Jhanwar-Barua's Identity-Based Encryption Revisited

Ibrahim Elashry, Yi Mu, and Willy Susilo

Centre for Computer and Information Security Research,
School of Computer Science and Software Engineering,
University of Wollongong, Wollongong NSW 2522, Australia
ifeae231@uowmail.edu.au, {ymu,wsusilo}@uow.edu.au

Abstract. In FOCS'07, Boneh, Gentry and Hamburg presented an identity-based encryption (IBE) system (BasicIBE) based on the quadratic residuosity (QR) assumption. A BasicIBE encryption of an l-bit message has a short ciphertext of $\log_2 N + 2l$ bits where N is a Blum integer. However, it is not time-efficient due to solving $l+1$ equations in the form $Rx^2 + Sy^2 \equiv 1 \pmod{N}$. Jhanwar and Barua presented a variant of BasicIBE in which the encryptor only solves $2\sqrt{l}$ such equations. The decryptor decrypts the message without solving any such equations. In addition, the decryption key is decreased to only one element in $\mathbb{Z}_N$. However, the ciphertext size increases from a single element to $2\sqrt{l}$ elements in $\mathbb{Z}_N$. In this paper, we revisit the Jhanwar-Barua (JB) system and review its security. We prove that this system is not IND-ID-CPA secure and present a solution to the security flaw of this system. We also point out a flaw in the security proof of the JB system and propose two different security proofs for the fixed system. We prove that it has the same security as the original BasicIBE system.

Keywords: Identity-based Encryption, Quadratic Residuosity Assumption, IND-ID-CPA.

1 Introduction

In 1985, Shamir [1] presented the notion of identity-based encryption (IBE) in which the user's identity represents his public key and consequently, no public key certificate is required. Additionally, the construction of identity-based signature was proposed in the same work, but the construction of identity-based encryption (IBE) was left as an open research problem. The design of a provable secure IBE remained an open problem for sixteen years until Boneh and Franklin [2] proposed a provably secure IBE in the random oracle model based on bilinear maps. Subsequently, there has been a rapid development in IBE based on bilinear maps, such as [3,4,5,6]. The notion of identity-based cryptography is very important in the real world application where the necessity of having to verify the certificates is not a viable solution. In the literature, many such applications have been proposed to date. However, all the previously mentioned IBEs are

M.H. Au et al. (Eds.): NSS 2014, LNCS 8792, pp. 271–284, 2014.

based on pairing operations. According to MIRACL benchmarks, a 512-bit Tate pairing takes 20 ms while a 1024-bit prime modular exponentiation takes 8.80 ms. The pairing computations are expensive compared to normal operations. The costly pairing computation limits it from being used in wide application, specially when time and power consumptions are a major concern such as in limited wireless sensor networks. Hence, the seek for a scheme that does not rely on pairings is desirable. Another approach to design IBEs is based on the quadratic residuosity (QR) assumption. The first IBE based on this approach is due to Cocks [7]. This system is IND-ID-CPA secure in the random oracle model. It is time-efficient compared to pairing-based IBEs, but it produces a long ciphertext of two elements in $\mathbb{Z}_N$ for every bit in the message. The design of efficient IBEs without pairings was an open problem until Boneh, Gentry and Hamburg [8] presented two space-efficient systems (BasicIBE and AnonIBE) in which the ciphertext is reduced from $2l$ elements to only one element in $\mathbb{Z}_N$. As in Cocks' IBE, the security of BasicIBE is based on the QR assumption in the random oracle model. Although the concrete instantiation of BasicIBE is highly space-efficient, this comes at the cost of less time-efficient encryption/decryption algorithms. To encrypt an l-bit message, BasicIBE solves $l+1$ equations in the form $Rx^2 + Sy^2 \equiv 1 \pmod{N}$ for known values of R, S and N [8]. Solving such an equation requires a 'solubility certificate' and obtaining these certificates requires the generation of primes [7,9,10]. The obtained certificates can be used to solve $Rx^2 + Sy^2 \equiv 1 \pmod{N}$ efficiently using the Cremona-Rusin algorithm [9]. The prime generation is a time-consuming process and it is the bottleneck in the BGH systems. Moreover, the decryption key is l elements in $\mathbb{Z}_N$ because the identity ID is hashed to a different value to encrypt each bit. AnonIBE is based on BasicIBE and it is Anon-IND-ID-CPA secure in the standard model under the interactive quadratic residuosity (IQR) assumption [8]. Moreover, the ciphertext length is reduced to one element in $\mathbb{Z}_N$ plus $l+1$ bits.

Jhanwar and Barua [11] made some significant observations on the BGH systems (for solving equations in the form $Rx^2+Sy^2 \equiv 1 \pmod{N}$) and proposed a trade-off system that reduces the private key length but increases the ciphertext length. They found that by knowing the value of $S \pmod{N}$, one can find a random solution to the equation $Rx^2 + Sy^2 \equiv 1 \pmod{N}$ using only one inversion in $\mathbb{Z}_N$. The sender solves only $2\sqrt{l}$ equations in the form $Rx^2+Sy^2 \equiv 1 \pmod{N}$ using only $2\sqrt{l}$ inversions in $\mathbb{Z}_N$ and thus, no prime generation is required. This increases the encryption/decryption speed dramatically. The private key is only one element in $\mathbb{Z}_N$. However, this system produces a large ciphertext of $2\sqrt{l}$ elements in $\mathbb{Z}_N$. The most interesting part of Jhanwar and Barua [11] is its time- and power-efficiency. It avoids the expensive prime generation operations and replaces it with only one inversion in $\mathbb{Z}_N$. Moreover, there is no expensive-computational operations such as pairing or even modular exponentiation. We compare between the (JB) system and some other efficient IBE systems such as Boneh-Boyen IBE [5] and IBE systems with more powerful adversary such as Boneh, Raghunathan, Segev (BRS) IBE [12]. We also compare it to other pairing-free IBE such as Cock's IBE [7] and BGH IBEs [8]. In the table, the

symbol m represents prime modular exponentiation while e and p represents pairing operation and prime generation respectively. l represents the message length. The simple x in the table represents a parameter in BRS IBE which is function of the security parameter, the length of the identity and a prime p [12]. The symbols G and G_T represents an element in two groups G and G_T such that $e : G \times G \to G_T$.

