Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles

Dan Boneh[1]*, and Xavier Boyen[2]

[1] Computer Science Department, Stanford University, Stanford CA 94305-9045
dabo@cs.stanford.edu
[2] Voltage Security, Palo Alto, California
xb@boyen.org

Abstract. We construct two efficient Identity Based Encryption (IBE) systems that are selective identity secure *without the random oracle model*. 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 the standard model the adversary is allowed to choose this identity adaptively. Our first secure IBE system extends to give a selective identity Hierarchical IBE secure without random oracles.

1 Introduction

Boneh and Franklin [BF01,BF03] recently defined a security model for Identity Based Encryption [Sha84] and gave a construction using bilinear maps. Cocks [Coc01] describes another construction using quadratic residues. Proving security for these systems requires the random oracle model [BR93]. A natural open question is to construct a secure IBE system without random oracles. No such system is currently known.

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. [CHK03,CHK04] recently 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. are able to construct 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 selective identity secure without the random oracle model. In both systems, encryption requires no bilinear map computation and decryption requires at most two. Our first

* Supported by NSF and the Packard Foundation.

construction 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 [HL02] and the first construction in the random oracle model was given by Gentry and Silverberg [GS02]. Our efficient HIBE construction is similar to the Gentry-Silverberg system, but we are able to prove security without using random oracles. Our second IBE construction is even more efficient, but is based on a new 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 q.

Canetti et al. [CHK04] recently 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. In particular, using our second system we obtain a CCA2-secure public key system that has comparable efficiency to the Cramer-Shoup system based on DDH.

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 [Sha84,BF01]: *Setup*, *KeyGen*, *Encrypt*, *Decrypt*. The *Setup* algorithm generates system parameters, denoted by *params*, and a master key *master-key*. The *KeyGen* algorithm uses the master key to generate the 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 [HL02,GS02] identities are vectors. A vector of dimension ℓ represents an identity at depth ℓ. Algorithm *KeyGen* 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 $\ell - 1$. It outputs the private key d_{ID} for identity ID. We refer to the *master-key* as the private key at depth 0 and note that an IBE system is an HIBE where all identities are at depth 1.

Boneh and Franklin [BF01,BF03] define chosen ciphertext security for IBE systems under a chosen identity attack. In their model the adversary is allowed to adaptively chose the public key it wishes to attack (the public key on which it will be challenged). Canetti, Halevi, and Katz [CHK03,CHK04] define a weaker notion of security in which the adversary commits ahead of time to the public key it will attack. We refer to this notion as selective identity, chosen ciphertext secure IBE ($\mathsf{IND\text{-}sID\text{-}CCA}$). More precisely, selective identity IBE and HIBE security is defined using the following game:

Init: The adversary outputs an identity ID^* where it wishes to be challenged.

Setup: The challenger runs the *Setup* algorithm. It gives the adversary the resulting system parameters *params*. It keeps the *master-key* to itself.

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

- Private key query $\langle \mathsf{ID}_i \rangle$ where $\mathsf{ID}_i \neq \mathsf{ID}^*$ and ID_i is not a prefix of ID^*. The challenger responds by running algorithm *KeyGen* to generate the private key d_i corresponding to the public key $\langle \mathsf{ID}_i \rangle$. It sends d_i to the adversary.
- Decryption query $\langle C_i \rangle$ for identity ID^* or any prefix of ID^*. The challenger responds by running algorithm *KeyGen* to generate the private key d corresponding to ID^* (or the relevant prefix thereof as requested). It then runs algorithm *Decrypt* to decrypt the ciphertext C_i using the private key d. It sends the resulting plaintext to the adversary.

These queries may be asked adaptively, that is, each query q_i may depend on the replies 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 queries $q_{m+1}, \ldots, q_n$ where q_i is one of:

- Private key query $\langle \mathsf{ID}_i \rangle$ where $\mathsf{ID}_i \neq \mathsf{ID}^*$ and ID_i is not a prefix of ID^*. The challenger responds as in Phase 1.
- Decryption query $\langle C_i \rangle \neq \langle C \rangle$ for ID^* or any prefix of ID^*. The challenger responds as in Phase 1.

These queries may be asked adaptively 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 $\mathsf{IND\text{-}sID\text{-}CCA}$ adversary. We define the advantage of the adversary $\mathcal{A}$ in attacking the scheme $\mathcal{E}$ as

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

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

Definition 1. *We say that an IBE or HIBE system $\mathcal{E}$ is $(t, q_{\scriptscriptstyle ID}, q_{\scriptscriptstyle C}, \epsilon)$-selective identity, adaptive chosen ciphertext secure if for any t-time* $\mathsf{IND\text{-}sID\text{-}CCA}$ *adversary $\mathcal{A}$ that makes at most $q_{\scriptscriptstyle ID}$ chosen private key queries and at most $q_{\scriptscriptstyle C}$ chosen decryption queries we have that $Adv_{\mathcal{E},\mathcal{A}} < \epsilon$. As shorthand, we say that $\mathcal{E}$ is $(t, q_{\scriptscriptstyle ID}, q_{\scriptscriptstyle C}, \epsilon)$* $\mathsf{IND\text{-}sID\text{-}CCA}$ *secure.*

Semantic Security. As usual, we define selective identity, chosen plaintext security for an IBE system as in the preceding game, except that the adversary is not allowed to issue any decryption queries. The adversary may still issue adaptive private key queries.

