\right| \geq \epsilon$$

where the probability is over the random choice of generators g of $\mathbb{G}$ and $\hat{g}$ of $\hat{\mathbb{G}}$, the random choice of $x \in \mathbb{Z}_p^*$, the random choice of $T \in \{0,1\}^m$, the random choice of function $H \in \mathcal{H}$, and the random bits consumed by $\mathcal{B}$.

In the decisional definition, we refer to the $(q+4)$-tuple distribution (over $\mathbb{G}^2 \times \hat{\mathbb{G}}^{q+1} \times \mathbb{G}_t$) in the *true* instance (on the left) as $\mathcal{P}_{BDHI}$, and in the *false* instance (on the right) as $\mathcal{R}_{BDHI}$.

Definition 3.2. We say that the (t, q, ϵ)-(Decision-)BDHI assumption holds in $(\mathbb{G}, \hat{\mathbb{G}})$ if no t-time algorithm has advantage at least ϵ in solving the q-(Decision-)BDHI problem in $(\mathbb{G}, \hat{\mathbb{G}})$.

We say that the $(t, q, \epsilon, \mathcal{H})$-Hash-BDHI assumption holds in $(\mathbb{G}, \hat{\mathbb{G}})$ if no t-time algorithm has advantage at least ϵ in solving the q-Hash-BDHI problem in $(\mathbb{G}, \hat{\mathbb{G}})$ with respect to $\mathcal{H}$.

Occasionally we drop the t and ϵ and refer to the q-BDHI and q-Decision-BDHI assumptions. It is not known whether the q-BDHI assumption, for $q > 1$, is equivalent to BDH. A closely related assumption called weak Diffie-Hellman was previously used in [51]. (See also Appendix A.)

3.3 Asymptotic Formulation of the Assumptions

For completeness, we give an asymptotic formulation of the BDH and BDHI assumptions. Since here we are interested in asymptotic behavior, we need a bilinear group generation algorithm $\mathcal{G}$.

Definition 3.3. A bilinear group generator $\mathcal{G}$ is a Probabilistic Polynomial Time (PPT) algorithm that, on input 1^λ, outputs the description of groups $\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_t$ and a bilinear map $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_t$, so that $(\mathbb{G}, \hat{\mathbb{G}})$ form a bilinear group pair.

We now define the asymptotic BDH assumptions. In this setting, the various quantities denoted by ϵ in Section 3.1 become functions of the security parameter λ.

Definition 3.4. Let $\mathcal{G}$ be a bilinear group generator. We say that the BDH, Decision BDH, and Hash BDH assumptions hold for $\mathcal{G}$ if, for every PPT algorithm $\mathcal{A}$, the respective functions $\mathsf{Adv}_{\mathcal{A}}^{\text{BDH}}(\lambda)$, $\mathsf{Adv}_{\mathcal{A}}^{\text{D-BDH}}(\lambda)$, and $\mathsf{Adv}_{\mathcal{A}}^{\text{HashBDH}}(\lambda)$ are negligible functions of λ.

We similarly define the asymptotic BDHI assumptions. As in the previous case, the various ϵ from Section 3.2 become functions of λ. Notice that this includes the parameter q.

Definition 3.5. Let $\mathcal{G}$ be a bilinear group generator. We say that the BDHI, Decision-BDHI, and Hash-BDHI assumptions hold for $\mathcal{G}$ if, for every PPT algorithm $\mathcal{A}$, and every polynomial $q \in \mathbb{Z}[X]$, the respective functions $\mathsf{Adv}_{q(\lambda),\mathcal{A}}^{\text{BDHI}}(\lambda)$, $\mathsf{Adv}_{q(\lambda),\mathcal{A}}^{\text{D-BDHI}}(\lambda)$, and $\mathsf{Adv}_{q(\lambda),\mathcal{A}}^{\text{HashBDHI}}(\lambda)$ are negligible functions of λ.

It is easy to interpret our results asymptotically using Definition 3.3 and the asymptotic assumptions.

3.4 A Discussion of our Asymmetric Formulation

Compared with the original, symmetric-pairing formulations of BDH [13] and BDHI [6], the definitions we give here involve problem instances containing two more elements. This is because some of the elements must be given explicitly in each group $\mathbb{G}$ and $\hat{\mathbb{G}}$, in accordance with our choice not to rely on the isomorphisms ϕ and ϕ^{-1} to be or not to be efficiency computable.

The BDH problem in symmetric bilinear groups was to compute (or recognize, or hash) $e(g,g)^{abc}$ given g, g^a, g^b, g^c. In asymmetric groups, we seek to find $e(g,\hat{g})^{abc}$ given g, g^a, g^c, and $\hat{g}, \hat{g}^a, \hat{g}^b$. This is not the weakest possible generalization, since the a-th power is given in both groups. This repetition (which is unnecessary for symmetric pairings since $g = \hat{g}$) is often needed in proofs using asymmetric pairings. We note that one can easily verify that an input tuple is well formed by using the pairing to check that $(g, \hat{g}, g^a, \hat{g}^a)$ is a valid Diffie-Hellman tuple.

4 BB_1 : Efficient IBE/HIBE From BDH Without Random Oracles

We construct an efficient HIBE system that is selective-identity chosen-plaintext secure without random oracles based on the Decision-BDH assumption. In particular, this implies an efficient selective-identity chosen-ciphertext secure HIBE based on Decision-BDH without random oracles, from one of the transformations given in [12, 20].

The challenge in building secure IBE and HIBE systems is that in the proof of security the simulator must answer private-key queries from the attacker. But this seems paradoxical: if the simulator knows all private keys then the simulator learns nothing new from the attacker. Fortunately, in the selective-security game the simulator needs to have the private keys for all identities

except for one: the challenge identity. In both our systems, the simulator uses a specially crafted method for generating private keys that enables it to generate keys for all identities, except for the challenge identity. Specifically, if one plugs the challenge identity into the simulator's key generation algorithm then the calculation results in a division by zero.

4.1 Core HIBE Construction

We are given a bilinear map $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_t$ over a bilinear group pair $(\mathbb{G}, \hat{\mathbb{G}})$ of prime order p, with respective generators $g \in \mathbb{G}$ and $\hat{g} \in \hat{\mathbb{G}}$. The size of p is determined by the security parameter.

For the time being we make two simplifying assumptions. First, we assume that public keys (IDs) at depth k are vectors of elements in $\mathbb{Z}_p^*$. We write $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_k) \in (\mathbb{Z}_p^*)^k$, where the j-th component corresponds to the identity at level j. We later extend the construction to arbitrary public keys in $\{0,1\}^*$ by first hashing each component I_j using a collision-resistant hash $H : \{0,1\}^* \to \mathbb{Z}_p^*$. In fact, we will show in Section 6 that for selective-ID security, a second-preimage-resistant hash is sufficient. Second, we temporarily assume that the messages to be encrypted are encoded as elements of $\mathbb{G}_t$. We later relax this assumption in Section 4.3.
The BB_1 HIBE system works as follows:

Setup(ℓ)**:** To generate system parameters for an HIBE system of maximum depth ℓ, given bilinear groups $(\mathbb{G}, \hat{\mathbb{G}})$ with generators $(g, \hat{g})$, the setup algorithm first selects a random $\alpha \in \mathbb{Z}_p$, and sets $g_1 = g^\alpha$ and $\hat{g}_1 = \hat{g}^\alpha$. It then picks ℓ random numbers $\delta_1, \ldots, \delta_\ell$ from $\mathbb{Z}_p$, and sets $h_i = g^{\delta_i}$ and $\hat{h}_i = \hat{g}^{\delta_i}$ for each $i = 1, \ldots, \ell$. Finally, it picks a random $\beta \in \mathbb{Z}_p$, sets $\hat{g}_0 = \hat{g}^{\alpha\beta}$, and computes $v = e(g, \hat{g}_0) = e(g, \hat{g})^{\alpha\beta}$. The public parameters *params* and the master secret mk are given by

$$params = \left(g, g_1, h_1, \ldots, h_\ell,\ \hat{g}, \hat{g}_1, \hat{h}_1, \ldots, \hat{h}_\ell,\ v \right) \in \mathbb{G}^{2+\ell} \times \hat{\mathbb{G}}^{2+\ell} \times \mathbb{G}_t$$
$$mk = \left(\hat{g}_0 \right) \in \hat{\mathbb{G}}$$

Extract(mk, ID)**:** To extract a private key d_{ID} for an identity $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j) \in (\mathbb{Z}_p^*)^j$ of depth $j \le \ell$, the authority holding the master key chooses random $r_1, \ldots, r_j \in \mathbb{Z}_p$ and outputs

$$d_{\mathsf{ID}} = \left(\hat{g}_0 \prod_{k=1}^{j} (\hat{g}_1^{\mathrm{I}_k} \hat{h}_k)^{r_k},\ \ \hat{g}^{r_1},\ \ \ldots,\ \ \hat{g}^{r_j} \right) \in \hat{\mathbb{G}}^{1+j} \tag{1}$$

Derive$(d_{\mathsf{ID}|j-1}, \mathsf{ID})$**:** The private key for ID can be generated hierarchically given a private key for the parent identity $\mathsf{ID}_{|j-1} = (\mathrm{I}_1, \ldots, \mathrm{I}_{j-1}) \in (\mathbb{Z}_p^*)^{j-1}$. Indeed, let $d_{\mathsf{ID}|j-1} = (d_0, \ldots, d_{j-1}) \in \hat{\mathbb{G}}^j$ be the private key for $\mathsf{ID}_{|j-1}$. To generate d_{ID}, pick random $r_1, \ldots, r_j \in \mathbb{Z}_p$ and output

$$d_{\mathsf{ID}} = \left(d_0 \cdot \prod_{k=1}^{j} (\hat{g}_1^{\mathrm{I}_k} \hat{h}_k)^{r_k},\ \ d_1 \cdot \hat{g}^{r_1},\ \ \ldots,\ \ d_{j-1} \cdot \hat{g}^{r_{j-1}},\ \ \hat{g}^{r_j} \right) \in \hat{\mathbb{G}}^{1+j}$$

Notice that this amounts to taking $d_{\mathsf{ID}} = \left(d_0 \cdot (\hat{g}_1^{\mathrm{I}_j} \hat{h}_j)^{r_j},\ d_1,\ \ldots,\ d_{j-1},\ \hat{g}^{r_j} \right)$, which is a valid key, and then randomizing it with the exponents $r_1, \ldots, r_{j-1}$ to ensure that the distribution of derived keys is the same as the distribution of keys generated by *Extract*.

Encrypt$(params, \mathsf{ID}, M)$**:** To encrypt a message $M \in \mathbb{G}_t$ under the public key $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j) \in (\mathbb{Z}_p^*)^j$, pick a random $s \in \mathbb{Z}_p$ and output

$$C = \left(M\, v^s,\ \ g^s,\ \ (g_1^{\mathrm{I}_1}\, h_1)^s,\ \ \ldots,\ \ (g_1^{\mathrm{I}_j}\, h_j)^s \right) \in \mathbb{G}_t \times \mathbb{G}^{1+j}$$

Decrypt(d_{ID}, C)**:** To decrypt a given ciphertext $C = (A, B, C_1, \ldots, C_j) \in \mathbb{G}_t \times \mathbb{G}^{1+j}$ using the private key $d_{\mathsf{ID}} = (d_0, d_1, \ldots, d_j) \in \hat{\mathbb{G}}^{1+j}$, output

$$A \cdot \prod_{k=1}^{j} e(C_k, d_k) \Big/ e(B, d_0) \ \in \mathbb{G}_t$$

The system in consistent. Indeed, for a valid ciphertext encrypted under the identity $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j)$ for which the private key d_{ID} is computed as in (1), we have

$$A \cdot \frac{\prod_{k=1}^{j} e(C_k, d_k)}{e(B, d_0)} = A \cdot \frac{\prod_{k=1}^{j} e(g_1^{\mathrm{I}_k}\, h_k, \hat{g})^{s\, r_k}}{e(g, \hat{g}_0)^s \prod_{k=1}^{j} e(g, \hat{g}_1^{\mathrm{I}_k}\, \hat{h}_k)^{s\, r_k}} = A \cdot \frac{1}{v^s} = M$$

4.2 Security Reduction

We prove security of our HIBE system under the Decision-BDH assumption in $(\mathbb{G}, \hat{\mathbb{G}})$ in the standard model.

