Identity-Based Encryption from $e$-th Power Residue Symbols

Xiaopeng Zhao¹, Jinwen Zheng¹, Zhenfu Cao *¹, Xiaolei Dong¹ and Nanyuan Cao²

¹School of Computer Science and Software Engineering, East China Normal University
²Department of Computer Science and Engineering, Shanghai Jiao Tong University

Abstract

Boneh, LaVigne and Sabin (BLS)’s scheme naturally generalizes Cocks’ scheme to higher power residue symbols, but it is less efficient, bandwidth-wise as computing the $e$-th power residue symbols is really time-consuming and ciphertexts are expressed in the form of polynomials. This paper, we improve the efficiency of BLS’s scheme through taking off the part of computing $e$-th power residue symbols in its encryption phase. This modification makes the encryption much more efficient than that in Cocks’ scheme. Our construction also widens BLS’s scheme to the case $e$ is square-free. Furthermore, we generalize the notable Galbraith’s test by introducing the general reciprocity law on function fields. With the help of the extended Galbraith’s test, we show BLS’s scheme is not anonymous in general. We also provide some methods for computing the $e$-th power residue symbols.

Keywords: identity-based encryption; $e$-th power residue symbol; the general reciprocity law on function fields; anonymity.

1 Introduction

Identity-based encryption (IBE) is a public key encryption system, originally proposed by Shamir in 1984 [26], while first achieved in 2001 [5][15]. The motivation of IBE is to solve some existing but unavoidable problems with classic public key encryption systems. For details, it substitutes the Public Key Infrastructure (PKI) for the Public Key Generator (PKG), and thus removes the overhead of certificate management and manages the keys centrally. As a result, IBE systems are more lightweight and scalable than classic ones. If an adversary learns nothing about the identity of a ciphertext, we call this IBE system anonymous. The notion of anonymity was put forward by Bellare et al. [3] as an additional security requirement of an encryption scheme. Constructing cryptosystems from higher power residue symbols has been explored in several studies by researchers. Cao [9] proposed a type of extension of the Goldwasser and Micali’s QR-based cryptosystem [18]. His scheme is based on $k^{th}$-power residues and enables segment encryption instead of bit encryption of Goldwasser and Micali’s cryptosystem. In 2013, Joye and Libert [20] revisited the Goldwasser and Micali’s QR-based cryptosystem using $2^k$-th power residue symbols and described

*Corresponding author: zfcao@sei.ecnu.edu.cn
the most efficient lossy trapdoor function based on quadratic residuosity. Subsequently, Cao [10] proposed a type of extension of Joye and Libert’s cryptosystem based on $k^{th}$-power residues. The extended scheme is more efficient than Joye and Libert’s cryptosystem on decryption speed. Recently, Brier et al. [8] introduced new $p^r q$-based one-way functions and companion signature schemes based on higher-power residue symbols.

1.1 Related Work

Cocks’ IBE scheme [15] is simple and a totally different approach. Encryption only needs several operations modulo an RSA modulus $N$ and the evaluation of Jacobi symbols. Its security is based on the standard quadratic residuosity assumption. Cocks’ scheme encrypts one bit plaintext into a ciphertext composed of a pair of two large integers, so it is not space efficient and used to encrypt short session keys in practice. Cocks’ scheme is known not anonymous due to the Galbraith’s test. In 2007, Boneh, Gentry and Hamburg [6] addressed the ciphertext expansion issue, they presented an anonymous IBE system which merely expands an $\ell$-bit plaintext to a ciphertext about a size of $\ell + \log_2 N$. However, the encryption in their scheme is not efficient. In 2013, Clear, Hughes, and Tewari [13] considered Cocks’ scheme over the polynomial quotient ring $\mathbb{Z}_N[x]/(x^2 - R_{id})$ because it is natural and convenient to view ciphertexts as elements in it. With the help of this sharp observation, they constructed a strongly XOR-homomorphic IBE scheme. In the same year, Boneh, LaVigne and Sabin [7] generalized Cocks’ scheme to $e^{th}$ residuosity so that it can encrypt more than one bit in a message. The downside of this generalization is that the ciphertext expansion is massive, which is intractable to optimize yet because any intuitive attempt of compression fails to be secure due to the attack found by Boneh, LaVigne and Sabin [13]. Recently, Clear and McGoldrick [14] extended BLS’s scheme so that it can use a hash function which can be securely instantiated.

1.2 Our Contributions

In this work, we investigate the IBE scheme of Boneh, LaVigne and Sabin (BLS’s scheme) [7] and make the following contributions.

Our first contribution is to improve BLS’s scheme in the following two aspects:

1. We omit the spare calculation of $e$-th power residue symbols, a very time-consuming part in the encryption phase of BLS’s scheme. Also, this modification does not influence the security.

2. In BLS’s scheme, $e$ must be a prime number. We leverage knowledge of classical number theory and extend BLS’s scheme to the case $e$ is a square-free number, which strengthens the flexibility and is more efficient in the case that it is converted into a public-key scheme.

Moreover, we give a proof that our proposed IBE scheme is semantically secure.

Our first contribution is to make the discussion about the incompressibility of BLS’s scheme in the original paper rigorous by introducing the general reciprocity law on function fields. Applying this technique, we successfully generalize the famous Galbraith’s test and show that BLS’s scheme is not anonymous when $e$ is small.

Our third contribution is to provide methods for computing $e$-th power residue symbols for BLS’s scheme and ours. Furthermore, we correct a theorem proposed in [17] and give an analogous conclusion which has the same effect.
2 Preliminaries

2.1 Notation

If $X$ is a finite set, the notation $\#X$ means the cardinality of $X$, writing $x \leftarrow X$ to indicate that $x$ is an element sampled from the uniform distribution over $X$. If $A$ is an algorithm, then we write $x \leftarrow A(y)$ to mean: "run $A$ on input $y$ and the output is assigned to $x$". PPT is short for "probabilistic polynomial time".

For a group $G$, the subgroup of $G$ generated by the set $X$ is denoted by $\langle X \rangle$. If $R$ is a ring, $a,b \in R$ and $I$ is an ideal of $R$, the relation $a - b \in I$ is written $a \equiv b \pmod{I}$. A finite field of size $q$ is denoted by $\mathbb{F}_q$. For a polynomial $f$, we denote as $\deg(f) = n$ to say $f$ has degree $n$. $\lg$ stands for the binary logarithm. $(\cdot)$ stands for Jacobi symbol. Let $\varphi$ be the Euler’s totient function.

2.2 Identity-Based Encryption

An identity-based encryption is defined as a tuple of four PPT algorithms $(\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$:

$\text{Setup}(1^\lambda)$ The setup algorithm $\text{Setup}$ is a randomized algorithm that takes a security parameter $1^\lambda$ as input, and outputs a tuple $(\text{mpk}, \text{msk})$, where the $\text{mpk}$ denotes the public parameters and $\text{msk}$ denotes the master secret key.

$\text{KeyGen}(\text{mpk}, \text{msk}, \text{id})$ The key generation algorithm $\text{KeyGen}$ is a deterministic algorithm that takes $\text{msk}$ and an identity $\text{id}$ as inputs, and outputs a decryption key $\text{sk}_{\text{id}}$ associated with the identity $\text{id}$.

$\text{Enc}(\text{mpk}, \text{id}, m)$ The encryption algorithm $\text{Enc}$ is a randomized algorithm that takes $\text{mpk}, \text{id}$ and a plaintext $m$ as inputs, and outputs a ciphertext $c$. That is, we encrypt plaintext $m$ with identity $\text{id}$ and achieve ciphertext $c$.

$\text{Dec}(\text{mpk}, \text{sk}_{\text{id}}, c)$ The key generation algorithm $\text{Dec}$ is a deterministic algorithm that takes $\text{mpk}, \text{sk}_{\text{id}}, c$ as inputs, and outputs the corresponding plaintext $m$ if $c$ is a valid ciphertext, and $\bot$ otherwise.

