An Efficient Variant of Boneh-Gentry-Hamburg's Identity-Based Encryption Without Pairing

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, Australia
ifeae231@uowmail.edu.au, {ymu,wsusilo}@uow.edu.au

Abstract. Boneh, Gentry and Hamburg presented an encryption system known as BasicIBE without incorporating pairings. This system has short ciphertext size but this comes at the cost of less time-efficient encryption/decryption algorithms in which their processing time increases drastically with the message length. Moreover, the private key size is l elements in $\mathbb{Z}_N$, where N is a Blum integer and l is the message length. In this paper, we optimize this system in two steps. First, we decrease the private key length from l elements in $\mathbb{Z}_N$ to only one element. Second, we present two efficient variants of the BasicIBE in terms of ciphertext length and encryption/decryption speed. The ciphertext is as short as the BasicIBE, but with more time-efficient algorithms which do not depend on the message length. The proposed system is very time efficient compared to other IBE systems and it is as secure as the BasicIBE system.

Keywords: Identity-based encryption · Quadratic residuosity assumption · IND-ID-CPA

1 Introduction

In 1985, Shamir [12] 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. Shamir successfully managed to design an identity-based signature based on the RSA algorithm but he was unable to design an IBE because sharing an RSA modulus between different users makes RSA insecure [12]. The design of a provable secure IBE remained an open problem for sixteen years until Boneh and Franklin [4] 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 [2,3,10,13].

However, all the previously mentioned IBEs are 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

K.-H. Rhee and J.H. Yi (Eds.): WISA 2014, LNCS 8909, pp. 257–268, 2015.
DOI: 10.1007/978-3-319-15087-1_20

limits it from being used in wide applications, 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 [6]. 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 [5] 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 [5]. Solving such an equation requires a 'solubility certificate' and obtaining these certificates requires the generation of primes [6–8]. The obtained certificates can be used to solve $Rx^2 + Sy^2 \equiv 1 \pmod N$ efficiently using the Cremona-Rusin algorithm [8]. 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 [5]. 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 $\mathbb{Z}_N$.

Our Contribution. In this paper, we first present some definitions and review Basic IBE. After that, we optimise BasicIBE in two steps. First, we prove that hashing the identity ID to a different value to encrypt each bit is as secure as hashing the identity once to encrypt the whole message and therefore, the private key length is reduced to one element in $\mathbb{Z}_N$. Then, we present a variant of BasicIBE (V-BasicIBE) which is both time- and space- efficient. Moreover, we prove that V-BasicIBE is as secure as BasicIBE. Although the proposed variant has the same ciphertext length as BasicIBE, it only solves two equations in the form $Rx^2 + Sy^2 \equiv 1 \pmod N$ regardless of the message length. We also present

another version of V-BasicIBE with a time-space trade-off. For V-BasicIBE, with only the cost of one more element in $\mathbb{Z}_N$, the sender can find a solution to $Rx^2+Sy^2 \equiv 1 \pmod N$ using only one inversion in $\mathbb{Z}_N$ and the receiver does not have to solve any of these equations. The proposed variant is time- and power-efficient compared to other IBE systems. It does not use expensive-computational operations such as pairing like Boneh-Boyen or Boneh-Franklin IBEs [2,4] or even a prime modular exponentiation such as RSA. Table 1 compares all systems in this paper, where V2-BasicIBE is the proposed systems with the trade-off applied. In this 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 symbols G and G_T represents an element in two groups G and G_T such that $e : G \times G \rightarrow G_T$.