Theorem 4.1. *Suppose the (t, ϵ)-Decision-BDH assumption holds in $(\mathbb{G}, \hat{\mathbb{G}})$. Then the previously defined "BB_1" ℓ-HIBE system is $(t', q_{\mathsf{ID}}, \epsilon)$-selective identity, chosen-plaintext* (IND-sID-CPA) *secure for arbitrary ℓ and q_{ID}, and all $t' < t - \Theta(\ell\, q_{\mathsf{ID}})$ expressed in the time unit of computing a general exponentiation in $\mathbb{G}$ or $\hat{\mathbb{G}}$.*

Concretely, if $\mathcal{A}$ is a q_{ID}-IND-sID-CPA adversary for BB_1 with advantage ϵ, then there exists a Decision-BDH adversary $\mathcal{B}$ for $(\mathbb{G}, \hat{\mathbb{G}})$ with the same advantage ϵ, and running in approximately the same time as $\mathcal{A}$.

Proof. Suppose $\mathcal{A}$ has advantage ϵ in attacking the HIBE system. We build an algorithm $\mathcal{B}$ that solves the Decision-BDH problem in $(\mathbb{G}, \hat{\mathbb{G}})$. Algorithm $\mathcal{B}$ is given as input a random 7-tuple $(g, g^a, g^c, \hat{g}, \hat{g}^a, \hat{g}^b, T)$ that is either sampled from $\mathcal{P}_{BDH}$ (where $T = e(g, \hat{g})^{abc}$) or from $\mathcal{R}_{BDH}$ (where T is uniform and independent in $\mathbb{G}_t$). Algorithm $\mathcal{B}$'s goal is to output 1 if $T = e(g, \hat{g})^{abc}$ and 0 otherwise. Set $g_1 = g^a$, $g_3 = g^c$ and similarly $\hat{g}_1 = \hat{g}^a$, $\hat{g}_2 = \hat{g}^b$. Algorithm $\mathcal{B}$ works by interacting with $\mathcal{A}$ in a selective identity game as follows:

Initialization. The selective-identity game begins with $\mathcal{A}$ first outputting an identity $\mathsf{ID}^* = (\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*) \in (\mathbb{Z}_p^*)^k$ of depth $k \le \ell$, that it intends to attack. If $k < \ell$ then $\mathcal{B}$ appends to ID^* a suffix of $\ell - k$ zeroes to make ID^* a vector of length ℓ; hence, from here on ID^* is assumed to be a vector in $(\mathbb{Z}_p)^\ell$ which we call the padded ID^*.

Setup. To generate the system parameters, algorithm $\mathcal{B}$ picks random $\alpha_1, \ldots, \alpha_\ell \in \mathbb{Z}_p$ and defines $h_j = g_1^{-\mathrm{I}_j^*}\, g^{\alpha_j} \in \mathbb{G}$ and $\hat{h}_j = \hat{g}_1^{-\mathrm{I}_j^*}\, \hat{g}^{\alpha_j} \in \hat{\mathbb{G}}$ for $j = 1, \ldots, \ell$. It also calculates $v = e(g_1, \hat{g}_2) \in \mathbb{G}_t$. Algorithm $\mathcal{B}$ gives $\mathcal{A}$ the system parameters $params = (g, g_1, h_1, \ldots, h_\ell, \hat{g}, \hat{g}_1, \hat{h}_1, \ldots, \hat{h}_\ell, v)$. The corresponding master key, which is unknown to $\mathcal{B}$, is equal to $\hat{g}_0 = \hat{g}^{ab} \in \hat{\mathbb{G}}$.

Phase 1. Algorithm $\mathcal{A}$ issues up to q_{ID} private key queries to $\mathcal{B}$, one at a time. Consider a query for the private key corresponding to $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_u) \in (\mathbb{Z}_p^*)^u$ where $u \leq \ell$. The only restriction is that ID not be a prefix of ID^* before we padded ID^* with zeroes. But since components of ID are restricted to $\mathbb{Z}_p^*$ it follows that ID cannot be a prefix of the padded ID^*. Hence, there must exist an index $j \in \{1, \ldots, u\}$ such that $\mathrm{I}_j \neq \mathrm{I}_j^*$. To respond to the query, algorithm $\mathcal{B}$ first derives a private key for the identity $(\mathrm{I}_1, \ldots, \mathrm{I}_j)$ from which it then constructs a private key for the requested identity $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j, \ldots, \mathrm{I}_u)$.

For the first step, $\mathcal{B}$ picks random elements $r_1, \ldots, r_j \in \mathbb{Z}_p$ and sets

$$d_0 = \hat{g}_2^{\frac{-\alpha_j}{\mathrm{I}_j - \mathrm{I}_j^*}} \prod_{n=1}^{j} (\hat{g}_1^{\mathrm{I}_n}\, \hat{h}_n)^{r_n}, \quad d_1 = \hat{g}^{r_1}, \quad \ldots, \quad d_{j-1} = \hat{g}^{r_{j-1}}, \quad d_j = \hat{g}_2^{\frac{-1}{\mathrm{I}_j - \mathrm{I}_j^*}}\, g^{r_j} \tag{2}$$

We claim that $(d_0, d_1, \ldots, d_j)$ is a correctly distributed and valid random private key for the identity $(\mathrm{I}_1, \ldots, \mathrm{I}_j)$. To see this, let $\tilde{r}_j = r_j - b/(\mathrm{I}_j - \mathrm{I}_j^*)$ in $\mathbb{Z}_p$. Then we have that

$$\hat{g}_2^{\frac{-\alpha_j}{(\mathrm{I}_j - \mathrm{I}_j^*)}} (\hat{g}_1^{\mathrm{I}_j}\, \hat{h}_j)^{r_j} = \hat{g}^{\frac{-b\,\alpha_j}{(\mathrm{I}_j - \mathrm{I}_j^*)}} (\hat{g}^{a\,(\mathrm{I}_j - \mathrm{I}_j^*)}\, \hat{g}^{\alpha_j})^{r_j} = \hat{g}^{a\,b}\, (\hat{g}^{a\,(\mathrm{I}_j - \mathrm{I}_j^*)} \hat{g}^{\alpha_j})^{r_j - \frac{b}{\mathrm{I}_j - \mathrm{I}_j^*}} = \hat{g}_0\, (\hat{g}_1^{\mathrm{I}_j}\, \hat{h}_j)^{\tilde{r}_j}$$

It follows that the private key $(d_0, d_1, \ldots, d_j)$ defined in (2) satisfies

$$d_0 = \hat{g}_0 \cdot \prod_{n=1}^{j-1} (\hat{g}_1^{\mathrm{I}_n}\, \hat{h}_n)^{r_n} \cdot (\hat{g}_1^{\mathrm{I}_j}\, \hat{h}_j)^{\tilde{r}_j}, \quad d_1 = \hat{g}^{r_1}, \quad \ldots, \quad d_{j-1} = \hat{g}^{r_{j-1}}, \quad d_j = \hat{g}^{\tilde{r}_j}$$

where the exponents $r_1, \ldots, r_{j-1}, \tilde{r}_j$ are uniform and independent in $\mathbb{Z}_p$. This matches the distribution of a private key generated by *Extract*. Hence, $(d_0, d_1, \ldots, d_j)$ is a valid private key for $(\mathrm{I}_1, \ldots, \mathrm{I}_j)$.

For the second step, algorithm $\mathcal{B}$ derives a private key for the requested $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j, \ldots, \mathrm{I}_u)$ by repeatedly applying the *Derive* procedure $u - j$ times starting from the private key $(d_0, d_1, \ldots, d_j)$ just constructed, and gives $\mathcal{A}$ the end result.

Note that this procedure will fail to produce a private key for any prefix of ID^*. Hence, $\mathcal{B}$ can generate private keys for all identities, except for prefixes of ID^*, as required.

Challenge. When $\mathcal{A}$ decides that Phase 1 is over, it outputs two messages $M_0, M_1 \in \mathbb{G}_t$ on which it wishes to be challenged. Algorithm $\mathcal{B}$ picks a random bit $\gamma \in \{0, 1\}$ and responds with the challenge ciphertext $C^* = (M_\gamma\, T,\ g_3,\ g_3^{\alpha_1}, \ldots,\ g_3^{\alpha_k})$. Since $g^{\alpha_i} = g_1^{\mathrm{I}_i^*}\, h_i$ for all i, we have that

$$C^* = \left(M_\gamma\, T,\ \ g^c,\ \ (g_1^{\mathrm{I}_1^*}\, h_1)^c,\ \ \ldots,\ \ (g_1^{\mathrm{I}_k^*}\, h_k)^c \right)$$

Hence, when $T = e(g, \hat{g})^{abc} = e(g, \hat{g}_0)^c = v^c$, i.e., when $\mathcal{B}$'s input 7-tuple is sampled from $\mathcal{P}_{BDH}$, then C^* is a valid encryption of M_γ under the (unpadded) public key $(\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*)$ initially chosen by the adversary. On the other hand, when T is uniform and independent in $\mathbb{G}_t$, i.e., when $\mathcal{B}$'s input 7-tuple is sampled from $\mathcal{R}_{BDH}$, then C^* is independent of γ in the adversary's view.

Phase 2. Algorithm $\mathcal{A}$ continues to adaptively issue queries not issued in Phase 1, up to a total of q_{ID} queries. Algorithm $\mathcal{B}$ responds as before.

Guess. Finally, $\mathcal{A}$ outputs a guess $\gamma' \in \{0,1\}$. Algorithm $\mathcal{B}$ concludes its own game by outputting a guess as follows. If $\gamma = \gamma'$ then $\mathcal{B}$ outputs 1 meaning $T = e(g,\hat{g})^{abc}$. Otherwise, it outputs 0 meaning $T \neq e(g,\hat{g})^{abc}$.

When $\mathcal{B}$'s input 7-tuple is sampled from $\mathcal{P}_{BDH}$ (where $T = e(g,\hat{g})^{abc}$) then $\mathcal{A}$'s view of the simulation is identical to a real attack, and therefore $\mathcal{A}$ must satisfy $|\Pr[\gamma = \gamma'] - 1/2| > \epsilon$. On the other hand, when $\mathcal{B}$'s input 7-tuple is sampled from $\mathcal{R}_{BDH}$ (where T is uniform in $\mathbb{G}_t$) then $\mathcal{A}$'s advantage is nil and thus $\Pr[\gamma = \gamma'] = 1/2$. Therefore, with a, b, c uniform in $\mathbb{Z}_p$, and T uniform in $\mathbb{G}_t$, we have that

$$\left| \Pr\left[\mathcal{B}(g\,, g^a, g^c, \hat{g}\,, \hat{g}^a, \hat{g}^b, e(g,g)^{abc}) = 0\right] - \Pr\left[\mathcal{B}(g\,, g^a, g^c, \hat{g}\,, \hat{g}^a, \hat{g}^b, T) = 0\right]\right| \;\geq\; \left|\frac{1}{2} \pm \epsilon - \frac{1}{2}\right| \;=\; \epsilon$$

as required. This completes the proof of Theorem 4.1. □

We note that even though the BDH assumptions stated in Section 3.1 required the group generators $g, \hat{g}$ to be random, our reduction does not rely on this property and works for arbitrary generators. Indeed, the simulator $\mathcal{B}$ passes the given generators $g, \hat{g}$ to $\mathcal{A}$ without modification. Thus, our system remains secure under a "Decision-BDH" assumption that might hold only for specific generators $(g, \hat{g})$, as long as $\mathcal{B}$ is given the generators.

4.3 (H)IBE with Security from the Hash-BDH Assumption

In Theorem 4.1 we proved security of our system based on the Decision-BDH assumption. Here we briefly observe that a simple variant of the BB_1 system can be proven secure under the weaker Hash-BDH assumption defined in Section 3.1.

Recall from Section 3.1 that Hash-BDH is defined with respect to a family of hash functions $\mathcal{H}$ of the form $H : \mathbb{G}_t \to \{0,1\}^m$ for some $m \in \mathbb{Z}_{>0}$. To prove security under Hash-BDH, we slightly modify the system to admit messages encoded as bit strings $M \in \{0,1\}^m$, and to hash the term v^s during the encryption process and similarly during decryption. For simplicity, we describe these modifications as they apply to the BB_1-IBE system; the modification to the BB_1-HIBE is analogous.

Setup(1)**:** To generate IBE system parameters, first, select a random $\alpha \in \mathbb{Z}_p$, and set $g_1 = g^\alpha$ and $\hat{g}_1 = \hat{g}^\alpha$. Next, select a random $\delta \in \mathbb{Z}_p$ and set $h = g^\delta$ and $\hat{h} = \hat{g}^\delta$. Then, pick a random $\beta \in \mathbb{Z}_p$, set $\hat{g}_0 = \hat{g}^{\alpha\beta}$, and compute $v = e(g, \hat{g}_0)$. Finally, pick a random hash function H from $\mathcal{H}$. The public parameters and the master key are respectively given by $params = (H,\ g,\ g_1,\ h,\ v)$ and $mk = \hat{g}_0$. (Without hierarchy, $\hat{g}$, $\hat{g}_1$, $\hat{h}$ may be omitted from *params*.)