Definition 2. *We say that an IBE or HIBE system $\mathcal{E}$ is (t, q_{ID}, ϵ)-selective identity, chosen plaintext secure if $\mathcal{E}$ is $(t, q_{ID}, 0, \epsilon)$-selective identity, chosen ciphertext secure. As shorthand, we say that $\mathcal{E}$ is (t, q_{ID}, ϵ)* IND-sID-CPA *secure.*

2.2 Bilinear Groups

We briefly review the necessary facts about bilinear maps and bilinear map groups. We follow the notation in [BF01]:

1. $\mathbb{G}$ and $\mathbb{G}_1$ are two (multiplicative) cyclic groups of prime order p;
2. g is a generator of $\mathbb{G}$.
3. e is a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$.

Let $\mathbb{G}$ and $\mathbb{G}_1$ be two groups as above. A bilinear map is a map $e : \mathbb{G}\times\mathbb{G} \to \mathbb{G}_1$ with the following properties:

1. Bilinear: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degenerate: $e(g, g) \neq 1$.

We say that $\mathbb{G}$ is a bilinear group if the group action in $\mathbb{G}$ can be computed efficiently and there exists a group $\mathbb{G}_1$ and an efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ as above. Note that $e(,)$ is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

3 Complexity Assumptions

Let $\mathbb{G}$ be a bilinear group of prime order p and g be a generator of $\mathbb{G}$. We review the standard Bilinear Diffie-Hellman (BDH) assumption and define the Bilinear Diffie-Hellman Inversion (BDHI) assumption.

3.1 Bilinear Diffie-Hellman Assumption

The BDH problem [Jou00,BF01] in $\mathbb{G}$ is as follows: given a tuple $g, g^a, g^b, g^c \in \mathbb{G}$ as input, output $e(g, g)^{abc} \in \mathbb{G}_1$. An algorithm $\mathcal{A}$ has advantage ϵ in solving BDH in $\mathbb{G}$ if

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

where the probability is over the random choice of a, b, c in $\mathbb{Z}_p^*$ and the random bits used by $\mathcal{A}$. Similarly, we say that an algorithm $\mathcal{B}$ that outputs $b \in \{0, 1\}$ has advantage ϵ in solving the *decision* BDH problem in $\mathbb{G}$ if

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

where the probability is over the random choice of a, b, c in $\mathbb{Z}_p^*$, the random choice of $T \in \mathbb{G}_1^*$, and the random bits of $\mathcal{B}$.

Definition 3. *We say that the (Decision) (t, ϵ)-BDH assumption holds in $\mathbb{G}$ if no t-time algorithm has advantage at least ϵ in solving the (Decision) BDH problem in $\mathbb{G}$.*

Occasionally we drop the t and ϵ and refer to the BDH and Decision BDH assumptions in $\mathbb{G}$.

3.2 Bilinear Diffie-Hellman Inversion Assumption

The q-BDHI problem in the group $\mathbb{G}$ is defined as follows: given the $(q+1)$-tuple $(g, g^x, g^{(x^2)}, \ldots, g^{(x^q)}) \in (\mathbb{G}^*)^{q+1}$ as input, compute $e(g,g)^{1/x} \in \mathbb{G}_1^*$. An algorithm $\mathcal{A}$ has advantage ϵ in solving q-BDHI in $\mathbb{G}$ if

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

where the probability is over the random choice of x in $\mathbb{Z}_p^*$ and the random bits of $\mathcal{A}$. Similarly, we say that an algorithm $\mathcal{B}$ that outputs $b \in \{0,1\}$ has advantage ϵ in solving the *decision* q-BDHI problem in $\mathbb{G}$ if

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

where the probability is over the random choice of x in $\mathbb{Z}_p^*$, the random choice of $T \in \mathbb{G}_1^*$, and the random bits of $\mathcal{B}$.

Definition 4. *We say that the (Decision) (t, q, ϵ)-BDHI assumption holds in $\mathbb{G}$ if no t-time algorithm has advantage at least ϵ in solving the (Decision) q-BDHI problem in $\mathbb{G}$.*

Occasionally we drop the t and ϵ and refer to the q-BDHI and Decision q-BDHI assumptions. It is easy to show that the 1-BDHI assumption is equivalent to the standard Bilinear Diffie-Hellman assumption (BDH). It is not known if the q-BDHI assumption, for $q > 1$, is equivalent to BDH.

