Space-Efficient Identity-Based Encryption:

Spelling out the Approach by Boneh-Gentry-Hamburg

Patrícia Lustosa V. Ribeiro[1], Ruy J. G. B. de Queiroz[1]

[1]Centro de Informática
Universidade Federal de Pernambuco (UFPE) – Recife, PE – Brazil

{plvr,ruy}@cin.ufpe.br

Abstract. *Identity-based encryption (also known as IBE) is a type of public key cryptography in which the public key of a user is some unique information about his identity. The initial motivation to the creation of IBE was to simplify key management in email systems. An open problem was the creation of a space-efficient IBE scheme that was not based in pairings on elliptic curves. Boneh, Gentry and Hamburg proposed such a system in 2007. The objective of this work is to do a critical analysis of how Boneh, Gentry and Hamburg scheme works, filling in some missing details whenever necessary.*

Resumo. *Encriptação Baseada em Identidade (mais conhecida como Identity-based encryption ou IBE) é um tipo de criptografia de chave pública tal que a chave pública de um usuário é alguma informação única sobre a sua identidade. A motivação inicial para a criação de IBE foi simplificar o gerenciamento de certificados em sistemas de email. Um problema em aberto era a criação de um esquema de IBE espaço-eficiente e sem recorrer a emparelhamentos em curvas elípticas. Boneh, Gentry e Hamburg propuseram tal sistema em 2007. O objetivo desse trabalho é fazer uma análise crítica do funcionamento do esquema de Boneh, Gentry e Hamburg, preenchendo detalhes quando necessário.*

1. Introduction

Identity-based encryption was first proposed by Adi Shamir in 1984[1]. The objective was to avoid the need to maintain the complex infra-structure of key management that exists in public-key systems and thus making encryption in e-mail systems easier. The scheme is based on public-key cryptography with an extra benefit: the user chooses some unique identifier as its public key instead of generating a random public/private key. This unique identifier can be his/her name, social number security, email address, or any other information that uniquely identifies him/her.

According to Martin [2], IBE systems are very similar to public-key systems in many aspects, but it also has significant differences. In public-key systems, a public-key certificate has all the information needed to encrypt the message. In IBE, a user needs to get a set of public key parameters from a trusted third party. Once he/she has these public parameters, he/she can use it to calculate public keys for any user he/she wants and uses it to encrypt the messages.

The recipient of an IBE-encrypted message needs his/her private key to decrypt the message. In order to obtain his/her private key, the recipient must authenticate himself/herself to a private key generator (PKG), a trusted third party that calculates the private keys. The PKG uses the identity of the user together with some secret information, called a master secret, to calculate the private key for that user. After that, the private key can be securely sent to the user.

In public-key cryptography, public-key certificates have a preset expiration date. This can be made in IBE-systems by setting the public key as the identifier concatenated with the current year. The user can only use his/her private key during that year. After that, he/she needs to obtain a new private key. Note that a user who wants to communicate with him/her does not have to obtain his/her public key every time his/her private key expires.

A problem with this approach is that during the validating period of some key, there is no way to revoke that key. To solve this problem, IBE systems typically use short-lived keys. This is not as precise as having the ability of revoking immediately a key but it makes key validation trivial [2].

An interesting use of identity-based encryption is to send messages in the future. The public key can be defined as the email address concatenated with the date. Thus, one could send an email that the recipient could only read in the future, in the date specified by the sender.

In IBE schemes there are four algorithms that are responsible for creating and using a public/private key pair. They are called Setup, KeyGen, Encryption and Decryption.

The Setup algorithm initializes the system parameters (also known as public parameters or PP) and the master key. Intuitively, the system parameters will be publicly known, while the master key will be kept in secret by the PKG. The KeyGen algorithm takes as inputs the master secret and the identity and returns the private key associated with this identity.

The Encryption algorithm takes as input the public parameters, the identity of the receiver and the message and returns the corresponding ciphertext. The Decryption algorithm takes as input the private key and a ciphertext and returns the original message.

