Efficient Identity-Based Encryption Without Random Oracles

Brent Waters

Stanford University
bwaters@cs.stanford.edu

Abstract. We present the first efficient Identity-Based Encryption (IBE) scheme that is fully secure without random oracles. We first present our IBE construction and reduce the security of our scheme to the decisional Bilinear Diffie-Hellman (BDH) problem. Additionally, we show that our techniques can be used to build a new signature scheme that is secure under the computational Diffie-Hellman assumption without random oracles.

1 Introduction

Identity-Based Encryption allows for a party to encrypt a message using the recipient's identity as a public key. The ability to use identities as public keys avoids the need to distribute public key certificates. This can be very useful in applications such as email where the recipient is often off-line and unable to present a public-key certificate while the sender encrypts a message.

The first efficient and secure method for Identity-Based Encryption was put forth by Boneh and Franklin [4]. They proposed a solution using efficiently computable bilinear maps that was shown to be secure in the random oracle model. Since then, there have been schemes shown to be secure without random oracles, but in a weaker model of security know as the Selective-ID model [9, 1]. Most recently, Boneh and Boyen [2] described a scheme that was proved to be fully secure without random oracles; the possibility of such a scheme was to that point an open problem. However, their scheme is too inefficient to be of practical use.

We present the first efficient Identity-Based Encryption scheme that is fully secure without random oracles. The proof of our scheme makes use of an algebraic method first used by Boneh and Boyen [1] and the security of our scheme reduces to the decisional Bilinear Diffie-Hellman (BDH) assumption.

We additionally show that our IBE scheme implies a secure signature scheme under the computational Diffie-Hellman assumption without random oracles. Previous practical signature schemes that were secure in the standard model relied on the Strong-RSA assumption [12, 11] or the Strong-BDH assumption [3].

1.1 Related Work

Shamir [16] first presented the idea of Identity-Based Encryption as a challenge to the research community. However, the first secure and efficient scheme of

Boneh and Franklin[4] did not appear until much later. The authors took a novel approach in using efficiently computable bilinear maps in order to achieve their result.

Canetti et. al. [9] describe a weaker model of security for Identity-Based Encryption that they term the *Selective-ID* model. In the Selective-ID model the adversary must first declare which identity it wishes to be challenged on before the global parameters are generated. The authors provide a scheme that is provably secure in the Selective-ID model without random oracles. Boneh and Boyen [1] improve upon this result by describing an efficient scheme that is secure in the Selective-ID model.

Finally, Boneh and Boyen [2] describe a scheme that is fully secure without random oracles. However, their construction is too inefficient to be of practical use.

1.2 Organization

We organize the rest of the paper as follows. In Section 2 we give our security definition. In Section 3 we describe our complexity assumptions. In Section 4 we present the construction of our IBE scheme and follow with a proof of security in Section 5. In Section 6 we discuss how our scheme can be extended to a hierarchical identity-based encryption scheme and how that can be used to achieve CCA-security. We discuss the transformation to a signature scheme in Section 7. Finally, we conclude in Section 8.

2 Security Definitions

In this section we present the definition of semantic security against passive adversaries for Identity-Based Encryption. This definition was first described by Boneh and Franklin [4]. Consider the following game played by an adversary. The game has four distinct phases:

Setup The challenger generates the master public parameters and gives them to the adversary.

Phase 1 The adversary is allowed to make a query for a private key, v, where v is an identity specified by the adversary. The adversary can repeat this multiple times for different identities.

Challenge The adversary submits a public key, v^*, and two messages M_0 and M_1. The adversary's choice of v^* is restricted to the identities that he did not request a private key for in Phase 1. The challenger flips a fair binary coin,γ, and returns an encryption of M_γ under the public key v^*.

Phase 2 Phase 1 is repeated with the restriction that the adversary cannot request the private key for v^*.

Guess The adversary submits a guess, γ', of γ.

Definition 1 (IBE Semantic Security). *An Identity-Based Encryption scheme is (t, q, ϵ)-semantically secure if all t-time adversaries making at most q private key queries have at most an ϵ in breaking our scheme.*

3 Complexity Assumptions

We briefly review the facts about groups with efficiently computable bilinear maps. We refer the reader to previous literature [4] for more details.

Let $\mathbb{G}, \mathbb{G}_1$ be s groups of prime order p and g be a generator of $\mathbb{G}_1$. We say $\mathbb{G}_1$ has an admissible bilinear map, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$, into $\mathbb{G}_1$ if the following two conditions hold. The map is bilinear; for all a, b we have $e(g^a, g^b) = e(g, g)^{ab}$. The map is non-degenerate; we must have that $e(g, g) \neq 1$.