4 Efficient Selective Identity HIBE Based on BDH Without Random Oracles

We construct an efficient HIBE system that is selective identity secure without random oracles based on the Decision BDH assumption. In particular, this implies an efficient selective identity, chosen ciphertext secure IBE based on Decision BDH without random oracles.

4.1 Construction

Let $\mathbb{G}$ be a bilinear group of prime order p and g be a generator of $\mathbb{G}$ (the security parameter determines the size of $\mathbb{G}$). Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ be the bilinear map. For now, we assume public keys (ID) of depth ℓ are vectors of elements in ${\mathbb{Z}_p}^\ell$. We write $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_\ell) \in {\mathbb{Z}_p}^\ell$. The j-th component corresponds to the identity at level j. We later extend the construction to public keys over $\{0,1\}^*$ by first hashing each component I_j using a collision resistant hash $H : \{0,1\}^* \to \mathbb{Z}_p$. We also assume messages to be encrypted are elements in $\mathbb{G}_1$. The HIBE system works as follows:

Setup(ℓ)**:** To generate system parameters for an HIBE of maximum depth ℓ, select a random $\alpha \in \mathbb{Z}_p^*$ and set $g_1 = g^\alpha$. Next, pick random elements $h_1, \ldots, h_\ell \in \mathbb{G}$ and a generator $g_2 \in \mathbb{G}^*$. The public parameters *params* and the secret *master-key* are given by

$$params = (g, g_1, g_2, h_1, \ldots, h_\ell) \;, \qquad master\text{-}key = g_2^\alpha$$

For $j = 1, \ldots, \ell$, we define $F_j : \mathbb{Z}_p \to \mathbb{G}$ to be the function: $\quad F_j(x) = g_1^x h_j$.

KeyGen$(d_{\mathsf{ID}|j-1}, \mathsf{ID})$**:** To generate the private key d_{ID} for an identity $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j) \in {\mathbb{Z}_p}^j$ of depth $j \le \ell$, pick random $r_1, \ldots, r_k \in \mathbb{Z}_p$ and output

$$d_{\mathsf{ID}} = \left(g_2^\alpha \cdot \prod_{k=1}^{j} F_k(\mathrm{I}_k)^{r_k}, \;\; g^{r_1}, \;\; \ldots, \;\; g^{r_j} \right)$$

Note that the private key for ID can be generated just given a private key for $\mathsf{ID}_{|j-1} = (\mathrm{I}_1, \ldots, \mathrm{I}_{j-1}) \in {\mathbb{Z}_p}^{j-1}$, as required. Indeed, let $d_{\mathsf{ID}|j-1} = (d_0, \ldots, d_{j-1})$ be the private key for $\mathsf{ID}_{|j-1}$. To generate d_{ID} pick a random $r_j \in \mathbb{Z}_p$ and output $d_{\mathsf{ID}} = (d_0 \cdot F_j(\mathrm{I}_j)^{r_j}, \, d_1, \ldots, \, d_{j-1}, \, g^{r_j})$.

Encrypt$($***params***, ID, **M**$)$**:** To encrypt a message $M \in \mathbb{G}_1$ 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

$$C = \left(e(g_1, g_2)^s \cdot M, \quad g^s, \;\; F_1(\mathrm{I}_1)^s, \;\; \ldots, \;\; F_j(\mathrm{I}_j)^s \right)$$

Note that $e(g_1, g_2)$ can be precomputed once and for all so that encryption does not require any pairing computations. Alternatively, $e(g_1, g_2)$ can be included in the system parameters.

Decrypt(d_{ID}, C)**:** Let $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j)$ be an identity. To decrypt a ciphertext $C = (A, B, C_1, \ldots, C_j)$ using the private key $d_{\mathsf{ID}} = (d_0, d_1, \ldots, d_j)$, output

$$A \cdot \frac{\prod_{k=1}^{j} e(C_j, d_j)}{e(B, d_0)} = M$$

Indeed, for a valid ciphertext, we have

$$\frac{\prod_{k=1}^{j} e(C_j, d_j)}{e(B, d_0)} = \frac{\prod_{k=1}^{j} e(F_k(\mathrm{I}_k), g)^{s r_k}}{e(g, g_2)^{s\alpha} \prod_{k=1}^{j} e(g, F_k(\mathrm{I}_k))^{s r_k}} = \frac{1}{e(g_1, g_2)^s}$$

4.2 Security

The HIBE system above is reminiscent of the Gentry-Silverberg HIBE which is only known to be secure in the random oracle model. Surprisingly, our choice of functions $F_1, \ldots, F_\ell$ enables us to prove security *without random oracles*. We prove security of our HIBE system under the standard Decision BDH assumption in $\mathbb{G}$.