Table 1. Comparison between Various IBEs and the JB IBE

	Expensive Mathematical Operations	Ciphertext Length
Cock's	0	$2l(\log N)$
The BasicIBE	$(l+1)p$	$\log N + 2l$
The AnonIBE	$(2l+1)p$	$\log N + l + 1$
Jhanwar-Barua	0	$2\sqrt{l}\log N + 2l$
Boneh-Boyen	e+3m	G_T+2G
BRS	m+(x)e	G+(x)G_T

Our Contributions. We revisit the JB system, and identify some security issues with the system. We prove that an IND-ID-CPA adversary can attack this system and hence it is not IND-ID-CPA secure. The attack comes from mistakenly reusing the same y to encrypt multiple bits and hence, these bits are encrypted using the same key. We also present a solution to the security flaw of this system. We also point to a flaw of the security proof of the JB system and present two security proofs for the fixed system. We prove that it is as secure as the BasicIBE system. We note here that the fixed JB system is as efficient as the original system.

2 Definitions

2.1 IND-ID-CPA

The IND-ID-CPA security model of an IBE system is described as a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ [1,2]. This game is as follows:

- Setup(λ): $\mathcal{C}$ generates the public parameters (PP) and sends them to $\mathcal{A}$ and keeps the master secret (MSK) to himself.
- Query Phase: In this phase, $\mathcal{A}$ sends private key queries to $\mathcal{C}$ for identities ID_s of his choice. These queries are adaptive based on previous queries.
- Challenge: Satisfied with private key queries, $\mathcal{A}$ sends to $\mathcal{C}$ two messages m_1 and m_2 for an identity ID^*. $\mathcal{C}$ tosses a coin $b \in [0,1]$ randomly and encrypts m_b using ID^*. Note that ID^* must not be queried in the query phase.
- Guess: $\mathcal{A}$ outputs $\overline{b} \in [0,1]$. $\mathcal{A}$ wins the game if $b = \overline{b}$.

The advantage of $\mathcal{A}$ to attack a system ξ and win this game is:

$$IBEAdv_{A,\xi}(\lambda) = |pr[\overline{b} = b] - \tfrac{1}{2}|.$$

If $\mathcal{A}$ submits two pairs of (ID_0, m_0) and (ID_1, m_1) in the challenge phase, then this game is called the ANON-IND-ID-CPA security model. The advantage of the adversary winning this game is the same as above.

2.2 QR Assumption and Jacobi Symbols

For a positive integer N, define the following set [8]:

$$J(N) = [a \in \mathbb{Z}_N : \left(\tfrac{a}{N}\right) = 1],$$

where $\left(\frac{a}{N}\right)$ is the Jacobi symbol of a w.r.t N. The Quadratic Residue set $QR(N)$ is defined as follows.

$$QR(N) = [a \in \mathbb{Z}_N : gcd(a, N) = 1 \wedge x^2 \equiv a \pmod{N} \text{ has a solution}].$$

Definition 1. *Quadratic Residuosity Assumption: Let RSAgen(λ) be a probabilistic polynomial time (PPT) algorithm. This algorithm generates two equal size primes p, q. The QR assumption holds for RSAgen if it cannot distinguish between the following two distributions for all PPT algorithms $\mathcal{A}$ [8].*

$$P_{QR}(\lambda) : (N, V)(p, q) \leftarrow RSAgen(\lambda), N = pq, V \in_R QR(N),$$
$$P_{NQR}(\lambda) : (N, V)(p, q) \leftarrow RSAgen(\lambda), N = pq, V \in_R J(N) \setminus QR(N).$$

In other words, the advantage of $\mathcal{A}$ against QR assumption $QRAdv_{\mathcal{A},RSAgen(\lambda)} =$

$$|\Pr[(N, V) \leftarrow P_{QR}(\lambda) : \mathcal{A}(N, V) = 1]| - |\Pr[(N, V) \leftarrow P_{NQR}(\lambda) : \mathcal{A}(N, V) = 1]|.$$

is negligible. i.e. $\mathcal{A}$ cannot distinguish between elements in $J(N) \setminus QR(N)$ and elements in $QR(N)$.

3 BasicIBE [8]

BasicIBE encrypts an l-bit message m using a square $S \equiv s^2 \pmod{N}$ where $s \in_R \mathbb{Z}_{\mathbb{N}}$, the user's identity ID and a pair of Jacobi symbols for each bit. It first hashes ID to different values $H(ID, i) = u^a R_i = r_i^2$ where $a \in \{0, 1\}$, $u \in J(N) \setminus QR(N)$ and i is the bit index. Then it solves the equations $R_i x_i^2 + S y_i^2 \equiv 1 \pmod{N}$ and $uR_i \overline{x}_i^2 + S\overline{y}_i^2 \equiv 1 \pmod{N}$ to get $(x_i, y_i, \overline{x}_i, \overline{y}_i)$. The ciphertext is $(S, c, \overline{c})$ where $c \leftarrow [c_1, c_2, c_3, ..., c_l]$, $c_i = m \cdot \left(\frac{2+2y_i s}{N}\right)$ and $\overline{c} \leftarrow [\overline{c}_1, \overline{c}_2, \overline{c}_3, ..., \overline{c}_l]$, $\overline{c}_i = m \cdot \left(\frac{2+2\overline{y}_i s}{N}\right)$. To decrypt, one needs to know the square-root of R_i or uR_i. If $R_i = r_i^2$, the message is $m_i = c_i \cdot \left(\frac{1+x_i r_i}{N}\right)$ and if $uR_i = r_i^2$, the message is $m_i = \overline{c}_i \cdot \left(\frac{1+\overline{x}_i r_i}{N}\right)$.