Extract(mk, ID)**:** To extract a private key d_{ID} for an identity $\mathsf{ID} \in \mathbb{Z}_p^*$, pick a random $r \in \mathbb{Z}_p$ and output $d_{\mathsf{ID}} = (\hat{g}_0\,(\hat{g}_1^{\mathsf{ID}}\hat{h})^r,\ \hat{g}^r)$.

Encrypt$(params, \mathsf{ID}, M)$**:** To encrypt a message $M \in \{0,1\}^m$ under the public key $\mathsf{ID} \in \mathbb{Z}_p^*$, pick a random $s \in \mathbb{Z}_p$ and output $C = \big(M \oplus H(v^s),\ g^s,\ (g_1^{\mathsf{ID}}h)^s\big)$.

Decrypt(d_{ID}, C)**:** To decrypt a given ciphertext $C = (A, B, C_1)$ using the private key $d_{\mathsf{ID}} = (d_0, d_1)$, output $A \oplus H(\,e(B, d_0)/e(C_1, d_1)\,) = M$.

Theorem 4.1 can now be restated in terms of Hash-BDH as follows. The proof is almost identical to the proof of Theorem 4.1.

Theorem 4.2. *Suppose the* $(t, \epsilon, \mathcal{H})$*-Hash-BDH assumption holds in* $(\mathbb{G}, \hat{\mathbb{G}})$*. Then the preceding "hashed-*BB_1*" IBE system is* $(t', q_{\mathsf{ID}}, \epsilon)$*-selective identity, chosen-plaintext* (IND-sID-CPA) *secure for arbitrary* q_{ID} *and* ϵ*, and any* $t' < t - \Theta(\ell\, q_{\mathsf{ID}})$ *expressed in the time unit of exponentiation in* $\mathbb{G}$ *or* $\hat{\mathbb{G}}$*.*

A benefit of the Hash-BDH system over the Decision-BDH one is that the ciphertext no longer contains any element of $\mathbb{G}_t$, whose representation is potentially much larger than the amount of information it can convey (for bilinear groups realized on elliptic curves, the representation of elements of $\hat{\mathbb{G}}$ and $\mathbb{G}_t$ without compression tends to be much larger than that of $\mathbb{G}$). In the Hash-BDH construction with ℓ levels of hierarchy, the ciphertext overhead amounts to $\ell + 1$ elements of $\mathbb{G}$.

4.4 Relaxed Hierarchical Key Derivation

As already discussed in Section 2.1, we adopted a strong HIBE security model that requires private keys to have the same distribution regardless of how the keys were created. In particular, it should be information-theoretically impossible to tell whether a private key d_{ID} for an identity $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_\ell)$ was derived from a parent private key $d_{\mathsf{ID}|\ell-1}$ or extracted independently from the master secret, even if the parent key is revealed.

For HIBE schemes such as BB_1 where private keys are not deterministic, the preceding requirement forces the hierarchical key derivation procedure to re-randomize fully all key components that are passed from parent to child.

The BB_1 hierarchical key derivation can be made slightly more efficient under the weaker HIBE notion discussed in Section 2.1.1 that does not require independence of the private keys from their derivation history. In this case, the BB_1 HIBE supports the following simplified derivation algorithm:

Derive$'(d_{\mathsf{ID}|j-1}, \mathsf{ID})$**:** To generate a functional (but incompletely re-randomized) private key for a child identity $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j) \in (\mathbb{Z}_p^*)^j$ given a private key $d_{\mathsf{ID}|j-1} = (d_0, \ldots, d_{j-1}) \in \hat{\mathbb{G}}^j$ for the parent identity $\mathsf{ID}_{|j-1} = (\mathrm{I}_1, \ldots, \mathrm{I}_{j-1}) \in (\mathbb{Z}_p^*)^{j-1}$, pick a random $r_j \in \mathbb{Z}_p$ and output

$$d_{\mathsf{ID}} = \Big(\; d_0 \cdot \hat{g}_1^{r_j \mathrm{I}_j} \, \hat{h}_j^{r_j}, \;\; d_1, \;\; \ldots, \;\; d_{j-1}, \;\; \hat{g}^{r_j} \; \Big) \; \in \hat{\mathbb{G}}^{1+j}$$

We omit the formal security model and security reduction for this relaxed notion of HIBE.

4.5 Efficiency Considerations

The BB_1 (H)IBE system of Sections 4.1 and its variant of Section 4.3 present a number of opportunities for fast implementation, which we now explore.

Encryption. We already noted that the *Encrypt* algorithm does not require any pairing computation. Furthermore, since all exponentiations (and multi-exponentiations) in the *Encrypt* algorithm use constant bases for a given set of public parameters *params*, these exponentiations can be done very efficiently by using pre-computations. By segmenting the exponents in W-bit windows, and by pre-computing the power $g^{y \cdot 2^k}$ for all W-bit integers $y = 0, \ldots, 2^W - 1$ and all $k < \log p$ such that $W | k$, one can implement an exponentiation using only $\lceil \frac{1}{W} \log p \rceil$ group multiplications, compared with the usual W-bit windowed exponentiation algorithm which requires about $\lceil \frac{W+1}{W} \log p \rceil$ multiplications. This observation results in much faster encryption than in Boneh-Franklin IBE, as well as in non-IBE public-key systems based on discrete logarithm, such as ElGamal, where (at least) one exponentiation base will vary from user to user.

Decryption. The main computational task of the *Decrypt* algorithm involves taking the product of $j+1$ pairings (or their inverses). We note that a product of j pairings can be computed more efficiently than computing j individual pairings, with a product or ratio of two pairings being almost as fast as a single pairing [60].

Private Key Extraction. In the case of a non-hierarchical IBE system without secret sharing, the *Extract* algorithm can be implemented very efficiently. The implementation makes use of the discrete logarithms $\alpha = \mathrm{dlog}_g(g_1)$ and $\delta = \mathrm{dlog}_g(h)$ (or $\delta_1 = \mathrm{dlog}_g(h_1)$), which is accommodated by slightly modifying the *Setup* algorithm to preserve the ephemeral exponents α and δ (or δ_1) along with the master key g_0 for future use. Under these assumptions, the private key d_{ID} returned by *Extract* for a given identity ID can be efficiently calculated as

$$d_{\mathsf{ID}} = \left(\hat{g}_0\,(\hat{g}_1^{\mathsf{ID}}\hat{h})^r,\ \hat{g}^r\right) = \left(\hat{g}^{\alpha\beta+r\,(\alpha\,\mathsf{ID}+\delta)},\ \hat{g}^r\right) = \left(\hat{g}_0\cdot\hat{g}^{\xi},\ \hat{g}^r\right)$$

where $\xi = r\,(\alpha\,\mathsf{ID}+\delta) \in \mathbb{Z}_p$. Since $\hat{g}_0$ is constant, the only variable quantities in the right-most expression are the two powers $\hat{g}^{\xi}$ and $\hat{g}^r$ respectively obtained by raising the common fixed base $\hat{g}$ to the known exponents ξ and r. Without precomputation, the cost of jointly computing both quantities slightly exceeds that of a single exponentiation in $\hat{\mathbb{G}}$, which for the sake of comparison is also the cost of key generation in the Boneh-Franklin IBE system. Furthermore, since here the base of exponentiation $\hat{g}$ is fixed, precomputation techniques can be exploited to reduce further the total cost to a fraction of that level. For example, using an exponentiation algorithm with W-bit windows, the asymptotic amortized cost of key extraction is found to be about $\frac{2}{W+1}$ that of the Boneh-Franklin system, i.e., a threefold speedup using 5-bit windows.

5 BB_2 : Efficient IBE From BDHI Without Random Oracles

We construct a second efficient IBE system that is selective identity, chosen plaintext secure without random oracles, based on the q-Decision-BDHI assumption defined in Section 3.2. This gives a very different approach to constructing IBE systems. The main benefit of this system is that decryption is slightly simpler than in the system of the previous section. Encryption efficiency and ciphertext size are nominally the same; and the public parameters are only three elements, as opposed to four in the previous system. The price to pay is the reliance on a stronger assumption and a lack of flexibility (e.g., more complex hierarchies and no support for threshold key generation).

5.1 IBE Construction

We are given a bilinear map $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_t$ over a bilinear group pair $(\mathbb{G}, \hat{\mathbb{G}})$ of prime order p, with respective generators $g \in \mathbb{G}$ and $\hat{g} \in \hat{\mathbb{G}}$. The size of p is determined by the security parameter.

For now, we assume that the public keys (ID) are elements in $\mathbb{Z}_p^*$, although we later show that arbitrary strings in $\{0,1\}^*$ can be used as identities by first hashing them using a collision resistant hash $H : \{0,1\}^* \to \mathbb{Z}_p^*$. We also assume that the messages to be encrypted are represented as elements in $\mathbb{G}_t$. The IBE system works as follows:

Setup(1)**:** To generate the IBE system parameters given a bilinear group pair $(\mathbb{G}, \hat{\mathbb{G}})$ with generators $(g, \hat{g})$, the setup algorithm first selects a random generator $\hat{h}$ of $\hat{\mathbb{G}}$. Next, the algorithm computes $v = e(g, \hat{h})$. It then selects random numbers $x, y \in \mathbb{Z}_p^*$, and defines $X = g^x$ and

$Y = g^y$. The public parameters *params* and the master secret mk are given by

$$params = \left(g,\ X,\ Y,\ v \right) \ \in \mathbb{G}^3 \times \mathbb{G}_t$$
$$mk = \left(x,\ y,\ \hat{h} \right) \ \in (\mathbb{Z}_p^*)^2 \times \hat{\mathbb{G}}$$

Extract(mk, ID)**:** To extract a private key for the public key $\mathsf{ID} \in \mathbb{Z}_p^*$, pick a random $r \in \mathbb{Z}_p$ such that $x + r\,y + \mathsf{ID} \neq 0 \pmod{p}$, compute $K = \hat{h}^{1/(\mathsf{ID}+x+ry)}$, and output the private key

$$d_{\mathsf{ID}} = \left(r,\ K \right) \ \in \mathbb{Z}_p \times \hat{\mathbb{G}}$$

Encrypt$(params, \mathsf{ID}, M)$**:** To encrypt a message $M \in \mathbb{G}_t$ under public key $\mathsf{ID} \in \mathbb{Z}_p^*$, pick a random $s \in \mathbb{Z}_p^*$ and output the ciphertext

$$\mathrm{CT} = \left(M\,v^s,\ \ Y^s,\ \ X^s\,g^{s\cdot\mathsf{ID}} \right) \ \in \mathbb{G}_t \times \mathbb{G}^2$$

Decrypt$(d_{\mathsf{ID}}, \mathrm{CT})$**:** To decrypt a ciphertext $\mathrm{CT} = (A, B, C)$ using the private key $d_{\mathsf{ID}} = (r, K)$, output

$$A \,/\, e(B^r C, K) \ \in \mathbb{G}_t$$

The system is consistent. Indeed, for a private key d_{ID} and a valid ciphertext encrypted under a matching public key ID, we have

$$\frac{A}{e(B^r C, K)} = \frac{A}{e(g^{s(ry+x+\mathsf{ID})}, \hat{h}^{1/(\mathsf{ID}+x+ry)})} = \frac{A}{e(g,\hat{h})^s} = \frac{A}{v^s} = M$$

Size and Performance. Provided that the same bilinear group pair $(\mathbb{G}, \hat{\mathbb{G}})$ is used, ciphertext size and encryption time are similar to our first IBE system described earlier, and supports the same pre-computation optimizations. Decryption requires one exponentiation followed by one pairing computation, as opposed to a product (or ratio) of two pairings in our earlier system, and is thus possibly slightly simpler. We note however that BB_2 is not necessarily more efficient than BB_1, as discussed in Section 5.3.

Hash-BDHI Construction. The system above may be modified like our first system as in Section 4.3, by requiring $M \in \{0,1\}^m$ and hashing the blinding factor v^s using some function $H \in \mathcal{H} = \{H : \mathbb{G}_t \to \{0,1\}^m\}$ during encryption, and similarly during decryption. The security of the resulting system would then rely on the q-Hash-BDHI assumption rather than q-Decision-BDHI. The modifications to the above construction, and to the security proof below, are straightforward.

