Anonymous IBE From Quadratic Residuosity With Fast Encryption

Xiaopeng Zhao[1], Zhenfu Cao[1,2(✉)], Xiaolei Dong[1], and Jinwen Zheng[1]

[1] Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China
52164500025@stu.ecnu.edu.cn, zfcao@sei.ecnu.edu.cn
dongxiaolei@sei.ecnu.edu.cn, jinwen.zheng@foxmail.com
[2] Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen and Shanghai Institute of Intelligent Science and Technology, Tongji University, China

Abstract. We develop two variants of Cocks' identity-based encryption. One variant has faster encryption, where the most time-consuming part only requires several modular multiplications. The other variant makes the first variant anonymous under suitable complexity assumptions, while its decryption efficiency is about twice lower than the first one. Both the variants have ciphertext expansion twice more extensive than the original Cocks' identity-based encryption. To alleviate the issue of the second variant's large ciphertext expansion, we consider using it to construct a public-key encryption with keyword search scheme with a fast encryption algorithm by means of the transform in [1].

Keywords: Public-key cryptography · Identity-based encryption · Quadratic residuosity · Cocks' scheme · Anonymous encryption · Public-key encryption with keyword search

1 Introduction

The notion of identity-based cryptography was first proposed by Shamir [19] in 1984. This new paradigm of cryptography aims at solving the issue of managing and recovering the public-key certificate by simplifying the key management. For example, users' identification information such as email addresses or names rather than digital certificates can be used as their public key to encrypt or verify digital signature. Shamir constructed an identity-based signature scheme using the RSA function, but developing identity-based encryption (IBE) schemes turns out to be much harder. Until the year 2001, Shamir's open problem was solved by Boneh and Franklin [5] and Cocks [13] independently. Recently, lattice was considered as an emergent system for constructing IBE schemes (e.g., as in [15]). The Boneh-Franklin IBE scheme makes use of bilinear maps and is truly practical. Therefore, this work has attracted tons of attention from researchers over the years. However, Cocks' IBE scheme received less attention because of the lack of algebraic structure. Although Cocks' IBE scheme is inefficient for

large messages, it is simple, elegant, and secure under the standard quadratic residuosity (QR) assumption in the random oracle model. It can be used to encrypt short session keys in practice, e.g., a 128-bit symmetric key. Thus, the scheme was followed up by some researchers [2, 6, 7, 10–12, 14, 17, 20].

In 2016, Joye [17] made Cocks' scheme amenable to applications including electronic voting, auction systems, private information retrieval, or cloud computing; Joye proved that Cocks' scheme is homomorphic by considering Cocks' ciphertext as elements of a certain algebraic group. A similar conclusion can also be reached by considering Cocks' scheme over the polynomial quotient ring $\mathbb{Z}_N[x]/(x^2 - R_{\mathsf{id}})$ for which N is an RSA modulus and R_{id} is the IBE public key of an identity id [10, 11]. Our two variants are based on the latter structure.

It is well-known that Cocks' scheme is not anonymous due to Galbraith's test [4]. The test has been well studied by several researchers [2, 20]. Despite the test, some researchers [2, 6, 12, 14, 17] managed to propose anonymous variants of Cocks' scheme. In [14], the anonymization of Cocks' scheme was achieved for the first time, and a public-key encryption with keyword search (PEKS) scheme was proposed based on a variant of the quadratic residuosity problem. In this work, we mainly follow the approach of Joye in [17], which does not increase Cocks' ciphertext size or sacrifice its security.

In this work, we use the time-space tradeoff method to propose two variants of Cocks' IBE scheme [13] in the following two aspects:

1. Our first variant omits the computation of the Jacobi symbol $\left(\frac{a}{b}\right)$ for κ-bit integers a and b, which has $\mathcal{O}\left(M(\kappa)\log\kappa\right)$[3] time complexity [8], and the modular multiplicative inverse in Cocks' encryption. In detail, the ciphertext extension is increased by a factor of 2, but the most time-consuming part of the encryption in our variant only requires several modular multiplications of time complexity $\mathcal{O}\left(M(\kappa)\right)$ (see [9, Section 2.4]). The variant can also be proved semantic secure under a complexity assumption slightly stronger than the QR assumption, moreover, this improvement hardly affects the decryption speed.
2. Inspired by the anonymous variant of Cocks' scheme, without ciphertext expansion, proposed in [17, Section 6.2], our second variant makes the first variant anonymous under suitable complexity assumptions. This improvement does not affect the ciphertext expansion either. To alleviate the issue of the second variant's large ciphertext expansion, we consider using this variant to construct a PEKS scheme with a fast PEKS-encryption algorithm by means of the transform in [1].