3.1 BGH Product Formula

BasicIBE [8] has to solve $2l$ equations in the form $Rx^2 + Sy^2 \equiv 1 \pmod N$ to encrypt/decrypt a message m of length l by computing pairs (x_i, y_i), $(\overline{x}_i, \overline{y}_i) \in \mathbb{Z}_N^2$ such that:

$$R_i x_i^2 + S y_i^2 \equiv 1 \pmod N \quad \textit{and} \quad uR_i\overline{x}_i^2 + S\overline{y}_i^2 \equiv 1 \pmod N. \tag{1}$$

Boneh, Gentry and Hamburg presented a product formula which only solves $l+1$ equations to encrypt/decrypt a message [8].

Lemma 1. *For $i = 1, 2$ let (x_i, y_i) be a solution to $R_i x^2 + Sy^2 \equiv 1 \pmod N$. Then (x_3, y_3) is a solution to*

$$R_1 R_2 x^2 + Sy^2 \equiv 1 \pmod N, \tag{2}$$

where $x_3 = \frac{x_1 x_2}{S y_1 y_2 + 1}$ and $y_3 = \frac{y_1 + y_2}{S y_1 y_2 + 1}$.

During encryption/decryption, BasicIBE solves the following equations:

$$R_i x_i^2 + S y_i^2 \equiv 1 \pmod N \textit{ and } ux^2 + Sy^2 \equiv 1 \pmod N \tag{3}$$

and then uses Lemma 1 to find solutions to $uR_i\overline{x}_i^2 + S\overline{y}_i^2 \equiv 1 \pmod N$.

4 The JB System [11]

4.1 JB Product Formula

Jhanwar and Barua [11] presented a variant of BasicIBE (the JB System). They used a variant of Lemma 1 to implement their system. This lemma states that:

Lemma 2. *For $i = 1, 2$ let (x_i, y_i) be a solution to $Rx^2 + S_i y^2 \equiv 1 \pmod N$. Then (x_3, y_3) is a solution to*

$$Rx^2 + S_1 S_2 y^2 \equiv 1 \pmod N, \tag{4}$$

where $x_3 = \frac{x_1 + x_2}{R x_1 x_2 + 1}$ and $y_3 = \frac{y_1 y_2}{R x_1 x_2 + 1}$.

4.2 The System Structure

The JB system is explained as follows.

- Setup(λ): Using RSAgen(λ), generate (p,q). Calculate the modulus $N \leftarrow pq$. Choose a random $u \in J(N) \setminus QR(N)$ and choose a hash function $H : ID \to J(N)$. The public parameters PP are $[N, u, H]$. The master secret MSK parameters are p, q and a secret key K for a pseudorandom function $F_K : ID \to \{0, 1, 2, 3\}$.
- KeyGen(MSK, ID): Calculate $R \leftarrow H(ID) \in J(N)$ and $w \leftarrow F_K(ID) \in \{0, 1, 2, 3\}$. Choose $a \in \{0, 1\}$ such that $u^a R \in QR(N)$. Let $[z_0, z_1, z_2, z_3]$ be the four square roots of $u^a R \in \mathbb{Z}_N$, then $r \leftarrow z_w$.

– Encryption(PP, ID, m): To encrypt a message $m \in \{-1,1\}^l$ execute the following algorithm:

$k \leftarrow \sqrt{l}$, $R \leftarrow H(ID) \in J(N)$
foreach $i \in [1,l]$ **do**
  if $i \leq k$ **then**
    $s_i \in_R \mathbb{Z}_N$, $s_i^2 \equiv S_i \pmod N$, $[x_i, y_i] \leftarrow Rx_i^2 + S_i y_i^2 \equiv 1 \pmod N$
    $[\overline{x}_i, \overline{y}_i] \leftarrow uR\overline{x}_i^2 + S_i\overline{y}_i^2 \equiv 1 \pmod N$
    $c_i \leftarrow m_i \cdot \left(\frac{2y_i s_i + 2}{N}\right)$, $\overline{c}_i \leftarrow m_i \cdot \left(\frac{2\overline{y}_i s_i + 2}{N}\right)$
  else
    $(j_1, j_2) \leftarrow i = k \cdot j_1 + j_2$, $y_{j_1,j_2} \leftarrow \frac{y_{j_1} y_{j_2}}{Rx_{j_1} x_{j_2} + 1}$, $\overline{y}_{j_1,j_2} \leftarrow \frac{\overline{y}_{j_1}\overline{y}_{j_2}}{uR\overline{x}_{j_1}\overline{x}_{j_2}+1}$
    $c_i \leftarrow m_i \cdot \left(\frac{2y_{j_1,j_2} s_{j_1} s_{j_2} + 2}{N}\right)$, $\overline{c}_i \leftarrow m_i \cdot \left(\frac{2\overline{y}_{j_1,j_2} s_{j_1} s_{j_2} + 2}{N}\right)$
  end
  $c \leftarrow [c_1, ..., c_l], \overline{c} \leftarrow [\overline{c}_1, ..., \overline{c}_l], x \leftarrow [x_1, ..., x_k]$ and $\overline{x} \leftarrow [\overline{x}_1, ..., \overline{x}_k]$
end