Table 1. Comparison between various IBEs and the proposed IBEs

	Expensive mathematical operations	Ciphertext length
Cock's	0	$2l(\log N)$
The BasicIBE	$(l+1)p$	$\log N + 2l$
The AnonIBE	$(2l+1)p$	$\log N + l + 1$
V-BasicIBE	$2p$	$\log_2 N + 2l$
V2-BasicIBE	0	$2\log_2 N + 2l$
Jhanwar-Barua	0	$2\sqrt{l}\log N + 2l$
Boneh-Boyen	e+3m	G_T+2G
Boneh-Franklin	e	G+l

2 Definitions

2.1 IND-ID-CPA

The IND-ID-CPA security model of an IBE is described as a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ [4,12]. 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:

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

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:

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

where $\left(\frac{a}{N}\right)$ is the Jacobi symbol of a w.r.t N [5]. 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 $\mathcal{A}$ [5].*

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

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

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 Review of the BasicIBE System [5]

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 $m_i = \overline{c}_i \cdot \left(\frac{1+\overline{x}_i r_i}{N}\right)$.

4 Optimization of BasicIBE

4.1 Optimization of the Private Key Length

As shown above, the BasicIBE system hashes the identity ID to different values $H(ID, i) = u^a R_i = r_i^2, a \in \{0, 1\}$. This has a negative impact on the system. First, the private key length is larger than the message by a factor of $\mathbb{Z}_N$ which consumes bandwidth and memory. Second, the Private Key Generator (PKG) must generate n private keys of l elements in $\mathbb{Z}_N$ where n is the number of users in the whole system. This overloads the PKG. Third, this not suitable for encrypting variable messages length.

In this section, we prove that hashing the identity ID to different values $R_i = H(ID, i)$ does not have a positive impact on the security of BasicIBE. Solving the equations $Rx_i^2 + Sy_i^2 \equiv 1 \pmod N$ is exactly equivalent to solving the equations $R_i x_i^2 + Sy_i^2 \equiv 1 \pmod N$. Consequently, there is no need for generating a long private key of l elements in $\mathbb{Z}_{\mathbb{N}}$.

Theorem 1. *Hashing the identity ID to a different value to encrypt each bit is as secure as hashing the identity once to encrypt the whole message.*

Proof. Jhanwar and Barua [1] showed that there is $N - 1$ solutions for the equation $Rx^2 + Sy^2 \equiv 1 \pmod N$ if $S, R \in QR(N)$. The solution (x, y) for that equation is in the form:

$$\left(\frac{-2st}{R + St^2}, \frac{R - St^2}{s(R + St^2)}\right)$$

for some $t \in \mathbb{Z}_N^*$ such that $R + St^2 \in \mathbb{Z}_N^*$.

$$Rx_i^2 + Sy_i^2 = R\left(\frac{-2st}{R + St^2}\right)^2 + Sy_i^2 = \left(\frac{4SR}{(R + St^2)^2}\right) t^2 + Sy_i^2 = R_i \overline{x}_i^2 + Sy_i^2$$

$$where \;\; R_i = t^2 \;\; and \;\; \overline{x}_i = \frac{-2sr}{R + St^2}.$$

Since t is random in $\mathbb{Z}_N^*$, R_i looks mathematically random exactly as $R_i = H(ID, i)$. □

4.2 V-BasicIBE

In this section, we explain how to implement a variant of BasicIBE (V-BasicIBE) that is both time and space efficient. Like any other IBE, V-BasicIBE consists of four algorithms; Setup, KeyGen, Encrypt and Decrypt.

- Setup(λ): Using RSAgen(λ), generate (p,q), calculate the modulus $N \leftarrow pq$, choose $u \in J(N) \setminus QR(N)$, and choose a hash function $H : ID \rightarrow J(N)$. 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 $F_K : ID \rightarrow [0, 1, 2, 3]$.

- KeyGen(MSK, ID, l): 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 $r \leftarrow z_w$.
- Encrypt(id, m): To encrypt a message $m \in \{-1,1\}^l$, V-BasicIBE calculates $[x_i, y_i, \overline{x}_i, \overline{y}_i]$, $i \in [0, l-1]$ such that these variables satisfy the following equations:

$$[x_i, y_i] \leftarrow Rx_i^2 + S^j y_i^2 \equiv 1 \pmod N \ , \ [\overline{x}_i, \overline{y}_i] \leftarrow uR\overline{x}_i^2 + S^j \overline{y}_i^2 \equiv 1 \pmod N$$

for an odd number $j = 2i + 1$. To solve these equations, we review a product formula presented by Boneh, Gentry and Hamburg [5].