The rest of the paper is organized as follows. In §2, we review the notion of semantic secure and the notion of anonymity. In §3, we describe our first variant and prove that it is semantic secure. In §4, we describe our second variant and prove that it is anonymous under reasonable complexity assumptions. In Appendix A, we give a suitable application of our second variant. Concluding remarks are given in §5.

[3] $M(\kappa)$ is the time to multiply κ-bit numbers.

2 Preliminaries

We write $x \xleftarrow{R} X$ for sampling at random an element x from the set X. If $\mathcal{A}$ is an algorithm, then we write $x \leftarrow \mathcal{A}(y)$ to mean: "run $\mathcal{A}$ on input y and the output is assigned to x".

2.1 Identity-Based Encryption

An *identity-based encryption* (IBE) scheme is defined as a tuple of probabilistic polynomial time (PPT) algorithms $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$:

$\mathsf{Setup}(1^\kappa)$ The setup algorithm Setup is a randomized algorithm that takes a security parameter 1^κ as input, and returns a tuple $(\mathsf{mpk}, \mathsf{msk})$, where mpk denotes the public parameters and msk denotes the master secret key. The message space is denoted by M.

$\mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$ The key generation algorithm KeyGen takes msk and an identity id as inputs, and returns a decryption key $\mathsf{sk}_{\mathsf{id}}$ associated with the identity id.

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, m)$ The encryption algorithm Enc is a randomized algorithm that takes the public parameters mpk, an identity id and a message $m \in \mathsf{M}$ as inputs, and returns a ciphertext C.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}, C)$ The decryption algorithm Dec takes the public parameters mpk, a secret key $\mathsf{sk}_{\mathsf{id}}$ (corresponding to the identity id) and a ciphertext C as inputs, and returns a message m if C is a valid ciphertext, and $\perp$ otherwise.

For any identity id and all messages $m \in \mathsf{M}$, the *correctness* property requires that

$$\mathsf{Dec}\left(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}, C \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, m)\right) = m.$$

2.2 Security Notions

The following notions are consistent with the notions described in [17, Section 2.2].

Semantic security. The semantic security property [16] states that it is infeasible for any adversary with the limited computation ability to get any information of a message given the corresponding ciphertext. The behaviors of an adversary $\mathcal{A}$ can be simulated by a pair of probabilistic PPT algorithms $(\mathcal{A}_1, \mathcal{A}_2)$. The adversary is allowed to adaptively make private key extraction queries to the key-extraction oracle $\mathsf{Extract}(\mathsf{mpk}, \mathsf{msk}, \cdot)$. The game between an adversary and a challenger contains the following five successive phases:

Initialization phase: The challenger takes a security parameter κ as input and runs the algorithm Setup. It then gives the public parameters mpk to the adversary $\mathcal{A}$ while keeping the master secret key msk to itself.

THE FIRST QUERY PHASE: After receiving mpk, $\mathcal{A}_1$ adaptively chooses an identity subspace ID_1 in the identity space ID, and issues the key generation queries to $\mathsf{Extract}(\mathsf{mpk}, \mathsf{msk}, \cdot)$ and obtains the private key corresponding to each identity in ID_1.

CHALLENGE PHASE: $\mathcal{A}_1$ fixes a challenge identity $\mathsf{id}^* \notin \mathsf{ID}_1$ and two different messages m_0, $m_1 \in \mathsf{M}$ of equal length. It then returns them along with some state information s. The challenger chooses uniformly at random a bit b and encrypts m_b with mpk and id^*. It then returns the corresponding ciphertext C as the challenge ciphertext to $\mathcal{A}_2$.