Hierarchies and Other Extensions. The BB_2 system can be generalized to support hierarchies, delegation, and other features, but not as easily as the BB_1 system. We refer to [19] for details.

5.2 Proving Security

We prove security of the above IBE scheme under the q-Decision-BDHI assumption from Section 3.2.

Theorem 5.1. *Suppose the (t, q, ϵ)-Decision-BDHI assumption holds in groups $(\mathbb{G}, \hat{\mathbb{G}})$ of size p. Then the "BB_2" IBE system is $(t', q_{\mathsf{ID}}, \epsilon)$-selective-identity, chosen-plaintext* (IND-sID-CPA) *secure for any $q_{\mathsf{ID}} < q$, and any $t' < t - \Theta(q^2)$ expressed in the time unit of exponentiation in $\mathbb{G}$ or $\hat{\mathbb{G}}$.*

Proof. Suppose an algorithm $\mathcal{A}$ has advantage ϵ in attacking the IBE system. We build an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to solve the q-Decision-BDHI problem in $(\mathbb{G}, \hat{\mathbb{G}})$. Algorithm $\mathcal{B}$ is given as input a random $(q+4)$-tuple $(g, g^\alpha, \hat{g}, \hat{g}^\alpha, \hat{g}^{(\alpha^2)}, \ldots, \hat{g}^{(\alpha^q)}, T) \in (\mathbb{G})^2 \times (\hat{\mathbb{G}})^{q+1} \times \mathbb{G}_t$, that is either sampled from $\mathcal{P}_{BDHI}$ (where $T = e(g,\hat{g})^{1/\alpha}$) or from $\mathcal{R}_{BDHI}$ (where T is uniform and independent in $\mathbb{G}_t$). Algorithm $\mathcal{B}$'s goal is to output 1 if $T = e(g,\hat{g})^{1/\alpha}$ and 0 otherwise. Algorithm $\mathcal{B}$ works by interacting with $\mathcal{A}$ in a selective identity game as follows:

Preparation. Algorithm $\mathcal{B}$ builds a generator $\hat{h} \in \hat{\mathbb{G}}$ for which it knows $q-1$ pairs of the form $(w_i,\ \hat{h}^{1/(\alpha+w_i)})$ for random $w_1, \ldots, w_{q-1} \in \mathbb{Z}_p^*$. This is done as follows:

1. Pick random $w_1, \ldots, w_{q-1} \in \mathbb{Z}_p^*$ and $\tau \in \mathbb{Z}_p^*$. Let $f(z)$ be the polynomial $f(z) = \tau \prod_{i=1}^{q-1}(z + w_i)$. Expand the terms of f to get $f(z) = \sum_{i=0}^{q-1} c_i z^i$. The constant term c_0 is non-zero.
2. Compute $\hat{h} = \prod_{i=0}^{q-1}(\hat{g}^{(\alpha^i)})^{c_i} = \hat{g}^{f(\alpha)}$. The random variable τ ensures that $\hat{h}$ is a random generator of $\hat{\mathbb{G}}$ independent of other values.
3. Check that $\hat{h} \neq 1$. If $\hat{h} = 1$ this would mean that $w_j = -\alpha$ for some easily identifiable w_j, at which point $\mathcal{B}$ would be able to solve the challenge BDHI problem directly. We thus assume that $w_j \neq -\alpha$ for all $j = 1, \ldots, q-1$.
4. Observe that it is easy for $\mathcal{B}$ to construct the pair $\left(w_i, \hat{h}^{1/(\alpha+w_i)}\right)$ for each $i = 1, \ldots, q-1$. To see this, write $f_i(z) = f(z)/(z+w_i) = \sum_{j=0}^{q-2} d_j z^j$, and verify that
$$\hat{h}^{1/(\alpha+w_i)} = \hat{g}^{f_i(\alpha)} = \prod_{j=0}^{q-2} \left(\hat{g}^{(\alpha^j)}\right)^{d_j}$$
which is easily computed from the input tuple.
5. Set $u = g^\alpha$ (the quantity g^α is provided as part of the input tuple)

In addition, algorithm $\mathcal{B}$ computes

$$T_h = T^{c_0} \cdot T_0 \qquad \text{where} \qquad T_0 = \prod_{j=1}^{q-1} e(g, \hat{g}^{c_j \alpha^{j-1}}) \ \in \mathbb{G}_t$$

Observe that $(T_0)^\alpha = \prod_{j=1}^{q-1} e(g, \hat{g}^{c_j \alpha^j}) = e(g, \hat{g}^{f(\alpha)-c_0})$. Thus, if $T = e(g,\hat{g})^{1/\alpha}$ then

$$T_h = T^{c_0} \cdot T_0 = e\left(g, \hat{g}^{f(\alpha)}\right)^{1/\alpha} = e(g,\hat{h})^{1/\alpha} \tag{3}$$

On the contrary, if T is uniform in $\mathbb{G}_t$, then so is T_h. We will be using the values $g, \hat{h}, u, T_h$, and the pairs $(w_i, \hat{h}^{1/(\alpha+w_i)})$ for $i = 1, \ldots, q-1$ throughout the simulation.

Initialization. The selective identity game begins with $\mathcal{A}$ outputting the identity $\mathsf{ID}^* \in \mathbb{Z}_p^*$ that it intends to attack.

Setup. To generate the system parameters, algorithm $\mathcal{B}$ does the following:

1. Pick random $a, b \in \mathbb{Z}_p^*$ under the constraint that $ab = \mathsf{ID}^*$.
2. Set $X = u^{-a} g^{-ab}$ and $Y = u$. Observe that $X = g^{-a(\alpha+b)}$ and $Y = g^\alpha$.

3. Compute $v = e(g, \hat{h})$.
4. Publish $params = (g, X, Y, v)$ as the public parameters. These are valid parameters; in particular the quantity $\hat{h}$ from which v is computed is uniformly distributed in $\hat{\mathbb{G}} \setminus \{1\}$. Note also that X and Y are independent of ID^* in the adversary's view.

We implicitly define $x = -a(\alpha + b)$ and $y = \alpha$ so that $X = g^x$ and $Y = g^y$. Algorithm $\mathcal{B}$ does not know the value of x or y, but does know the value of $x + ay = -ab = -\mathsf{ID}^*$.

Phase 1. The adversary $\mathcal{A}$ issues up to $q_{\mathsf{ID}} < q$ private key queries, one at a time. Consider the i-th query for the private key corresponding to public key $\mathsf{ID}_i \neq \mathsf{ID}^*$. Algorithm $\mathcal{B}$ needs to respond with a private key $(r, \hat{h}^{1/(\mathsf{ID}_i + x + ry)})$ for a uniformly distributed $r \in \mathbb{Z}_p \setminus \{\frac{\mathsf{ID}_i + x}{-y}\}$. Algorithm $\mathcal{B}$ responds to the query as follows:

1. Let $\left(w_i, \hat{h}^{1/(\alpha + w_i)}\right)$ be the i-th pair constructed during the preparation step. Define $\hat{h}_i = \hat{h}^{1/(\alpha + w_i)}$.
2. Algorithm $\mathcal{B}$ first constructs an $r \in \mathbb{Z}_p$ satisfying $(r - a)(\alpha + w_i) = \mathsf{ID}_i + x + ry$. To see how, let us substitute the expressions for x and y and rewrite the equation as

$$(r - a)(\alpha + w_i) = \mathsf{ID}_i - a(\alpha + b) + r\alpha$$

   After expanding, we see that the unknown α cancels from the equation, and we get $r = a + \frac{\mathsf{ID}_i - ab}{w_i} \in \mathbb{Z}_p$, which $\mathcal{B}$ can evaluate.
3. Now, $\left(r,\ \hat{h}_i^{1/(r-a)}\right)$ is a valid random private key for ID_i for two reasons. First,

$$\hat{h}_i^{1/(r-a)} = \left(\hat{h}^{1/(\alpha + w_i)}\right)^{1/(r-a)} = \hat{h}^{1/(r-a)(\alpha + w_i)} = \hat{h}^{1/(\mathsf{ID}_i + x + ry)}$$

   as required. Second, r is uniformly distributed among all elements in $\mathbb{Z}_p$ for which $\mathsf{ID}_i + x + ry \neq 0$ and $r \neq a$. This is true since w_i is uniform in $\mathbb{Z}_p \setminus \{0, -\alpha\}$ and is currently independent of $\mathcal{A}$'s view. Algorithm $\mathcal{B}$ gives $\mathcal{A}$ the private key $\left(r,\ \hat{h}_i^{1/(r-a)}\right)$.
4. For completeness, we note that $\mathcal{B}$ can construct the private key for ID_i such that $r = a$ as $\left(r,\ \hat{h}^{1/(\mathsf{ID}_i - \mathsf{ID}^*)}\right)$. Hence, the randomizer r in the private key given to $\mathcal{A}$ can be made uniform among all $r \in \mathbb{Z}_p$ for which $\mathsf{ID}_i + x + ry \neq 0$ as required.

We point out that this procedure will fail to produce the private key for $\mathsf{ID}_i = \mathsf{ID}^*$ since in that case we get $r = a$ in step (2) and therefore $\mathsf{ID}_i + x + ry = 0$. Hence, $\mathcal{B}$ can generate private keys for all public keys except for ID^*.

Challenge. The adversary $\mathcal{A}$ outputs two messages $M_0, M_1 \in \mathbb{G}_t$. Algorithm $\mathcal{B}$ picks a random bit $\gamma \in \{0, 1\}$ and a random $\rho \in \mathbb{Z}_p^*$. It responds with the ciphertext $C^* = (A, B, C) = \left(M_\gamma \cdot (T_h)^\rho,\ g^\rho,\ g^{-a\rho}\right)$. Define $s = \rho/\alpha$. On the one hand, if $T_h = e(g, \hat{h})^{1/\alpha}$ we have by (3) that

$$\begin{aligned}(T_h)^\rho &= e(g, \hat{h})^{\rho/\alpha} = e(g, \hat{h})^s = v^s \\ B &= g^\rho = Y^{\rho/\alpha} = Y^s \\ C &= g^{-a\rho} = g^{-a\alpha(\rho/\alpha)} = g^{(x+ab)(\rho/\alpha)} = g^{(x+\mathsf{ID}^*)(\rho/\alpha)} = X^s g^{s \cdot \mathsf{ID}^*}\end{aligned}$$

It follows that in this case C^* is a valid encryption of M_γ under ID^*, with the uniformly distributed randomization value $s = \rho/\alpha \in \mathbb{Z}_p^*$. On the other hand, when T_h is uniform in $\mathbb{G}_t$, then, in the adversary's view, C^* is independent of the bit γ.

Phase 2. The adversary $\mathcal{A}$ issues additional private key queries, for a total of at most $q_{\mathsf{ID}} < q$. Algorithm $\mathcal{B}$ responds as in Phase 1.

Guess. Finally, $\mathcal{A}$ outputs a guess $\gamma' \in \{0,1\}$. If $\gamma = \gamma'$ then $\mathcal{B}$ outputs 1, meaning $T = e(g,\hat{g})^{1/\alpha}$. Otherwise, it outputs 0, meaning $T \neq e(g,\hat{g})^{1/\alpha}$.

The reduction shows that when the input tuple is sampled from $\mathcal{P}_{BDHI}$ (where $T = e(g,\hat{g})^{1/\alpha}$) then $T_h = e(g,\hat{h})^{1/\alpha}$ in which case $\mathcal{A}$ must satisfy $|\Pr[\gamma = \gamma'] - 1/2| > \epsilon$. On the other hand, when the input tuple is sampled from $\mathcal{R}_{BDHI}$ (where T is uniform in $\mathbb{G}_t$) then T_h is uniform and independent in $\mathbb{G}_t$ in which case $\Pr[\gamma = \gamma'] = 1/2$. Therefore, for uniform generators $g, \hat{g}$, uniform α in $\mathbb{Z}_p^*$, and T uniform in $\mathbb{G}_t$, we have that

$$\left| \Pr\left[\mathcal{B}\begin{pmatrix} g, g^\alpha, \\ \hat{g}, \hat{g}^\alpha, \ldots, \hat{g}^{(\alpha^q)}, e(g,\hat{g})^{1/\alpha} \end{pmatrix} = 0\right] - \Pr\left[\mathcal{B}\begin{pmatrix} g, g^\alpha, \\ \hat{g}, \hat{g}^\alpha, \ldots, \hat{g}^{(\alpha^q)}, T \end{pmatrix} = 0\right] \right| \;\geq\; \left| \frac{1}{2} \pm \epsilon - \frac{1}{2} \right| \;=\; \epsilon$$

as required. This completes the proof of Theorem 5.1. □

The BB_2-IBE system described in this section is related to an IBE construction due to Sakai and Kasahara [58, Sect. 3.1]. Following our work [6], Chen and Cheng [25] presented a security proof for the Sakai-Kasahara scheme in the random-oracle model. Their proof is essentially a subset of the security proof given in Section 5.2, and is similarly based on the q-BDHI assumption. In the system of Sakai and Kasahara, and its adaptation by Chen and Cheng, the algorithm for generating user private keys is deterministic. In the BB_2 system, the identity-based key extraction is randomized and this randomization is essential for the proof of security in the standard model.