3.1 Decisional Bilinear Diffie-Hellman (BDH) Assumption

The challenger chooses $a, b, c, z \in \mathbb{Z}_p$ at random and then flips a fair binary coin β. If $\beta = 1$ it outputs the tuple $(g, A = g^a, B = g^b, C = g^c, Z = e(g, g)^{abc})$. Otherwise, if $\beta = 0$, the challenger outputs the tuple $(g, A = g^a, B = g^b, C = g^c, Z = e(g, g)^z)$. The adversary must then output a guess β' of β.

An adversary, $\mathcal{B}$, has at least an ϵ advantage in solving the decisional BDH problem if

$$\Big| \Pr\left[\mathcal{B}\left(g, g^a, g^b, g^c, e(g,g)^{abc}\right) = 1\right] - \Pr\left[\mathcal{B}\left(g, g, g^a, g^b, g^c, e(g,g)^{z}\right) = 1\right] \Big| \geq 2\epsilon$$

where the probability is over the randomly chosen a, b, c, z and the random bits consumed by $\mathcal{B}$. We refer to the left hand side as $\mathcal{P}_{BDH}$ and the right hand side as $\mathcal{R}_{BDH}$.

Definition 2. *The decisional (t, ϵ)-BDH assumption holds if no t-time adversary has at least ϵ advantage in solving the above game.*

3.2 Computational Diffie-Hellman (DH) Assumption

The challenger chooses $a, b \in \mathbb{Z}_p$ at random and outputs $(g, A = g^a, B = g^b)$. The adversary then attempts to output $g^{ab} \in \mathbb{G}$. An adversary, $\mathcal{B}$, has at least an ϵ advantage if

$$Pr\left[\mathcal{B}\left(g, g^a, g^b\right) = g^{ab}\right] \geq \epsilon$$

where the probability is over the randomly chosen a, b and the random bits consumed by $\mathcal{B}$.

Definition 3. *The computational* (t, ϵ)*-DH assumption holds if no* t*-time adversary has at least* ϵ *advantage in solving the above game.*

4 Construction

Our construction can be viewed as a modification of the Boneh-Boyen [1] scheme. We first present our construction then describe its relation to the Boneh-Boyen scheme.

Let $\mathbb{G}$ be a group of prime order, p, for which there exists an efficiently computable bilinear map into $\mathbb{G}_1$. Additionally, let $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$ denote the bilinear map and g be the corresponding generator. The size of the group is determined by the security parameter. Identities will be represented as bitstrings of length n, a separate parameter unrelated to p. We can also let identities be bitstrings of arbitrary length and n be the output length of a collision-resistant hash function, $H : \{0,1\}^* \rightarrow \{0,1\}^n$. Our construction follows.

Setup The system parameters are generated as follows. A secret $\alpha \in \mathbb{Z}_p$ is chosen at random. We choose a random generator, $g \in \mathbb{G}$, and set the value $g_1 = g^\alpha$ and choose g_2 randomly in $\mathbb{G}$. Additionally, the authority chooses a random value $u' \in \mathbb{G}$ and a random n-length vector $U = (u_i)$, whose elements are chosen at random from $\mathbb{G}$. The published public parameters are g,g_1, g_2,u', and U. The master secret is g_2^α.

Key Generation Let v be an n bit string representing an identity, v_i denote the *ith* bit of v, and $\mathcal{V} \subseteq \{1, \ldots, n\}$ be the set of all i for which $v_i = 1$. (That is $\mathcal{V}$ is the set of indicies for which the bitstring v is set to 1.) A private key for identity v is generated as follows. First, a random $r \in \mathbb{Z}_p$ is chosen. Then the private key is constructed as:

$$d_v = \left(g_2^\alpha \left(u' \prod_{i \in \mathcal{V}} u_i \right)^r, g^r \right).$$

Encryption A message $M \in \mathbb{G}_1$ is encrypted for an identity v as follows. A value $t \in \mathbb{Z}_p$ is chosen at random. The ciphertext is then constructed as

$$C = \left(e(g_1, g_2)^t M, g^t, \left(u' \prod_{i \in \mathcal{V}} u_i \right)^t \right).$$

Decryption Let $C = (C_1, C_2, C_3)$ be a valid encryption of M under the identity v. Then C can be decrypted by $d_v = (d_1, d_2)$ as:

$$C_1 \frac{e(d_2, C_3)}{e(d_1, C_2)} = \left(e(g_1, g_2)^t M\right) \frac{e(g^r, \left(u' \prod_{i \in \mathcal{V}} u_i\right)^t)}{e(g_2^\alpha \left(u' \prod_{i \in \mathcal{V}} u_i\right)^r, g^t)}$$

$$= \left(e(g_1, g_2)^t M\right) \frac{e(g, \left(u' \prod_{i \in \mathcal{V}} u_i\right)^{rt})}{e(g_1, g_2)^t e(\left(u' \prod_{i \in \mathcal{V}} u_i\right)^{rt}, g)} = M.$$

4.1 Efficiency

If the value of $e(g_1, g_2)$ is cached then encryption requires on average $\frac{n}{2}$ (and at most n) group operations in $\mathbb{G}$, two exponentiations in $\mathbb{G}$, one exponentiation in $\mathbb{G}_1$, and one group operation in $\mathbb{G}_1$. Decryption requires two bilinear map computations, one group operation in $\mathbb{G}_1$ and one inversion in $\mathbb{G}_1$.