THE SECOND QUERY PHASE: Just like THE FIRST QUERY PHASE, $\mathcal{A}_2$ can adaptively issue more key generation queries in the identity space $\mathsf{ID}_2 \subseteq \mathsf{ID}$ which does not contain id^*.

GUESS PHASE: The goal of $\mathcal{A}_2$ is to guess the bit b from C and s. It outputs a guess b' of b.

Formally, an IBE scheme is said to be semantically secure if the advantage

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{IND\text{-}ID\text{-}CPA}}(1^\kappa) = \left| \Pr\left[\begin{array}{l} (\mathsf{mpk}, \mathsf{msk}) \xleftarrow{R} \mathsf{Setup}(1^\kappa), \\ (\mathsf{id}^*, m_0, m_1, \mathsf{s}) \leftarrow \mathcal{A}_1^{\mathsf{Extract}(\mathsf{mpk},\mathsf{msk},\cdot)}, \\ b \xleftarrow{R} \{0,1\}, C \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}^*, m_b) \end{array} : \mathcal{A}_2^{\mathsf{Extract}(\mathsf{mpk},\mathsf{msk},\cdot)}(C, \mathsf{s}) = b \right] - \frac{1}{2} \right|$$

is negligible in the security parameter κ for any PPT adversary $\mathcal{A}$. The semantic security can also be called indistinguishable, chosen-identity, chosen-plaintext ($\mathsf{IND\text{-}ID\text{-}CPA}$) security.

Anonymity. The notion of *anonymity* [3] is a strong requirement of privacy: it is infeasible for any adversary with the limited computation ability to get the identity of the recipient from the ciphertext. Anonymous IBE can be used for searchable encryption [1, 4]. The behaviors of an adversary $\mathcal{A}$ can also be simulated by a pair of probabilistic PPT algorithms $(\mathcal{A}_1, \mathcal{A}_2)$. The game between an adversary and a challenger contains the following five successive phases:

INITIALIZATION PHASE: The same as that in §2.2.

THE FIRST QUERY PHASE: The same as that in §2.2.

CHALLENGE PHASE: The adversary chooses two distinct challenge identities $\mathsf{id}_0^*, \mathsf{id}_1^* \notin \mathsf{ID}_1$ and a message $m \in \mathsf{M}$. It then returns them along with some state information s. The challenger chooses a random bit b and encrypts m with mpk and id_b^*. It then sends the corresponding ciphertext C to $\mathcal{A}_2$.

THE SECOND QUERY PHASE: Just like THE FIRST QUERY PHASE, $\mathcal{A}_2$ can issue more key generation queries in the identity space $\mathsf{ID}_2 \subseteq \mathsf{ID}$ which does not contain id_0^* and id_1^*.

GUESS PHASE: The same as that in §2.2.

Formally, an IBE scheme is said to be *anonymous* if the advantage

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{ANO\text{-}ID\text{-}CPA}}(\kappa) = \left| \Pr \left[\begin{array}{l} (\mathsf{mpk}, \mathsf{msk}) \xleftarrow{R} \mathsf{Setup}(1^\kappa), \\ (\mathsf{id}_0^*, \mathsf{id}_1^*, m, \mathsf{s}) \leftarrow \mathcal{A}_1^{\mathsf{Extract}(\mathsf{mpk},\mathsf{msk},\cdot)}, : \ \mathcal{A}_2^{\mathsf{Extract}(\mathsf{mpk},\mathsf{msk},\cdot)}(\mathsf{s}, C) = b \\ b \xleftarrow{R} \{0,1\}, C \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}_b^*, m) \end{array} \right] - \frac{1}{2} \right|$$

is negligible in the security parameter κ for any PPT adversary $\mathcal{A}$.

2.3 Complexity Assumption

Let N be a product of two RSA primes p and q. Let $\mathbb{J}_N = \left\{ x \in \mathbb{Z}_N^* \mid \left(\frac{x}{N}\right) = 1 \right\}$, i.e., the set of integers whose Jacobi symbols are 1. Let $\mathbb{QR}_N = \{x \mid \exists y \in \mathbb{Z}_N^*, x \equiv y^2 \pmod N\}$. The following complexity assumption slightly modifies the QR assumption.