5.3 Concrete Security under Generic Attacks

The q-BDHI assumption used in the proof of BB_2 is a more complicated assumption than the decision BDH assumption used for BB_1. To study the assumption we can analyze its complexity in the generic group model [64]. The results from [8, 9] imply that a generic algorithm for the q-BDHI problem in a group of prime order p must take at least $\Omega(\sqrt[3]{p})$ time provided $q^3 \leq p$. When $q^3 > p$ no generic lower bound is known for q-BDHI, and one should avoid using such large q. From here on we always assume $q^3 \leq p$.

Interestingly, the best known algorithm for q-BDHI that works in all groups involves computing discrete-log. Hence, the best generic upper bound for q-BDHI takes time $O(\sqrt{p})$, showing a gap between the best known generic lower bound and the best known generic upper bound.

For special cases, better generic algorithms for q-BDHI are known. When q divides $p-1$, Brown and Gallant [21] and independently Cheon [26] show that given group elements g, g^α, and $g^{(\alpha^d)}$ there is an algorithm than recovers α with $O(\sqrt{p/d}+\sqrt{d})$ exponentiations in $\mathbb{G}$. Cheon [26] further shows that, when $d|p+1$, then α can be computed from $g^{(\alpha^i)}$, $i = 0, \ldots, d$ with $O(\sqrt{p/d}+d)$ exponentiations in $\mathbb{G}$.

These bounds show that for the largest allowable q, namely $q^3 \approx p$, when $q|p \pm 1$ the q-BDHI problem can be solved using a generic algorithm in $O(\sqrt[3]{p})$ exponentiations, matching the generic lower bound from [8, 9]. For other groups it is still open whether the generic lower bound or the algorithmic upper bound can be improved.

Security Implications. These generic algorithms do not imply that systems based on q-BDHI are in any way insecure. However, one should be careful when selecting the size of $\mathbb{G}$ and $\hat{\mathbb{G}}$ to meet a specific security requirement.

- A heuristic approach, allowing one to continue to rely on the discrete-log generic hardness $\Omega(\sqrt{p})$ to select the size of p, is to require that $p-1$ and $p+1$ have no divisor d in the range $\log_2 q < d \leq q$ when $q^3 \leq p$.
- A more prudent approach is to select the size of p based on the q-BDHI generic lower bound $\Omega(\sqrt[3]{p})$, regardless of the divisors of $p-1$ and $p+1$.

Security-wise, the first approach will guard against the specific algorithms discovered in [21] and [26], whereas the second approach will remain secure against all *generic* attacks, present and future, at the prescribed security level. Efficiency-wise, for a security parameter ℓ, the first approach requires a group of prime order $p \geq 2^{2\ell}$ with additional constraints on the divisors of $p-1$ and $p+1$, whereas the second approach requires a group order $p \geq 2^{3\ell}$ without such constraints. In a real implementation, these differences will likely not significantly affect the choice of elliptic curves hosting the bilinear groups, because the requirement that $|\mathbb{G}| = |\hat{\mathbb{G}}| = |\mathbb{G}_t| > 2^{\Omega(\ell)}$ is often dominated by the requirement that discrete-log in the finite field containing $\mathbb{G}_t$ have security 2^ℓ. This is especially true when using curves with embedding degree 2. Hence, requiring $p \geq 2^{3\ell}$ instead of $p \geq 2^{2\ell}$ will only make the second approach $3/2$ times slower due to the increase in the size of the exponents.

6 Arbitrary Length Identities

Recall that the basic IBE and HIBE schemes described in the previous sections require the identities or identity components to be integers in $\mathbb{Z}_p^*$. In this section we show how to allow identities to be arbitrary strings using hashing, and in particular show how to do this in the selective-identity model using universal one-way hash function families, instead of collision resistance.

6.1 Arbitrary Identities From Collision Resistance

A simple way to extend the BB_1-HIBE of Section 4 to handle identities $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_\ell)$ with $\mathrm{I}_j \in \{0,1\}^*$ (as opposed to $\mathrm{I}_j \in \mathbb{Z}_p^*$) is to first hash each I_j using a collision resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_p^*$ prior to key generation and encryption. We can similarly extend the BB_2-IBE of Section 5 to handle arbitrary identities $\mathsf{ID} \in \{0,1\}^*$. A standard argument shows that if the underlying IBE of HIBE scheme is selective-identity, chosen-plaintext (respectively, chosen-ciphertext) secure, then so is the scheme with the additional hash function.

6.2 Arbitrary Identities From UOWHF Hashing

It is well known that collision-resistant hash functions can be constructed in groups where the discrete logarithm problem is hard; which is clearly the case under the present assumptions. However, in practice, it is advantageous to use faster hash constructions such as SHA-256, whose collision resistance is heuristic. It is natural to ask whether our IBE systems can be extended to accept arbitrary length identities using a weaker assumption than collision resistance. We observe that a universal one-way hash family (UOWHF) of functions $H_k : \{0,1\}^* \to \mathcal{I}$ is sufficient to transform any selective-identity secure IBE with a finite identity set $\mathcal{I}$ into one that accepts arbitrary identities of any length in $\{0,1\}^*$.

Recall that a (t', ϵ')-UOWHF [54] is a family of functions $H_k : \{0,1\}^* \to \mathcal{I}$ with indices in $\mathcal{K}$, such that, for any randomized algorithm $(\mathcal{A}_1, \mathcal{A}_2)$ running in time t', it holds that:

$$\Pr\left[H_k(x) = H_k(x') \ : \ (x,\sigma) \stackrel{\mathrm{R}}{\leftarrow} \mathcal{A}_1,\ k \stackrel{\mathrm{R}}{\leftarrow} \mathcal{K},\ x' \stackrel{\mathrm{R}}{\leftarrow} \mathcal{A}_2(x,k,\sigma)\right] < \epsilon'$$

The probability is taken over the random choice of k and the random coins used by $(\mathcal{A}_1, \mathcal{A}_2)$. For practical purposes, it is assumed that the index k ranges in a finite set $\mathcal{K}$ and has a compact representation.

Let $\mathcal{E}$ be an IBE system that admits identities $\mathsf{ID} \in \mathcal{I}$. Suppose that $\mathcal{H} = \{H_k\}$ is a (t, ϵ')-UOWHF of functions $H_k : \{0,1\}^* \to \mathcal{I}$ indexed by $k \in \mathcal{K}$. We construct a new IBE system $\mathcal{E}'$ that takes arbitrary length identities $\mathsf{ID} \in \{0,1\}^*$ as follows:

Setup$_{\mathcal{E}'}$ runs $Setup_{\mathcal{E}}$ to obtain *params* and *mk*, selects a random index $k \in \mathcal{K}$, and returns $params' = (params, k)$ as public parameters and *mk* as master secret.

Extract$_{\mathcal{E}'}(params', mk, \mathsf{ID}')$ returns $d_{\mathsf{ID}'} \leftarrow Extract_{\mathcal{E}}(params, mk, H_k(\mathsf{ID}'))$.

Encrypt$_{\mathcal{E}'}(params', \mathsf{ID}', M)$ returns $Encrypt_{\mathcal{E}}(params, \mathsf{ID}, M)$ where $\mathsf{ID} = H_k(\mathsf{ID}')$.

Decrypt$_{\mathcal{E}'}(params', d_{\mathsf{ID}'}, C)$ returns $Decrypt_{\mathcal{E}}(params, d_{\mathsf{ID}'}, C)$.

The following theorem shows that $\mathcal{E}'$ is a selective-identity secure IBE system. This is a consequence of the fact that in the selective security model the attacker must commit to the challenge identity before seeing the public parameters, and in particular, before seeing the hash function key k. Consequently, the attacker cannot cause a hash collision with the challenge identity. The proof is immediate and is omitted.

Theorem 6.1. *Let $\mathcal{E}$ be a $(t, q_{\mathsf{ID}}, \epsilon)$-selective-identity secure IBE system* (IND-sID-CPA) *that admits identities $\mathsf{ID} \in \mathcal{I}$. Suppose that $\mathcal{H} = \{H_k\}$ is a (t, ϵ')-UOWHF of functions $H_k : \{0,1\}^* \to \mathcal{I}$ indexed by $k \in \mathcal{K}$. Then, $\mathcal{E}'$ is a $(t, q_{\mathsf{ID}}, \epsilon + \epsilon')$-selective-identity secure IBE system that admits identities $\mathsf{ID}' \in \{0,1\}^*$.*

We mention that the same transformation applies to HIBE systems in the selective-identity security model. In this case the published random hash key k is used at all levels in the hierarchy. The argument is otherwise identical.

6.3 Selective-Message Signature Schemes

For completeness, we note that each of our IBE systems, BB_1 and BB_2, gives a selectively secure signature scheme (secure against adaptive chosen-message attacks) in the standard model: view the identities as the messages, and use the key extraction as the signing algorithm. Arbitrarily long messages can be signed by suitable use of hashing, exactly as in the IBE case. By selectively secure signature [41], we mean that the adversary is required to announce the message on which it intends to forge a signature, before it is given a public key that must be statistically independent of the target message. This is analogous to the selective-identity IBE security model.

For signatures, computational (rather than decisional) versions of the assumptions are sufficient for the security reductions. In Section 7 we describe two techniques to turn selective into adaptive identity IBE. The same techniques can be used to turn selectively secure signatures into existentially unforgeable signatures.

7 Fully Secure Identity Based Encryption

Until now we only discussed selective-identity security for IBE systems where the adversary commits ahead of time to the identity ID^* it wants to attack. In the "full" adaptive-identity IBE semantic security model [13] (denoted IND-ID-CPA) and its chosen-ciphertext counterpart (denoted IND-ID-CCA2), the attacker is allowed to choose adaptively which identity to attack by specifying ID^* in the challenge phase rather than in the setup phase. Giving the adversary more power this way makes it harder to construct *adaptive-ID secure* IBE systems, or full IBE for short.

We briefly show that any selective-ID secure IBE is also an adaptive-ID secure IBE, but the reduction is not polynomial time. First, we note that the selective-identity security of an IBE system is not weakened if additional restrictions on the identities are imposed (indeed, this only tightens the constraints on the adversary and relaxes those on the simulator). Identities in the systems of Sections 4 and 5 range natively over $\mathbb{Z}_p^*$, but by the preceding remark it is safe to restrict them to the set of integers $\{1, \ldots, 2^n\}$ for $2^n < p$, represented as binary strings of length n. As noted in Section 6.1, we can then expand our IBE schemes to arbitrary identities in $\{0,1\}^*$ by first hashing identities using a collision resistant function with n-bit output, such as SHA-256 whose output is 256 bits. Hence, for an appropriately large p, taking $n = 256$ as the length of identities in the underlying IBE is a natural choice.

7.1 From Selective Identity To Adaptive Identity

Let thus N be the number of allowed identities in the underlying IBE, where for example $N = 2^{256}$. The reduction from selective-identity IBE to adaptive-identity IBE introduces a factor of N in the security parameters of the system, as described in Theorem 7.1 below. Consequently, if the IBE system has sufficiently high selective-identity security (which requires using a bilinear group of sufficiently large size p) then the system is also a fully secure IBE with adequate security. This means that the selective-ID secure IBE system of [22] as well as the two systems described earlier in this paper are fully secure IBE systems in their own right, assuming we use a large enough group so that the Decision-BDH and Decision-BDHI problems are sufficiently difficult. Extension to arbitrary identities is done using collision-resistant hashing, in which case N must be at least 2^{256}. We note that collision resistance appears to be necessary to preserve adaptive-identity security, contrarily to the case discussed in Section 6.2 where UOWHF was enough to preserve selective-identity security.

Concretely, an immediate corollary of Theorem 7.1 below is that, using 256-bits identities and using a group where no t-time adversary can break Decision-BDH with advantage 2^{-384}, the IBE system of Section 4 is a $(t, q_{\mathsf{ID}}, 2^{-128})$-adaptive-ID secure IBE for any q_{ID}. The system can be expanded to arbitrary identities in $\{0,1\}^*$ by first hashing identities using a collision-resistant hash function with a 256-bit output.