Theorem 1. *Suppose the (t, ϵ)-Decision BDH assumption holds in $\mathbb{G}$. Then the previously defined ℓ-HIBE system is (t', q_S, ϵ)-selective identity, chosen plaintext* (IND-sID-CPA) *secure for arbitrary ℓ and q_S and any $t' < t - o(t)$.*

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 $\mathbb{G}$. On input (g, g^a, g^b, g^c, T) algorithm $\mathcal{B}$'s goal is to output 1 if $T = e(g,g)^{abc}$ and 0 otherwise. Let $g_1 = g^a,\ g_2 = g^b,\ g_3 = g^c$. 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 necessary, $\mathcal{B}$ appends random elements in $\mathbb{Z}_p$ to ID^* so that ID^* is a vector of length ℓ.

Setup. To generate the system parameters, algorithm $\mathcal{B}$ picks $\alpha_1, \ldots, \alpha_\ell \in \mathbb{Z}_p$ at random and defines $h_j = g_1^{-\mathrm{I}_j^*} g^{\alpha_j} \in \mathbb{G}$ for $j = 1, \ldots, \ell$. It gives $\mathcal{A}$ the system parameters $params = (g, g_1, g_2, h_1, \ldots, h_\ell)$. Note that the corresponding master-key, which is unknown to $\mathcal{B}$, is $g_2^a = g^{ab} \in \mathbb{G}^*$. As before, for $j = 1, \ldots, \ell$ we define $F_j : \mathbb{Z}_p \to \mathbb{G}$ to be the function

$$F_j(x) = g_1^x h_j = g_1^{x - \mathrm{I}_j^*} g^{\alpha_j}$$

Phase 1. $\mathcal{A}$ issues up to q_S private key queries. Consider a query for the private key corresponding to $\mathsf{ID} = (\mathrm{I}_1, \ldots, I_u) \in {\mathbb{Z}_p}^u$ where $u \le \ell$. The only restriction is that ID is not a prefix of ID^*. Let j be the smallest index such that $\mathrm{I}_j \ne \mathrm{I}_j^*$. Necessarily $1 \le j \le u$. 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 $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j, \ldots, \mathrm{I}_u)$. Algorithm $\mathcal{B}$ picks random elements $r_1, \ldots, r_j \in \mathbb{Z}_p$ and sets

$$d_0 = g_2^{\frac{-\alpha_j}{\mathrm{I}_j - \mathrm{I}_j^*}} \prod_{v=1}^{j} F_v(\mathrm{I}_v)^{r_v}, \quad d_1 = g^{r_1}, \quad \ldots, \quad d_{j-1} = g^{r_{j-1}}, \quad d_j = g_2^{\frac{-1}{\mathrm{I}_j - \mathrm{I}_j^*}} g^{r_j}$$

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

$$g_2^{\frac{-\alpha_j}{(\mathrm{I}_j - \mathrm{I}_j^*)}} F_j(\mathrm{I}_j)^{r_j} = g_2^{\frac{-\alpha_j}{(\mathrm{I}_j - \mathrm{I}_j^*)}} (g_1^{\mathrm{I}_j - \mathrm{I}_j^*} g^{\alpha_j})^{r_j} = g_2^a (g_1^{\mathrm{I}_j - \mathrm{I}_j^*} g^{\alpha_j})^{r_j - \frac{b}{\mathrm{I}_j - \mathrm{I}_j^*}} = g_2^a F_j(\mathrm{I}_j)^{\tilde{r}_j}$$

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

$$d_0 = g_2^a \cdot (\prod_{v=1}^{j-1} F_v(\mathrm{I}_v)^{r_v}) \cdot F_j(\mathrm{I}_j)^{\tilde{r}_j}, \quad d_1 = g^{r_1}, \quad \ldots, \quad d_{j-1} = g^{r_{j-1}}, \quad d_j = g^{\tilde{r}_j}$$

where $r_1, \ldots, r_{j-1}, \tilde{r}_j$ are uniform in $\mathbb{Z}_p$. This matches the definition for a private key for $(\mathrm{I}_1, \ldots, \mathrm{I}_j)$. Hence, $(d_0, d_1, \ldots, d_j)$ is a valid private key for $(\mathrm{I}_1, \ldots, \mathrm{I}_j)$. Algorithm $\mathcal{B}$ derives a private key for the requested ID from the private key $(d_0, d_1, \ldots, d_j)$ and gives the result to $\mathcal{A}$.

Challenge. When $\mathcal{A}$ decides that Phase 1 is over, it outputs two messages $M_0, M_1 \in \mathbb{G}_1$. Algorithm $\mathcal{B}$ picks a random bit $b \in \{0,1\}$ and responds with the ciphertext $C = (M_b \cdot T,\ g_3,\ g_3^{\alpha_1}, \ldots,\ g_3^{\alpha_k})$. Since $F_i(\mathrm{I}_i^*) = g^{\alpha_i}$ for all i, we have that

$$C = (M_b \cdot T,\ \ g^c,\ \ F_1(\mathrm{I}_1^*)^c,\ \ \ldots,\ \ F_k(\mathrm{I}_k^*)^c)$$

Hence, if $T = e(g,g)^{abc} = e(g_1, g_2)^c$ then C is a valid encryption of M_b under the public key $\mathsf{ID}^* = (\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*)$. Otherwise, C is independent of b in the adversary's view.