Definition 1 (Strong Quadratic Residuosity (SQR) Assumption). *Given a security parameter κ. A PPT algorithm $\mathsf{RSAGen}(1^\kappa)$ generates two RSA primes p and q such that $p \equiv -q \mod 4$ and their product $N = pq$. $\mathsf{RSAGen}(\kappa)$ also chooses $u \xleftarrow{R} \mathbb{J}_N \setminus \mathbb{QR}_N$. The strong quadratic residuosity assumption with respect to $\mathsf{RSAGen}(\kappa)$ asserts that the advantage $\mathsf{Adv}_{\mathcal{A},\mathsf{RSAGen}}^{\mathsf{SQR}}(\kappa)$ defined as*

$$\left| \Pr \left[\mathcal{A}(N, u, x) = 1 \;\middle|\; x \xleftarrow{R} \mathbb{QR}_N \right] - \Pr \left[\mathcal{A}(N, u, x) = 1 \;\middle|\; x \xleftarrow{R} \mathbb{J}_N \setminus \mathbb{QR}_N \right] \right|$$

is negligible for any PPT adversary $\mathcal{A}$; the probabilities are taken over the experiment of running $(N, p, q, u) \leftarrow \mathsf{RSAGen}(\kappa)$ and choosing at random $x \in \mathbb{QR}_N$ and $x \in \mathbb{J}_N \setminus \mathbb{QR}_N$.

Remark 1. The only difference between the SQR assumption and the assumption on which Cocks' scheme relies is the choice of p and q. In the latter assumption, $N = pq$ where $p \equiv q \equiv 3 \pmod 4$, and $-1 \in \mathbb{J}_N \setminus \mathbb{QR}_N$ is public. Hence, we believe that breaking one is as intractable as breaking the other.

3 A Variant of Cocks' IBE Scheme with Fast Encryption

Our first scheme can be viewed as a variant of the classical Cocks' scheme. Define the function

$$\mathcal{J}_N(x) = \begin{cases} \perp, & \text{if } \gcd(x, N) \neq 1; \\ i, & \text{if } \gcd(x, N) = 1 \text{ and } \left(\frac{x}{N}\right) = (-1)^i. \end{cases}$$

Our first scheme proceeds as follows.

$\mathsf{Setup}(1^\kappa)$ Given a security parameter κ, Setup generates two RSA primes p and q such that $p \equiv -q \mod 4$ and their product $N = pq$. Setup also samples an element $u \xleftarrow{R} \mathbb{J}_N \setminus \mathbb{QR}_N$. The public parameters is $\mathsf{mpk} = \{N, u, \mathsf{H}\}$ where H is a publicly available cryptographic hash function mapping an arbitrary binary string to $\mathbb{J}_N$. The master secret key is $\mathsf{msk} = \{p, q\}$.

$\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$ Using mpk and msk, KeyGen sets $R_{\mathsf{id}} = \mathsf{H}(\mathsf{id})$. If $R_{\mathsf{id}} \in \mathbb{QR}_N$, KeyGen computes $r_{\mathsf{id}} = R_{\mathsf{id}}^{1/2} \bmod N$; otherwise it computes $r_{\mathsf{id}} = (uR_{\mathsf{id}})^{1/2} \bmod N$. Finally, KeyGen returns $\mathsf{sk}_{\mathsf{id}} = \{r_{\mathsf{id}}\}$ as user's private key.

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, m)$ On inputting mpk, an identity id and a message $m \in \{0, 1\}$, Enc derives the hash value $R_{\mathsf{id}} = \mathsf{H}(\mathsf{id})$. Enc then chooses at random two polynomials $f(x), \overline{f}(x)$ of degree 1 from $\mathbb{Z}_N[x]$ and calculates

$$g(x) = f(x)^2 \bmod (x^2 - R_{\mathsf{id}}) \quad \text{and} \quad \overline{g}(x) = \overline{f}(x)^2 \bmod (x^2 - uR_{\mathsf{id}}).$$