The ciphertext is $C \leftarrow (c, \overline{c}, x, \overline{x})$

– Decrypt(C, r): To decrypt $C \leftarrow (c, \overline{c}, x, \overline{x})$, execute the following algorithm:

foreach $i \in [1,l]$ **do**
  if $r^2 = R$ **then**
    if $i > k$ **then**
      $(j_1, j_2) \leftarrow i = k \cdot j_1 + j_2$, $x_i \leftarrow \frac{x_{j_1} + x_{j_2}}{Rx_{j_1} x_{j_2} + 1}$
    end
    $m_i \leftarrow c_i \cdot \left(\frac{x_i r + 1}{N}\right)$
  end
  if $r_i^2 = uR$ **then**
    if $i > k$ **then**
      $(j_1, j_2) \leftarrow i = k \cdot j_1 + j_2$, $\overline{x}_i \leftarrow \frac{\overline{x}_{j_1} + \overline{x}_{j_2}}{uR\overline{x}_{j_1}\overline{x}_{j_2+1}}$
    end
    $m_i \leftarrow \overline{c}_i \cdot \left(\frac{\overline{x}_i r + 1}{N}\right)$
  end
end

5 The Security Flaw of the JB System

The idea behind the JB system is based on a time-space trade-off of BasicIBE [8]. To decrease the number of $y, \overline{y}$ elements, the system solves only two sets of $k = \sqrt{l}$ equations. Each set is used to generate c and $\overline{c}$ respectively. A bit $m_{i<k}$ is encrypted with $y_i, \overline{y}_i$ while a bit $m_{i>k}$ is encrypted with $y_{j_1,j_2} \leftarrow (y_{j_1}, y_{j_2})$ where $i = k \cdot j_1 + j_2$. Assume that there are two bits $m_{i_1>k}, m_{i_2>k}$ where $i_1 = k \cdot j_1 + j_2$ and $i_2 = k \cdot j_2 + j_1$, then $x_{j_1,j_2} = x_{j_2,j_1} = \frac{x_{j_1} + x_{j_2}}{Rx_{j_1} x_{j_2} + 1}$ and $y_{j_1,j_2} = y_{j_2,j_1} = \frac{y_{j_1} y_{j_2}}{Rx_{j_1} x_{j_2} + 1}$. Consequently, bits i_1, i_2 are encrypted/decrypted using the same key. The same idea goes for $\overline{x}_{j_1,j_2}, \overline{y}_{j_1,j_2}$. Based on this security flaw, an IND-ID-CPA adversary can win an IND-ID-CPA game against this system as follows:

- An adversary $\mathcal{A}$ chooses $i_1, i_2 > k$ such that $i_1 = k \cdot j_1 + j_2$ and $i_2 = k \cdot j_2 + j_1$.
- In the challenge phase, $\mathcal{A}$ sends to the challenger $\mathcal{C}$ two messages $m, \overline{m}$. These messages are chosen at random with $m_{i_1} = m_{i_2}$ and $\overline{m}_{i_1} \neq \overline{m}_{i_2}$.
- In the guess phase, the adversary $\mathcal{A}$ checks the bits c_{i_1}, c_{i_2}. If $c_{i_1} = c_{i_2}$ then $b = 0$ and if $c_{i_1} \neq c_{i_2}$ then $b = 1$.

To overcome this security flaw, j_1 must not be equal to j_2 for all values of j_1 and j_2. i.e., $j_1 \neq j_2$ for all $[j_1, j_2] \in [1, k]$. This means that the number of k equations (i.e. the number of y elements) required for encrypting a message with length l is more than $\sqrt{l}$. Next, we deduce the relation between k and l in order to make the JB system IND-ID-CPA secure. Fig. 1 represents a message m as a table.

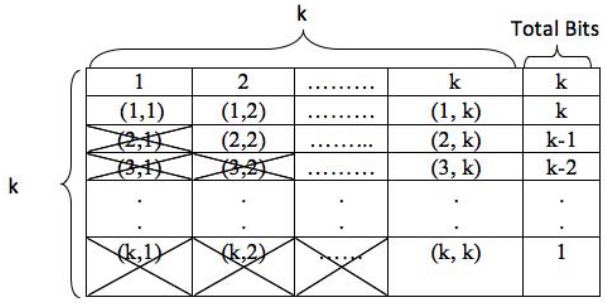

Fig. 1. The maximum number of l-bit encrypted by k elements of y_{j_1,j_2}

Each row is encrypted using k elements of y_i. The first row is encrypted by the first k elements of y_i. The second row is encrypted by the combination of y_1 and all values of $y_1, ..., y_k$. The third row is encrypted by the combination of y_2 and all values of $y_1, ..., y_k$ and so on. In the third row, the value $y_{2,1}$ is eliminated because it is equal to $y_{1,2}$. In the fourth row, the values of $y_{3,1}$ and $y_{3,2}$ are eliminated because they are equal to $y_{1,3}$ and $y_{2,3}$ respectively. Symmetrically, one can find the number of eliminated bits in each row until the last row, where only $y_{k,k}$ is used. The maximum number of bits that can be encrypted using k values of y is:

$$l \leq k + k + k - 1 + k - 2 + ... + 1 \leq \frac{k^2 + 3k}{2}. \tag{5}$$

For example, if the message length is 100 bits, then the minimum number of solved equations must be $200 \leq k^2 + 3k$, $k \geq 13$, which is larger than $\sqrt{l} = \sqrt{100} = 10$.