Phase 2. $\mathcal{A}$ issues its complement of private key queries not issued in Phase 1. Algorithm $\mathcal{B}$ responds as before.

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

When $T = e(g,g)^{abc}$ then $\mathcal{A}$ must satisfy $|\Pr[b = b'] - 1/2| > \epsilon$. When T is uniform in $\mathbb{G}_1^*$ then $\Pr[b = b'] = 1/2$. Therefore, when a, b, c are uniform in $\mathbb{Z}_p^*$ and T is uniform in $\mathbb{G}_1^*$ we have that

$$\begin{aligned}\Big|\Pr\big[\mathcal{B}(g, g^a, g^b, g^c, e(g,g)^{abc}) = 0\big] - \Pr\big[\mathcal{B}(g, g^a, g^b, g^c, T) = 0\big]\Big| \\ \geq\ |(\frac{1}{2} \pm \epsilon) - \frac{1}{2}|\ =\ \epsilon\end{aligned}$$

as required. This completes the proof of Theorem 1. □

4.3 Chosen Ciphertext Security

A recent result of Canetti et al. [CHK04] gives an efficient way to build a selective identity, chosen ciphertext ℓ-HIBE from a selective identity, chosen plaintext $(\ell+1)$-HIBE. In combination with the above construction, we obtain a selective identity, chosen ciphertext ℓ-HIBE for any ℓ. In particular, we can easily construct an efficient selective identity, chosen ciphertext secure IBE without random oracles.

4.4 Arbitrary Identities

We can extend our HIBE above to handle identities $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_\ell)$ with $\mathrm{I}_j \in \{0,1\}^*$ (as opposed to $I_j \in \mathbb{Z}_p$) by first hashing each I_j using a collision resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_p$ prior to key generation and encryption. A standard argument shows that if the scheme above is selective identity, chosen ciphertext secure then so is the scheme with the additional hash function. We note that there is no need for a full domain hash into $\mathbb{Z}_p$; for example, a collision resistant hash function $H : \{0,1\}^* \to \{1, \ldots, 2^b\}$ where $2^b < p$ is sufficient for the security proof.

5 More Efficient Selective Identity IBE Based on BDHI Without Random Oracles

We construct an efficient IBE system that is selective identity, chosen plaintext secure without random oracles based on the Decision q-BDHI assumption (see Section 3.2). The resulting IBE system is more efficient that the IBE construction in the previous section.

5.1 Basic Construction

Let $\mathbb{G}$ be a bilinear group of prime order p and g be a generator of $\mathbb{G}$. For now, we assume that the public keys (ID) are elements in $\mathbb{Z}_p^*$. We show later that arbitrary identities in $\{0,1\}^*$ can be used by first hashing ID using a collision resistant hash $H : \{0,1\}^* \to \mathbb{Z}_p^*$. We also assume that the messages to be encrypted are elements in $\mathbb{G}_1$. The IBE system works as follows:

***Setup*:** To generate IBE parameters, select random elements $x, y \in \mathbb{Z}_p^*$ and define $X = g^x$ and $Y = g^y$. The public parameters *params* and the secret *master-key* are given by

$$\textit{params} = (g, g^x, g^y)\ , \qquad \textit{master-key} = (x, y)$$

***KeyGen*(*master-key*,ID):** To create a private key for the public key $\mathsf{ID} \in \mathbb{Z}_p^*$:

1. pick a random $r \in \mathbb{Z}_p$ and compute $K = g^{1/(\mathsf{ID}+x+ry)} \in \mathbb{G}$,
2. output the private key $d_{\mathsf{ID}} = (r, K)$.

In the unlikely event that $x + ry + \mathsf{ID} = 0 \pmod p$, try again with a new random value for r.

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

$$C = \left(g^{s \cdot \mathsf{ID}} X^s, \quad Y^s, \quad e(g,g)^s \cdot M\right)$$

Note that $e(g,g)$ can be precomputed once and for all so that encryption does not require any pairing computations.

***Decrypt*(d_{ID}, C):** To decrypt a ciphertext $C = (A, B, C)$ using the private key $d_{\mathsf{ID}} = (r, K)$, output $C/e(AB^r, K)$. Indeed, for a valid ciphertext we have

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

Performance. In terms of efficiency, we note that the ciphertext size and encryption time are similar to the IBE system of the previous section. However, decryption requires only one pairing computation, as opposed to two in the previous section.