The objective of this work is to describe Boneh, Gentry and Hamburg IBE. In the second chapter, we present the concept of security of IBE systems. The following chapter talks about the motivation for the creation of Boneh, Gentry and Hamburg IBE and describe its algorithms. The fourth chapter presents the proof of security of such scheme.

2. Security of IBE systems

Chosen ciphertext security (IND-CCA) [3] is the standard acceptable model of security for public key schemes. Hence, it is natural to require this notion of security to identity-based encryption schemes. However, it is not enough for an IBE scheme to be IND-CCA secure. The reason is that, when an adversary attacks an identity, he may already have the corresponding private keys to other identities. The system should remain secure despite the adversary being able to obtain private keys for any identity of his/her

choice (other than the identity being attacked). The adversary is also allowed to choose the identity being attacked [4].

An IBE scheme may also be required to be anonymous. It means that the ciphertext reveals nothing about the identity of the user used to create it. The following IBE security game, as described in [5], captures chosen ciphertext security, private key queries and anonymity:

Setup: The challenger runs $Setup(\lambda)$ and gives the adversary the resulting public parameters PP. It keeps the master-key (MSK) to itself. We set ID_0^*, $ID_1^* \leftarrow \perp$ and $C^* \leftarrow \perp$.

Queries: The adversary can issue adaptive queries of the following types:

- Private key query ‹ID_i ›: the challenger returns the resulting private key d_i = KeyGen(*MSK*, ID_i) to the adversary. ID_i must be different from both ID_0^* and ID_1^*.
- Decryption query (ID_i ,C_i): the challenger responds by running KeyGen(*MSK,* ID_i) to obtain the private key d_i and Decrypt(d_i,C_i) to obtain the plaintext and then sends it to the adversary. (ID_i ,C_i) must be different from both (ID_0^*,C^*) and (ID_1^*,C^*).
- A single encryption query ((ID_0 ,m_0), (ID_1 ,m_1)): ID_0, ID_1 are distinct from all previous key queries and m_0, m_1 are two equal length plaintexts. The challenger picks a random bit b $\overset{R}{\leftarrow}\{0,1\}$ and sets

$$C^* \leftarrow Encrypt(PP, ID_b, m_b),\ ID_0^* \leftarrow ID_0\ , ID_1^* \leftarrow ID_1$$

It sends *C** to the adversary.

Guess: Eventually, the adversary outputs $b' \overset{R}{\leftarrow}\{0,1\}$. The adversary wins if $b = b'$.

Note that, initially, there is no value set to ID_0^*, ID_1^* and C^*. So the adversary can make arbitrary private key queries and decryption queries. When he/she does his single encryption query, the two identities he/she chose must be different from all previous identities he/she queried for private keys. After that, the private key queries and decryption queries have restrictions to avoid the adversary to obtain the private key for ID_0^* or ID_1^* and to decrypt the challenge ciphertext with one of these two identities.

We call the adversary $\mathcal{A}$ and define its advantage in attacking the scheme $\mathcal{E}$ as

$$\text{IBEAdv}_{\mathcal{A},\mathcal{E}(\lambda)} = \left|\Pr[b = b'] - \frac{1}{2}\right|$$

An adversary in this game is called an ANON-IND-ID-CCA adversary. ANON stands for anonymous, IND stands for indistinguishability, ID refers to the ability of the adversary to make private key queries and CCA stands for chosen ciphertext attack.

We will also consider three types of weaker adversaries:

- If $\mathcal{A}$ makes no decryption queries we say that $\mathcal{A}$ is an ANON-IND-ID-CPA adversary. This models an anonymous IBE under a chosen plaintext attack.

- If in the single encryption query the adversary uses $ID_0 = ID_1$, then we say that the adversary is an IND-ID-CCA adversary. This models a chosen ciphertext secure IBE that is not necessarily anonymous.
- If $\mathcal{A}$ makes no decryption queries and uses $ID_0 = ID_1$ we say that $\mathcal{A}$ is IND-ID-CPA adversary. This is the standard IBE security model under a chosen plaintext attack.