2.3 Security Notions

2.3.1 Correctness

The correctness property issues the fact that any valid ciphertext can be decrypted to recover the corresponding plaintext. For a formal definition, we denotes $M, I, D, C$ as the plaintext space, the identity space and the ciphertext space respectively. An identity-based encryption $(\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$ is said correct if $\forall m \in M, \forall \text{id} \in I, \text{mpk}, \text{id}, \text{sk}_{\text{id}}$ obtained from $\text{Setup}$ and $\text{KeyGen}$, it satisfies:

$$\Pr[\text{Dec} (\text{mpk}, \text{sk}_{\text{id}}, \text{Enc} (\text{mpk}, \text{id}, m)) = m] = 1.$$
second query phase, guess phase. We describe each phase to capture the attack.

**Initialization phase:** The challenger runs the algorithm Setup, keeps the master secret key msk and gives the public parameters mpk to the adversary.

The first query phase: The adversary receives mpk. Thus it knows the plaintext space M, the identity space ID and the ciphertext space C. It then chooses a subset ID1 ⊆ ID, issues key generation queries and receives back the the private keys corresponding to each identity in ID1. The queries can be asked adaptively so the adversary can update and rich its knowledge of the scheme, which denotes by the state s.

**Challenge phase:** The adversary chooses a challenge identity id* ∉ ID1 and two different plaintexts m0, m1 of the same length. It sends them to the challenger.

The second query phase: This phase is the same to the first query phase excepts that the query identity subset ID2 cannot contain id*.

**Guess phase:** The challenger chooses a random bit b and encrypts mb received from the adversary with msk, id*. It then sends the corresponding ciphertext c to the adversary. The adversary tries to guess the bit b. It wins the game (carries a successful attack) when guessing right.

Formally, an identity-based encryption is said semantically secure if

\[ \Pr \left[ \left( \text{mpk,msk} \right) \leftarrow \text{Setup}(1^\lambda) \right| \left( \text{id}^*,m_0,m_1,s \right) \leftarrow \text{KeyGen}(\text{mpk,msk},s) \right| \left( b \leftarrow \{0,1\} \right| c \leftarrow \text{Enc}(\text{mpk,msk,}^*,m_b) \right) = b \right] - \frac{1}{2} \]

is negligible, where \( A_1 \) denotes the behaviors of the adversary in two query phases and challenge phase, \( A_2 \) denotes the behaviors of the adversary in guess phase.

Because the adversary can choose the challenge identity and plaintexts as it likes and tries to distinguish the challenge ciphertext, the semantic security can be also called indistinguishable chosen-identity chosen-plaintext security (IND-ID-CPA).

### 2.4 \( e \)-th Power Residue Symbol

Let \( K \) be a number field, and \( \mathcal{O}_K \) be the ring of integers in \( K \), and \( e \geq 1 \) be an integer. We say a prime ideal \( p \) in \( \mathcal{O}_K \) is prime to \( e \) if \( p \nmid e\mathcal{O}_K \). It is easy to see that \( p \) is relatively prime to \( e \) if and only if \( \gcd(q,e) = 1 \), where \( q = p^f = \text{Norm}(p) \) for some \( f \in \mathbb{N} \). For every \( \alpha \in \mathcal{O}_K \), \( \alpha \notin p \), we have

\[ \alpha^{q-1} \equiv 1 \ (p) \]

Let \( \zeta_e = \exp^{2\pi i/e} \) be an \( e \)-th root of unity. If \( \zeta_e \in K \) and \( p \) is relatively prime to \( e \), the order of the subgroup of \( \text{#}(\mathcal{O}_K/p)^\times \) generated by \( \zeta_e \mod p \) is \( e \). This indicates that \( e \) divides \( q-1 \), hence we can define the \( e \)-th power residue symbol \( \left( \frac{\alpha}{p} \right)_e \) as follows:

1. \( \left( \frac{\alpha}{p} \right)_e = 0 \) if \( \alpha \in p \).

2. If \( \alpha \in p \), \( \left( \frac{\alpha}{p} \right)_e \) is the unique \( e \)-th root of unity such that \( \alpha^{\text{Norm}(p)-1} \equiv \left( \frac{\alpha}{p} \right)_e \ (p) \).

Next, we extend the symbol multiplicatively to all ideals. Suppose \( a \subset \mathcal{O}_K \) is an ideal prime to \( e \). Let \( a = p_1p_2 \cdots p_m \) be the prime decomposition of \( a \). For \( \alpha \in \mathcal{O}_K \) define \( \left( \frac{\alpha}{a} \right)_e = \prod_{i=1}^{m} \left( \frac{\alpha}{p_i} \right)_e \). If \( \beta \in \mathcal{O}_K \) and \( \beta \) is prime to \( e \), we define \( \left( \frac{\alpha}{\beta} \right)_e = \left( \frac{\alpha}{(\beta)} \right)_e \). See \([19, 22, 23]\) for more properties about
the $e$-th power residue symbol.

In the following, we only consider the case $K = \mathbb{Q}(\zeta_e)$. It’s well-known that $\mathcal{O}_K = \mathbb{Z}[\zeta_e]$. Let $N = pq$ be a product of two primes satisfying $p \equiv 1 \pmod{n}$, $q \equiv 1 \pmod{n}$, then both $p$ and $q$ split completely in $K$. If $\mu \in \mathbb{Z}_N^*$ is a primitive $e$-th root of unity modulo $p$ and modulo $q$, we say it a *non-degenerate* primitive $e$-th root of unity modulo $N$.

**Lemma 2.1 (Freeman et al. [17]).** Let $e$ be a positive integer, $N = pq$ be a product of two primes $p, q$ with $p \equiv q \equiv 1 \pmod{e}$. Let $\mu \in \mathbb{Z}_N^*$ be a non-degenerate primitive $e$-th root of unity modulo $N$. For each $i$ in $\{1, 2, \ldots, e\}$ with $\gcd(i, e) = 1$, let $a_i = N\mathcal{O}_K + (\zeta_e - \mu^i)\mathcal{O}_K$, $p_i = p\mathcal{O}_K + (\zeta_e - \mu^i)\mathcal{O}_K$ and $q_i = q\mathcal{O}_K + (\zeta_e - \mu^i)\mathcal{O}_K$. Then, we have $\text{Norm}(a_i) = N$, $a_i = p_i q_i$ for all $i$, and

$$
p\mathcal{O}_K = \prod_{\gcd(i,e)=1} p_i \quad q\mathcal{O}_K = \prod_{\gcd(i,e)=1} q_i \quad N\mathcal{O}_K = \prod_{\gcd(i,e)=1} a_i
$$

We define a function $\mathcal{J}_{N,e}: \mathbb{Z}_N \mapsto \{0, \ldots, e - 1\}$ as follows:

$$
\mathcal{J}_{N,e}(x) = \begin{cases} 
0, & \text{if } \gcd(x, N) \neq 1, \\
i, & \text{if } \gcd(x, N) = 1 \text{ and } \left(\frac{x}{a_i}\right)_e = \zeta_e. 
\end{cases}
$$

Obviously, if $x, y \in \mathbb{Z}_N^*$, then $\mathcal{J}_{N,e}(xy) = \mathcal{J}_{N,e}(x)\mathcal{J}_{N,e}(y)$. Squirrel [29] gave a polynomial algorithm computing $e$-th power residue symbols, which requires expensive precomputations. Boer [16] proposed an improved algorithm that does not rely on heavy precomputations and runs fast in experiments. However, he could not give a rigorous proof that it runs in polynomial time.