5.2 Proving Security

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

Theorem 2. *Suppose the (t, q, ϵ)-Decision BDHI assumption holds in $\mathbb{G}$. Then the previously defined IBE system is (t', q_S, ϵ)-selective identity, chosen plaintext* (IND-sID-CPA) *secure for any $q_S < q$ and $t' < t - o(t)$.*

Proof. Suppose $\mathcal{A}$ has advantage ϵ in attacking the IBE system. We build an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to solve the decision q-BDHI problem in $\mathbb{G}$. On input $(g, g^{\alpha}, g^{\alpha^2}, \ldots, g^{\alpha^q}, T) \in (\mathbb{G}^*)^{q+1} \times \mathbb{G}_1^*$ for some unknown $\alpha \in \mathbb{Z}_p^*$, the goal of $\mathcal{B}$ is to output 1 if $T = e(g, g)^{1/\alpha}$ and 0 otherwise. It does so by interacting with $\mathcal{A}$ in a selective identity game as follows:

Preparation. Algorithm $\mathcal{B}$ builds a generator $h \in \mathbb{G}^*$ for which it knows $q - 1$ pairs of the form $(w_i, h^{1/(\alpha+w_i)})$ 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 let $f(z)$ be the polynomial $f(z) = \prod_{i=1}^{q-1}(z + w_i)$. Expand the terms of f to get $f(z) = \sum_{i=0}^{q-1} c_i x^i$. The constant term c_0 is non-zero.
2. Compute $h = \prod_{i=0}^{q-1}(g^{(\alpha^i)})^{c_i} = g^{f(\alpha)}$ and $u = \prod_{i=1}^{q}(g^{(\alpha^i)})^{c_{i-1}} = g^{\alpha f(\alpha)}$. Note that $u = h^{\alpha}$.
3. Check that $h \in \mathbb{G}^*$. Indeed if we had $h = 1$ in $\mathbb{G}$ 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 directly. We thus assume that all $w_j \neq -\alpha$.
4. Observe that for any $i = 1, \ldots, q-1$, it is easy for $\mathcal{B}$ to construct the pair $\left(w_i, h^{1/(\alpha+w_i)}\right)$. To see this, write $f_i(z) = f(z)/(z + w_i) = \sum_{i=0}^{q-2} d_i z^i$. Then $h^{1/(\alpha+w_i)} = g^{f_i(\alpha)} = \prod_{i=0}^{q-2}(g^{(\alpha^i)})^{d_i}$.
5. Next, $\mathcal{B}$ computes

$$T_h = T^{(c_0^2)} \cdot T_0 \qquad \text{where} \qquad T_0 = \prod_{i=0}^{q-1}\prod_{j=0}^{q-2} e\left(g^{(\alpha^i)}, g^{(\alpha^j)}\right)^{c_i c_{j+1}}$$

   Observe that if $T = e(g, g)^{1/\alpha}$ then $T_h = e\left(g^{f(\alpha)/\alpha}, g^{f(\alpha)}\right) = e(h, h)^{1/\alpha}$. On the contrary, if T is uniform in $\mathbb{G}_1^*$, then T_h is uniform in $\mathbb{G}_1 \setminus \{T_0\}$.

We will be using the values h, u, T_h and the pairs $(w_i, h^{1/(\alpha+w_i)})$ for $i = 1, \ldots, q-1$ throughout the simulation.

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

Setup. To generate the system parameters *params* $= (g, X, Y)$, algorithm $\mathcal{B}$ does the following:

1. Pick random $a, b \in \mathbb{Z}_p^*$ under the constraint that $ab = \mathsf{ID}^*$.
2. Compute $X = u^{-a}h^{-ab} = h^{-a(\alpha+b)}$ and $Y = u = h^{\alpha}$.
3. Publish *params* $= (h, X, Y)$ as the public parameters. Note that X, Y are independent of ID^* in the adversary's view.

4. We implicitly define $x = -a(\alpha + b)$ and $y = \alpha$ so that $X = h^x$ and $Y = h^y$. 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. $\mathcal{A}$ issues up to $q_S < q$ private key queries. Consider the i-th query for the private key corresponding to public key $\mathsf{ID}_i \neq \mathsf{ID}^*$. We need to respond with a private key $(r, h^{1/(\mathsf{ID}_i+x+ry)})$ for a uniformly distributed $r \in \mathbb{Z}_p$. Algorithm $\mathcal{B}$ responds to the query as follows:

1. Let $\left(w_i, h^{1/(\alpha+w_i)}\right)$ be the i-th pair constructed during the preparation step. Define $h_i = h^{1/(\alpha+w_i)}$.
2. $\mathcal{B}$ first constructs an $r \in \mathbb{Z}_p$ satisfying $(r-a)(\alpha + w_i) = \mathsf{ID}_i + x + ry$. Plugging in the values of x and y the equation becomes

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

We see that α cancels from the equation and we get $r = a + \frac{\mathsf{ID}_i - ab}{w_i} \in \mathbb{Z}_p$.

3. Now, $(r,\ h_i^{1/(r-a)})$ is a valid private key for ID_i for two reasons. First,

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

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 $(r,\ h_i^{1/(r-a)})$.