A single encryption query with different identities, as in the original game, guarantees that the system is anonymous. This happens because, if the adversary was able to extract some information about the ID from the ciphertext, he/she could use this information to find out the right value of b, increasing his chance of winning the game. On the other hand, if we change the game to use the same identity for both m_0 e m_1 in the encryption query, there is no guarantee that the ciphertext reveals no information about the identity used to create it. Hence, the second type of adversary is not anonymous.

Definition 2.1: *Let $\mathcal{S}$ be one of {IND-ID-CPA, IND-ID-CCA, ANON-IND-ID-CPA, ANON-IND-ID-CCA}. We say that an IBE system $\mathcal{E}$ is $\mathcal{S}$-secure if for all polynomial time $\mathcal{S}$ adversaries $\mathcal{A}$ we have that $IBEAdv_{\mathcal{A},\mathcal{E}(\lambda)}$ is a negligible function.*

3. Boneh-Gentry-Hamburg Scheme

In 2001, the first fully functional identity-based encryption scheme was proposed by Dan Boneh and Matthew Franklin [4]. Their work is based on bilinear maps between groups and they prove its security based on the random oracle model. Boneh-Franklin IBE requires the calculation of a pairing, an expensive calculation that accounts for almost all the computation required for decryption and most of the computation required for encryption. Besides that, the assumptions about the hardness of problems in certain elliptic curves groups are relatively new compared to other cryptographic assumptions.

In the same year, Clifford Cocks [6] invented another IBE scheme. The security of Cocks' IBE is based on both the computational difficulty of integer factorization and on the quadratic residuosity problem. Cocks IBE is efficient with respect to time but is not space efficient. Ciphertexts in this system contain two elements of $\mathbb{Z}/N\mathbb{Z}$ for each bit of the plaintext. Therefore the encryption of an l-bit message is $2l.log_2N$ long.

Because of the uncertainty about the cryptographic assumption about pairings in elliptic curves and the space inefficiency of the only IBE scheme that was not based on elliptic curves [6], it was an open problem to find a space efficient IBE scheme that wasn't based on pairings on elliptic curves.

In 2007, Boneh, Gentry and Hamburg [5] found such a scheme. In their system, the encryption on an l-bit message consists of a single element in $\mathbb{Z}/N\mathbb{Z}$ plus $(l + 1)$ additional bits. Hence, ciphertext size is about $l + log_2N$.

To construct the IBE system, we are going to start constructing a deterministic algorithm with the following properties.

Definition 3.1. *Let $\mathcal{Q}$ be a deterministic algorithm that takes as input (N, R, S) where $N \in \mathbb{Z}^+$ and $R, S \in \mathbb{Z}/N\mathbb{Z}$. The algorithm outputs two polynomials $f, g \in \mathbb{Z}/N\mathbb{Z}[x]$. We say that $\mathcal{Q}$ is IBE compatible if the following two conditions hold:*

- *Condition 1: If R and S are quadratic residues, then $f(r)g(s)$ is a quadratic residue for all square roots r of R and s of S.*
- *Condition 2: If R is a quadratic residue, then $f(r)f(-r)S$ is a quadratic residue for all square roots r of R.*

We now describe the scheme that encrypts multi-bit messages, called BasicIBE.

$\boldsymbol{Setup}(\boldsymbol{\lambda})$: Generate $(p, q) \leftarrow RSAgen(\lambda)$, $N \leftarrow pq$ and pick a random $u \xleftarrow{R} J(N) \setminus QR(N)$. Output public parameters $PP = (N, u, H)$ where H is a hash function $H : \mathcal{ID} \times [1, l] \rightarrow J(N)$. The master key MSK is the factorization of N and a random key K for a pseudorandom function $F_K: \mathcal{ID} \times [1, l] \rightarrow \{0,1,2,3\}$.