6 The JB System Security Proof

In this section, we first point out a flaw in the security proof presented in [11] for the JB system, then we present two rigorous security proofs for the fixed system. In the JB system security proof, the authors assumed that, if an adversary $\mathcal{A}$ guessed the first k Jacobi symbols on the form $\left(\frac{2y_{j_1}s_{j_1}+2}{N}\right)$ and $\left(\frac{2y_{j_2}s_{j_2}+2}{N}\right)$, he will be able to guess the distribution of the rest $l-k$ Jacobi symbols $\left(\frac{2y_{j_1,j_2}s_{j_1}s_{j_2}+2}{N}\right)$. That is obviously because $y_{j_1,j_2}s_{j_1}s_{j_2}$ depends on $y_{j_1}s_{j_1}$ and $y_{j_2}s_{j_2}$ and consequently, the JB security is reduced by a factor of $\frac{1}{2^k}$. We prove that this claim needs revision. In fact, we prove that guessing the Jacobi symbols $\left(\frac{2y_{j_1,j_2}s_{j_1}s_{j_2}+2}{N}\right)$ from $\left(\frac{2y_{j_1}s_{j_1}+2}{N}\right)$ and $\left(\frac{2y_{j_2}s_{j_2}+2}{N}\right)$ is as hard as guessing them from other independent Jacobi symbols $\left(\frac{2y_js_j+2}{N}\right)$ and $\left(\frac{2y_is_i+2}{N}\right)$. That is because Damgard [14] showed that the distribution of Jacobi symbols sequences is random. If an adversary knows $\left(\frac{a}{N}\right)$, it is a hard problem for him to find $\left(\frac{a+1}{N}\right)$ for an unknown value a. Although a and $a+1$ are highly related, the Jacobi symbols $\left(\frac{a}{N}\right)$ and $\left(\frac{a+1}{N}\right)$ look random and indistinguishable from the adversary point of view. Based on the above, we present the following Lemma.

Lemma 3. *The distribution of the last $l-k$ bits of the JB system encryption key in the form of $\left(\frac{2y_{j_1,j_2}s_{j_1}s_{j_2}+2}{N}\right)$ does not depend on the distribution of the first k bits in the form $\left(\frac{2y_{j_1}s_{j_1}+2}{N}\right)$ and $\left(\frac{2y_{j_2}s_{j_2}+2}{N}\right)$.*

Proof. Damgard proved that the following is a hard problem [14].

Theorem 1. *Let J be the Jacobi sequence modulo N with a starting point a and length $P(k)$, for a security parameter k and polynomial P. Given J, find $\left(\frac{a+P(k)+1}{N}\right)$.*

This means that, knowing $\left(\frac{a}{N}\right), \left(\frac{a+1}{N}\right), \left(\frac{a+2}{N}\right), ..., \left(\frac{a+a_1}{N}\right), ..., \left(\frac{a+a_2}{N}\right), ..., \left(\frac{a+P}{N}\right)$, it is a hard problem to find $\left(\frac{a+P+1}{N}\right)$.

We first choose a and P such that $a+P+1 = 2y_{j_1,j_2}s_{j_1}s_{j_2}+2$, then we can write the above sequence in two different forms:

$$\left(\frac{a}{N}\right), \left(\frac{a+1}{N}\right), \left(\frac{a+2}{N}\right), ..., \left(\frac{2y_{j_1}s_{j_1}+2}{N}\right), ..., \left(\frac{2y_{j_2}s_{j_2}+2}{N}\right), ..., \left(\frac{a+P}{N}\right)$$

where $a_1 = 2y_{j_1}s_{j_1}+2-a,\ a_2 = 2y_{j_2}s_{j_2}+2-a$.

$$\left(\frac{a}{N}\right), \left(\frac{a+1}{N}\right), \left(\frac{a+2}{N}\right), ..., \left(\frac{2y_js_j+2}{N}\right), ..., \left(\frac{2y_is_i+2}{N}\right), ..., \left(\frac{a+P}{N}\right)$$

where $a_1 = 2y_js_j+2-a,\ a_2 = 2y_is_i+2-a$.

Since $\mathbb{Z}_N$ is an additive group, the values of a_1, a_2 and P exist in both sequences for any value y or s. From the above equations, guessing the Jacobi symbol $\left(\frac{2y_i s_{j_1} s_{j_2}+2}{N}\right)$ from $\left(\frac{2y_{j_1} s_{j_1}+2}{N}\right)$ and $\left(\frac{2y_{j_2} s_{j_2}+2}{N}\right)$ is as hard as guessing them from independent Jacobi symbols. □

We note here that in the JB system, it is much harder to guess the Jacobi symbols $\left(\frac{2y_{j_1,j_2} s_{j_1} s_{j_2}+2}{N}\right)$ than the Damgard problem because the only available Jacobi symbols in the whole sequence are $\left(\frac{2y_{j_1} s_{j_1}+2}{N}\right)$ and $\left(\frac{2y_{j_2} s_{j_2}+2}{N}\right)$.

We now present two different security proofs for the fixed JB system.

Theorem 2. *Suppose the QR assumption holds for RSAgen and F is a secure PRF. Then the proposed JB system is IND-ID-CPA secure based on the QR assumption when H is modelled as a random oracle. In particular, suppose $\mathcal{A}$ is an efficient IND-ID-CPA adversary, then there exist efficient algorithms B_1, B_2 whose running time is the same as that of $\mathcal{A}$ such that:*

$$IBEAdv_{\mathcal{A},JB}(\lambda) \leq 2QRAdv_{B_2,RSAgen}(\lambda) + PRFAdv_{B_1,F}(\lambda).$$

To prove this theorem, we first introduce Lemma 4.