The returned ciphertext is $C = ((-1)^m \cdot g(x), (-1)^m \cdot \overline{g}(x))$.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}, C)$ On inputting mpk, a secret key $\mathsf{sk}_{\mathsf{id}} = \{r_{\mathsf{id}}\}$ and a ciphertext $C = (c(x), \overline{c}(x))$, Dec computes

$$m' = \begin{cases} \left(\frac{c(r_{\mathsf{id}})}{N}\right) & \text{if } r_{\mathsf{id}}^2 \equiv \mathsf{H}(\mathsf{id}) \pmod N; \\ \left(\frac{\overline{c}(r_{\mathsf{id}})}{N}\right) & \text{otherwise.} \end{cases}$$

and recovers the message m as $\mathcal{J}_N(m')$.

Correctness. The correctness of the decryption follows by noticing that when $r_{\mathsf{id}}^2 \equiv \mathsf{H}(\mathsf{id}) \pmod N$ we have

$$m' = \left(\frac{c(r_{\mathsf{id}})}{N}\right) = \left(\frac{(-1)^m f(r_{\mathsf{id}})^2}{N}\right) = (-1)^m,$$

and thus we can recover the message m by the function $\mathcal{J}_N$. When $r_{\mathsf{id}}^2 \equiv u\mathsf{H}(\mathsf{id}) \pmod N$, we can proceed similarly.

Remark 2. In the encryption, if we set $f(x) = ax + b$, we have

$$g(x) = f(x)^2 = (ax + b)^2 \equiv a^2 R_{\mathsf{id}} + b^2 + 2abx \pmod{x^2 - R_{\mathsf{id}}}.$$

Thus, calculating $g(x)$ needs two squares, two general multiplications and one addition modulo N. In the decryption, we need one more modular multiplication than Cocks' decryption. However, this hardly affects the decryption speed because computing one general 1024-bit Jacobi symbol is about 27 times slower than calculating one general 1024-bit modular multiplication according to the running times in [6, Table 1].

Before proving that the above scheme is semantic secure, we need the following theorem.

Theorem 1. *Let $t \in \mathbb{Z}_N^*$ and R an element in $\mathbb{J}_N \setminus \mathbb{QR}_N$. If $c(x) = \frac{f(x)^2}{t} \bmod (x^2 - R)$ for some $f(x) \in \mathbb{Z}_N[x]$ is a polynomial of degree 1, then the sets*

$$\Omega_k = \left\{ g(x) \in \mathbb{Z}_N[x] \;\middle|\; \deg g(x) = 1,\ \frac{g(x)^2}{k} \bmod (x^2 - R) = c(x) \right\}$$

are of the same size for each $k \in \mathbb{Z}_N^*$.

Proof. Consider the two sets $\Omega_t, \Omega_{\bar{t}}$, to prove the theorem, it suffices to prove that $\#\Omega_t = \#\Omega_{\bar{t}}$ for fixed t and any $\bar{t} \in \mathbb{Z}_N^*$. Suppose that $\left(\frac{t^{-1}\bar{t}}{p}\right) = (-1)^{i_t}$ and $\left(\frac{t^{-1}\bar{t}}{q}\right) = (-1)^{j_t}$ for $i_t, j_t \in \{0,1\}$. Since

$$\left(\frac{R^{i_t}}{p}\right) = \left(\frac{t^{-1}\bar{t}}{p}\right) \qquad \text{and} \qquad \left(\frac{R^{j_t}}{q}\right) = \left(\frac{t^{-1}\bar{t}}{q}\right),$$

there exist $W_p \in \mathbb{Z}_p^*$ and $W_q \in \mathbb{Z}_q^*$ such that

$$\begin{aligned} W_p^2 R^{i_t} &\equiv t^{-1}\bar{t} \pmod{p} \\ W_q^2 R^{j_t} &\equiv t^{-1}\bar{t} \pmod{q}. \end{aligned}$$

According to the Chinese Remainder Theorem, we have

$$\mathbb{Z}[x]/(N, x^2 - R) \cong \mathbb{Z}[x]/(p, x^2 - R) \oplus \mathbb{Z}[x]/(q, x^2 - R).$$