$\boldsymbol{KeyGen}(\boldsymbol{MSK}, \boldsymbol{ID}, \boldsymbol{l})$: It generates a private key for encrypting l-bit messages. Takes as input the master secret key, the identity of the user and the length l. For $j = 1, \dots, l$ do:

$R_j \leftarrow H(ID, j) \in J(N)$ and $w \leftarrow F_K(ID, j) \in \{0,1,2,3\}$.

Let $a \in \{0,1\}$ be such that $u^a R_j \in QR(N)$.

Let $\{z_0, z_1, z_2, z_3\}$ be the square roots of $u^a R_j$ in $\mathbb{Z}/N\mathbb{Z}$.

Set $r_j \leftarrow z_w$.

Output the decryption key $d_{ID} \leftarrow (PP, r_1, \dots, r_l)$. The PRF F guarantees that the same square roots is output for a given *ID*, but an adversary can't tell ahead of time which one will be output.

Notice that the master secret key (MSK) is used to find the four square roots of $u^a R_j$. Without the factorization, we wouldn't be able to determine whether $u^a R_j$ is a quadratic residue, which is easier than finding the square roots.

$\boldsymbol{Encrypt}(\boldsymbol{PP}, \boldsymbol{ID}, \boldsymbol{m})$: Takes as input the public parameters, the *ID* of the user and the message to be encrypted $m = m_1 \dots m_l \in \{\pm 1\}^l$. It picks a random $s \in \mathbb{Z}/N\mathbb{Z}$ and computes $S \leftarrow s^2$. For $j = 1, \dots, l$ do:

$R_j \leftarrow H(ID, j)$, $(f_j, g_j) \leftarrow \mathcal{Q}(N, R_j, S)$ and $(\bar{f}_j, \bar{g}_j) \leftarrow \mathcal{Q}(N, uR_j, S)$

$$c_j \leftarrow m_j \,.\left(\frac{g_j(s)}{N}\right) \quad \text{and} \quad \bar{c}_j \leftarrow m_j \,.\left(\frac{\bar{g}_j(s)}{N}\right).$$

Set $c \leftarrow c_1 \dots c_l$ and $\bar{c} \leftarrow \bar{c}_1 \dots \bar{c}_l$ and output the ciphertext $C \leftarrow (S, c, \bar{c})$.

$\boldsymbol{Decrypt(C, d_{ID})}$: Takes as input the ciphertext C and the decryption key $d_{ID} = (PP, r_1, \dots, r_l)$. For $j = 1, \dots, l$ let $R_j \leftarrow H(ID, j)$ and do:

if $r_j^2 = R_j$ run $(f_j, g_j) \leftarrow \mathcal{Q}(N, R_j, S)$ and set $m_j \leftarrow c_j.\left(\frac{f_j(r_j)}{N}\right)$

if $r_j^2 = uR_j$ run $(\bar{f}_j, \bar{g}_j) \leftarrow \mathcal{Q}(N, uR_j, S)$ and set $m_j \leftarrow \bar{c}_j.\left(\frac{\bar{f}_j(r_j)}{N}\right)$

Output $m = m_1 \dots m_l$.

This completes the description of BasicIBE.

The same value of S can be used to encrypt all l bits of the message. To encrypt an l-bit message we hash ID multiple times by computing $R_i \leftarrow H(ID, i)$ for $i = 1, \dots, l$. Now each pair (S, R_i) can be used to encrypt one message bit. The length of the ciphertext is the size of S plus 2 bits for each message bit. Hence, when encrypting an l-bit message, the length of the ciphertext $(S, (c_1, c_1'), \dots, (c_l, c_l'))$ is $log_2 N + 2l$ bits.

4. Security of Boneh-Gentry-Hamburg scheme

In this section, we prove the security of BasicIBE in the random oracle model based on the QR assumption.