We point out that this procedure will fail to produce the private key for ID^* since in that case we get $r - a = 0$. Hence, $\mathcal{B}$ can generate private keys for *all* public keys except for ID^*.

Challenge. $\mathcal{A}$ outputs two messages $M_0, M_1 \in \mathbb{G}_1$. Algorithm $\mathcal{B}$ picks a random bit $b \in \{0,1\}$ and a random $\ell \in \mathbb{Z}_p^*$. It responds with the ciphertext $CT = (h^{-a\ell},\ h^\ell,\ T_h^\ell \cdot M_b)$. Define $s = \ell/\alpha$. On the one hand, if $T_h = e(h,h)^{1/\alpha}$ we have

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

It follows that CT is a valid encryption of M_b under ID^*, with the uniformly distributed randomization value $s = \ell/\alpha \in \mathbb{Z}_p^*$. On the other hand, when T_h is uniform in $\mathbb{G}_1 \setminus \{T_0\}$, then, in the adversary's view, CT is independent of the bit b.

Phase 2. $\mathcal{A}$ issues more private key queries, for a total of at most $q_S < q$. Algorithm $\mathcal{B}$ responds as before.

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

We showed that when the input T satisfies $T = e(g,g)^{1/\alpha}$ then $T_h = e(h,h)^{1/\alpha}$ in which case $\mathcal{A}$ must satisfy $|\Pr[b = b'] - 1/2| > \epsilon$. On the other hand, when T is uniform and independent in $\mathbb{G}_1^*$ then T_h is uniform and independent in $\mathbb{G}_1 \setminus \{T_0\}$ in which case $\Pr[b = b'] = 1/2$. Therefore, when x is uniform in $\mathbb{Z}_p^*$ and P is uniform in $\mathbb{G}_1^*$ we have that

$$\left|\Pr\left[\mathcal{B}(g, g^x, \ldots, g^{(x^q)}, e(g,g)^{1/x}) = 0\right] - \Pr\left[\mathcal{B}(g, g^x, \ldots, g^{(x^q)}, P) = 0\right]\right|$$
$$\geq \ |(\frac{1}{2} \pm \epsilon) - \frac{1}{2}| \ = \ \epsilon$$

as required. This completes the proof of Theorem 2. □

5.3 Chosen-Ciphertext Security and Arbitrary Identities

Canetti et al. [CHK03, Section 2.2] describe a general method for converting a selective identity, chosen plaintext secure IBE into a selective identity, chosen ciphertext secure IBE. The method is based on [NY90,Sah99,Lin03]. Since it is generic, it applies to our system as well. In particular, the method can be used to render the IBE system above secure against chosen ciphertext attacks. The result is an IND-sID-CCA secure IBE without random oracles. However, the resulting system is inefficient since it relies on generic non-interactive zero-knowledge (NIZK) constructions.

As before, a standard argument shows that we can extend the IBE above to handle arbitrary identities $\mathsf{ID} \in \{0,1\}^*$ by first hashing ID using a collision resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_p^*$ prior to key generation and encryption. If the underlying scheme is selective identity, chosen plaintext (resp. ciphertext) secure, then so is the scheme with the additional hash function.

5.4 An Efficient CCA2-Secure Public-Key System

A recent result of Canetti et al. [CHK04] gives a general method for constructing a CCA2 public key system from any selective identity, chosen plaintext IBE. Essentially the same result was used in Section 4 to transform our first HIBE construction into a chosen ciphertext secure HIBE of lesser depth.

When used on the construction of this section, we obtain a new efficient CCA2 public key system. We briefly summarize its characteristics:

1. Encryption time: Dominated by three exponentiations in $\mathbb{G}$.
2. Decryption time: Dominated by one pairing computation.
3. Ciphertext size: Composed of three elements of $\mathbb{G}$ plus a public key and signature of a one-time signature scheme.

In terms of performance, this is comparable to, though not quite as efficient as, the Cramer-Shoup [CS98] CCA2-secure public key system which is proven secure in the standard model.

The ciphertext size can be further reduced by using the short signature scheme recently proposed by Boneh and Boyen [BB04] instead of the one-time signatures suggested by Canetti et al. [CHK04]. The Boneh-Boyen signature scheme is existentially unforgeable in the strong sense (sUF-CMA) without random oracle, and thus satisfies the requirements of the CCA2 construction. Here, strong existential unforgeability means that it is infeasible for an adversary to forge a new signature even on messages for which one or more valid signatures are already known.

6 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 [MSK02] where it was called weak Diffie-Hellman.