4.2 Similarity to Boneh-Boyen

Our construction is a modification of Boneh and Boyen's [1] in that that for an identity v we evaluate $u' \prod_{i \in \mathcal{V}} u_i$ whereas in their scheme they evaluate $u' g_1^v$, where v is interpreted as an integer. (We technically are referring to the first scheme presented in Boneh-Boyen [1] when only a level one hierarchy is used. Although, our scheme can be extended to be a hierarchical scheme in an analogous manner.) In the next section we show that, remarkably, this small modification is all that is needed to make the scheme fully secure.

5 Security

We now prove the security of our scheme.

Theorem 1. *Our IBE- scheme is (t, q, ϵ) secure assuming the decisional $(t + O(\epsilon^{-2} \ln(\epsilon^{-1}) \lambda^{-1} \ln(\lambda^{-1})), \frac{\epsilon}{32(n+1)q})$ BDH assumption holds, where $\lambda = \frac{1}{8(n+1)q}$.*

Proof. Suppose there exists a (t, q, ϵ)-adversary, $\mathcal{A}$ against our scheme. We construct a simulator, $\mathcal{B}$, to play the decisional BDH game. The simulator will take BDH challenge $(g, A = g^a, B = g^b, C = g^c, Z)$ and outputs a guess, β', as to whether the challenge is a BDH tuple. The simulator runs $\mathcal{A}$ executing the following steps.

5.1 Simulator Description

Setup The simulator first sets an integer, $m = 4q$, and chooses an integer, k, uniformly at random between 0 and n. It then chooses a random n-length vector, $\overrightarrow{x} = (x_i)$, where the elements of $\overrightarrow{x}$ are chosen uniformly at random from the integers between 0 and $m-1$ and a value, x', chosen uniformly at random between 0 and $m-1$. Let X^* denote the pair $(x', \overrightarrow{x})$ Additionally, the simulator chooses a random $y' \in \mathbb{Z}_p$ and an n-length vector, $\overrightarrow{y} = (y_i)$, where the elements of $\overrightarrow{y}$ are chosen at random in $\mathbb{Z}_p$. These values are all kept internal to the simulator.

Again, for an identity v we will let $\mathcal{V} \subseteq \{1, \ldots, n\}$ be the set of all i for which $v_i = 1$ For ease of analysis we define three functions. We define $F(v) = (p - mk) + x' + \sum_{i \in \mathcal{V}} x_i$ and define $J(v) = y' + \sum_{i \in \mathcal{V}} y_i$. Finally, we define a binary function $K(v)$ as

$$K(v) = \begin{cases} 0, & \text{if } x' + \sum_{i \in \mathcal{V}} x_i \equiv 0 \pmod{m} \\ 1, & \text{otherwise.} \end{cases}$$

The simulator assigns $g_1 = A$ and $g_2 = B$. It then assigns the public parameters $u' = g_2^{p-km+x'} g^{y'}$ and U as $u_i = g_2^{x_i} g^{y_i}$. From the perspective of the adversary the distribution of the public parameters is identical to the real construction.

Phase 1 The adversary, $\mathcal{A}$, will issue private key queries. Suppose the adversary issues a query for an identity v. If $K(v) = 0$ the simulator aborts and randomly chooses its guess β' of the challenger's value β.

Otherwise, the simulator chooses a random $r \in \mathbb{Z}_p$. Using the technique described by Boneh and Boyen [1] it constructs the private key, d, as

$$d = (d_0, d_1) = \left(g_1^{\frac{-J(v)}{F(v)}} (u' \prod_{i \in \mathcal{V}} u_i)^r, g_1^{\frac{-1}{F(v)}} g^r \right).$$

Let $\tilde{r} = r - \frac{a}{F(v)}$. Then we have

$$\begin{aligned} d_0 &= g_1^{\frac{-J(v)}{F(v)}} (u' \textstyle\prod_{i \in v} u_i)^r \\ &= g_1^{\frac{-J(v)}{F(v)}} (g_2^{F(v)} g^{J(v)})^r \\ &= g_2^a (g_2^{F(v)} g^{J(v)})^{-\frac{a}{F(v)}} (g_2^{F(v)} g^{J(v)})^r \\ &= g_2^a (u' \textstyle\prod_{i \in \mathcal{V}} u_i)^{r - \frac{a}{F(v)}} \\ &= g_2^a (u' \textstyle\prod_{i \in \mathcal{V}} u_i)^{\tilde{r}}. \end{aligned}$$

Additionally, we have

$$d_1 = g_1^{\frac{-1}{F(v)}} g^r = g^{r - \frac{a}{F(v)}} = g^{\tilde{r}}.$$

This simulator will be able to perform this computation iff $F(v) \neq 0 \mod p$. For ease of analysis the simulator will only continue (not abort) in the sufficient condition where $K(v) \neq 0$. (If we have $K(v) \neq 0$ this implies $F(v) \neq 0 \mod p$ since we can assume $p > nm$ for any reasonable values of p, n, and m).