Assuming $e \geq 2$, we say an integer $x \in \mathbb{Z}_N^*$ is an $e$-th residue modulo $N$ if there exists an integer $y \in \mathbb{Z}_N^*$ such that $y^e \equiv x \pmod{N}$. Note that if $x$ is an $e$-th residue, then $\left(\frac{x}{p_i}\right)_e = \left(\frac{x}{q_i}\right)_e = 1$ holds for every $i$ relatively prime to $e$. We denote the set of all $e$-th residues in $\mathbb{Z}_N^*$ by $\mathcal{ER}_{N,e}$. $\mathcal{PR}_{N,e}$ is defined by

$$
\mathcal{PR}_{N,e} = \begin{cases} 
\left\{ x \in \mathbb{Z}_N^* \mid \left(\frac{x}{a_i}\right)_e = 1 \right\}, & i = 0, \\
\left\{ x \in \mathbb{Z}_N^* \mid \left(\frac{x}{a_i}\right)_e = 1, \left(\frac{x}{p_i}\right)_e \text{ and } \left(\frac{x}{q_i}\right)_e \text{ are primitive} \right\} \cup \mathcal{ER}_{N,e}, & i = 1.
\end{cases}
$$

We alter the MER assumption defined in [7] as follows.

**Definition 2.2 (Modified $e$-th Residue ($\text{MER}_i$, $i \in \{0, 1\}$) Assumption).** For a PPT algorithm $\text{RSA}_\lambda$ that generates two equally sized primes $p, q$ and a square-free integer $e$ such that $p \equiv q \equiv 1 \pmod{e}$, $\gcd\left(\frac{p+q-2}{e}, e\right) = 1$, and picks $\mu \in \mathbb{Z}_N^*$ a non-degenerate primitive $e$-th root of unity to $N = pq$. We define the following two distributions relative to $\text{RSA}_\lambda$ as:

$$
\mathcal{D}^{\lambda}_{\text{ER}} : \left\{ (N, v, e, \mu) : (p, q, e, \mu) \leftarrow \text{RSA}_\lambda, v \leftarrow \mathcal{PR}_{N,e} \right\}
$$

$$
\mathcal{D}^{\lambda}_{\text{ER}} : \left\{ (N, v, e, \mu) : (p, q, e, \mu) \leftarrow \text{RSA}_\lambda, v \leftarrow \mathcal{PR}_{N,e} \setminus \mathcal{ER}_{N,e} \right\}
$$

The $\text{MER}_i$ assumption relative to $\text{RSA}_\lambda$ asserts that the advantage $\text{Adv}_{\lambda, \text{RSA}_\lambda}^{\text{MER}_i}(\lambda)$ defined as

$$
\left| \Pr[A(N, v, e, \mu) = 1 \mid (N, v, e, \mu) \leftarrow \mathcal{D}^{\lambda}_{\text{ER}}(\lambda)] - \Pr[A(N, v, e, \mu) = 1 \mid (N, v, e, \mu) \leftarrow \mathcal{D}^{\lambda}_{\text{ER}}(\lambda)] \right|
$$

is negligible for any PPT adversary $A$. 

5
Both of two assumptions are natural extensions of the standard quadratic residue assumption ($\zeta_2 = -1$ and $\mu = -1$ when $e = 2$). Therefore, we believe that it is intractable to break both of them. Obviously, $MER_1$ implies $MER_0$. The following lemma further elaborates the relation between the two assumptions.

**Lemma 2.3.** If there exists a PPT distinguisher $\mathcal{A}$ has $\epsilon$ advantage against $MER_1$ assumption where $\epsilon > \frac{e}{\varphi(e)+1} - 1$, then there exists a PPT distinguisher $\mathcal{B}$ has at least $\frac{\varphi(e)+1}{e}(1+\epsilon) - 1$ advantage against $MER_0$ assumption with comparable running time.

**Proof.** Let $U = \{1, \zeta_e, \ldots, \zeta_e^{e-1}\}$ denotes the subgroup of roots of unity in $O_K$. The map $\theta : \mathbb{Z}_p^* \to U$ given by $x \mapsto \left(\frac{x}{p}\right)_e$ is an homomorphism. Let $\mathbb{E}R_{p,e} = \{ y \in \mathbb{Z}_p^* \mid y \equiv x^e \pmod{p} \}$ be the subgroup composed of $e$-th residues in $\mathbb{Z}_p^*$. It's easy to verify that the cardinality of $\mathbb{E}R_{p,e}$ is $\frac{p-1}{e}$. Therefore, an integer $z \in \mathbb{Z}_p^*$ satisfying $\left(\frac{z}{p}\right)_e = 1$ must be in $\mathbb{E}R_{p,e}$. Hence the kernel of $\theta$ is $\mathbb{E}R_{p,e}$ and we have the following isomorphism

$$\mathbb{Z}_p^*/\mathbb{E}R_{p,e} \cong U$$

due to the equality of cardinality. Of course, elements in different cosets of $\mathbb{E}R_{p,e}$ in $\mathbb{Z}_p^*$ have different $e$-th power residue symbols, whence there is a one to one correspondence between cosets of $\mathbb{E}R_{p,e}$ in $\mathbb{Z}_p^*$ and $e$-th roots of unity via the $e$-th power residue symbol. The above arguments are also valid in $\mathbb{Z}_q^*$. Hence, we have

$$\frac{\#\mathbb{PR}_{N,e}}{\#\mathbb{E}R_{N,e}} = \frac{(\varphi(\epsilon)+1)\varphi(N)}{e\varphi(\epsilon)^2} = \frac{\varphi(e)+1}{e},$$

When $\mathcal{B}$ is given the tuple $\{N, v, e, \mu\}$ as input, it invokes $\mathcal{A}$ and acts as follow: $\mathcal{B}(N, v, e, \mu) = 1$ if and only if $\mathcal{A}(N, v, e, \mu) = 1$. We have

$$\text{Adv}^{MER_0}_{\mathcal{B}, \text{RSAgen}} = \frac{\text{Pr}^{\mathcal{A}(N,v,e,\mu)=1;\mathcal{A}(N,v,e,\mu)=1}}{\#\mathbb{E}R_{N,e}} \times \frac{\#\mathbb{PR}_{N,e}}{\#\mathbb{E}R_{N,e}} + \frac{\text{Pr}^{\mathcal{A}(N,v,e,\mu)=1;\mathcal{A}(N,v,e,\mu)=1}}{\#\mathbb{E}R_{N,e}} \times \left(1 - \frac{\#\mathbb{PR}_{N,e}}{\#\mathbb{E}R_{N,e}}\right)$$

$$- \text{Pr}^{\mathcal{A}(N,v,e,\mu)=1;\mathcal{E}R_{N,e}=\mathcal{E}R_{N,e}} \times \frac{\#\mathbb{PR}_{N,e}}{\#\mathbb{E}R_{N,e}} - \frac{\#\mathbb{PR}_{N,e}}{\#\mathbb{E}R_{N,e}} \times \left(1 - \frac{\#\mathbb{PR}_{N,e}}{\#\mathbb{E}R_{N,e}}\right)$$

$$\geq \frac{\#\mathbb{PR}_{N,e}}{\#\mathbb{E}R_{N,e}} \times (1 + \epsilon) - 1 = \frac{\varphi(e)+1}{e}(1+\epsilon) - 1$$

When the inequality $\epsilon > \frac{e}{\varphi(e)+1} - 1$ holds, $\mathcal{B}$ has at least $\frac{\varphi(e)+1}{e}(1+\epsilon) - 1$ advantage against $MER_0$ assumption, as desired.

We close this section by showing that the precondition of Proposition 4.3 proposed in [17] can be relaxed as follows.

**Proposition 2.4.** Let $e$ be an integer. Let $N = pq$ where $p \equiv q \equiv 1 \pmod{e}$. Suppose that $\gcd(\frac{p-1}{e}, e) = \gcd(\frac{q-1}{e}, e)$. Then there is a $\nu$ such that

1. $\nu$ is a non-degenerate primitive $e$-th root of unity modulo $N$.
2. $\left(\frac{\nu}{\mathfrak{a}}\right)_e = 1$ for every ideal $\mathfrak{a} \subset O_K$ as in Lemma 2.1.
The returned ciphertext is primitive and calculates multiple bits at a time.

