Efficient Identity-Based Encryption Without Random Oracles

Brent Waters
bwaters@cs.stanford.edu

Abstract

We present the first efficient Identity-Based Encryption (IBE) scheme that is secure in the full model 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 via the usual transformation of an IBE scheme to a signature scheme.

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 while the message has been sent.

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 [6, 1]. Most recently, Boneh and Boyen [2] described a scheme that was proved to be secure in full model 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 secure in the full model 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. Previous practical signature schemes that were secure in the standard model relied on the Strong-RSA assumption [9, 8] or the Strong-BDH assumption [3].

1.1 Related Work

Shamir [13] first presented the idea of Identity-Based Encryption as a challenge to the research community. However, the first secure and efficient scheme by 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. [6] describe a somewhat 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 ciphertext 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. One property of their security reduction is that in the simulator they construct there exist specific bits, which if set in the challenge identity result in the simulator needing to abort.

This results in the need to apply an error-correcting code to identities in order to guarantee an adequate Hamming separation between different identities. In contrast, in our reduction there are no specific identity bits that correspond to an identity being able to serve as a challenge. Therefore, we are able to avoid applying an error-correcting code to identities and can construct an efficient scheme.

1.2 Organization

We organize the rest of the paper as follows. In Section 2 we give the definitions of IBE definition of security. 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. We discuss the transformation to a signature scheme in Section 6. Finally, we conclude in Section 7.

2 Security Definitions

In this section we present the definition for a semantically secure 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 which are:

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

Phase 1 The adversary is allowed to make multiple queries 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 for all t-time adversaries making at most q queries have at most an ϵ in breaking our scheme.*

3 Complexity Assumptions

Let $\mathbb{G}$ be a group of prime order p with an admissible bilinear map, e, into $\mathbb{G}_1$ and g be a generator of $\mathbb{G}$.

We define the security of respective IBE and signature schemes in terms of the following two games.

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

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

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

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 $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ 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. We can also let identities be bitstrings of arbitrary length and n be the output length of a collision-resistant hash function, $H : \{0,1\}^* \to \{0,1\}^n$.

We now describe the construction of our scheme.

Setup The system parameters are generated as follows. A secret $\alpha \in \mathbb{Z}_p$ is chosen at random. We set the value $g_1 = g^\alpha$ and choose g_2 randomly in $\mathbb{G}$. Additionally, the simulator 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_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 and let v_i denote the *ith* bit of v and let $\mathcal{V} \subseteq \{0, \ldots, n\}$ 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 secret key for 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:

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

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 $\mathbb{G}_1$.

5 IBE Security

We now prove the security of our scheme.

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

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 $v_i = 1$ First, we define $F(v) = (p - mk) + x' + \sum_{i \in \mathcal{V}} x_i$. Next, we 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 $u' = g_2^{p-km+x'} g^{y'}$ and the parameter 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. We first check if $K(v) = 0$. If this is the cases 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 algebraic methods first 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

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

Aditionally, we have

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

This simulator will be able to perform this computation iff $F(v) \neq 0 \mod p$. For ease of analysis the simulator will only 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. Even though $\mathcal{A}$ has advantage ϵ in the game defined by our security definition this does not imply that if the adversary has not aborted by this point that $\Pr[\gamma = \gamma'] - \frac{1}{2} = \epsilon$.

The adversary's probability making of causing the simulator to abort is not necessarily independent of his probability of making a correct guess of γ. Indeed, one might expect the adversary to be more successful when it makes more queries and thus has a higher probability of aborting.

In this stage the simulator accounts for this by adding on an artificial abort possibility. 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. (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

$$\tau(X', \overrightarrow{v}, v^*) = \begin{cases} 0, & \text{if } K(v_1) = 1 \wedge \ldots \wedge K(v_q) = 1 \wedge x' + \sum_{i=1}^n x_{i,v_i^*} = km \\ 1, & \text{otherwise.} \end{cases}$$

The function $\tau(X', \overrightarrow{v}, v^*)$ will evaluate to 0 if the private key and challenge queries $\overrightarrow{v}, v^*$ will 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 $\eta = \Pr_{X'}[\tau(X', \overrightarrow{v}, v^*)]$.