Challenge The adversary next will submit two messages $M_0, M_1 \in \mathbb{G}_1$ and an identity, v^*. If $x' + \sum_{i \in \mathcal{V}^*} x_i \neq km$ the simulator will abort and submit a random guess for β'. Otherwise, we have $F(v^*) \equiv 0 \pmod{p}$ and the simulator will flip a fair coin, γ, and construct the ciphertext

$$T = (ZM_\gamma, C, C^{J(v^*)}).$$

Suppose that the simulator was given a BDH tuple, that is $Z = e(g, g)^{abc}$. Then we have

$$T = \left(e(g, g)^{abc} M_\gamma, g^c, g^{cJ(v^*)}\right) = \left(e(g_1, g_2)^c M_\gamma, g^c, (u' \prod_{i \in \mathcal{V}^*} u_i)^c\right).$$

We see that T is a valid encryption of M_γ.

Otherwise, we have that Z is a random element of $\mathbb{G}$. In that case the ciphertext will give no information about the simulator's choice of γ.

Phase 2 The simulator repeats the same method it used in Phase 1.

Guess Finally, the adversary $\mathcal{A}$ outputs a guess γ' of γ.

Artificial Abort At this point the simulator is still unable to use the output from the adversary. An adversary's probability of success could be correlated with the probability that the simulator needs to abort. This stems from the fact that two different sets of q private key queries may cause the simulator to abort with different probabilities. In the worst case we might worry that $\Pr[\gamma = \gamma' | \overline{\texttt{abort}}] - \frac{1}{2} = 0$ (or some negligible value) in the simulation even if $\Pr[\gamma = \gamma'] - \frac{1}{2} = \epsilon$ for some non-negligible ϵ.

The simulator corrects for this by forcing all possible sets of queries of the adversary to cause the simulator to abort with (almost) the same probability (1-λ), where (1-λ) is a lower bound on any set of private key queries causing an abort before this stage.

Let $\overrightarrow{v} = v_1, \ldots v_q$ denote the private key queries made in Phase 1 and Phase 2 and let v^* denote the challenge identity and we let $\mathcal{V}^* \subseteq \{1, \ldots, n\}$ be the set of all i for which $v_i^* = 1$. (All of these values are defined at this point in the simulation.) First, we define the function $\tau(X', \overrightarrow{v}, v^*)$, where X' is a set of simulation values $x', x_1, \ldots, x_n$, as

$$\tau(X', \overrightarrow{v}, v^*) = \begin{cases} 0, & \text{if } (\bigwedge_{i=1}^{q} K(v_i) = 1) \wedge x' + \sum_{i \in \mathcal{V}^*} x_i = km \\ 1, & \text{otherwise.} \end{cases}$$

The function $\tau(X', \overrightarrow{v}, v^*)$ will evaluate to 0 if the private key and challenge queries $\overrightarrow{v}, v^*$ will not cause an abort for a given choice of simulation values, X'. We can now consider the probability over the simulation values for a given set of queries, $\overrightarrow{v}, v^*$, as $\eta = \Pr_{X'}[\tau(X', \overrightarrow{v}, v^*) = 0]$.

The simulator samples $O(\epsilon^{-2} \ln(\epsilon^{-1}) \lambda^{-1} \ln(\lambda^{-1}))$ times the probability η by choosing a random X' and evaluating $\tau(X', \overrightarrow{v}, v^*)$ to compute an estimate η'. We emphasize that the sampling does not involve running the adversary again.

Let $\lambda = \frac{1}{8nq}$, be the lower bound on the probability of not aborting for any set of queries. (We show how to calculate λ below.) Then if $\eta' \geq \lambda$ the simulator will abort with probability $\frac{\eta' - \lambda}{\eta'}$ (not abort with probability $\frac{\lambda}{\eta'}$) and take a random guess β'. Otherwise, the simulator will not abort.

If the simulator has not aborted at this point it will take check to see if the adversary's guess, $\gamma' = \gamma$. If $\gamma' = \gamma$ then the simulator outputs a guess $\beta' = 1$, otherwise it outputs $\beta = 0$.

This concludes the description of the simulator.

5.2 Analysis

Our simulator is difficult to analyze directly since it might abort before all of the queries are made. For ease of exposition we now describe a second simulation, which we will use to reason about the output distribution of the first simulation.

Setup The simulator chooses the secret key g_2^{α} as in the construction and then chooses $X^*, \overrightarrow{y}$ as in the first simulation and derives u', U in the same way. It then runs the adversary.