Therefore, the map $\phi : \Omega_t \to \Omega_{\bar{t}}$ given by $h(x) \mapsto g(x)$ where $h(x) \in \Omega_t, g(x) \in \Omega_{\bar{t}}$ and

$$\begin{aligned} g(x) &\equiv W_p x^{i_t} h(x) \pmod{(p, x^2 - R)} \\ g(x) &\equiv W_q x^{j_t} h(x) \pmod{(q, x^2 - R)} \end{aligned}$$

is well defined. In the other direction, the inverse map $\psi : \Omega_{\bar{t}} \to \Omega_t$ is given by $g(x) \mapsto h(x)$ where

$$\begin{aligned} h(x) &\equiv W_p^{-1} \left(R^{-1}x\right)^{i_t} g(x) \pmod{(p, x^2 - R)} \\ h(x) &\equiv W_q^{-1} \left(R^{-1}x\right)^{j_t} g(x) \pmod{(q, x^2 - R)} \end{aligned}$$

It is straightforward to verify that the composite map $\psi \circ \phi = 1_{\Omega_t}$ and $\phi \circ \psi = 1_{\Omega_{\bar{t}}}$ where 1_{Ω_t} and $1_{\Omega_{\bar{t}}}$ denote the identity maps on Ω_t and $\Omega_{\bar{t}}$ respectively. This establishes the bijection and completes the proof. □

Theorem 2. *Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary against the* IND-ID-CPA *security of the scheme in §3, making q_{H} queries to the random oracle* H *that are not followed by (private key) extraction queries before the* Challenge phase. *Then, there exists an adversary $\mathcal{B}$ against the* SQR *assumption such that*

$$\mathsf{Adv}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\mathcal{A}}(\kappa) = \frac{q_{\mathsf{H}}}{2} \cdot \mathsf{Adv}^{\mathsf{SQR}}_{\mathcal{B},\mathsf{RSAGen}}(\kappa)$$

The security proof is obtained by following the proof of [17, Appendix A].

Proof. Suppose that $\mathcal{B}$ is given a tuple $(N, u) \leftarrow \mathsf{RSAGen}(\kappa)$ and a random element $w \in \mathbb{J}_N$, and is asked to determine whether $w \in \mathbb{J}_N \setminus \mathbb{QR}_N$. $\mathcal{B}$ sets $\mathsf{mpk} = \{N, u, \mathsf{H}\}$ and gives it to $\mathcal{A}_1$, who has oracle access to hash queries and extraction queries, i.e., asking the private key corresponding to each identity in the chosen set ID_1. $\mathcal{B}$ answers the oracle queries as follows:

Hash queries Initially, $\mathcal{B}$ maintains a counter ctr initialized to 0 and a list $\mathcal{S}_{\mathsf{H}} \leftarrow \emptyset$ whose entry is in the form $(\mathsf{id}, R_{\mathsf{id}}, r_{\mathsf{id}})$. In addition, $\mathcal{B}$ selects $i^* \stackrel{R}{\leftarrow} \{1, 2, \ldots, q_{\mathsf{H}}\}$.

When $\mathcal{A}$ queries oracle H on an identity id, $\mathcal{B}$ increments ctr and checks whether there is an entry whose first component is id. If so, it returns R_{id}; otherwise,

1. If $ctr = i^*$, it returns w and appends the entry $(\mathsf{id}, w, \perp)$ to $\mathcal{S}_{\mathsf{H}}$;
2. Otherwise, it returns $h = u^{-j} r^2 \bmod N$ for which $r \stackrel{R}{\leftarrow} \mathbb{Z}_N$ and $j \stackrel{R}{\leftarrow} \{0, 1\}$, and appends the entry (id, h, r) to $\mathcal{S}_{\mathsf{H}}$.

Extraction queries When $\mathcal{A}$ queries the secret key on id, $\mathcal{B}$ first checks whether there is an entry whose first component is id. If not, it invokes $\mathsf{H}(\mathsf{id})$ to generate such an entry $(\mathsf{id}, R_{\mathsf{id}}, r_{\mathsf{id}})$. Finally, if $r_{\mathsf{id}} = \perp$, it aborts; otherwise, it returns r_{id}.