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 $\frac{\eta' - \lambda}{\lambda}$ (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

We now need to calculate the distributions $\mathcal{P}_{BDH}$ and $\mathcal{R}_{BDH}$ for our simulator. One difficulty in analyzing our simulator directly is that 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.

The second simulation proceeds as follows:

Setup The simulator chooses a 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 1. *The probabilities* $\Pr[\beta' = \beta]$ *are the same in both the first and second distributions 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 does the takes a random guess also. 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 purposes of exposition we will now derive the success of the simulator in terms of the second simulator. However, due to Claim 1 the discussion applies to both simulators equally.

Claim 2. *We calculate a lower bound,* λ*, on the probability of not aborting by the guess phase for any sequence of private key queries and challenge identity as* $\lambda = \frac{1}{8(n+1)q}$.

Proof. During the second simulation the adversary's view is completely independent of X^* since the queries are satisfied using the master key and the adversary is simulated to the end. Therefore, we can calculate a lower bound, λ as the lower bound of $\Pr_{X'}[\tau(X', \overrightarrow{v}, v^*)] = 0$ for any $\overrightarrow{v}, v^*$. Without loss of generality we can assume the adversary always makes the maximum number of queries, q, as we can always simulate the adversary making more queries at the end.

For any set of q queries $v_1, \ldots, v_q$ and challenge identity, v^*, we have:

$$
\begin{aligned}
\Pr[\overline{\texttt{abort}}] &= \Pr[(\bigwedge_{i=1}^{q} K(v_i) = 1) \wedge \sum_{i=1}^{n} x_{i,v_i^*} = km] && \text{(1a)}\\
&= (1 - \Pr[\bigvee_{i=1}^{q} K(v_i) = 0]) \Pr[\sum_{i=1}^{n} x_{i,v_i^*} = km | \bigwedge_{i=1}^{q} K(v_i) = 1] && \text{(1b)}\\
&\geq (1 - \sum_{i=1}^{q} \Pr[K(v_i) = 0]) \Pr[\sum_{i=1}^{n} x_{i,v_i^*} = km | \bigwedge_{i=1}^{q} K(v_i) = 1] && \text{(1c)}\\
&= (1 - \frac{q}{m}) \Pr[\sum_{i=1}^{n} x_{i,v_i^*} = km | \bigwedge_{i=1}^{q} K(v_i) = 1] && \text{(1d)}\\
&= \frac{1}{n+1}(1 - \frac{q}{m}) \Pr[K(v^*) = 0 | \bigwedge_{i=1}^{q} K(v_i) = 1] && \text{(1e)}\\
&= \frac{1}{n+1}(1 - \frac{q}{m}) \frac{\Pr[K(v^*) = 0]}{\Pr[\bigwedge_{i=1}^{q} K(v_i) = 1]} \Pr[\bigwedge_{i=1}^{q} K(v_i) = 1 | K(v^*) = 0] && \text{(1f)}\\
&\geq \frac{1}{(n+1)m}(1 - \frac{q}{m}) \Pr[\bigwedge_{i=1}^{q} K(v_i) = 1 | K(v^*) = 0] && \text{(1g)}\\
&= \frac{1}{(n+1)m}(1 - \frac{q}{m})(1 - \Pr[\bigvee_{i=1}^{q} K(v_i) = 0 | K(v^*) = 0]) && \text{(1h)}\\
&\geq \frac{1}{(n+1)m}(1 - \frac{q}{m})(1 - \sum_{i=1}^{q} \Pr[K(v_i) = 0 | K(v^*) = 0]) && \text{(1i)}\\
&= \frac{1}{(n+1)m}(1 - \frac{q}{m})^2 && \text{(1j)}\\
&\geq \frac{1}{(n+1)m}(1 - 2\frac{q}{m}) && \text{(1k)}
\end{aligned}
$$