Phase 1 The simulator responds to private key queries by using the master key as in the construction, in this way all queries can be answered.

Challenge The simulator receives the challenge messages M_0, M_1. The second simulator will flip two coins β and γ. If $\beta = 0$ then it encrypts a random message and if $\beta = 1$ it encrypts M_{γ}.

Phase 2 Same as Phase 1.

Guess The simulator receives a guess γ' from the adversary. At this point the simulator has seen as the private key queries and the challenge query $(\overrightarrow{v}, v^*)$. It evaluates the function $\tau(X^*, \overrightarrow{v}, v^*)$ and aborts if it evaluates to 1, outputting a random guess of β'.

Artificial Abort The last step is done in exactly the same way as the first simulation. This ends the description.

We first equate the probabilities of the both simulators with the following claim.

Claim. The probabilities $\Pr[\beta' = \beta]$ are the same in both the first and second simulations we described.

Proof. The second simulation runs the adversary completely and receives all of its queries. In the guess phase it checks if $\tau(X^*, \overrightarrow{v}, v^*) = 1$ and aborts if so. The check decides if there was a point where the first simulator would have needed to abort during the simulator and take a random guess. If so the second simulator aborts and takes a random guess itself. Additionally, all public parameters, private key queries, and challenge ciphertexts have the same distribution up to the point of a possible abortion. The artificial abort stages are also identical. Therefore, we can reason that the output distributions will be the same. □

For purpose of exposition, we will now derive the success of the simulator in terms of the second simulator. However, due to Claim 5.2 the discussion applies to both simulators equally.

Claim. The probability of the simulation not aborting by the guess phase is at least $\lambda = \frac{1}{8(n+1)q}$.

Proof. We calculate a lower bound, λ as the lower bound of $\Pr_{X'}[\tau(X', \overrightarrow{v}, v^*) = 0]$ for all $\overrightarrow{v}, v^*$. Without loss of generality we can assume the adversary always makes the maximum number of queries, q. For any set of q queries $v_1, \ldots, v_q$ and challenge identity, v^*, we have $\Pr[\overline{\texttt{abort}}] = \Pr[(\bigwedge_{i=1}^{q} K(v_i) = 1) \wedge \sum_{i \in \mathcal{V}^*} x_i = km]$. We can then lower bound the probability of not aborting as follows.

$$\Pr[(\bigwedge_{i=1}^{q} K(v_i) = 1) \wedge \sum_{i \in \mathcal{V}^*} x_i = km] \tag{1a}$$

$$= (1 - \Pr[\bigvee_{i=1}^{q} K(v_i) = 0]) \Pr[\sum_{i \in \mathcal{V}^*} x_i = km | \bigwedge_{i=1}^{q} K(v_i) = 1] \tag{1b}$$

$$\geq (1 - \sum_{i=1}^{q} \Pr[K(v_i) = 0]) \Pr[\sum_{i \in \mathcal{V}^*} x_i = km | \bigwedge_{i=1}^{q} K(v_i) = 1] \tag{1c}$$

$$= (1 - \frac{q}{m}) \Pr[\sum_{i \in \mathcal{V}^*} x_i = km | \bigwedge_{i=1}^{q} K(v_i) = 1] \tag{1d}$$

$$= \frac{1}{n+1}(1 - \frac{q}{m}) \Pr[K(v^*) = 0 | \bigwedge_{i=1}^{q} K(v_i) = 1] \tag{1e}$$

$$= \frac{1}{n+1}(1 - \frac{q}{m}) \frac{\Pr[K(v^*) = 0]}{\Pr[\bigwedge_{i=1}^{q} K(v_i) = 1]} \Pr[\bigwedge_{i=1}^{q} K(v_i) = 1 | K(v^*) = 0] \tag{1f}$$

$$\geq \frac{1}{(n+1)m}(1 - \frac{q}{m}) \Pr[\bigwedge_{i=1}^{q} K(v_i) = 1 | K(v^*) = 0] \tag{1g}$$

$$= \frac{1}{(n+1)m}(1 - \frac{q}{m})(1 - \Pr[\bigvee_{i=1}^{q} K(v_i) = 0 | K(v^*) = 0]) \tag{1h}$$

$$\geq \frac{1}{(n+1)m}(1 - \frac{q}{m})(1 - \sum_{i=1}^{q} \Pr[K(v_i) = 0 | K(v^*) = 0]) \tag{1i}$$

$$= \frac{1}{(n+1)m}(1 - \frac{q}{m})^2 \tag{1j}$$

$$\geq \frac{1}{(n+1)m}(1 - 2\frac{q}{m}) \tag{1k}$$

Equations 1d and 1g come from the fact that $\Pr[K(v)=0]=\frac{1}{m}$ for any query, v. The $\frac{1}{n+1}$ factor of Equation 1e comes from the simulator taking a guess of k. Equation 1j is derived from the pairwise independence of the probabilities that $K(v)=0, K(v')=0$ for any pair of different queries v, v'. The probabilities are pairwise independent since the sums $x' + \sum_{i \in V} x_i \pmod m$ and $x' + \sum_{i \in V'} x_i \pmod m$ will differ in at least one random x_j.