Lemma 4. *Let $N = pq$ be an RSA modulus, $S, R \in J(N)$. Then*

- *1-When $R \in J(N) \setminus QR(N)$, $S \in QR(N)$, the Jacobi symbols $\left(\frac{g(s)}{N}\right)$ for any function g are uniformly distributed in $\{\pm 1\}$, where s is a random variable uniformly chosen among the four square roots of S modulo N and $g(s)g(-s)R \in QR(N)$ for all the four values of s.*
- *2-When $S \in J(N) \setminus QR(N)$, $R \in QR(N)$, the Jacobi symbols $\left(\frac{f(r)}{N}\right)$ for any function f are uniformly distributed in $\{\pm 1\}$, where r is a random variable uniformly chosen among the four square roots of R modulo N and $f(r)f(-r)S \in QR(N)$ for all the four values of r.*
- *3-When $S, R \in QR(N)$, the Jacobi symbols $\left(\frac{g(s)}{N}\right)$ and $\left(\frac{f(r)}{N}\right)$ are constant, i.e. the same for all four values of r and s .*

Proof. Let $s, \overline{s}$ be the four square roots of $S \in QR(N)$ such that $\overline{s} = s \pmod{p}$ and $\overline{s} = -s \pmod{q}$, then the four square roots of S are $\{\pm\overline{s}, \pm s\}$. We can assume the same for $R \in QR(N)$ and the four square roots are $\{\pm\overline{r}, \pm r\}$, where $\overline{r} = r \pmod{p}$ and $\overline{r} = -r \pmod{q}$.

Case 1

$$\left(\frac{g(s)g(-s)R}{N}\right) = \left(\frac{g(s)g(-s)R}{p}\right) = \left(\frac{g(s)g(-s)R}{q}\right) = 1.$$
$$\left(\frac{R}{p}\right) = \left(\frac{R}{q}\right) = -1,$$
$$\left(\frac{g(s)g(-s)}{p}\right) = \left(\frac{g(s)g(-s)}{q}\right) = -1,$$
$$\left(\frac{g(s)}{p}\right) = -\left(\frac{g(-s)}{p}\right) \ \textit{and} \ \left(\frac{g(s)}{q}\right) = -\left(\frac{g(-s)}{q}\right),$$
$$\left(\frac{g(s)}{N}\right) = \left(\frac{g(-s)}{N}\right).$$
$$\left(\frac{g(\overline{s})}{p}\right) = \left(\frac{g(s)}{p}\right).$$
$$\left(\frac{g(\overline{s})}{q}\right) = \left(\frac{g(-s)}{q}\right) = -\left(\frac{g(s)}{q}\right),$$
$$\left(\frac{g(\overline{s})}{p}\right)\left(\frac{g(\overline{s})}{q}\right) = -\left(\frac{g(s)}{p}\right)\left(\frac{g(s)}{q}\right),$$
$$\left(\frac{g(\overline{s})}{N}\right) = -\left(\frac{g(s)}{N}\right),$$
$$\left(\frac{g(\overline{s})}{N}\right) = \left(\frac{g(-\overline{s})}{N}\right) = -\left(\frac{g(s)}{N}\right) = -\left(\frac{g(-s)}{N}\right).$$

That means that among the four Jacobi symbols $\left(\frac{g(\overline{a})}{N}\right), \left(\frac{g(-\overline{a})}{N}\right), \left(\frac{g(a)}{N}\right), \left(\frac{g(-a)}{N}\right)$ two are $+1$ and two are -1. Case 2 and Case 3 can be proven similarly to Case 1.

(*High Level Idea of the Proof*). Before presenting the formal proof, we first illustrate the idea of the proof as follows. The security proof is based on successfully proving that the distribution of the Jacobi symbols $\left(\frac{2y_i s_i+2}{N}\right)$, $\left(\frac{2\overline{y}_i s_i+2}{N}\right)$, $\left(\frac{x_i r+1}{N}\right)$ and $\left(\frac{\overline{x}_i r+1}{N}\right)$ are random in $\{\pm 1\}$ and thus, the ciphertext is indistinguishable from random. This is achieved by replacing the variables u, R, S in the equation $uRx_i^2 + Sy_i^2 = 1 \pmod N$ with other variables based on the QR problem such that one of Case 1 or Case 2 in Lemma 4 holds.

We define two sequences of games and let W_i represents the winning of the i_{th} game and $\overline{W}_i$ represents the winning of the $\overline{i}_{th}$ game by the adversary $\mathcal{A}$. Any of these sequences proves the security of the JB system. These games are defined as follows.

- **Game-0**. This game is the usual adversarial game.
- **Game-1**. This game replaces the PRF F with a truly random function.
- **Game-2**. This game explains how to simulate the hash function H.

- **Game-3**. This game sets $u \in QR(N)$.
- **Game-4**. This game explains how to respond to an encryption query from $\mathcal{A}$.
- **Game-$\overline{4}$**. This game explains how to respond to an encryption query using the *Decrypt* algorithm.
- **Game-5**. This game sets $R \in J(N) \setminus QR(N)$.
- **Game-$\overline{5}$**. This game sets $S_i \in J(N) \setminus QR(N)$.
- **Game-6** and **Game-$\overline{6}$** replace the message m with a random number z.

The detail of the proof is as follows.