We now describe the IBE scheme presented by Boneh, LaVigne and Sabin \[7\]. The scheme encrypts e

- **Setup** \((1^\lambda)\)  Given a security parameter \(\lambda\), Setup selects a prime \(e\), then generates an RSA modulus \(N = pq\) a product of two large primes \(p\) and \(q\) such that \(e \mid p - 1, e \mid q - 1\). The public parameters are \(mpk = \{N, e, \mu, H\}\) where \(\mu\) is a non-degenerate primitive \(e\)-th root of unity in \(\mathbb{Z}_N\), \(H\) is a publicly available hash maps an arbitrary binary string to an \(e\)-th residue in \(\mathbb{Z}_N^*\). The master secret key is \(msk = \{p, q\}\).

- **KeyGen\((mpk, msk, id)\)**  Using the hash function \(H\) and \(p, q\), KeyGen sets \(R_{id} = H(id)\), then calculates \(r_{id} = H(id)\frac{t}{e} \mod N\). Finally, KeyGen returns \(usk = \{r_{id}\}\) as user’s private key.

- **Enc\((mpk, id, m)\)**  To encrypt a message \(m \in \{0, \ldots, e - 1\}\) for a user with identity \(id\), Enc derives the hash value \(R_{id} = H(id)\). It then chooses a random polynomial \(f\) of degree \(e - 1\) from \(\mathbb{Z}_N[x]\) and calculates \(g(x) = f(x)^e \mod (x^e - R_{id}) = \sum_{i=0}^{e-1} a_i x^i\). Next, it chooses a transport key \(t \leftarrow \mathbb{Z}_N^*\). The returned ciphertext is

\[
C = \left\{ \frac{a_0}{t}, \frac{a_1}{t}, \ldots, \frac{a_{e-1}}{t}, (m + J_{N,e}(t)) \mod e \right\}.
\]

- **Dec\((mpk, usk, C)\)**  When a user with \(usk = \{r_{id}\}\) receives a ciphertext set \(C\), it parses \(C\) as \(\{c_0, c_1, \ldots, c_{e-1}, c\}\), Dec recovers the plaintext \(m\) as

\[
m = \left( J_{N,e} \left( \sum_{i=0}^{e-1} c_i r_{id}^i \right) + c \right) \pmod{e}
\]

Remark 3.1. BLS’s scheme extends Cocks’ scheme to higher residue case. To see this, pick \(e = 2\) and \(f(x) = t + x\), there is \(\frac{g(x)}{t} = t + \frac{R_{id}}{t} + 2x\), which behaves like that in Cocks’ scheme.
Remark 3.2. In Cocks’ scheme, the PKG can easily derive a user’s secret key by several efficient probabilistic algorithms taking square roots in finite fields such as Cipolla-Lehmer [21], Tonelli-Shanks [27] and Adleman-Manders-Miller [1]. These methods can also be applied to general situations, e.g. [11], the extended Adleman-Manders-Miller’s algorithm can extract an \( e \)-th root modulo a prime \( p \) in \( \mathcal{O}(\log^4 p + e \log^3 p) \) time complexity.

Since it’s unclear how to implement such a hash function \( H \), BLS’s scheme was not given a formally security proof in the original paper. In [14], the authors proposed an approach which is akin to Cocks’ scheme to circumventing the issue, but at the cost of a lower efficiency and a larger ciphertext extension.

3.2 Our IBE Scheme

We find that it is redundant to compute \( e \)-th power residue symbols in the encryption phase of BLS’s scheme. We presume that there exists a hash function mapping an arbitrary binary string to an \( e \)-th residue in \( \mathbb{Z}_N^* \), e.g., The PKG selects a large number of \( e \)-th residues beforehand, and constructs it by the Lagrange’s interpolation formula. Our improved IBE scheme for a square-free integer \( e \) is defined as follows:

- **Setup** (1) Given a security parameter \( \lambda \), Setup generates an RSA modulus \( N = pq \) a product of two large primes \( p \) and \( q \), and selects a square-free integer \( e \) with the prime decomposition \( e = \prod_{i=1}^{\ell} e_i \) such that \( e \parallel p - 1 \), \( e \parallel q - 1 \) and \( \gcd(p+q-2,e) = 1 \). The settings of \( \mu \) and \( H \) are the same as for in BLS’s scheme. The public parameters are \( mpk = \{ N, e, \mu, J_{N,e}(\mu), H \} \). The master secret key is \( msk = \{ p, q \} \).

- **KeyGen** (mpk, msk, id) Using the hash function \( H \) and \( p, q \), KeyGen sets \( R_{id} = H(id)^\frac{1}{e} \mod N \). Finally, KeyGen returns \( usk = \{ r_{id} \} \) as user’s private key.

- **Enc** (mpk, id, m) To encrypt a message \( m \in \{ 0, \ldots, e - 1 \} \) for a user with identity \( id \), Enc first derives the hash value \( R_{id} = H(id) \). Then, it generates a transport key \( t = \mu^k \) where \( k \leftarrow \{0, \ldots, e - 1\} \). We define the following sub-algorithm \( E \) which takes a prime number \( \mathcal{P} \) and a public key \( R_{id} \) as inputs:

  **Algorithm 1: \( E \)**

  **Input:** a prime number \( \mathcal{P} \), a public key \( R_{id} \)
  **Output:** a polynomial

  1. Generate a uniform random polynomial \( f(x) \leftarrow \mathbb{Z}_N^*[x] \) of degree \( \mathcal{P} - 1 \)
  2. Compute \( g(x) = f(x)^\mathcal{P} \mod x^\mathcal{P} - R_{id} \)
  3. Output the polynomial \( c(x) = g(x)_{\mu^k \mod \mathcal{P}} \)

  The returned ciphertext is

  \[
  C = \{ E(e_1), \ldots, E(e_{\ell}), (m + J_{N,e}(t)) \mod e \}.
  \]

- **Dec** (mpk, usk, C) When a user with \( usk = \{ r_{id} \} \) receives a ciphertext set \( C \), it parses \( C \) as \( \{ c_1(x), \ldots, c_{\ell}(x), c \} \), Dec recovers the plaintext \( m \) as

  \[
  m = \left( J_{N,e} \left( \prod_{i=1}^{\ell} c_i \left( r_{id}^e \right)^{\frac{e}{e_i}} \mod N \right) + c \right) \mod e
  \]
Remark 3.3. The condition \( \gcd(p^2 - 2, e) = 1 \) ensures that \( J_{N,e}(\mu) \) is primitive by the proof of Proposition 2.4. In the encryption phase, computing \( J_{N,e}(t) = (k J_{N,e}(\mu) \mod e) \) can be very convenient.

Remark 3.4. It is worth mentioning that in the case \( e = 2 \), the encryption is much more efficient than Cocks’ scheme’s, as a result of eliminating the operations about computing the inverses modulo \( \mathbb{Z}_N^* \) and computing the Jacobi symbols.

Remark 3.5. Unfortunately, we could not omit to compute \( e \)-th power residue symbols in the decryption phase. One method is to utilize existing algorithms to compute power residue symbols with respect to each prime factor of \( e \) and to apply the Chinese remainder theorem (see Appendix B). The other is to keep the PKG online for answering queries about evaluating \( \left( \frac{R}{p} \right)_e \) from each user. In this case, a secure channel between the PKG and each user should be established.

Correctness  Correctness can be verified directly as follows.

\[
\text{Dec}(\text{mpk}, \text{sk}_i, (\text{Enc}(id, m))) = \sum_{i=1}^\ell \mathcal{J}_{N,e} \left( \left( \frac{c_i(r_i^{i+1})}{\mu^k \mod e_i} \right) \right) + m + \mathcal{J}_{N,e}(\mu^k)
\]

\[
\equiv \sum_{i=1}^\ell \mathcal{J}_{N,e} \left( \frac{1}{\mu^k \mod e_i} \right) + m + \mathcal{J}_{N,e}(\mu^k)
\]

\[
\equiv \mathcal{J}_{N,e} \left( \frac{\mu^k}{\mu^k \mod e_i} \right) + m \equiv m \pmod{e}
\]

Security  To simplify the security proof, we first prove the following theorem.