We can optimize the last equation by setting $m = 4q$ (as we did in the simulation), where q is the maximum number of queries. (If the adversary makes less queries the probability of not aborting can only be greater). Solving for this gives us a lower bound $\lambda = \frac{1}{8(n+1)q}$. $\square$

We now can calculate the distributions $\mathcal{P}_{BDH}$ and $\mathcal{R}_{BDH}$. The distribution $\mathcal{R}_{BDH}$ is simply $\frac{1}{2}$. When the simulator is given a random element as the last term in the tuple the simulator will either abort (and guess $\beta' = 1$ with probability $\frac{1}{2}$) or it will guess $\beta' = 1$ when the adversary is correct in guessing γ. However, the γ will be completely hidden from the adversary in this case so the adversary will be correct with probability $\frac{1}{2}$.

The calculation of $\mathcal{P}_{BDH}$ is somewhat more complicated. In the second simulation the adversary's view of the simulation will be identical to the real game. We want to know the probability that the guess $\beta' = 1$.

We then break the event into the abort and non-abort cases and see that $\Pr[\beta'=1]$ is the sum of $\Pr[\beta'=1|\mathtt{abort}]\Pr[\mathtt{abort}]$ and $Pr[\beta'=1|\overline{\mathtt{abort}}]\Pr[\overline{\mathtt{abort}}]$. We observe that $\Pr[\beta'=1|\mathtt{abort}] = \frac{1}{2}$ and that when the simulator does not abort $\beta' = 1$ when the adversary correctly guesses $\gamma' = \gamma$. Then, we have $\mathcal{P}_{BDH} = \frac{1}{2} + \frac{1}{2}(\Pr[\overline{\mathtt{abort}}|\gamma'=\gamma]\Pr[\gamma'=\gamma] - \Pr[\overline{\mathtt{abort}}|\gamma' \neq \gamma]\Pr[\gamma' \neq \gamma])$. By our assumption, this is equal to $\frac{1}{2} + \frac{1}{2}(\Pr[\overline{\mathtt{abort}}|\gamma'=\gamma](\frac{1}{2}+\epsilon) - \Pr[\overline{\mathtt{abort}}|\gamma' \neq \gamma](\frac{1}{2}-\epsilon))$. All that is left to do is to both lower and upper bound the probability of not aborting in our simulation. We state the following claim.

Claim. If the simulator takes takes $O(\epsilon^{-2}\ln(\epsilon^{-1})\lambda^{-1}\ln(\lambda^{-1}))$ samples when computing the estimate η', then $(\frac{1}{2}+\epsilon)\Pr[\overline{\mathtt{abort}}|\gamma'=\gamma] - (\frac{1}{2}-\epsilon)\Pr[\overline{\mathtt{abort}}|\gamma'=\gamma] \geq \frac{3}{2}\lambda\epsilon$.

We prove the claim in Appendix A.

Plugging in the claim we have $\mathcal{P}_{BDH} \geq \frac{1}{2} + \frac{3}{4}\lambda\epsilon$. Then, $\frac{1}{2}(\mathcal{P}_{BDH} - \mathcal{R}_{BDH}) \geq \frac{3}{4}\lambda\epsilon \geq \frac{\epsilon}{32(n+1)q}$. $\square$

We note that if there was a way for the simulator to efficiently compute the abort probability, η, for a given set of queries (as opposed to sampling) then we could improve the time component of our reduction could be significantly improved in addition to simplifying our analysis.

6 Hierarchical IBE and CCA Security

In Section 4 we discussed the similarity of our scheme to the 1-level hierarchical IBE (HIBE) scheme of Boneh and Boyen [1]. We can further take advantage of the similarity of our schemes to construct an ℓ-level HIBE scheme in an obvious manner. (For each level we must generate new parameters u' and U.)

The problem with using our techniques to construct an HIBE scheme is that the reduction becomes inefficient for all but small values of ℓ. In particular to construct a scheme in which any efficient adversary has at most ϵ advantage it must be true that all efficient adversaries have at most an $O((nq)^{\ell}\epsilon)$ advantage in the decisional BDH game. The intuition behind this is that in the simulation the setup must be "match" the challenge identity at all ℓ different levels in order to not abort. (However, our reduction still provides a stronger reduction than that of Boneh and Boyen [1] for a fully secure HIBE scheme.) For this reason we still consider the construction of a fully secure HIBE scheme without random oracles to be an open problem.

Recent results of Canetti et al. [10], further improved upon by Boneh and Katz [6], show how to build a CCA-secure Identity-Based encryption scheme from a 2-level HIBE scheme. We can actually build a hybrid 2-level HIBE [15, 13] scheme that uses our scheme at the first level and the scheme of Boneh and Boyen [1] at the second level. Since the transformations [10, 6] only require Selective-ID security at the second level our hybrid construction is CCA secure without any significant further degradation in the security reduction relative to our non-hierarchical construction.