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 $\Pr[\beta' = 1 | \texttt{abort}] \Pr[\texttt{abort}] + Pr[\beta' = 1 | \overline{\texttt{abort}}] \Pr[\overline{\texttt{abort}}]$. We observe that $\Pr[\beta' = 1 | \texttt{abort}] = \frac{1}{2}$ and that when the simulator does not abort $\beta' = 1$ when the adversary correctly guesses $\gamma' = \gamma$. Then, we have $\mathcal{P}_{BDH} = \frac{1}{2} + \frac{1}{2}(\Pr[\overline{\texttt{abort}} | \gamma' = \gamma] \Pr[\gamma' = \gamma] - \Pr[\overline{\texttt{abort}} | \gamma' \neq \gamma] \Pr[\gamma' \neq \gamma])$. By our assumption, this is equal to $\frac{1}{2} + \frac{1}{2}(\Pr[\overline{\texttt{abort}} | \gamma' = \gamma](\frac{1}{2} + \epsilon) - \Pr[\overline{\texttt{abort}} | \gamma' \neq \gamma](\frac{1}{2} - \epsilon))$. 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 3. *If the simulator takes takes $O(\epsilon^{-2}\ln(\epsilon^{-1})\lambda^{-1}\ln(\lambda^{-1}))$ samples when computing the estimate η', then* $(\frac{1}{2}+\epsilon)\Pr[\texttt{abort}|\gamma'=\gamma] - (\frac{1}{2}-\epsilon)\Pr[\texttt{abort}|\gamma'=\gamma] \geq \frac{3}{2}\lambda\epsilon$.

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, $\frac{1}{2}(\mathcal{P}_{BDH}-\mathcal{R}_{BDH}) \geq \frac{3}{4}\lambda\epsilon \geq \frac{\epsilon}{32(n+1)q}$. □

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 the t component of our reduction could be significantly improved.

5.3 CCA Security

Recent results of Canetti et al. [7], further improved upon by Boneh and Katz [5], show how Selective-ID security implies CCA security. We can build a hybrid 2-level HIBE [12, 10] scheme that uses our scheme at the first level and the scheme of Boneh and Boyen [1] at the second level. Since the transformations [7, 5] only require Selective-ID security at the second level our hybrid construction is CCA secure in the full model without any significant further degradation in the security reduction.

6 Signatures

From the usual transformation from an Identity-Based Encryption system to a signature scheme we see that our scheme implies an efficient signature scheme. The security of our scheme relies only on the Computational Diffie-Hellman assumption in groups with efficiently computable bilinear maps. We omit the proof and note that it is analogous to the one given for our IBE scheme, with the exception that the artificial abort stage is not needed since we are reducing to a computational assumption. The result is that a t-time adversary that makes at most q signature queries and succeeds with probability ϵ in our signature scheme implies a $(t, \frac{\epsilon}{16(n+1)q})$-CDH adversary.

One downside of our signature scheme is that the number of group elements needed to specify a user's public key is linear in the message size (or output of the hash function). While, this is tolerable for an Identity-Based Encryption scheme where there will be few authorities in practice it seems less practical to let each user have such large public keys in a signature scheme.

Other efficient schemes that are secure against existential forgery under an adaptive chosen-message attack [11] in the standard model depend upon the Strong-RSA assumption [9, 8] or the Strong Diffie-Hellman assumption [3].

7 Conclusions and Open Problems

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 implies 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.

8 Acknowledgements

We would like to thank Mihir Bellare and Dan Boneh 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 Jonathan Katz. Improved efficiency for cca-secure cryptosystems built using identity based encryption. In *To Appear in RSA-CT 2005*, 2005.

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

[7] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In *Proceedings of Eurocrypt 2004*. Springer-Verlag, 2004.

[8] Ronald Cramer and Victor Shoup. Signature schemes based on the strong rsa assumption. *ACM Trans. Inf. Syst. Secur.*, 3(3):161–185, 2000.

[9] Rosario Gennaro, Shai Halevi, and Tal Rabin. Secure hash-and-sign signatures without the random oracle. *Lecture Notes in Computer Science*, 1592:123++, 1999.

[10] 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.

[11] 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.

[12] Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In *Advances in Cryptology: EUROCRYPT 2002*, pages 466–481, 2002.

[13] 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 $(\frac{1}{2}+\epsilon)\Pr[\overline{\mathtt{abort}}|\gamma'=\gamma]-(\frac{1}{2}-\epsilon)\Pr[\overline{\mathtt{abort}}|\gamma'=\gamma]\geq\frac{3}{2}\lambda\epsilon$ we first upper bound the term $(\frac{1}{2}+\epsilon)\Pr[\overline{\mathtt{abort}}|\gamma'=\gamma]$.

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

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

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

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

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