Theorem 3.6. Let \( e \) be a prime number, \( t \in \mathbb{Z}_N^* \) a transport key, \( R \) an element of \( \mathbb{Z}_N^* \) such that \( \left( \frac{R}{p_1} \right)_e = \zeta_2^{i+1}, \left( \frac{R_1}{q_1} \right)_e = \zeta_2^{j+1} \) where \( i, j \) are relatively prime to \( e \). If \( f(x) = \frac{f(x)^e}{t} \mod (x^e - R) \) where \( f(x) \leftarrow \mathbb{Z}_N^*[x] \) is a polynomial of degree \( e - 1 \), then

\[
\Omega_\ell = \left\{ g(x) \in \mathbb{Z}_N^*[x] \mid \deg g(x) = e - 1, \frac{g(x)^e}{t} \mod (x^e - R) = c(x) \right\}
\]

has the same cardinality for each transport key \( \ell \in \mathbb{Z}_N^* \).

Proof. Suppose \( \left( \frac{\bar{t}^{i+1}}{p_1} \right)_e = \zeta_2^{i+1}, \left( \frac{\bar{t}^{-1}}{q_1} \right)_e = \zeta_2^{j+1} \). Since

\[
\left( \frac{R^{i+1}}{\bar{p}_1} \right)_e = \left( \frac{\bar{t}^{-1}}{p_1} \right)_e, \quad \left( \frac{R^{i+1}}{\bar{q}_1} \right)_e = \left( \frac{\bar{t}^{-1}}{q_1} \right)_e,
\]

by Lemma 2.3 there exist \( W_p \in \mathbb{Z}_p^* \) and \( W_q \in \mathbb{Z}_q^* \) such that

\[
W_p R^{i+1} \equiv t^{-1} \pmod{p}, \quad W_q R^{i+1} \equiv t^{-1} \pmod{q}.
\]

Since \( \mathbb{Z}_N[x]/(x^e - R) \cong \mathbb{Z}_p[x]/(x^e - R) \oplus \mathbb{Z}_q[x]/(x^e - R) \), the map \( \phi : \Omega_\ell \to \Omega_\ell \) given by \( h(x) \mapsto g(x) \) where

\[
g(x) \equiv W_p x^{i+1} h(x) \pmod{p},
\]

\[
g(x) \equiv W_q x^{i+1} h(x) \pmod{q}
\]
is well defined for a fixed $t$. Likewise, the inverse map $\psi: \Omega_t \rightarrow \Omega_t$ is given by $g(x) \mapsto h(x)$ where

\[
h(x) \equiv W_p^{-1} \left( R^{-1} x^{e-1} \right)^{i_k} g(x) \pmod{p}
\]

\[
h(x) \equiv W_q^{-1} \left( R^{-1} x^{e-1} \right)^{i_{j}} g(x) \pmod{q}
\]

It is straightforward to verify $\psi \phi = 1_{\Omega_t}$ and $\phi \psi = 1_{\Omega_t}$ where $1_{\Omega_t}$ and $1_{\Omega_t}$ denote the identity maps on $\Omega_t$ and on $\Omega_t$ respectively. The proof is completed.

We are now in a position to investigate the security of our IBE scheme.

**Theorem 3.7.** Let $A = (A_1, A_2)$ be an adversary against IND-ID-CPA security of our IBE scheme, making at most $q_H$ queries to the random oracle $H$ and a single query to the Challenge phase. Then, there exists an adversary $B$ against the $\text{MER}_1$ assumption such that

\[
\text{Adv}_{\text{ind-cpa}}^A(\lambda) = q_H \cdot \text{Adv}_{B, \text{RSAgen}}^{\text{MER}_1}(\lambda)
\]

**Proof.** We prove it by defining a sequence of three games. For simplicity, we omit the procedure of $\text{Enc}$ in the Challenge phase.

**Game$_1^A(\lambda)$:** This game is the real attack against our IBE scheme.

**Game$_2^A(\lambda)$:** In this game, we guess the number of the challenge identity and abort the game if the guess is wrong.

**Game$_3^A(\lambda)$:** We change the simulation of the $H$ phase so that it returns a random element in $\mathbb{PR}_{1, e}$ for the $i^*$-th query.

We claim:

**Claim 1:** $\text{Adv}_{\text{ind-cpa}}^A(\lambda) = \left| \Pr[\text{Game}_1^A(\lambda) = \text{true}] - \frac{1}{2} \right|.$

**Claim 2:** $\Pr[\text{Game}_2^A(\lambda) = \text{true}] = \frac{1}{2}(1 - \frac{1}{q_1}) + \frac{1}{q_1} \Pr[\text{Game}_1^A(\lambda) = \text{true}].$

**Claim 3:** $\Pr[\text{Game}_3^A(\lambda) = \text{true}] - \Pr[\text{Game}_2^A(\lambda) = \text{true}] \leq \text{Adv}_{B, \text{RSAgen}}^{\text{MER}_1}(\lambda).$

**Claim 4:** $\Pr[\text{Game}_3^A(\lambda) = \text{true}] = \frac{1}{2}.$

**Proof.** **Claim 1** follows immediately by the definition of semantic security. **Claim 2** is derived from Bayes’ theorem. **Claim 3** follows from Difference Lemma [28]. If the public key $Y$ of the challenge identity $id^*$ is chosen from $\mathbb{PR}_{1, e}$, then $\left( \frac{Y}{p_1} \right)_{e_j}$ and $\left( \frac{Y}{q_1} \right)_{e_j}$ are both primitive $e_j$-th root of unity for every $1 \leq j \leq \ell$. Since $\left( \frac{\mu}{a_1} \right)_{e}$ is primitive $e$-th root of unity, the set

\[
\left\{ \left( \frac{\mu^k \pmod{e}}{a_1} \right)_{e_1}, \ldots, \left( \frac{\mu^k \pmod{e}}{a_1} \right)_{e_\ell} \right\} \quad 0 \leq k < e
\]

takes over combinations of all $e_j$-th roots of unity for each $1 \leq j \leq \ell$. Hence, by Theorem 3.6, ciphertexts are statistically indistinguishable to an adversary, which completes the proof of **Claim 4**.

Combining all claims above gives this theorem.
Game\textsuperscript{A} (\(\lambda\))

\textbf{phase Setup}(\(\lambda\))

\begin{align*}
    & b \leftarrow \{0, 1\} \\
    & \mathcal{S}_H \leftarrow \emptyset; \; ctr \leftarrow 0 \\
    & \text{msk} \leftarrow \{p, q\} \\
    & \text{mpk} \leftarrow \{N, e, \mu, J_{N,e}(\mu), \ell, e_1, \ldots, e_\ell\} \\
    & \text{return mpk}
\end{align*}

\textbf{phase KeyGen}(id)

\begin{align*}
    & \text{if } (ctr, id, Y, \cdot) \notin \mathcal{S}_H \; \mathcal{H}(id) \\
    & \text{read } (ctr, id, Y, \cdot) \in \mathcal{S}_H \\
    & \text{usk} \leftarrow Y \frac{1}{2} \text{ mod } N \\
    & \text{return usk}
\end{align*}

\textbf{phase H}(id)

\begin{align*}
    & \text{if } (ctr, id, Y, \cdot) \in \mathcal{S}_H \text{ return } Y \\
    & \text{ctr} \leftarrow ctr + 1 \\
    & Y \leftarrow \mathcal{E}_N \\
    & \mathcal{S}_H \leftarrow \mathcal{S}_H \cup \{(ctr, id, Y, \bot)\} \\
    & \text{return } Y
\end{align*}