Lemma 4.1: *Let $N = pq$ be as RSA modulus, $X \in QR(N)$ and $S \in J(N)$. Let x be a random variable uniformly chosen from among the four square roots of X. Let f be a polynomial such that $f(x)f(-x)X$ is a quadratic residue for all four values of x. Then:*

- *when $S \notin QR(N)$ the Jacobi symbol $(f(x)/N)$ is uniformly distributed in $\{\pm 1\}$;*
- *when $S \in QR(N)$ then $(f(x)/N)$ is constant, namely the same for all four values of x.*

Theorem 4.2: *Suppose the QR assumption holds for RSAgen and $\boldsymbol{F}$ is a secure PRF. Then the system BasicIBE is IND-ID-CPA secure when $\boldsymbol{H}$ is modeled as a random oracle. In particular, suppose $\boldsymbol{\mathcal{A}}$ is an efficient IND-ID-CPA adversary. Then there exist efficient algorithms $\boldsymbol{\mathcal{B}_1}$, $\boldsymbol{\mathcal{B}_2}$ (whose running time is about the same as that of $\boldsymbol{\mathcal{A}}$) such that*

$$IBEAdv_{\mathcal{A},BasicIBE}(\lambda) \leq 2.QRAdv_{\mathcal{B}_1,RSAgen}(\lambda) + PRFAdv_{\mathcal{B}_2,F}(\lambda)$$

Proof: We present the proof as a sequence of games. We let $\boldsymbol{W_i}$ denote the event that the adversary $\boldsymbol{\mathcal{A}}$ wins the game $\boldsymbol{i}$.

Game 0: This game is identical to the one defined in section 3.1. Hence, we know that

$$\left|\Pr[W_0] - \frac{1}{2}\right| = IBEAdv_{\mathcal{A},BasicIBE}(\lambda)$$

The challenger chooses the random oracle $H : \mathcal{ID} \times [1, l] \to J(N)$ at random from the set of all such functions.

Game 1: This game differs from the past game in the way it generates private keys. Instead of using a pseudorandom function F, the challenger uses a truly random function. If F is a secure pseudorandom function, the adversary won't notice the difference between the two games. In particular, there exists an algorithm $\mathcal{B}_2$ (whose running time is about the same as that of $\mathcal{A}$) such that

$$|\Pr[W_1] - \Pr[W_0]| = PRFAdv_{\mathcal{B}_2,F}(\lambda)$$

Game 2: In game 1 the public parameters given to $\mathcal{A}$ contain (N, u, H) where u is uniform in $J(N)\backslash QR(N)$, as in the original system. Moreover, the random oracle H is a random function $H : \mathcal{ID} \times [1, l] \to J(N)$. In game 2, we change H in the following manner: H outputs $H(ID, j) = u^a v^2$ by choosing at random $a \overset{R}{\leftarrow} \{0,1\}$ and $v \overset{R}{\leftarrow} \mathbb{Z}/N\mathbb{Z}$. We know that $u \in J(N)\backslash QR(N)$ so $\left(\frac{u}{N}\right) = 1$. Since $v^2 \in QR(N)$, $\left(\frac{v^2}{N}\right) = 1$. If $a = 0$, $H(ID, j) = v^2 \in J(N)$. If $a = 1$, $H(ID, j) = uv^2$ and we have $\left(\frac{uv^2}{N}\right) = \left(\frac{u}{N}\right).\left(\frac{v^2}{N}\right) = 1.1 = 1$. In both cases, we have $H(ID, j) \in J(N)$. So, H implements a random function $H : \mathcal{ID} \times [1, l] \to J(N)$.

Let $R_j \leftarrow H(ID, j)$ for some (ID, j). In game 1 the challenger responds to private key queries by outputting a random square root of R_j or uR_j for $j = 1, \ldots l$. In game 2 let $R_j \leftarrow H(ID, j) = u^a v^2$. The challenger responds to private key queries by outputting either ${R_j}^{1/2} = v$ (used if $a = 0$) or $\left(uR_j\right)^{1/2} = uv$ (used if $a = 1$) for $j = 1, \ldots, l$. Since v is uniform in $\mathbb{Z}/N\mathbb{Z}$, r_j is uniform in the set of the square roots of R_j or uR_j, just like in game 1. Because of this, from $\mathcal{A}$'s view, games 1 and 2 are identical and therefore

$$\Pr[W_2] = \Pr[W_1]$$

Note that in game 2 the challenger no longer needs the factorization of N to respond to $\mathcal{A}$'s queries. In game 1, the challenger needs it to calculate the square roots of R_j.