Afterwards, $\mathcal{A}_1$ selects a challenge identity $\mathsf{id}^* \notin \mathsf{ID}_1$. If $\mathsf{H}(\mathsf{id}^*) \neq w$, $\mathcal{B}$ returns $b \stackrel{R}{\leftarrow} \{0, 1\}$; otherwise, $\mathcal{B}$ does the following process:

1. Choose at random two polynomials $f(x), \overline{f}(x)$ of degree 1 from $\mathbb{Z}_N[x]$ and $b \stackrel{R}{\leftarrow} \{0, 1\}$. Calculate

$$g(x) = f(x)^2 \bmod (x^2 - w)$$
$$\overline{g}(x) = \overline{f}(x)^2 \bmod (x^2 - uw)$$

The corresponding ciphertext is

$$C_b = \begin{cases} (g(x),\ -\overline{g}(x)), & \text{if } b = 0; \\ (-g(x),\ \overline{g}(x)), & \text{otherwise.} \end{cases}$$

2. Give C_b to $\mathcal{A}_2$. $\mathcal{A}_2$ may issue more hash queries and extraction queries on identities except for id^*. Finally, $\mathcal{A}_2$ returns a bit b'.

3. If $b = b'$ return 1; otherwise return 0.

We first analyze the subcase that $w \neq \mathsf{H}(\mathsf{id}^*)$. In this case $\mathcal{B}$ returns a random bit, regardless of what w is. Therefore, we have $\Pr[\mathcal{B}(N, u, w) = 1 \mid w \in \mathbb{QR}_N \wedge w \neq \mathsf{H}(\mathsf{id}^*)] = \Pr[\mathcal{B}(N, u, w) = 1 \mid w \in \mathbb{J}_N \setminus \mathbb{QR}_N \wedge w \neq \mathsf{H}(\mathsf{id}^*)] = 1/2$. We now consider the subcase that $w = \mathsf{H}(\mathsf{id}^*)$. If $w \in \mathbb{QR}_N$, according to the fact that $uw \in \mathbb{J}_N \setminus \mathbb{QR}_N$ and Theorem 1, we conclude that C_b is a valid ciphertext for b. For the same reason, if $w \in \mathbb{J}_N \setminus \mathbb{QR}_N$, we conclude that C_b is a valid ciphertext for $1 - b$; in this case, $\mathcal{B}$ returns 1 if and only if $\mathcal{A}$ loses the IND-ID-CPA game. Let $\epsilon = \Pr[\mathcal{B}(N, u, w) = 1 \mid w \in \mathbb{QR}_N \wedge w = \mathsf{H}(\mathsf{id}^*)]$, and hence $\Pr[\mathcal{B}(N, u, w) = 1 \mid w \in \mathbb{J}_N \setminus \mathbb{QR}_N \wedge w = \mathsf{H}(\mathsf{id}^*)] = 1 - \epsilon$. We have

$$\begin{aligned} &\Pr[\mathcal{B}(N, u, w) = 1 \mid w \in \mathbb{QR}_N] \\ &= \Pr[w = \mathsf{H}(\mathsf{id}^*)] \cdot \Pr[\mathcal{B}(N, u, w) = 1 \mid w \in \mathbb{QR}_N \wedge w = \mathsf{H}(\mathsf{id}^*)] \\ &\quad + \Pr[w \neq \mathsf{H}(\mathsf{id}^*)] \cdot \Pr[\mathcal{B}(N, u, w) = 1 \mid w \in \mathbb{QR}_N \wedge w \neq \mathsf{H}(\mathsf{id}^*)] \\ &= \frac{\epsilon}{q_{\mathsf{H}}} + \left(1 - \frac{1}{q_{\mathsf{H}}}\right) \cdot \frac{1}{2} \end{aligned}$$