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*

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

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

Proof. By directly substituting the values of x_3 and y_3 in the equation $R_1 R_2 x^2 + Sy^2 \equiv 1 \pmod N$.

Jhanwar and Barua [11] presented 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*

$$Rx^2 + S_1 S_2 y^2 \equiv 1 \pmod N,$$

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

Proof. Same as Lemma 1.

To solve these equations, BasicIBE calculates $[x_0, y_0]$ and then uses Lemma 2 to find $[x_i, y_i]$ as follows.

$$\hat{x} = \frac{2x_0}{Rx_0^2 + 1} \ , \ \hat{y} = \frac{y_0^2}{Rx_0^2 + 1}, \ \ x_i = \frac{\hat{x} + x_{i-1}}{R\hat{x}x_{i-1} + 1} \ , \ y_i = \frac{\hat{y} y_{i-1}}{R\hat{x}x_{i-1} + 1},$$

where $[\hat{x}, \hat{y}]$ is a solution to $R\hat{x}^2 + S^2\hat{y}^2 \equiv 1 \pmod N$. Similarly, $[\overline{x}_i, \overline{y}_i]$ are generated as shown above.
The message $m \leftarrow [m_0, m_1, ..., m_{l-1}]$ is encrypted using the following formula:

$$c_i \leftarrow m_i \cdot \left(\frac{2y_i s^j + 2}{N} \right) \ , \ \overline{c}_i \leftarrow m_i \cdot \left(\frac{2\overline{y}_i s^j + 2}{N} \right).$$

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

– Decrypt(C, r): The message can be retrieved from the ciphertext as follows.

$$m_i \leftarrow c_i \cdot \left(\frac{x_i r + 1}{N}\right) \quad \text{if } r^2 = R \text{ and } m_i \leftarrow \overline{c}_i \cdot \left(\frac{\overline{x}_i r + 1}{N}\right) \quad \text{if } r^2 = uR.$$

Correctness: As in [5], it is easy to prove that:

$$\begin{aligned}(x_i r + 1) \cdot (2y_i s^j + 2) &= 2x_i r y_i s^j + 2x_i r + 2y_i s^j + 2 + (Rx_i^2 + S^j y_i^2 - 1) \\ &= (x_i r + y_i s^j + 1)^2,\end{aligned}$$

$$\left(\frac{x_i r + 1}{N}\right) \cdot \left(\frac{2y_i s^j + 2}{N}\right) = 1, \quad \left(\frac{x_i r + 1}{N}\right) = \left(\frac{2y_i s^j + 2}{N}\right).$$

4.3 V-BasicIBE Security

Theorem 2. *Suppose the quadratic residuosity assumption holds for RSAgen and F is a secure PRF. Then the proposed V-BasicIBE 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:*

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

We first introduce Lemma 3 [5].

Lemma 3. *Let $N = pq$ be an RSA modulus, $S_i, R \in J(N)$. Then*

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

Proof. Let $s_i, \overline{s}_i$ be the four square roots of $S_i \in QR(N)$ such that $\overline{s}_i = s_i \pmod p$ and $\overline{s}_i = -s_i \pmod q$, then the four square roots of S_i are $\{\pm\overline{s}_i, \pm s_i\}$. 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

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

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.