Theorem 7.1. *Let $\mathcal{E}$ be a $(t, q_{\mathsf{ID}}, \epsilon)$-IND-sID-CPA (selective identity) secure IBE system. Suppose $\mathcal{E}$ is restricted to admit N distinct identities. Then $\mathcal{E}$ is also a $(t, q_{\mathsf{ID}}, N\epsilon)$-IND-ID-CPA (adaptive identity) secure IBE.*

Proof. Suppose algorithm $\mathcal{A}$ has advantage $N\epsilon$ in breaking the full security of the IBE system. We build an algorithm $\mathcal{B}$ that has advantage ϵ in breaking selective-ID security of the system. Algorithm $\mathcal{B}$ works as follows:

Init. $\mathcal{B}$ picks a random $\mathsf{ID}^* \in \{0,1\}^n$ and reveals it to the challenger as the identity that it wishes to attack.

Setup. The challenger gives $\mathcal{B}$ the public parameters for an IBE system. $\mathcal{B}$ forwards these parameters to $\mathcal{A}$. Observe that, by definition of the selective-identity security model, the public parameters generated by the challenger are distributed independently of ID^*.

Phase 1. $\mathcal{A}$ issues private key queries. Consider the i-th query for identity ID_i. If $\mathsf{ID}_i \neq \mathsf{ID}^*$, algorithm $\mathcal{B}$ forwards the query to its challenger. Since the query is valid ($\mathsf{ID}_i \neq \mathsf{ID}^*$), the challenger responds with the private key for ID_i which $\mathcal{B}$ then forwards to $\mathcal{A}$. Note that the challenger's response is created using *Extract* and is therefore independent of ID^*. Thus, the only information about ID^* revealed by $\mathcal{B}$'s response is that $\mathsf{ID}^* \neq \mathsf{ID}_i$.

In the unlikely event that $\mathsf{ID}_i = \mathsf{ID}^*$, algorithm $\mathcal{B}$ cannot respond to this query. In this case, $\mathcal{B}$ terminates the simulation, picks a random bit $b' \in \{0,1\}$, and outputs b' as its guess for the challenger's bit b in the challenge phase.

Challenge. Once phase 1 is over $\mathcal{A}$ outputs an identity $\mathsf{ID}_0^* \in \{0,1\}^n$ and two equal length messages M_0, M_1. Algorithm $\mathcal{B}$ forwards M_0, M_1 to its challenger and receives back the challenge ciphertext C^*. We consider two cases:

1. If $\mathsf{ID}^* \neq \mathsf{ID}_0^*$ then algorithm $\mathcal{B}$ picks a random bit $\gamma' \in \{0,1\}$, outputs γ' as its guess for γ, and terminates.
2. Otherwise, $\mathsf{ID}^* = \mathsf{ID}_0^*$ in which case C^* is a proper encryption of one of M_0 or M_1 under ID_0^* as expected by $\mathcal{A}$. Algorithm $\mathcal{B}$ gives C^* to $\mathcal{A}$ and continues to Phase 2.

Phase 2. $\mathcal{A}$ continues to issue private-key queries. $\mathcal{B}$ responds as before. Since now the queries cannot equal ID^* these queries cannot cause $\mathcal{B}$ to abort.

Output. Finally, $\mathcal{A}$ outputs its guess $\gamma' \in \{0,1\}$ for γ. $\mathcal{B}$ outputs the same γ' as its guess for γ.

Next, we analyze $\mathcal{B}$'s advantage in guessing γ. Let $q_1 \leq q_{\mathsf{ID}} < N$ be the number of distinct queries that $\mathcal{A}$ issued during phase 1. Let $\mathsf{success}_1$ denote the event that during phase 1 $\mathcal{A}$ did not issue a query for ID^*. Then $\Pr[\mathsf{success}_1] = 1 - (q_1/N)$ since for any sequence of q_1 queries from the adversary, the probability (over the choice of ID^*) that ID^* collides with one the queries is q_1/N. Let $\mathsf{success}$ denote the event that both $\mathsf{success}_1$ occurred and $\mathsf{ID}^* = \mathsf{ID}_0^*$. Then

$$\Pr[\mathsf{success}] = \Pr[\mathsf{success}_1] \cdot \Pr[\mathsf{ID}^* = \mathsf{ID}_0^* \mid \mathsf{success}_1] = \left(1 - \frac{q_1}{N}\right) \frac{1}{N - q_1} = \frac{1}{N}$$

When event $\mathsf{success}$ happens, $\mathcal{A}$'s view is identical to its view in a real attack game and therefore $|\Pr[\gamma = \gamma' | \mathsf{success}] - 1/2| \geq N\epsilon$. Furthermore, by definition of $\mathcal{B}$ we have $\Pr[\gamma = \gamma' | \overline{\mathsf{success}}] = 1/2$. It follows that

$$\begin{aligned}\left|\Pr[\gamma = \gamma'] - \frac{1}{2}\right| &= \left|\Big(\Pr[\gamma = \gamma' | \mathsf{success}] \cdot \Pr[\mathsf{success}]\Big) + \Big(\Pr[\gamma = \gamma' | \overline{\mathsf{success}}] \cdot \Pr[\overline{\mathsf{success}}]\Big) - \frac{1}{2}\right| \\ &= \left|\Pr[\gamma = \gamma' | \mathsf{success}] \cdot \frac{1}{N} + \frac{1}{2} \cdot \frac{N-1}{N} - \frac{1}{2}\right| \;=\; \left|\Pr[\gamma = \gamma' | \mathsf{success}] - \frac{1}{2}\right| \cdot \frac{1}{N} \;\;\geq\;\; \epsilon\end{aligned}$$

as required. This completes the proof of Theorem 7.1. □

The Hierarchical Case. Theorem 7.1 has an immediate corollary for HIBE systems, which follows from the same proof. Here, the loss factor N is the total number of identities throughout the hierarchy.

Corollary 7.2. *Let $\mathcal{E}$ be a $(t, q_{\mathsf{ID}}, \epsilon)$-IND-sID-CPA (selective-identity) secure HIBE system. Suppose $\mathcal{E}$ admits N distinct identities across the entire hierarchy. Then $\mathcal{E}$ is also a $(t, q_{\mathsf{ID}}, N\epsilon)$-IND-ID-CPA (adaptive-identity) secure HIBE system.*

In general N will grow exponentially with the depth of the hierarchy and therefore Corollary 7.2 is most useful when the hierarchy is shallow or when the hierarchy is sparsely populated.

Theorem 7.1 and Corollary 7.2 are tight. One can give a simple example of a system that has $1/N$ security in the selective-ID model, but is insecure in the adaptive-ID settings. For example, the *Setup* algorithm can include the private key of a random identity in the public parameters. This will only help a selective attacker with probability $1/N$, but will enable an adaptive attacker to win the security game with probability one [36].

Fully Secure BB1 and BB2 in Practice. A consequence of the preceding theorem and corollary is that, with a judicious choice of parameters, one can turn any selective-ID (H)IBE scheme (such as BB_1 or BB_2) into a very simple and reasonably efficient (H)IBE system with full-fledged adaptive-ID security in the standard model.

Suppose that we seek a fully secure IND-ID-CPA IBE with security parameter $\lambda \in \mathbb{N}$, thus operating at security level 2^λ, and accepting $N = 2^\lambda$ identities. To realize it, one takes a selective IND-sID-CPA IBE at security level $2^{2\lambda}$ and restricts its allowable identities to a set of size $2^{2\lambda}/2^\lambda = 2^\lambda$, such as $\{0,1\}^\lambda$. The reason why this works despite the exponential security loss factor $N = 2^\lambda$, is that such loss is exactly compensated by doubling the IND-sID-CPA security parameter of the basic scheme (from λ to 2λ) — which for all known bilinear groups increases the computational and representational costs by constant factors, which we calculate below.

Cost of the Transformation. Recall that, for a given family of curves, the time cost $T(p, n)$ of computing a pairing using Miller's algorithms and its variations grows as:

$$T(p, n) = \Theta((\log p) \cdot (\log n)^2)$$

where $p = |\mathbb{G}| = |\hat{\mathbb{G}}| = |\mathbb{G}_t|$ is the group order and $n = |\mathbb{F}|$ is the size of the finite field extension in which the target group $\mathbb{G}_t$ is embedded. Preventing generic discrete-log attacks in $\mathbb{G}$ and number-field-sieve attacks in $\mathbb{F}$ in fewer than 2^λ elementary operations simultaneously requires that:

$$2^\lambda < L_p(1, 1/2) = 2^{(1/2+o(1))\cdot(\log p)}$$
$$2^\lambda < L_n(1/3, 1/c) = 2^{(1/c+o(1))\cdot(\log n)^{1/3}\cdot(\log\log n)^{2/3}}$$

for some constant c. Asymptotically, this can be achieved by selecting bitsizes for p and n according to the relations:

$$\log p \approx 2\lambda \qquad \text{and} \qquad \log n \approx \Theta(\lambda^3)$$

Comparing the complexity of a pairing computation (or, equivalently, that of a general exponentiation in $\mathbb{F}$ with exponent in $\mathbb{Z}_p$) for security parameters $\lambda_1 = \lambda$ versus $\lambda_2 = 2\lambda$, we see that $\log p$ must be doubled and $\log n$ must be multiplied by 8. Therefore,

$$T(p_2, n_2)/T(p_1, n_1) = 2 \cdot 8^2 = 128$$

Likewise, depending on the group, the pairing, and the elliptic-curve family on which it is realized, the representation size of elements in $\mathbb{G}$, $\hat{\mathbb{G}}$, or $\mathbb{G}_t$ will grow by a factor between

$$\log p_2 / \log p_1 = 2 \qquad \text{and} \qquad \log n_2 / \log n_1 = 8$$

In other words, operating BB_1 and BB_2 with full adaptive-ID security at security level 2^λ, requires a constant $128\times$ more time, and a constant between $2\times$ and $8\times$ more space, than operating the same scheme in basic selective-ID mode at the same security level 2^λ. This applies for all values of the (true, intended) security parameter λ.

Limitations. Even though the transformation above is sound and the resulting schemes secure, the drawback of this method is that it introduces a large loss factor N in the security reduction. To compensate we need to increase the security parameters and degrade performance.

To avoid this issue, in [7] we constructed a fully secure IBE scheme based on BB_1 and whose security reduction only carried a polynomial loss factor of $\tilde{O}({q_{\mathsf{ID}}}^2)$, as opposed to $N = 2^\lambda$ in the generic transformation; however that construction was impractical and mostly served as a proof of concept. Subsequently, Waters [69] constructed an elegant and much more efficient adaptive-ID generalization of BB_1 with a security reduction bearing a loss factor of roughly $O(q_{\mathsf{ID}})$. While that system has large public parameters, more recent constructions are adaptive-ID secure with shorter public parameters [37, 70].

7.2 Fully Secure IBE Using Random Oracles

For completeness we note that a random oracle H can convert a selective-identity IBE scheme $\mathcal{E}$ into an adaptively secure one by the process of hashing the identity ID with H before using it. We denote the resulting system by $\mathcal{E}_H$. We assume that the base system $\mathcal{E}$ does not use the random oracle H.

Theorem 7.3. *Let H be a hash function $H : \{0,1\}^* \to \{0,1\}^n$ modeled as a random oracle. Let $\mathcal{E}$ be a $(t, q_{\mathsf{ID}}, \epsilon)$ selective-ID secure IBE that does not call H. Suppose identities in $\mathcal{E}$ are n-bits long. Then $\mathcal{E}_H$ is a $(t, q_{\mathsf{ID}}, \epsilon')$ adaptive-ID secure IBE (in the random oracle model) for $\epsilon' = q_{\mathrm{H}}\,\epsilon + ({q_{\mathrm{H}}}^2/2^n)$ $\approx q_{\mathrm{H}}\,\epsilon$, where q_{H} such that $q_{\mathsf{ID}} \le q_{\mathrm{H}} < 2^{n/2}$ is the maximum number of oracle calls to H that the adversary can make (including those needed to answer private key queries).*

The proof is a straightforward adaptation of the proof of Theorem 7.1 and is omitted. Note that any collision on the hash function enables the attacker to win the IBE security game and is captured in the term ${q_{\mathrm{H}}}^2/2^n$. We also point out that in the proof, the random oracle is "programmed" at only one point.