7 A Signature Scheme

Boneh and Franklin [5] describe a generic method for converting any Identity-Based Encryption scheme into a signature scheme. The public key of the signature scheme corresponds to the global parameters of the IBE scheme. To sign a message, M, in the signature scheme the signer gives an IBE private key of M as the signature of M. To verify a signature of M the verifier encrypts a random value, R, to the identity M, then attempts to decrypt the ciphertext with the private key. The signature is accepted if the decryption successfully decrypts to R. We note that this is a randomized verification algorithm.

In the generic transformation the security of the resulting signature scheme reduces to the security of the Identity-Based Encryption scheme. Thus, we immediately have a signature scheme which is secure as the decisional BDH problem. However, we can use the bilinear map in order to deterministically verify a signature and get a signature scheme that reduces to the weaker computational Diffie-Hellman assumption. We note that similar techniques have been used previously. For example, the signatures in the scheme of Boneh, Lynn, Shacham [7] correspond to private keys of the Boneh-Franklin IBE system. We describe our signature scheme for completeness.

7.1 Construction

Let $\mathbb{G}$ be a group of prime order, p, for which there exists an efficiently computable bilinear map into $\mathbb{G}_1$. Additionally, let $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$ denote the bilinear map and g be the corresponding generator. The size of the group is determined by the security parameter. We will sign messages of n bits; again, we can use a collision-resistant hash function, $H : \{0,1\}^* \to \{0,1\}^n$, to sign messages of arbitrary length.

Setup The public key is generated as follows. A secret $\alpha \in \mathbb{Z}_p$ is chosen at random. We choose a random generator, g, and set the value $g_1 = g^\alpha$ and choose g_2 randomly in $\mathbb{G}$. Additionally, the algorithm chooses a random value $u' \in \mathbb{G}$ and a random n-length vector $U = (u_i)$, whose elements are chosen at random from $\mathbb{G}$. The published public key is g,g_1, g_2,u', and U. The secret key is g_2^α.

Signing Let M be an n-bit message to be signed and M_i denote the ith bit of M, and $\mathcal{M} \subseteq \{1, \ldots, n\}$ be the set of all i for which $M_i = 1$. A signature of M is generated as follows. First, a random $r \in \mathbb{Z}_p$ is chosen. Then the signature is constructed as:

$$\sigma_M = \left(g_2^\alpha \left(u' \prod_{i \in \mathcal{M}} u_i \right)^r , g^r \right).$$

Verification Suppose we wish to check if $\sigma = (\sigma_1, \sigma_2)$ is a signature for a message M. The signature is accepted if $e(\sigma_1, g)/e(\sigma_2, u' \prod_{i \in \mathcal{M}} u_i) = e(g_1, g_2)$.

7.2 Security

Theorem 2. *The signature scheme is (t, q, ϵ) existentially unforgeable assuming the decisional-$(t, \frac{\epsilon}{16(n+1)q})$ BDH assumption holds, where $\lambda = \frac{1}{8(n+1)q}$.*

We omit the proof of this theorem, but note that it is analogous to the proof our IBE scheme. The fact that the adversary returns a forgery results in two important differences though. First, the forgery is used to solve the computational Diffie-Hellman problem. Secondly, since a forgery is returned there is no need for an artificial abort stage as in the previous reduction.

Other efficient schemes that are secure against existential forgery under an adaptive chosen-message attack [14] in the standard model depend upon the Strong-RSA assumption [12, 11] or the Strong Diffie-Hellman assumption [3]. Additionally Boneh, Mironov, and Shoup [8] describe a tree-based signature scheme based on the computational Diffie-Hellman assumption.

8 Conclusions

We presented the first efficient Identity-Based Encryption scheme that is secure in the full model without random oracles. We proved our the security of our scheme by reducing it to the decisional Bilinear Diffie-Hellman problem. Additionally, we showed how our Identity-Based encryption scheme can be converted to an efficient signature scheme that depends only upon the computational Diffie-Hellman assumption in the standard model.

This work motivates two interesting open problems. The first is to find an efficient Identity-Based Encryption system (without random oracles) that has short public parameters. The second, is to find an IBE system with a tight reduction in security. Such a solution would also likely permit an efficient reduction for an analogous HIBE scheme.

Acknowledgments

We would like to thank Mihir Bellare, Dan Boneh, and the anonymous reviewers of Eurocrypt 2005 for giving helpful suggestions.