- Game-0. This is the usual adversarial game for defining the IND-ID-CPA security of IBE protocols. The challenger picks the random oracle $H : ID \to J(N)$ at random from the set of all such functions in the *Setup* algorithm and allows $\mathcal{A}$ to query H at arbitrary points. Thus, we have
$$|\Pr[W_0] - \frac{1}{2}| = IBEAdv_{\mathcal{A},JB}(\lambda).$$
- Game-1. This is the same as Game-0, with the following change. In *Setup* algorithm, instead of using a PRF F to respond to $\mathcal{A}$'s private key queries, we use a truly random function $f : ID \to \{0, 1, 2, 3\}$. If F is a secure PRF, $\mathcal{A}$ will not notice the difference between Game-0 and Game-1. In particular, there exists an algorithm B_1 (whose running time is about the same as that of $\mathcal{A}$) such that
$$|\Pr[W_1] - \Pr[W_0]| = PRFAdv_{B_1,F}(\lambda).$$
- Game-2. (N, u, H) are the public parameters PP given to $\mathcal{A}$ in the previous game where u is uniform in $J(N) \setminus QR(N)$ and the random oracle H is a random function $H : ID \to J(N)$. We make the following change in the random oracle H in this game. The challenger responds to a query to $H(ID)$ by picking $a \in_R \{0, 1\}$ and $v \in_R \mathbb{Z}_N$ and setting $H(ID) = u^a v^2$. Thus the challenger implements a random function $H : ID \to J(N)$ as in the previous game. The challenger responds to a private key query as follows.
Suppose $R = H(ID) = u^a v^2$ for some $a \in_R \{0, 1\}$ and $v \in_R \mathbb{Z}_N$. The challenger responds to a private key query for ID by setting either $R^{\frac{1}{2}} = v$ (when $a = 0$) or $uR^{\frac{1}{2}} = uv$ (when $a = 1$). Since v is uniform in $\mathbb{Z}_N$ this will produce a square root of R or uR which is also uniform among the four square roots, as in the previous game. Thus, $\mathcal{A}$'s views in Game-1 and Game-2 are identical and therefore,
$$|\Pr[W_1] = \Pr[W_2]|.$$
- Game-3. In this game, the challenger chooses u uniformly in $QR(N)$ instead of $J(N) \setminus QR(N)$. Since this is the only change between Game-2 and Game-3, $\mathcal{A}$ will not notice the difference assuming that the QR assumption holds for RSAgen. In particular, there exists an algorithm B_2 (whose running time is about the same as that of $\mathcal{A}$) such that:
$$|\Pr[W_3] - \Pr[W_2]| = QRAdv_{B_2,RSAgen}(\lambda).$$

– Game-4. We describe below in detail how, in this game, the challenger responds to an encryption query from $\mathcal{A}$.
  - He chooses $R \in QR(N)$ and sets $H(ID) = R$. (*)
  - He chooses $s \in_R \mathbb{Z}_N$ and computes $S_i = s_i^2$.
  - He sets $c \leftarrow Encrypt(PP, ID, m)$.
  - He sends (S, c) to $\mathcal{A}$.

  Since this game is the same as Game-3, thus:

$$|\Pr[W_4] = \Pr[W_3]|.$$

– Game-$\overline{4}$. This game is the same as Game-3 except that the challenger handles encryption queries from $\mathcal{A}$ differently. To encrypt a message m for an identity ID the challenger does the following.
  - He chooses $R \in QR(N)$ and sets $H(ID) = R$.
  - He chooses $s \in_R \mathbb{Z}_N$ and computes $S_i = s_i^2$. (*)
  - He sets $c \leftarrow Decrypt(r, PP, (S, m))$.
  - He sends (S, c) to $\mathcal{A}$.

  It is easy to see that $c_i = m_i \cdot \left(\frac{1+x_i r}{N}\right)$ is a unique encryption of m. Since this game is the same as Game-3, thus:

$$|\Pr[\overline{W}_4] = \Pr[W_3]|.$$

– Game-5. In this game, we make a change in the challenge phase. We replace the line (*) in Game-4 with the following:
  - He chooses $R \in J(N) \setminus QR(N)$ and sets $H(ID) = R$.

  Since the only difference between Game-5 and Game-4 is that $R \in J(N) \setminus QR(N)$ in Game-5 instead of $R \in QR(N)$ in Game-4, $\mathcal{A}$ will not notice the difference assuming that the QR assumption holds for RSAgen. In particular, there exists an algorithm B_2 (whose running time is about the same as that of $\mathcal{A}$) such that:

$$|\Pr[W_5] - \Pr[W_4]| = QRAdv_{B_2,RSAgen}(\lambda).$$

– Game-$\overline{5}$. This game is similar to Game-5. We replace the line (*) in Game-$\overline{4}$ with the following:
  - He chooses $s \in_R \mathbb{Z}_N$ and computes $S_i = -s_i^2$ for the first k bits and sets $S_i = -S_{j_1} S_{j_1} = -s_i^2 = -(s_{j_1} s_{j_1})^2, i = k \cdot j_1 + j_2$ for the last $l - k$ bits.

  Since $S_i = -s_i^2$, $S_i \in J(N) \setminus QR(N)$. The only difference between Game-$\overline{5}$ and Game-$\overline{4}$ is that $S_i \in J(N) \setminus QR(N)$ in Game-$\overline{5}$ instead of $S_i \in QR(N)$ in Game-$\overline{4}$ so $\mathcal{A}$ will not notice the difference assuming that the QR assumption holds for RSAgen. In particular, there exists an algorithm B_2 (whose running time is about the same as that of $\mathcal{A}$) such that:

$$|\Pr[\overline{W}_5] - \Pr[\overline{W}_4]| = QRAdv_{B_2,RSAgen}(\lambda).$$

– Game-6: In this game, we replace the message $m^{(b)}$ by a random string $z \in_R \{-1, 1\}^l$ i.e., $c_i = z_i \cdot \left(\frac{2y_i s_i + 2}{N}\right)$ and $\overline{c}_i = z_i \cdot \left(\frac{2\overline{y}_i s_i + 2}{N}\right)$ where $y_i = y_{j_1, j_2}$ and $s_i = s_{j_1} s_{j_2}$, $i = k \cdot j_1 + j_2$ for the last $l - k$ bits. We first prove that $(2y_i s_i + 2)(-2y_i s_i + 2)R \in QR(N)$.