A consequence of Theorems 7.1 and 7.3 is that, using a collision-resistant function $H : \{0,1\}^* \to \{0,1\}^n$ for a properly chosen n satisfying $2^{256} \le 2^n \ll p$, the same hash IBE scheme $\mathcal{E}_H$ can be proven secure in both the standard model and the random-oracle model. Specifically, the scheme features $(t, q_{\mathsf{ID}}, 2^n\epsilon)$-full IBE security in the standard model, which is boosted to $(t, q_{\mathsf{ID}}, q_{\mathrm{H}}\epsilon)$ when H is viewed as a random oracle.

Hierarchical Adaptive Identity Security. A hierarchical analogue to Theorem 7.3 can be stated, but in the HIBE case it is necessary to apply the random-oracle conversion at every level of the hierarchy, so that the total degradation factor for an identity at level ℓ is a product of ℓ factors.

To avoid hash collisions across levels, we compute the identity ID for the underlying HIBE from the input identity $\mathsf{ID}' = (\mathrm{I}'_1, ..., \mathrm{I}'_\ell)$ as follows:

$$\mathsf{ID} = \big(I_1 = H(\mathrm{I}'_1),\ I_2 = H(\mathrm{I}'_1, \mathrm{I}'_2),\ \ldots,\ I_\ell = H(\mathrm{I}'_1, \mathrm{I}'_2, \ldots,\ I'_\ell) \big)$$

For completeness, we state the following corollary.

Corollary 7.4. *Let $\mathcal{E}$ be a $(t, q_{\mathsf{ID}}, \epsilon)$ selective-ID secure HIBE of maximum depth ℓ. Suppose identity components in $\mathcal{E}$ are n-bits long. Let $H : \{0,1\}^* \to \{0,1\}^n$ be a hash function modeled as a random oracle. Then $\mathcal{E}_H$ is a $(t, q_{\mathsf{ID}}, \epsilon')$ adaptive-ID secure HIBE (in the random-oracle model) for $\epsilon' = {q_{\mathrm{H}}}^{\ell}\, \epsilon + ({q_{\mathrm{H}}}^2/2^n)$, where q_{H} such that $q_{\mathsf{ID}} \leq q_{\mathrm{H}} < 2^{n/2}$ is the total number of oracle calls to H that the adversary can make (including those needed to answer private key queries).*

Note that security degrades exponentially in the hierarchy depth, so that only a logarithmic number of levels can be meaningfully allowed. For a long time all HIBE constructions in the adaptive-ID setting suffered from a similar problem. Recently, Gentry and Halevi [38] and Waters [70] presented HIBE systems for hierarchies of polynomial depth without restriction on the number of identities.

8 Adaptive Chosen Ciphertext Security

In this section, we turn to the orthogonal issue of hardening semantically secure systems (against chosen plaintext attacks, denoted CPA) into ones that withstand adaptive chosen ciphertext attacks (denoted CCA2 or CCA for short). We review a number of tight reductions with and without random oracles.

8.1 Chosen Ciphertext Security in the Standard Model

Canetti, Halevi, and Katz [22, Section 2.2] describe a general method for converting a selective-identity IBE that is chosen-plaintext secure into one that is chosen-ciphertext secure. The method is based on [55, 57, 47]. Since it is generic, it applies to both of our systems as well. In particular, the method can be used to render the IBE system of Section 5 secure against chosen-ciphertext attacks. The result is an $\mathsf{IND\text{-}sID\text{-}CCA2}$ secure IBE without random oracles. However, the resulting system is inefficient since the transformation from [22] relies on generic non-interactive zero-knowledge (NIZK) constructions.

A much more efficient way of constructing a CCA2-secure IBE is to start from an HIBE system such as that of Section 4. A method from Canetti, Halevi, and Katz [23] works by appending a one-time signature to the ciphertext, which is encrypted to an identity equal to the verification key. A variant due to Boneh and Katz [16] uses MAC instead of signatures. A direct approach given by Boyen, Mei, and Waters [20] exploits the structure of the BB_1 ciphertext to avoid MACs and signatures altogether. All of these methods can be used to build a selective identity, chosen ciphertext secure ℓ-HIBE system (or ℓ-HIB-KEM) starting from the basic BB_1 selective-identity, chosen-plaintext secure $(\ell + 1)$-HIBE scheme of Section 4. This works for any ℓ we choose, by applying either of the [23, 16, 20] methods on the last element of the identity vector. In particular, starting from a basic 2-level HIBE we obtain an efficient selective-identity, chosen-ciphertext secure IBE without random oracles (which can then be rendered fully secure as described in the previous section). We refer the reader to [12] and [20] for a performance analysis of these new CCA2 systems, in comparison with previous systems such as Cramer-Shoup [29] and Kurosawa-Desmedt [46].

8.2 Chosen Ciphertext Security in the Random Oracle Model

In the random oracle model, the hybrid construction of Fujisaki and Okamoto [33] and its tight variant [34] can be used to turn a semantically secure public-key system into a CCA2-secure one by substituting a random oracle for its random coins; these constructions also apply to our identity-based systems.

8.3 A Non-Interactive CCA2-Secure Threshold Public Key System

In the IBE system of Section 4 it is easy to distribute the master key among n parties so that any t parties can be used to derive the private key for a given identity. When applying the technique of [23] to the resulting threshold IBE system, we obtain an efficient CCA2-secure *threshold* public key system in the standard model. The full details are provided in [10]. Alternatively, we can use the method of [20] to turn BB_1 IBE specifically into a threshold public-key KEM that is more efficient; the drawback is that the functionality we obtain is (threshold) key encapsulation (of a random key) rather than complete encryption (of any selected plaintext). These constructions resolve an open problem posed by Shoup and Gennaro [65].

8.4 Labeled Encryption and Labeled IBE

A feature of encryption systems that is related to chosen ciphertext security, and is often desirable when constructing larger protocols is that of labeled encryption [65]. Essentially, in a labeled encryption system, the encryption algorithm takes an extra argument, the *label*, which is a string in $\{0,1\}^*$, and so does the decryption algorithm. The idea is that decryption should only succeed if the label given to the decryption algorithm matches the one used to produce the ciphertext to decrypt. More precisely, the additional security property we require is that it be infeasible for a polynomially bounded adversary to cause the decryption algorithm to decrypt a ciphertext that has been encrypted using a different label.

We observe that by using the identity as a label, IBE immediately gives us labeled encryption. Furthermore, since many protocols that require labels admit simulation proofs in which the label is known at the beginning, selective-ID security is sufficient for this purpose. In general, by using the last component of an identity vector as label, we can turn $(\ell+1)$-HIBE into a labeled ℓ-HIBE.

9 Conclusions

We constructed two IBE systems that are secure against selective-identity attacks in the standard model, i.e., without using random oracles.

The first scheme, BB_1, is based on the standard Decision (or hash) Bilinear Diffie-Hellman (BDH) assumption. Since its introduction in 2004, the BB_1 scheme has evolved into a useful paradigm based on which many generalizations of identity-based encryption have flourished. The second scheme, BB_2, is based on the stronger Bilinear Diffie-Hellman Inversion (BDHI) assumption. It is slightly more efficient in some contexts and was extended to an HIBE and beyond in [19].

Both our constructions can be transformed into efficient CCA2-secure public-key systems without random oracles that are almost as efficient as the Cramer-Shoup public-key system. We also observed that any selective-ID secure IBE system implies a full adaptive-ID secure IBE system. While the resulting security reduction is not polynomial, the parameters can be adjusted so that the resulting system is fully secure in the standard model. If one does tolerate the use of random oracles, then hashing identity strings is sufficient to turn any selective-ID secure (H)IBE system into

an adaptive-ID secure one. Operated this way, the BB_1 system is an efficient and highly practical IBE system.

We conclude by noting that very recently two lattice-based IBE constructions were also shown to secure without random oracles. Cash et al. [24] use a lattice analogue of the "bit-by-bit" mechanism of Canetti et al. [23], while Agrawal et al. [1] use a lattice analogue of the "all-at-once" mechanism of the BB_1 system.

Acknowledgments

The comment on Hash-BDH is due to Victor Shoup. The requirement for delegation history independence in the HIBE security model resulted from a discussion with Brent Waters. We also thank Shai Halevi, Jonathan Katz, Aleksandr Yampolskiy, and the anonymous referees for numerous helpful comments on this work. This work was supported by NSF and the Packard foundation.

References

[1] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In *Proceedings of Eurocrypto 2010*, 2010.

[2] Paulo S. L. M. Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In *Selected Areas in Cryptography—SAC 2005*, volume 3897 of *LNCS*, pages 319–331, 2005.

[3] Mihir Bellare and Phil Rogaway. Random oracle are practical: A paradigm for designing efficient protocols. In *Proceedings of the First ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[4] Eli Biham, Dan Boneh, and Omer Reingold. Breaking generalized Diffie-Hellman modulo a composite is no easier than factoring. *Information Processing Letters*, 70:83–7, 1999.

[5] Ian Blake, Gadiel Seroussi, and Nigel Smart. *Elliptic Curves in Cryptography*, volume 265 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 1999.

[6] Dan Boneh and Xavier Boyen. Efficient selective-ID identity based encryption without random oracles. In *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–38. Springer-Verlag, 2004.

[7] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In Matt Franklin, editor, *Advances in Cryptology—CRYPTO 2004*, volume 3152 of *LNCS*, pages 443–59. Springer-Verlag, 2004.

[8] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73. Springer-Verlag, 2004.

[9] Dan Boneh and Xavier Boyen. Short signatures without random oracles and the SDH assumption in bilinear groups. *Journal of Cryptology*, 21(2):149–77, 2008.

[10] Dan Boneh, Xavier Boyen, and Shai Halevi. Chosen ciphertext secure public key threshold encryption without random oracles. In *Topics in Cryptology—CT-RSA 2006*, volume 3860 of *LNCS*, pages 226–43. Springer-Verlag, 2006.

[11] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *Advances in Cryptology—CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, 2004.

[12] Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. *SIAM J. of Computing (SICOMP)*, 36(5):915–942, 2006. Journal version of [23] and [16].

[13] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *Advances in Cryptology—CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–29. Springer-Verlag, 2001.

[14] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. *SIAM Journal of Computing*, 32(3):586–615, 2003.

[15] Dan Boneh, Craig Gentry, and Michael Hamburg. Space-efficient identity based encryption without pairings. In *Proceedings of FOCS 2007*, pages 647–657, 2007.

[16] Dan Boneh and Jonathan Katz. Improved efficiency for CCA-secure cryptosystems built using identity based encryption. In *Proceedings of CT-RSA 2005*, volume 3376 of *LNCS*. Springer-Verlag, 2005.

[17] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In *Advances in Cryptology—ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 514–32. Springer-Verlag, 2001.

[18] Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, 324:71–90, 2003.

[19] Xavier Boyen. General ad hoc encryption from exponent inversion IBE. In *Advances in Cryptology—EUROCRYPT 2007*, volume 4515 of *LNCS*, pages 394–411. Springer-Verlag, 2007.

[20] Xavier Boyen, Qixiang Mei, and Brent Waters. Direct chosen ciphertext security from identity-based techniques. In *ACM Conference on Computer and Communications Security—CCS 2005*. ACM Press, 2005.

[21] Daniel Brown and Robert Gallant. The static Diffie-Hellman problem. Cryptology ePrint Archive, Report 2004/306, 2004. `http://eprint.iacr.org/`.

[22] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In *Advances in Cryptology—EUROCRYPT 2003*, volume 2656 of *LNCS*. Springer-Verlag, 2003.

[23] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–22. Springer-Verlag, 2004.

[24] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In *Proceedings of Eurocrypto 2010*, 2010.

[25] Liqun Chen and Zhaohui Cheng. Security proof of Sakai-Kasahara's identity-based encryption scheme. In *Cryptography and Coding, 10th IMA International Conference*, pages 442–459, 2005.

[26] Jung Hee Cheon. Security analysis of the strong Diffie-Hellman problem. In *Advances in Cryptology—EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 1–11. Springer-Verlag, 2006.

[27] Céline Chevalier, Pierre-Alain Fouque, David Pointcheval, and Sébastien Zimmer. Optimal randomness extraction from a Diffie-Hellman element. In *Advances in Cryptology—EUROCRYPT 2009*, pages 572–589, 2009.

[28] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *Proceedings of the 8th IMA International Conference on Cryptography and Coding*, pages 26–8, 2001.

[29] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attacks. In Hugo Krawczyk, editor, *Advances in Cryptology—CRYPTO 1998*, volume 1462 of *LNCS*, pages 13–25. Springer-Verlag, 1998.

[30] Giovanni Di Crescenzo and Vishal Saraswat. Public key encryption with searchable keywords based on jacobi symbols. In *Proceedings of INDOCRYPT 2007*, pages 282–296, 2007.