References

1. Dan Boneh and Xavier Boyen. Efficient selective-id secure identity based encryption without random oracles. In *Proceedings of the International Conference on Advances in Cryptology (EUROCRYPT '04)*, Lecture Notes in Computer Science. Springer Verlag, 2004.
2. Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In *Proceedings of the Advances in Cryptology (CRYPTO '04)*, 2004.
3. Dan Boneh and Xavier Boyen. Short signatures without random oracles. In *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73. Berlin: Springer-Verlag, 2004.
4. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, pages 213–229. Springer-Verlag, 2001.
5. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.
6. Dan Boneh and Jonathan Katz. Improved efficiency for cca-secure cryptosystems built using identity based encryption. In *In Proceedings of RSA-CT 2005*, 2005.
7. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. In *ASIACRYPT '01: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security*, pages 514–532. Springer-Verlag, 2001.
8. Dan Boneh, Ilya Mironov, and Victor Shoup. A secure signature scheme from bilinear maps. In *CT-RSA*, pages 98–110, 2003.
9. Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In *Proceedings of Eurocrypt 2003*. Springer-Verlag, 2003.
10. Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In *Proceedings of Eurocrypt 2004*. Springer-Verlag, 2004.
11. Ronald Cramer and Victor Shoup. Signature schemes based on the strong rsa assumption. *ACM Trans. Inf. Syst. Secur.*, 3(3):161–185, 2000.
12. Rosario Gennaro, Shai Halevi, and Tal Rabin. Secure hash-and-sign signatures without the random oracle. *Lecture Notes in Computer Science*, 1592:123++, 1999.

13. Craig Gentry and Alice Silverberg. Hierarchical id-based cryptography. In *Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security*, pages 548–566. Springer-Verlag, 2002.
14. Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.
15. Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In *Advances in Cryptology: EUROCRYPT 2002*, pages 466–481, 2002.
16. Adi Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 47–53. Springer-Verlag New York, Inc., 1985.

A Proof of Claim 3

In order to show that $(\frac{1}{2}+\epsilon)\Pr[\overline{\texttt{abort}}|\gamma'=\gamma] - (\frac{1}{2}-\epsilon)\Pr[\overline{\texttt{abort}}|\gamma'=\gamma] \geq \frac{3}{2}\lambda\epsilon$ we first upper bound the term $(\frac{1}{2}+\epsilon)\Pr[\overline{\texttt{abort}}|\gamma'=\gamma]$.

Let η be the probability of not aborting associated for a set of private key queries and challenge query on a particular run where $\gamma'=\gamma$. The simulator will make $O(\epsilon^{-2}\ln(\epsilon^{-1})\lambda^{-1}\ln(\lambda^{-1}))$ samples to calculate η' and we can use Chernoff bounds to show that $\Pr[\eta' > \eta(1+\frac{\epsilon}{8})] < \lambda\frac{\epsilon}{8}$. We then have

$$\Pr[\overline{\texttt{abort}}|\gamma'=\gamma] \geq (1-\lambda\frac{\epsilon}{8})\eta\frac{\lambda}{\eta(l+\frac{\epsilon}{8})} \geq \lambda(1-\frac{\epsilon}{8})^2 \geq \lambda(1-\frac{1}{4}\epsilon)$$

where the probability calculation is taken of the sampling of η. We now have

$$(\frac{1}{2}+\epsilon)\Pr[\overline{\texttt{abort}}|\gamma'=\gamma] \geq \lambda(\frac{1}{2}+\frac{3}{4}\epsilon).$$

(Note that the artificial abort stage aborts with probability $\frac{\lambda}{\max(\lambda,\eta')}$. Since $\eta(1+\frac{\epsilon}{8}) > \lambda$, we were able to ignore the maximum function.)

We now lower bound the term $(\frac{1}{2}+\epsilon)\Pr[\overline{\texttt{abort}}|\gamma'\neq\gamma]$. The simulator will make $O(\epsilon^{-2}\ln(\epsilon^{-1})\lambda^{-1}\ln(\lambda^{-1}))$ samples to calculate the estimate η' and we can use Chernoff bounds to show that $\Pr[\eta' < \eta(1-\frac{\epsilon}{8})] < \lambda\frac{\epsilon}{8}$. We then have

$$\Pr[\overline{\texttt{abort}}|\gamma'\neq\gamma] \leq \lambda\frac{\epsilon}{8} + \lambda\frac{\eta}{\eta(1-\frac{\epsilon}{8})} \leq \lambda\frac{\epsilon}{8} + \lambda(1+\frac{2\epsilon}{8}) = \lambda(1+\epsilon\frac{3}{8})$$

where the probability calculation is taken of the sampling of η. We now have

$$(\frac{1}{2}-\epsilon)\Pr[\overline{\texttt{abort}}|\gamma'=\gamma] \leq \lambda(\frac{1}{2}-\frac{3}{4}\epsilon).$$

We now see that $(\frac{1}{2}+\epsilon)\Pr[\overline{\texttt{abort}}|\gamma'=\gamma] - (\frac{1}{2}-\epsilon)\Pr[\overline{\texttt{abort}}|\gamma'=\gamma] \geq \frac{3}{2}\lambda\epsilon$. $\square$