Proof. Let $g(s_i) = (2y_i s_i + 2)$, then we have

$$g(s_i)g(-s_i)R = 4(y_i s_i + 1)(-y_i s_i + 1)R,$$
$$g(s_i)g(-s_i)R = 4(1 - (y_i s_i)^2),$$
$$g(s_i)g(-s_i)R = 4(Rx_i^2)R = (2Rx_i)^2 \in QR(N).$$

Similarly, we can prove that $(2\overline{y}_i s_i + 2)(-2\overline{y}_i s_i + 2)uR \in QR(N)$.
Since $S_i \in QR(N)$, $R \in J(N) \setminus QR(N)$, $(2y_i s_i + 2)(-2y_i s_i + 2)R \in QR(N)$ and $(2\overline{y}_i s_i + 2)(-2\overline{y}_i s_i + 2)uR \in QR(N)$ and based on Lemma 3, Case 1 in lemma 4 can be applied and the distribution of the Jacobi symbols $\left(\frac{2y_i s_i+2}{N}\right)$ and $\left(\frac{2\overline{y}_i s_i+2}{N}\right)$ are random in $\{\pm 1\}$. Thus, $\mathcal{A}$ will not be able to distinguish between Game-5 and Game-6. i.e.

$$|\Pr[W_6] = \Pr[W_5]|.$$

– Game-$\overline{6}$: In this game, we replace the message $m^{(b)}$ by a random string $z \in_R \{-1, 1\}^l$ i.e., $c_i = z_i \cdot \left(\frac{x_i r+1}{N}\right)$ and $\overline{c}_i = z_i \cdot \left(\frac{\overline{x}_i r+1}{N}\right)$. We first prove that $(xr + 1)(-xr + 1)S_i \in QR(N)$.

Proof. Let $f(r) = (xr + 1)$, then we have

$$f(r)f(-r)S_i = (x_i r + 1)(-x_i r + 1)S_i,$$
$$f(r)f(-r)S_i = (1 - (x_i r)^2) = 1 - x_i^2 R,$$
$$f(r)f(-r)S_i = (S_i y_i^2)S_i = (S_i y_i)^2 \in QR(N).$$

Similarly, we can prove that $(\overline{x}_i r + 1)(-\overline{x}_i r + 1)S_i \in QR(N)$.
Since $R \in QR(N)$, $S_i \in J(N) \setminus QR(N)$, $(xr + 1)(-xr + 1)S_i \in QR(N)$ and $(\overline{x}_i r + 1)(-\overline{x}_i r + 1)S_i \in QR(N)$ and based on Lemma 3, Case 2 in lemma 4 can be applied and the distribution of the Jacobi symbols $\left(\frac{x_i r+1}{N}\right)$ and $\left(\frac{\overline{x}_i r+1}{N}\right)$ are random in $\{\pm 1\}$. Thus, $\mathcal{A}$ will not be able to distinguish between Game-$\overline{5}$ and Game-$\overline{6}$. i.e.

$$|\Pr[\overline{W}_6] = \Pr[\overline{W}_5]|.$$

– Clearly in Game-6 and Game-$\overline{6}$ we have

$$|\Pr[W_6] = \Pr[\overline{W}_6] = \frac{1}{2}|.$$

Combining all the previous equations proves theorem.

7 Conclusion

In this paper, we reviewed the security of the JB system. We showed that this system is not IND-ID-CPA secure. We also presented a solution to overcome this security flaw. We also pointed out a flaw of the security proof of the JB system and presented two security proofs that show that the fixed JB system is as secure as the original BasicIBE system.

References

1. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
2. Boneh, D., Franklin, M.: Identity-based encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
3. Gentry, C.: Practical Identity-Based Encryption Without Random Oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
4. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
5. Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
6. Boneh, D., Boyen, X.: Secure Identity Based Encryption Without Random Oracles. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 443–459. Springer, Heidelberg (2004)
7. Cocks, C.: An identity based encryption scheme based on quadratic residues. In: Honary, B. (ed.) Cryptography and Coding 2001. LNCS, vol. 2260, pp. 360–363. Springer, Heidelberg (2001)
8. Boneh, D., Gentry, C., Hamburg, M.: Space-Efficient Identity Based Encryption Without Pairings. In: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2007, pp. 647–657. IEEE Computer Society (2007)
9. Cremona, J.E., Rusin, D.: Efficient solution of rational conics. Math. Comput. 72, 1417–1441 (2003)
10. Cohen, H.: A course in computational algebraic number theory. Springer-Verlag New York, Inc., New York (1993)
11. Jhanwar, M., Barua, R.: A Variant of Boneh-Gentry-Hamburg's Pairing-Free Identity Based Encryption Scheme. In: Yung, M., Liu, P., Lin, D. (eds.) Inscrypt 2008. LNCS, vol. 5487, pp. 314–331. Springer, Heidelberg (2009)
12. Boneh, D., Raghunathan, A., Segev, G.: Function-private identity-based encryption: Hiding the function in functional encryption. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 461–478. Springer, Heidelberg (2013)
13. Barua, R., Jhanwar, M.: On the number of solutions of the equation $Rx^2 + Sy^2 = 1 mod N$. Sankhya A - Mathematical Statistics and Probability 72, 226–236 (2010), 10.1007/s13171-010-0010-9
14. Damgård, I.B.: On the Randomness of Legendre and Jacobi Sequences. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 163–172. Springer, Heidelberg (1990)