Game 3: We slightly modify game 2 by choosing a random u in $QR(N)$ instead of in $J(N)\backslash QR(N)$. The adversary won't notice any differences assuming QR assumption holds for RSAgen, since it is the only change between the two games. In particular, there exists an efficient algorithm $\mathcal{B}_1$ such that

$$|\Pr[W_3] - \Pr[W_2]| = QRAdv_{\mathcal{B}_1,RSAgen}(\lambda)$$

We note that since $H(ID, j) = u^a v^2$ and $u \in QR(N)$, H will always output elements in $QR(N)$. Let u_0 be a square root of u.

Game 4: We slightly change the way that the challenger builds the ciphertext C^*. We pick C^* in a similar way to the one used in the proof of lemma 4.1. To respond to the encryption query (ID, m_0, m_1) from $\mathcal{A}$ the challenger chooses $b \overset{R}{\leftarrow} \{0,1\}$ and does:

$R_i \leftarrow H(ID, j) = u^{a_i}.v_i^2$ and $r_i \leftarrow u_0^{a_i}.v_i$ for $i = 1, \dots, l$

(then r_i is a root of R_i and $u_0 r_i$ is a root of uR_i)

$(*)$ $s \overset{R}{\leftarrow} \mathbb{Z}/N\mathbb{Z}$ and $S \leftarrow s^2$

write $m^{(b)} = m_1 \dots m_l \in \{\pm 1\}^l$

for $k = 1, \dots, l$ do:

$(f_k, g_k) \leftarrow \mathcal{Q}(N, R_k, S)$ and $(\bar{f}_k, \bar{g}_k) \leftarrow \mathcal{Q}(N, uR_k, S)$

$(**)$ $c_k \leftarrow m_k . \left(\frac{f_k(r_k)}{N}\right)$ and $\bar{c}_k \leftarrow m_k . \left(\frac{\bar{f}_k(u_0 r_k)}{N}\right)$.

$c \leftarrow c_1 \dots c_l$ and $\bar{c} \leftarrow \bar{c}_1 \dots \bar{c}_l$.

Send $\mathcal{A}$ the challenge ciphertext $C \leftarrow (S, c, \bar{c})$.

Since S, R_k, uR_k are all in $QR(N)$, we know by condition (1) of definition 3.1 that $\left(\frac{f_k(r_k)}{N}\right) = \left(\frac{g_k(s)}{N}\right)$ for all $k = 1, \dots, l$ and also $\left(\frac{\overline{f_k}(u_0 r_k)}{N}\right) = \left(\frac{\overline{g_k}(s)}{N}\right)$. Hence, the ciphertext C^* created in this way is identical to the challenge ciphertext created in game 3. Therefore,

$$Pr[W_3] = Pr[W_4]$$

It is important to note that s is not used in the creation of C^*.

Game 5: We slightly modify the challenger in game 4 by choosing S uniformly in $J(N) \backslash QR(N)$ instead of $QR(N)$. That is, we change the line marked with $(*)$ in Game 4 into

$(*)$ $S \leftarrow RJ(N) \backslash QR(N)$

Since this is the only difference between the games, the adversary will not notice the difference, assuming the QR assumption holds for RSAgen. In particular, there exists an algorithm $\mathcal{B}_1$ such that

$$|\Pr[W_5] - \Pr[W_4]| = QRAdv_{\mathcal{B}_1, RSAgen}(\lambda)$$

Game 6: We can now change Game 5 and make the challenge ciphertext C^* be independent of the challenge bit b. We change the line marked $(**)$ in Game 4 as follows:

$(**)$ $z_k \overset{R}{\leftarrow} \{\pm 1\}$, $c_k \leftarrow z_k . \left(\frac{f_k(r_k)}{N}\right)$ and $\bar{c}_k \leftarrow z_k . \left(\frac{\bar{f}_k(u_0 r_k)}{N}\right)$.