\textbf{phase Challenge}(id\textsuperscript{*}, m_0, m_1)

\begin{align*}
    & C \leftarrow \text{Enc}(\text{mpk}, id\textsuperscript{*}, m_b) \\
    & \text{return C}
\end{align*}

\textbf{phase Guess}(b')

\begin{align*}
    & \text{return } b' = b
\end{align*}

Game\textsuperscript{B} (\(\lambda\))

\textbf{phase Setup}(\(\lambda\))

\begin{align*}
    & b \leftarrow \{0, 1\} \\
    & i^* \leftarrow \{1, \ldots, q_H\} \\
    & \mathcal{S}_H \leftarrow \emptyset; \; ctr \leftarrow 0 \\
    & \text{msk} \leftarrow \{p, q\} \\
    & \text{mpk} \leftarrow \{N, e, \mu, J_{N,e}(\mu), \ell, e_1, \ldots, e_\ell\} \\
    & \text{return mpk}
\end{align*}

\textbf{phase KeyGen}(id)

\begin{align*}
    & \text{if } (ctr, id, Y, \cdot) \notin \mathcal{S}_H \; \mathcal{H}(id) \\
    & \text{read } (ctr, id, Y, \cdot) \in \mathcal{S}_H \\
    & \text{if } y = \bot \; \text{abort} \\
    & \text{usk} \leftarrow y \\
    & \text{return usk}
\end{align*}

\textbf{phase H}(id)

\begin{align*}
    & \text{if } (ctr, id, Y, y) \in \mathcal{S}_H \text{ return } Y \\
    & \text{ctr} \leftarrow ctr + 1 \\
    & \text{if } ctr = i^* \\
    & \text{y} \leftarrow \mathcal{Z}_N; \; Y = y^* \text{ mod } N \\
    & \mathcal{S}_H \leftarrow \mathcal{S}_H \cup \{(ctr, id, Y, \bot)\} \\
    & \text{else} \\
    & \text{y} \leftarrow \mathcal{Z}_N; \; Y = y^* \text{ mod } N \\
    & \mathcal{S}_H \leftarrow \mathcal{S}_H \cup \{(ctr, id, Y, y)\} \\
    & \text{return } Y \\
\end{align*}

\textbf{phase Challenge}(id\textsuperscript{*}, m_0, m_1)

\begin{align*}
    & \text{if } (i^*, id, Y, y) \in \mathcal{S}_H \text{ and } id = id^* \\
    & C \leftarrow \text{Enc}(\text{mpk}, id^*, m_b) \\
    & \text{else abort} \\
    & \text{return C}
\end{align*}

\textbf{phase Guess}(b')

\begin{align*}
    & \text{return } b' = b
\end{align*}

\begin{equation*}
    GT(a, c) = \left(\frac{c^2 - 4a}{N}\right)
\end{equation*}

\section{3.3 Anonymity}

In this section, we will generalize the famous Galbraith’s test \((4)\) to \(e\)-th power residue situation in order to prove that neither BLS’s scheme nor our IBE scheme is anonymous when \(e\) is small. Let \(a = H(id), N, c\) be the public key of user \(id\), the modulus and the ciphertext as in Cocks’ scheme respectively. Galbraith constructed the following elegant test:

\begin{equation*}
    GT(a, c) = \left(\frac{c^2 - 4a}{N}\right)
\end{equation*}
to distinguish the identity of a ciphertext. The reason it can be successful is: if the ciphertext $c$ is generated by the user $id$ with public key $a$, then $c^2 - 4a$ must be a square, but not necessarily if the public key $a$ is replaced by another one. In [2], Ateniese and Gasti proved that Galbraith’s test is the best test against the anonymity of Cocks’ scheme. Following this method, we find if $g(x) = f(x)^e \mod (x^e - R_{id})$ is a ciphertext polynomial encrypted by the user $id$ in BLS’s scheme, it is uncertain whether $g(x)$ can be encrypted by another user $id'$ if the modulus $x^e - R_{id}$ is replaced by $x^e - R_{id'}$. With the notation and the technique as in Appendix A, an adversary can obtain $c_N$ and $\gamma$ by continuously applying Theorem [A.4]. Therefore, we define the $e$-th Galbraith’s test as

$$\text{GT}(R_{id}, C)_e = \left( \frac{c_N \gamma}{a_1} \right)_e = \left( \frac{\left( \frac{t^{-1}g(x)}{x^e - R_{id}} \right)^{\frac{e}{p}}}{p_1} \right)_e \left( \frac{\left( \frac{t^{-1}g(x)}{x^e - R_{id}} \right)^{\frac{e}{q}}}{q_1} \right)_e. $$

Now that the ciphertext $C$ is generated by the user $id$, the equation $\text{GT}(R_{id}, C)_e = 1$ must hold with all but negligible probability. While for another user $id'$, we believe the value $\text{GT}(R_{id'}, C)_e$ is statistically close to the uniform distribution on $\{\zeta_i^e \mid i \in \{0, 1, \ldots, e - 1\}\}$. We also naturally conjecture that the $e$-th Galbraith’s test is the most effective test against the anonymity of BLS’s scheme.

Remark 3.8. When $e = 2$, let $c_0, c_1 \in \mathbb{Z}_N^*$ and $c(x) = c_1 x + c_0$ be the ciphertext polynomial, then

$$x^2 - R_{id} \equiv (c_1^{-1} c_0)^2 - R_{id} \pmod{c_1 x + c_0}. $$

By Theorem [A.4], we have $c_N = c_1^2$ and $\gamma = (c_1^{-1} c_0)^2 - R_{id}$. Hence, the $2$-th Galbraith’s test is

$$\text{GT}(R_{id}, C)_2 = \left( \frac{c_N \gamma}{a_1} \right)_2 = \left( \frac{c_0^2 - c_1^2 R_{id}}{N} \right), $$

as mentioned in [13].

Example 3.9. Assume that all parameters of BLS’s scheme are set as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>4331</td>
<td>$r_{id}$</td>
<td>67</td>
</tr>
<tr>
<td>$p$</td>
<td>61</td>
<td>$R_{id'}$</td>
<td>467</td>
</tr>
<tr>
<td>$q$</td>
<td>71</td>
<td>$r_{id'}$</td>
<td>51</td>
</tr>
<tr>
<td>$e$</td>
<td>5</td>
<td>$t$</td>
<td>7</td>
</tr>
<tr>
<td>$\mu$</td>
<td>1900</td>
<td>$f(x)$</td>
<td>$x^4 + 2x^3 + 3x^2 + 4x + 6$</td>
</tr>
<tr>
<td>$R_{id}$</td>
<td>822</td>
<td>$g(x)$</td>
<td>$3184x^4 + 3485x^3 + 1183x^2 + 3757x + 1193$</td>
</tr>
</tbody>
</table>

Here, the ciphertext polynomial $\frac{g(x)}{t}$ is generated by the user $id$. To distinguish the identity of $\frac{g(x)}{t}$ between $id$ and $id'$, an adversary performs the following calculations first.
Next, it derives
\[ c_N = (3184 \times 3855 \times 29 \times 3938)^2 \pmod{4331} \quad c'_N = (3184 \times 3855 \times 29 \times 3512)^2 \pmod{4331} \]
\[ \gamma = 55 \quad \gamma' = 4315 \]
and computes \((\frac{c_N}{\alpha_1})_5^\gamma = 1\) and \((\frac{c'_N}{\gamma'})_5 = \zeta_5^3 \neq 1\). Finally, it can determine that the identity of \(\frac{y(x)}{x}\) is id. Actually, there is \((\frac{c_N}{\alpha_1})_5^\gamma = (\frac{c'_N}{\gamma'})_5^\gamma = 1\).