– **Security Proof.** We define a sequence of games and let W_i represents the winning of the i_{th} game by the adversary $\mathcal{A}$. 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-5.** This game sets $R \in J(N) \setminus QR(N)$.
  - **Game-6.** This game sets $S_i = s_i^2$ for each bit.
  - **Game-7** replaces the message m with a random number z.
– Game-0. This is the usual adversarial game for defining the IND-ID-CPA security of IBE protocols. The challenger picks the random oracle $H : ID \rightarrow$

$J(N)$ 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

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

– 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 $S^j = s^{2j}$ for an odd value j.
  - He sets $c \leftarrow Encrypt(PP, ID, m_b)$.
  - He sends (S, c) to $\mathcal{A}$.
– 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-6: In this game, we encrypt the message by choosing $s_i \in \mathbb{Z}_N$ independently and randomly for each bit. In other words, we replace the Jacobi symbols $\left(\frac{2y_i s^j+2}{N}\right)$ and $\left(\frac{2\overline{y}_i s^j+2}{N}\right)$ with the Jacobi symbols $\left(\frac{2y_i s_i+2}{N}\right)$ and $\left(\frac{2\overline{y}_i s_i+2}{N}\right)$ respectively i.e. $c_i = m_i \cdot \left(\frac{2y_i s_i+2}{N}\right)$ and $\overline{c}_i = m_i \cdot \left(\frac{2\overline{y}_i s_i+2}{N}\right)$. To prove that Game-6 is indistinguishable from Game-5, we present the following Theorem.

Theorem 3. *The distribution of the Jacobi symbols* $\left(\frac{2y_i s^j+2}{N}\right)$ *is indistinguishable from the distribution the Jacobi symbols* $\left(\frac{2y_i s_i+2}{N}\right)$.

The proof of this theorem is based on the work of Damgard [9]. He proved that the Jacobi sequences are indistinguishable from random. i.e. if an adversary knows the value of $\left(\frac{a}{N}\right)$, it is a hard problem to find $\left(\frac{a+1}{N}\right)$ for an unknown value a. Although the values of a and $a+1$ are highly correlated and dependent, that does not mean that their Jacobi symbols are correlated. We now present a formal proof for the above theorem.

Proof. Damgard proved that the following is a hard problem [9].

Lemma 4. *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 $a + P + 1 = 2y_i s^j + 2$, then we can write the above sequence in two different forms:

$$\left(\frac{a}{N}\right), \left(\frac{a+1}{N}\right), \left(\frac{a+2}{N}\right), ..., \left(\frac{2y_{i_1} s^{j_1}+2}{N}\right), ..., \left(\frac{2y_{i_2} s^{j_2}+2}{N}\right), ..., \left(\frac{a+P}{N}\right)$$

where $a_1 = 2y_{i_1} s^{j_1} + 2 - a$, $a_2 = 2y_{i_2} s^{j_2} + 2 - a$, and $j_1 < j_2 < j$.

$$\left(\frac{a}{N}\right), \left(\frac{a+1}{N}\right), \left(\frac{a+2}{N}\right), ..., \left(\frac{2y_{i_1} s_{j_1}+2}{N}\right), ..., \left(\frac{2y_{i_2} s_{j_2}+2}{N}\right), ..., \left(\frac{a+P}{N}\right)$$

where $a_1 = 2y_{i_1} s_{j_1} + 2 - a$, $a_2 = 2y_{i_2} s_{j_2} + 2 - a$.

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 which means that both sequences represent the Damgard hard problem. Moreover, guessing the Jacobi symbol $\left(\frac{2y_i s^j+2}{N}\right)$

from the sequence $\left(\frac{2y_is+2}{N}\right)$, $\left(\frac{2y_is^2+2}{N}\right)$,...,$\left(\frac{2y_is^{j-1}+2}{N}\right)$ is as hard as guessing the same Jacobi symbol from the sequence $\left(\frac{2y_is_1+2}{N}\right)$,$\left(\frac{2y_is_2+2}{N}\right)$,...,$\left(\frac{2y_is_j+2}{N}\right)$. The same holds for $\left(\frac{2\overline{y}_is^j+2}{N}\right)$ and $\left(\frac{2\overline{y}_is_i+2}{N}\right)$. □

Based on Theorem 3, $\mathcal{A}$ will not be able to distinguish between Game-5 and Game-6. i.e.