As a result, the challenge ciphertext C^* is an encryption of a random message $z_1, \dots, z_l$, independent of the bit b.

We argue that because S is a non-residue, Games 5 and 6 are indistinguishable due to Condition 2 of definition 3.1 and lemma 4.1. The challenge ciphertext is created

by using $2l$ elements $\{R_1, uR_1, \dots, R_l, uR_l\}$ all in $QR(N)$. For each R the adversary does not know which of the four square roots of R is used in the creation of C^*. This is used to satisfy the condition of lemma 4.1 which says that r is a random variable over the four square roots of R.

Consider now a specific $k \in \{1, \dots, l\}$ and let $x = \left(\frac{f_k(r_k)}{N}\right)$ and $y = \left(\frac{\overline{f_k}(u_0 r_k)}{N}\right)$. Using condition (2) of definition 3.1 and lemma 4.1, we have that x is uniformly distributed in $\{\pm 1\}$ and the same happens with y. With this, we have that

$$Pr[(x,y) = (1,1)] = Pr[(x,y) = (-1,-1)] \text{ and}$$

$$Pr[(x,y) = (1,-1)] = Pr[(x,y) = (-1,1)]$$

It follows that the pair (x,y), which is an encryption of $+1$, is distributed identically as the pair $(-x,-y)$, which is an encryption of -1. Hence, in $\mathcal{A}$'s view, the bits $(ck, \overline{ck})$ are distributed identically whether the plaintext is $+1$ or -1. Since this holds for all $k = 1, \dots, l$ it follows that C^* is distributed identically in Games 5 and 6. As a result, we have:

$$Pr[W_6] = Pr[W_5]$$

End. In game 6 we have the ciphertext as the encryption of a random message z. Because of this, we clearly have

$$Pr[W_6] = \frac{1}{2}$$

Combining the equations, we have that

$$IBEAdv_{\mathcal{A},BasicIBE}(\lambda) \leq 2.QRAdv_{\mathcal{B}_1,RSAgen}(\lambda) + PRFAdv_{\mathcal{B}_2,F}(\lambda)$$

Thus, the theorem is proved. □

The scheme defined here is abstract, because we don't present a concrete instantiation of the IBE compatible algorithm Q. A concrete instantiation was given in [5]. However, it requires the generation of primes of the order of $\sqrt{N}$. This makes the encryption time quartic in the security parameter per message bit, while the decryption time is cubic in the security parameter. Most practical public-key systems, like RSA and the existing IBE schemes including Cocks' IBE, are cubic in the security parameter. Thus the encryption time in this scheme is not ideal.

5. Conclusion

We have described with more details the space-efficient IBE scheme without pairings based on the quadratic residuosity assumption in the random oracle model presented in [5]. Also in [5], Boneh, Gentry and Hamburg described an IBE scheme with the same properties that is also anonymous. The scheme described here is more space efficient than Cocks' IBE but is less time efficient. In order to make the scheme more efficient, one needs to look at other instantiations of the IBE compatible algorithm Q that are faster than the one proposed by Boneh, Gentry and Hamburg.

References

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

[2] Luther Martin. *Identity-Based Encryption*. *Information Security and Privacy Series*. Artech House, 2008.

[3] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman & Hall/CRC, 2008.

[4] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. *SIAM Journal of Computing*, 32(3):586–615, 2003. Extended abstract in *CRYPTO'01*.

[5] D. Boneh, C. Gentry, and M. Hamburg. Space-Efficient Identity Based Encryption Without Pairings. In *Proceedings of FOCS 2007*, pp. 647-657, 2007

[6] 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.