4 Computing \(\left(\frac{\cdot}{\alpha_1}\right)_e\) and \(\left(\frac{\cdot}{\gamma_1}\right)_e\)

In this section, we consider the public-key scheme converted from our IBE scheme. Computing \(e\)-th residue symbols seems to be easier in the decryption phase, for the factorization \(a_1 = p_1 q_1\) is known. The following simple theorem demonstrates that computing \(\left(\frac{y}{p_1}\right)_e\) for an integer \(y\) is somewhat related to solving the discrete logarithm problem in a certain cyclic group. Recall that the discrete logarithm problem (DLP) is defined as: given a finite cyclic group \(G\) of order \(n\) with a generator \(\alpha\) and an element \(\beta \in G\), find the integer \(x \in \mathbb{Z}_n\) such that \(\alpha^x = \beta\).

**Theorem 4.1.** \(\left(\frac{y}{p_1}\right)_e = \zeta_e^x\) if and only if \(\mu^x = y^{\frac{e-1}{e}}\) in \(\mathbb{F}_p^\ast\). Therefore, the solution of DLP in the finite cyclic subgroup \((\mu)\) of order \(e\) means the computation of \(\left(\frac{y}{p_1}\right)_e\).

**Proof.** \(\Leftarrow\) If \(\mu^x = y^{\frac{e-1}{e}}\), then \(y^{\frac{e-1}{e}} - \zeta_e = \mu^x - \zeta_e^x \in p_1\). Thus \(\left(\frac{y}{p_1}\right)_e = \zeta_e^x\).

\(\Rightarrow\) If \(\left(\frac{y}{p_1}\right)_e = \zeta_e^x\) for some \(x \in \mathbb{Z}_e\), that is \(y^{\frac{e-1}{e}} - \zeta_e \in p_1\). Since the order of \(y^{\frac{e-1}{e}}\) divides \(e\), \(y^{\frac{e-1}{e}}\) can be expressed as \(\mu^x\) with an integer \(z \in \mathbb{Z}_e\), which implies \(\mu^x - \mu^z \in p_1\). The fact that the order of \(\mu\) is \(e\) forces \(x = y\).

Although DLP is considered to be intractable in general, it can be quickly solved in a few special cases, e.g., if the order of \(G\) is smooth, Pohlig-Hellman algorithm \([24]\) is much more efficient. Note that computing \(\left(\frac{y}{p_1}\right)_e\) can be very fast in the case \(e\) is small (we can generate a lookup table or use baby-step giant-step algorithm). When \(e\) is medium large and smooth, using Pohlig-Hellman algorithm is appropriate. Specifically, if \(\prod_{i=1}^f e_i\) is the prime factorization of \(e\), the running time of computing \(\left(\frac{y}{p_1}\right)_e\) is \(O\left(\sum_{i=1}^f (\ell e + \sqrt{e_i})\right)\) group multiplications.
Since the encryption only involves one evaluation of $e$-th power residue symbols, computing modular exponentiations of polynomials becomes the most time-consuming part. Therefore, pre-computing modular exponentiations for different random polynomials and saving the results are necessary.

5 Some thoughts and Problems

All known algorithms for computing $e$-th power residue symbols are suitable for small $e$. In the previous section, we have given an algorithm by solving DLP in the finite cyclic subgroup of order $e$ if the factorization $a_1 = p_1q_1$ is already known. Without the knowledge of the factorization $a_1 = p_1q_1$, there is likely no algorithm running in low degree polynomial time in $e$, so is the problem of computing $e$-th power residue symbols hard for large $e$? And, if yes, is the public-key scheme converted from the space-efficient variation of BLS’s scheme (see Appendix A) secure for sufficiently large $e$?

Acknowledgments

We are grateful to Jun Shao for useful discussions. This work is supported in part by the National Natural Science Foundation of China (Grant No. 61632012, 61672239, 61602180), in part by Natural Science Foundation of Shanghai (Grant No. 16ZR1409200) and in part by China Postdoctoral Science Foundation (Grant No. 2017M611502).

References


A Incompressibility of BLS’s Scheme

BLS’s scheme has a space-efficient variation (see [7]) that seems to work. If a trusted PKG sets the user’s secret key to the root of $x^\delta - R_{id}$ for some $\delta$ satisfying $2 \leq \delta < e$, and operations of polynomials are performed in the quotient ring $\mathbb{Z}_N[x]/(x^\delta - v)$ in the encryption phase, then the number of elements in the ciphertext can be reduced to $\delta + 1$. However, this ambitious method makes the scheme insecure. Moreover, there even exists an attack that recovers the decrypted messages from the reciprocity law on $\mathbb{F}_q[t]$, the polynomial ring over a finite field $\mathbb{F}_q$. This attack also shows that it is incompressible for any generalization of similar methods. We start by explaining notation to be used and give crucial definitions and results due to Carlitz [12]. We here refer to Chapter 3 in [25].

Every element in $\mathbb{F}_q[t]$ has the form $f(t) = \alpha_n t^n + \alpha_{n-1} t^{n-1} + \cdots + \alpha_0$. In this case we set $sgn(f) = \alpha_n$ and call it the sign of $f$. Let $P \in \mathbb{F}_q[t]$ of degree $\gamma$ be an irreducible polynomial and $e$ a divisor of $q - 1$. Note that there is a unique $\alpha \in \mathbb{F}_q^*$ such that $a e^{-\gamma} \equiv \alpha \pmod{P}$.

**Definition A.1.** If $a \in \mathbb{F}_q[t]$ and $P$ does not divide $a$, let $(\frac{a}{P})_e$ be the unique element of $\mathbb{F}_q^*$ such that $a e^{-\gamma} \equiv (\frac{a}{P})_e (\mod P)$.

If $P \mid a$ define $(\frac{a}{P})_e = 0$. The symbol $(\frac{a}{P})_e$ is called the $e$-th power residue symbol.

**Proposition A.2.** The $e$-th power residue symbol has the following properties:

1. $(\frac{a}{P})_e = (\frac{b}{P})_e$ if $a \equiv b \pmod{P}$. 
2. \((\frac{ab}{P})_e = (\frac{a}{P})_e (\frac{b}{P})_e\).

3. Let \(\alpha \in \mathbb{F}_q\). Then, \((\frac{\alpha}{P})_e = \alpha^{q-1} \gamma\).

Just as the Jacobi symbol, the definition of the \(e\)-th power residue symbol can be extended to the case that \(P\) is an arbitrary non-zero element \(b \in \mathbb{F}_q[t]\) with the prime decomposition \(b = sgn(b)Q_1^{f_1} \cdots Q_s^{f_s}\), and define

\[
(\frac{a}{b})_e = \prod_{j=1}^{s} (\frac{a}{Q_j^{f_j}})_e.
\]

**Proposition A.3.** The symbol \((\frac{a}{b})_e\) has the following properties:

1. If \(a_1 \equiv a_2 \pmod{b}\), then \((\frac{a_1}{b})_e = (\frac{a_2}{b})_e\).
2. \((\frac{a_1a_2}{b})_e = (\frac{a_1}{b})_e (\frac{a_2}{b})_e\).
3. \((\frac{a}{b_1b_2})_e = (\frac{a}{b_1})_e (\frac{a}{b_2})_e\).
4. \((\frac{a}{b})_e \neq 0\) if and only if \(a\) is relatively prime to \(b\).
5. If \(x^e \equiv a \pmod{b}\) is solvable, then \((\frac{a}{b})_e = 1\).

The following pretty theorem is the general reciprocity law for \(\mathbb{F}_q[t]\).

**Theorem A.4.** [*The general reciprocity law*] Let \(a, b \in \mathbb{F}_q[t]\) be relatively prime, non-zero elements. Then,

\[
(\frac{a}{b})_e = (\frac{b}{a})_e (-1)^{deg(a)deg(b)sgn(a)deg(b)sgn(b)-deg(a)}^{\frac{q-1}{e}}
\]

An adversary who intercepts ciphertexts has the ability of recreating the polynomial \(\frac{g(x)}{t^e} = \frac{f(x)}{t^e} \mod (x^\delta - R_{id})\). Let \(x^\delta - R_{id} = \prod_{j=1}^{m} \eta_j^{p_j}\) be the prime decomposition of \(x^\delta - R_{id}\) in \(\mathbb{F}_p[x]\).