[31] David Freeman. Constructing pairing-friendly elliptic curves with embedding degree 10. In *Proceedings of ANTS 2006*, pages 452–465, 2006.

[32] David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. Cryptology ePrint Archive, Report 2006/372, 2006. `http://eprint.iacr.org/`.

[33] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology—CRYPTO 1999*, LNCS, pages 537–54. Springer-Verlag, 1999.

[34] Eiichiro Fujisaki and Tatsuaki Okamoto. How to enhance the security of public-key encryption at minimum cost. *IEICE Transactions on Fundamentals*, E83-9(1):24–32, 2000.

[35] Steven Galbraith, Kenneth Paterson, and Nigel Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[36] David Galindo. A separation between selective and full-identity security notions for identity-based encryption. *ICCSA*, 3:318–326, 2006.

[37] Craig Gentry. Practical identity-based encryption without random oracles. In *Advances in Cryptology—EUROCRYPT 2006*, LNCS. Springer-Verlag, 2006.

[38] Craig Gentry and Shai Halevi. Hierarchical identity based encryption with polynomially many levels. In *Theory of Cryptography—TCC 2009*, volume 5444 of *LNCS*, pages 437–56. Springer-Verlag, 2009.

[39] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of STOC 2008*, pages 197–206, 2008.

[40] Craig Gentry and Alice Silverberg. Hierarchical ID-based cryptography. In *Advances in Cryptology—ASIACRYPT 2002*, LNCS. Springer-Verlag, 2002.

[41] Shafi Goldwasser, Silvio Micali, and Ron Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.

[42] Jeremy Horwitz and Ben Lynn. Towards hierarchical identity-based encryption. In *Advances in Cryptology—EUROCRYPT 2002*, LNCS, pages 466–81. Springer-Verlag, 2002.

[43] Russell Impagliazzo, Leonid Levin, and Michael Luby. Pseudo random generation from one-way functions. In *Proceedings of the 21st ACM Symposium on Theory of Computing*, 1989.

[44] Antoine Joux. A one round protocol for tripartite Diffie-Hellman. In Wieb Bosma, editor, *Proceedings of ANTS IV*, volume 1838 of *LNCS*, pages 385–94. Springer-Verlag, 2000.

[45] Eike Kiltz. From selective-ID to full security: The case of the inversion-based Boneh-Boyen IBE scheme. Cryptology ePrint Archive, Report 2007/033, 2007. `http://eprint.iacr.org/`.

[46] Kaoru Kurosawa and Yvo Desmedt. A new paradigm of hybrid encryption scheme. In *Advances in Cryptology—CRYPTO 2004*, volume 3152 of *LNCS*, pages 426–42. Springer-Verlag, 2004.

[47] Yehuda Lindell. A simpler construction of CCA2-secure public-key encryption under general assumptions. In *Advances in Cryptology—EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 241–54, 2003.

[48] Anna Lysyanskaya. Unique signatures and verifiable random functions from the DH-DDH separation. In *Advances in Cryptology—CRYPTO 2002*, LNCS. Springer-Verlag, 2002.

[49] Ueli M. Maurer and Yacov Yacobi. A non-interactive public-key distribution system. *Designs, Codes and Cryptography*, 9(3):305–16, November 1996.

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

[51] Shigeo Mitsunari, Ryuichi Sakai, and Masao Kasahara. A new traitor tracing. *IEICE Transactions on Fundamentals*, E85-A(2):481–4, 2002.

[52] Atsuko Miyaji, Masaki Nakabayashi, and Shunzou Takano. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Trans. Fundamentals*, E84-A(5):1234–43, 2001.

[53] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, pages 458–67, 1997.

[54] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *Proceedings of the 21st ACM Symposium on Theory of Computing*. ACM, 1989.

[55] Moni Naor and Moti Yung. Public key cryptosystems provable secure against chosen ciphertext attacks. In *Proceedings of the 22nd ACM Symposium on Theory of Computing*, pages 427–37. ACM, 1990.

[56] Karl Rubin and Alice Silverberg. Supersingular abelian varieties in cryptology. In *Advances in Cryptology—CRYPTO 2002*, pages 336–353, 2002.

[57] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science*, 1999.

[58] Ryuichi Sakai and Masao Kasahara. ID based cryptosystems with pairing over elliptic curve. Cryptology ePrint Archive, Report 2003/054, 2003. http://eprint.iacr.org/.

[59] Ryuichi Sakai, Kiyoshi Ohgishi, and Masao Kasahara. Cryptosystems based on pairings. In *Proceedings of the Symposium on Cryptography and Information Security—SCIS 2000*, Japan, 2000.

[60] Michael Scott. Computing the Tate pairing. In *Proceedings of CT-RSA 2005*, volume 3376 of *LNCS*, pages 293–304. Springer-Verlag, 2005.

[61] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology—CRYPTO 1984*, volume 196 of *LNCS*, pages 47–53. Springer-Verlag, 1984.

[62] Emily Shen. Making the BB2-IBE scheme fully secure. Unpublished note, 2006.

[63] Elaine Shi and Brent Waters. Delegating capabilities in predicate encryption systems. In *ICALP*, pages 560–578, 2008.

[64] Victor Shoup. Lower bounds for discrete logarithms and related problems. In *Advances in Cryptology—EUROCRYPT 1997*, volume 1233 of *LNCS*, pages 256–66. Springer-Verlag, 1997.

[65] Victor Shoup and Rosario Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. *Journal of Cryptology*, 15(2):75–96, 2002. Extended abstract in Eurocrypt '98.

[66] Michael Steiner, Gene Tsudik, and Michael Waidner. Diffie-Hellman key distribution extended to groups. In *Proceedings of the ACM Conference on Computer and Communications Security*, 1996.

[67] Hatsukazu Tanaka. A realization scheme for the identity-based cryptosystem. In *Advances in Cryptology—CRYPTO 1987*, volume 293 of *LNCS*, pages 341–49. Springer-Verlag, 1987.

[68] Shigeo Tsujii and Toshiya Itoh. An ID-based cryptosystem based on the discrete logarithm problem. *IEEE Journal on Selected Areas in Communication*, 7(4):467–73, 1989.

[69] Brent Waters. Efficient identity-based encryption without random oracles. In *Advances in Cryptology—EUROCRYPT 2005*, volume 3494 of *LNCS*. Springer-Verlag, 2005.

[70] Brent Waters. Dual key encryption: Realizing fully secure IBE and HIBE under simple assumption. In *Advances in Cryptology—CRYPTO 2009*, 2009.

A DHI and Generalized Diffie-Hellman

In Section 3.2 we defined the q-BDHI problem in a bilinear group. A closely related problem is the q-Diffie-Hellman Inversion (q-DHI) problem: given a tuple $(g, g^x, g^{(x^2)}, \ldots, g^{(x^q)}) \in \mathbb{G}^{q+1}$ as input, output $g^{1/x} \in \mathbb{G}$. Here, $\mathbb{G}$ need not be a bilinear group. Loosely speaking, the q-DHI assumption states that the q-DHI problem is intractable in $\mathbb{G}$. This assumption was previously used in [51] where it was called weak Diffie-Hellman.

Many cryptographic constructions rely on the Generalized Diffie-Hellman assumption (GenDH) for security [66, 53, 4, 48, 18]. In this section we show that the q-DHI assumption implies the $(q+1)$-Generalized Diffie-Hellman assumption. Thus, constructions that rely on Generalized Diffie-Hellman could instead rely on q-DHI which appears to be a more natural complexity assumption, and is easier to state since the problem description does not require an oracle.

We first review the GenDH assumption. The assumption says that, for a random generator g of $\mathbb{G}$, given $g^{a_1}, \ldots, g^{a_q}$ in $\mathbb{G}$ and given all the subset products $g^{\prod_{i \in S} a_i} \in \mathbb{G}$ for all strict subsets $S \subset \{1, \ldots, q\}$, it is hard to compute $g^{a_1 \cdots a_q} \in \mathbb{G}$. Since the number of subset products is exponential in q, access to them is provided through an oracle. For a vector $\vec{a} = (a_1, \ldots, a_q) \in {\mathbb{Z}_p}^q$, define $\mathcal{O}_{g,\vec{a}}$ to be an oracle that for any strict subset $S \subset \{1, \ldots, q\}$ responds with

$$\mathcal{O}_{g,\vec{a}}(S) = g^{\prod_{i \in S} a_i} \in \mathbb{G}.$$

Define the advantage of algorithm $\mathcal{A}$ in solving the generalized Diffie-Hellman problem to be the probability that $\mathcal{A}$ is able to compute $g^{a_1 \cdots a_q}$ given access to the oracle $\mathcal{O}_{g,\vec{a}}(S)$. In other words,

$$\mathrm{Adv}_{\mathcal{A},q} = \Pr\left[\mathcal{A}^{\mathcal{O}_{g,\vec{a}}} = g^{a_1 \cdots a_q} \;:\; g \leftarrow \mathbb{G} \setminus \{1\},\ \ \vec{a} = (a_1, \ldots, a_q) \leftarrow (\mathbb{Z}_p)^q\right]$$

Note that the oracle only answers queries for strict subsets of $\{1, \ldots, q\}$.

Definition A.1. We say that $\mathbb{G}$ satisfies the (t, q, ϵ)-Generalized Diffie-Hellman assumption if for all t-time algorithms $\mathcal{A}$ we have $\mathrm{Adv}_{\mathcal{A},q} < \epsilon$.

Theorem A.2. *Suppose the $(t, q-1, \epsilon)$-DHI assumption holds in $\mathbb{G}$. Then the (t, q, ϵ)-GenDH assumption also holds in $\mathbb{G}$.*

Proof. Suppose $\mathcal{A}$ is an algorithm that has advantage ϵ in solving the q-GenDH problem. We construct an algorithm $\mathcal{B}$ that solves $(q-1)$-DHI with the same advantage ϵ. Algorithm $\mathcal{B}$ is given $g, g^x, g^{(x^2)}, \ldots, g^{(x^{q-1})} \in \mathbb{G}$ and its goal is to compute $g^{1/x} \in \mathbb{G}$. Let $h = g^{(x^{q-1})}$ and $y = x^{-1} \in \mathbb{Z}_p$. Then the input to $\mathcal{B}$ can be re-written as $h, h^y, h^{(y^2)}, \ldots, h^{(y^{q-1})} \in \mathbb{G}$ and $\mathcal{B}$'s goal is to output $h^{(y^q)} = g^{1/x}$.

Algorithm $\mathcal{B}$ first picks q random values $c_1, \ldots, c_q \in \mathbb{Z}_p$. It then runs algorithm $\mathcal{A}$ and simulates the oracle $\mathcal{O}_{h,\vec{a}}$ for $\mathcal{A}$. The vector $\vec{a}$ that $\mathcal{B}$ will use is $\vec{a} = (y + c_1, \ldots, y + c_q)$. Note that $\mathcal{B}$ does not know $\vec{a}$ explicitly since $\mathcal{B}$ does not have y. When $\mathcal{A}$ issues a query for $\mathcal{O}_{h,\vec{a}}(S)$ for some strict subset $S \subset \{1, \ldots, q\}$ algorithm $\mathcal{B}$ responds as follows:

1. Define the polynomial $f(z) = \prod_{i \in S}(z + c_i)$ and expand the terms to obtain $f(z) = \sum_{i=0}^{|S|} b_i z^i$.
2. Compute $t = \prod_{i=0}^{|S|}(h^{(y^i)})^{b_i} = h^{f(y)}$. Since $|S| < q$ all the values $h^{(y^i)}$ in the product are known to $\mathcal{B}$.
3. By construction we know that $t = h^{\prod_{i \in S}(y + c_i)}$. Algorithm $\mathcal{B}$ responds by setting $\mathcal{O}_{h,\vec{a}}(S) = t$.

The responses to all of the adversary's oracle queries are consistent with the hidden vector $\vec{a} = (y + c_1, \ldots, y + c_q)$. Therefore, eventually, $\mathcal{A}$ will output $T = h^{\prod_{i=1}^{q}(y + c_i)}$. Define the polynomial $f(z) = \prod_{i=1}^{q}(z + c_i)$ and expand the terms to get $f(z) = z^q + \sum_{i=0}^{q-1} b_i z^i$. To conclude, $\mathcal{B}$ outputs

$$T \Big/ \prod_{i=0}^{q-1}(h^{(y^i)})^{b_i} = h^{(y^q)}$$

which is the required value. □

The same property as in Theorem A.2 also holds for the decision versions of the DHI and GenDH problems. The q-DHI assumption is easier to state than the q-GenDH assumption since there is no need for an oracle. When appropriate, constructions that depend on GenDH for security could instead use the DHI assumption.