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

– Game-7: In this game, we replace the message $m^{(b)}$ by a random string $z \in_R \{-1,1\}^l$ i.e., $c_i = z_i \cdot \left(\frac{2y_is_i+2}{N}\right)$ and $\overline{c}_i = z_i \cdot \left(\frac{2\overline{y}_is_i+2}{N}\right)$. We first prove that $(2y_is_i+2)(-2y_is_i+2)R \in QR(N)$.

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

$$\begin{aligned} g(s_i)g(-s_i)R &= 4(y_is_i+1)(-y_is_i+1)R, \\ g(s_i)g(-s_i)R &= 4(1-(y_is_i)^2)R, \\ g(s_i)g(-s_i)R &= 4(Rx_i^2)R = (2Rx_i)^2 \in QR(N). \end{aligned}$$

Similarly, we can prove that $(2\overline{y}_is_i+2)(-2\overline{y}_is_i+2)uR \in QR(N)$.

Since $s_i \in QR(N)$, $R \in J(N) \setminus QR(N)$, $(2y_is_i+2)(-2y_is_i+2)R \in QR(N)$ and $(2\overline{y}_is_i+2)(-2\overline{y}_is_i+2)uR \in QR(N)$ then Case 1 in Lemma 3 can be applied and the distribution of the Jacobi symbols $\left(\frac{2y_is_i+2}{N}\right)$ and $\left(\frac{2\overline{y}_is_i+2}{N}\right)$ are random in $\{\pm 1\}$. Thus, $\mathcal{A}$ will not be able to distinguish between Game-6 and Game-7. i.e.

$$|\Pr[W_7] = \Pr[W_6]|.$$

– Clearly in Game-7 we have

$$|\Pr[W_7] = \frac{1}{2}|.$$

Combining all the previous equations proves theorem.

5 Space-Time Tradeoff

In this section, we present a trade-off between the time and the ciphertext length of the proposed systems. For V-BasicIBE, instead of sending S along with c and $\overline{c}$ as the full ciphertext C, the sender sends $C = (x_0, \overline{x}_0, c, \overline{c})$. Thus, he can solve $Rx^2 + Sy^2 \equiv 1 \pmod N$ using only one inversion in $\mathbb{Z}_N$. This results in high encryption speed. In the decryption, the receiver does not have to solve any equations and he can generate x_i or $\overline{x}_i$ (based on if $r^2 = R$ or uR) using Lemma 2. This, of course, comes at the cost of sending one more element in $\mathbb{Z}_N$.

6 Conclusion

This paper proposed a variant of BasicIBE. The proposed variant is more efficient (in terms of computation time) than previous IBE systems. We also proved that the proposed variant has the same security level as the BasicIBE system. Moreover, the proposed systems have only one element in the $\mathbb{Z}_N$ private key instead of l elements in $\mathbb{Z}_N$ as in BasicIBE. We also produced a time-space trade-off variant that is both time- and space-efficient.

References

1. Barua, R., Jhanwar, M.: On the number of solutions of the equation $Rx^2 + Sy^2 = 1 \bmod N$. Sankhya A Math. Stat. Probab. **72**, 226–236 (2010). doi:10.1007/s13171-010-0010-9
2. Boneh, D., Boyen, X.: Efficient selective-ID secure identity-based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
3. 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)
4. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, p. 213. Springer, Heidelberg (2001)
5. 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 '07, pages 647–657. IEEE Computer Society, Washington, DC, USA (2007)
6. 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)
7. Cohen, H.: A Course in Computational Algebraic Number Theory. Springer, New York (1993)
8. Cremona, J.E., Rusin, D.: Efficient solution of rational conics. Math. Comput. **72**(243), 1417–1441 (2003)
9. 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)
10. Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
11. Jhanwar, M.P., 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. 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)
13. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)