Many cryptographic constructions rely on the Generalized Diffie-Hellman assumption (GenDH) for security [MSW96,NR97,BBR99,Lys02,BS03]. 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 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 any strict subset $S \subset \{1, \ldots, q\}$, 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 all these subset products is provided through an oracle. For a vector $\boldsymbol{a} = (a_1, \ldots, a_q) \in {\mathbb{Z}_p}^q$, define $\mathcal{O}_{g,\boldsymbol{a}}$ to be an oracle that for any strict subset $S \subset \{1, \ldots, q\}$ responds with

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

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 $\mathcal{O}_{g,\boldsymbol{a}}(S)$. In other words,

$$\mathrm{Adv}_{\mathcal{A},q} = \Pr[\mathcal{A}^{\mathcal{O}_{g,\boldsymbol{a}}} = g^{a_1 \cdots a_q} \ : \ \boldsymbol{a} = (a_1, \ldots, a_q) \leftarrow {\mathbb{Z}_p}^q]$$

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

Definition 5. *We say that $\mathbb{G}$ satisfies the (t, q, ϵ)-Generalized Diffie-Hellman assumption if for all t-time algorithms $\mathcal{A}$ we have $Adv_{\mathcal{A},q} < \epsilon$.*

Theorem 3. *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 $\mathcal{O}_{h,\boldsymbol{a}}$ for $\mathcal{A}$. The vector $\boldsymbol{a}$ that $\mathcal{B}$ will use is $\boldsymbol{a} = (y + c_1, \ldots, y + c_q)$. Note that $\mathcal{B}$ does not know $\boldsymbol{a}$ explicitly since $\mathcal{B}$ does not have y. When $\mathcal{A}$ issues a query for $\mathcal{O}_{h,\boldsymbol{a}}(S)$ 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 $\mathcal{O}_{h,\boldsymbol{a}}(S) = t$.

The responses to all of the adversary's oracle queries are consistent with the hidden vector $\boldsymbol{a} = (y + c_1, \ldots, y + c_q)$. 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 3 also holds for the decision version 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.

7 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 construction is based on the now classic BDH assumption. It extends readily to give a selective identity HIBE without random oracles, that can efficiently be made chosen ciphertext secure using a technique of [CHK04]. The second construction is based on the Bilinear Diffie-Hellman Inversion assumption. The same technique of [CHK04] converts both our constructions into efficient CCA2-secure public key systems without random oracles that are almost as efficient as the Cramer-Shoup public key system.

Currently, the problem of constructing a fully secure IBE (against adaptive identity attacks) without resorting to random oracles is still open. We hope to see this question resolved soon.

References

[BB04] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *Proceedings of Eurocrypt 2004*, LNCS. Springer-Verlag, 2004.

[BBR99] Eli Biham, Dan Boneh, and Omer Reingold. Breaking generalized Diffie-Hellman modulo a composite is no easier than factoring. *IPL*, 70:83–87, 1999.

[BF01] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *Proceedings of Crypto 2001*, volume 2139 of *LNCS*, pages 213–29. Springer-Verlag, 2001.

[BF03] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. *SIAM J. of Computing*, 32(3):586–615, 2003.

[BR93] 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.

[BS03] Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, 324:71–90, 2003.

[CHK03] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In *Proceedings of Eurocrypt 2003*, volume 2656 of *LNCS*. Springer-Verlag, 2003.

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In *Proceedings of Eurocrypt 2004*, LNCS, 2004. `http://eprint.iacr.org/2003/182/`.

[Coc01] 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.

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attacks. In Hugo Krawczyk, editor, *Proceedings of Crypto 1998*, volume 1462 of *LNCS*, pages 13–25. Springer-Verlag, 1998.

[GS02] Craig Gentry and Alice Silverberg. Hierarchical ID-based cryptography. In *Proceedings of Asiacrypt 2002*, 2002.

[HL02] J. Horwitz and B. Lynn. Towards hierarchical identity-based encryption. In *Proceedings of Eurocrypt 2002*, pages 466–481, 2002.

[Jou00] 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.

[Lin03] Yehuda Lindell. A simpler construction of CCA2-secure public-key encryption under general assumptions. In *Proceeings of Eurocrypt '03*, volume 2656 of *LNCS*, pages 241–254, 2003.

[Lys02] Anna Lysyanskaya. Unique signatures and verifiable random functions from the DH-DDH separation. In *Proceedings of Crypto 2002*, LNCS. Springer-Verlag, 2002.

[MSK02] Shigeo Mitsunari, Ryuichi Sakai, and Masao Kasahara. A new traitor tracing. *IEICE Trans. Fundamentals*, E85-A(2):481–484, 2002.

[MSW96] G. Tsudik M. Steiner and M. Waidner. Diffie-Hellman key distribution extended to groups. In *Proceedings 1996 ACM Conference on Computer and Communications Security*, 1996.

[NR97] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *Proceeings 38th IEEE Symp. on Foundations of Computer Science*, pages 458–467, 1997.

[NY90] Moni Naor and Moti Yung. Public key cryptosystems provable secure against chosen ciphertext attacks. In *STOC '90*, pages 427–437. ACM, 1990.

[Sah99] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *Proceeings 40 IEEE Symp. on Foundations of Computer Science*, 1999.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of Crypto '84*, volume 196 of *LNCS*, pages 47–53. Springer-Verlag, 1984.