We obtain

\[
\left(\frac{t^e g(x)}{x^\delta - R_{id}}\right)_{e,\mathbb{F}_p} = \left(\frac{t^{-1} f(x)^e}{x^\delta - R_{id}}\right)_{e,\mathbb{F}_p} = \prod_{j=1}^{m} \left(\frac{t^{-1}}{\eta_j} \right)^{p_j}_{e,\mathbb{F}_p} = \prod_{j=1}^{m} t^{\frac{p_j}{e} \deg(\eta_j)}
\]

\[
= t^{-\frac{p_1}{e} \delta} \equiv \left(\frac{t^{-1}}{p_1}\right)^\delta_e (p_1).
\]

Similarly,

\[
\left(\frac{t^{-1} g(x)}{x^\delta - R_{id}}\right)_{e,\mathbb{F}_q} \equiv \left(\frac{t^{-1}}{q_1}\right)^\delta_e (q_1).
\]

Notice that all the following three situations occur with overwhelming probability.

1. \(\gcd(x^\delta - R_{id}, f(x)) = 1\).

2. The term \(x^{e-1} \frac{g(x)}{t}\) has non-zero coefficient.
3. Each term of each polynomial involved has coefficient relatively prime to $N$.

Therefore, we think all of them hold by default. By continuously applying Theorem A.4, we can get

$$\left(\frac{t^{-1}g(x)}{x^\delta - R_{id}}\right)_{e,\mathbb{F}_p} \equiv \left(\frac{c_p}{p_1}\right)_e \left(\frac{\alpha}{\Phi(x)}\right)_{e,\mathbb{F}_p} (p_1), \quad \left(\frac{t^{-1}g(x)}{x^\delta - R_{id}}\right)_{e,\mathbb{F}_q} \equiv \left(\frac{c_q}{q_1}\right)_e \left(\frac{\beta}{\Psi(x)}\right)_{e,\mathbb{F}_q} (q_1) \quad (4)$$

where $\alpha, c_p \in \mathbb{F}_p$, $\beta, c_q \in \mathbb{F}_q$ and $\Phi(x) \in \mathbb{F}_p[x]$, $\Psi(x) \in \mathbb{F}_q[x]$, $\text{deg}(\Phi(x)) = \text{deg}(\Psi(x)) = 1$. An adversary can also do the above steps, but in $\mathbb{Z}_N[x]$. That is, it can get $c_N, \gamma$ and $\Theta(x)$ such that

$$c_N \equiv c_p \pmod{p} \quad \gamma \equiv \alpha \pmod{p} \quad \Theta(x) \equiv \Phi(x) \pmod{p}$$
$$c_N \equiv c_q \pmod{q} \quad \gamma \equiv \beta \pmod{q} \quad \Theta(x) \equiv \Psi(x) \pmod{q}$$

Combining (1), (3), (4) yields

$$\left(\frac{t^{-1}g(x)}{x^\delta - R_{id}}\right)_e = \left(\frac{c_p}{p_1}\right)_e \left(\frac{\alpha}{p_1}\right)_e, \quad \left(\frac{t^{-1}g(x)}{x^\delta - R_{id}}\right)_e = \left(\frac{c_q}{q_1}\right)_e \left(\frac{\beta}{q_1}\right)_e.$$  \quad (5)

Since $\delta < e$ and $e$ is a prime number, an adversary gains $\left(\frac{t^{-1}}{a_1}\right)_e$ by calculating $\left(\frac{c_N \gamma}{a_1}\right)_e^{\delta^e - 2} \pmod{e}$.

**Example A.5.** Finally, we give a toy example to show how an adversary attacks the space-efficient variation of BLS’s scheme. Assume that all parameters are set as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>4331</td>
<td>$R_{id}$</td>
<td>158</td>
</tr>
<tr>
<td>$p$</td>
<td>61</td>
<td>$r_{id}$</td>
<td>67</td>
</tr>
<tr>
<td>$q$</td>
<td>71</td>
<td>$f(x)$</td>
<td>$x^4 + 2x^3 + 3x^2 + 4x + 6$</td>
</tr>
<tr>
<td>$e$</td>
<td>5</td>
<td>$t$</td>
<td>7</td>
</tr>
<tr>
<td>$\mu$</td>
<td>1900</td>
<td>$g(x)$</td>
<td>$2102x + 3769$</td>
</tr>
<tr>
<td>$\delta$</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

By calculation, we learn $\left(\frac{7}{p_1}\right)_5 = \zeta_5^4$, $\left(\frac{7}{q_1}\right)_5 = \zeta_5$. An adversary first analyzes as

$$x^2 - 158 \equiv 2102^{-2}3769^2 - 158 = 1416 \pmod{2102x + 3769},$$

then gets $c_N = ((-1)^22102^2)$, $\gamma = 1416$, and finally derives $\left(\frac{c_N \gamma}{a_1}\right)_5^3 = 1 = \left(\frac{7}{a_1}\right)_5$. Actually, there is $\left(\frac{c_N \gamma}{p_1}\right)_5 = \left(\frac{7}{p_1}\right)_5^3$, $\left(\frac{c_N \gamma}{q_1}\right)_5 = \left(\frac{7}{q_1}\right)_5^3$.

**B Computing $(\frac{-\alpha}{a_1})_e$ for Large Values of $e$**

In [17], to compute the $e$-th power residue symbol, the authors constructed a “compatibility” identity and stated that it holds for all ideals in $\mathbb{Z}[\zeta_e]$. But this is not correct, e.g., If $\mathfrak{a}$ is a
prime ideal in $\mathbb{Z}[\zeta_e]$ and $\mathfrak{B} = \mathfrak{U} \cap \mathbb{Z}[\zeta_f]$ is a prime ideal in $\mathbb{Z}[\zeta_f]$ where $f \mid e$, the argument $\text{Norm}_{\mathbb{Z}[\zeta_e]}(\mathfrak{U}) = \text{Norm}_{\mathbb{Z}[\zeta_f]}(\mathfrak{B})$ is not always true. In fact, when $\mathfrak{B}$ is singular, the local-global principle makes the “compatibility” identity hold, see Chapter 1 in [16]. Furthermore, note that in the case $\text{Norm}_{\mathbb{Z}[\zeta_e]}(\mathfrak{U}) = p - 1$, it also holds due to the inclusion map $\iota : \mathbb{Z}[\zeta_e]/\mathfrak{U} \mapsto \mathbb{Z}[\zeta_f]/\mathfrak{B}$. Hence, we formalize the following revised theorem.

**Theorem B.1.** Let $e, f$ be integers with $f \mid e$. Let $p_1$ be as Lemma 2.1, and let $x \in \mathbb{Z}[\zeta_e]$. Then

$$\left( \frac{x}{p_1 \cap \mathbb{Z}[\zeta_f]} \right)_f = \left( \frac{x}{p_1} \right)^{\zeta^e}_e.$$

One can verify that $p_1 \cap \mathbb{Z}[\zeta_f] = p\mathbb{Z}[\zeta_e] + (\zeta_f - \mu^{e/f})\mathbb{Z}[\zeta_e]$ due to the fact that $\mu^{e/f}$ is a non-degenerate primitive $f$-th root of unity modulo $N$. Therefore, we are able to learn the value of $\left( \frac{x}{a_1} \right)_e$ by computing $\left( \frac{x}{N\mathbb{Z}[\zeta_f] + (\zeta_f - \mu^{e/f})\mathbb{Z}[\zeta_f]} \right)_f$ for each prime factor $f$ of $e$ and applying the Chinese remainder theorem.