and similarly,

$$\begin{aligned}
&\Pr\left[\mathcal{B}\left(N,u,w\right)=1 \mid w\in \mathbb{J}_N\setminus\mathbb{QR}_N\right]\\
&=\Pr\left[w=\mathsf{H}(\mathsf{id}^*)\right]\cdot\Pr\left[\mathcal{B}\left(N,u,w\right)=1 \mid w\in \mathbb{J}_N\setminus\mathbb{QR}_N\wedge w=\mathsf{H}(\mathsf{id}^*)\right]\\
&\quad+\Pr\left[w\neq\mathsf{H}(\mathsf{id}^*)\right]\cdot\Pr\left[\mathcal{B}\left(N,u,w\right)=1 \mid w\in \mathbb{J}_N\setminus\mathbb{QR}_N\wedge w\neq\mathsf{H}(\mathsf{id}^*)\right]\\
&=\frac{1-\epsilon}{q_{\mathsf{H}}}+\left(1-\frac{1}{q_{\mathsf{H}}}\right)\cdot\frac{1}{2}
\end{aligned}$$

Consequently, we have

$$\begin{aligned}
&\mathsf{Adv}^{\mathsf{SQR}}_{\mathcal{B},\mathsf{RSAGen}}(\kappa)\\
&=\left|\Pr\left[\mathcal{B}\left(N,u,w\right)=1 \mid w\in \mathbb{QR}_N\right]-\Pr\left[\mathcal{B}\left(N,u,w\right)=1 \mid w\in \mathbb{J}_N\setminus\mathbb{QR}_N\right]\right|\\
&=\left|\frac{\epsilon}{q_{\mathsf{H}}}+\left(1-\frac{1}{q_{\mathsf{H}}}\right)\cdot\frac{1}{2}-\left(\frac{1-\epsilon}{q_{\mathsf{H}}}+\left(1-\frac{1}{q_{\mathsf{H}}}\right)\cdot\frac{1}{2}\right)\right|\\
&=\frac{2}{q_{\mathsf{H}}}\cdot\left|\epsilon-\frac{1}{2}\right|\\
&=\frac{2}{q_{\mathsf{H}}}\mathsf{Adv}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\mathcal{A}}(\kappa).
\end{aligned}$$

This completes the proof. □

4 An Anonymous Variant of Cocks' IBE Scheme with Fast Encryption

Galbraith developed a *test* which shows that Cocks' scheme is not anonymous. It was rigorously proved in [2, 20] that the test can distinguish the identity of the recipient from the ciphertext with overwhelming probability. It is not difficult to see that the scheme in §3 is also not anonymous when we simply modify *Galbraith's test* as:

$$\mathcal{GT}_N(R_{\mathsf{id}},C_i(x))=\left(\frac{c_{i0}^2-c_{i1}^2\alpha_i R_{\mathsf{id}}}{N}\right),\quad i=1,2.$$

where $\alpha_1=1,\alpha_2=u$, and $C=(C_1(x),C_2(x))=(c_{10}+c_{11}x,c_{20}+c_{21}x)$ represents the ciphertext (we still call it Galbraith's test in what follows). We should generate two types of ciphertexts whose Galbraith's tests are -1 and $+1$ separately to avoid this attack. Multiplying the ciphertext polynomial by a scalar does not work since the corresponding Galbraith's tests do not change. What about multiplying a polynomial? A polynomial x is feasible since

$$\mathcal{GT}_N(R_{\mathsf{id}},C_i'(x)=xC_i(x))=-\mathcal{GT}_N(R_{\mathsf{id}},C_i(x)),\quad i=1,2.$$

Therefore, inspired by the anonymous variant of Cocks' scheme, without ciphertext expansion in [17, Section 6.2], we can construct the following anonymous variant of the scheme in §3, without ciphertext expansion. Our second scheme proceeds as follows.

$\mathsf{Setup}(1^\kappa)$ Given a security parameter κ, Setup generates two RSA primes p and q such that $p \equiv -q \mod 4$ and their product $N = pq$. Setup samples an element $u \xleftarrow{R} \mathbb{J}_N \setminus \mathbb{QR}_N$. The public parameters is $\mathsf{mpk} = \{N, u, \mathsf{H}\}$ where H is a publicly available cryptographic hash function mapping an arbitrary binary string to $\mathbb{J}_N$. The master secret key is $\mathsf{msk} = \{p, q\}$.

$\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$ Using mpk and msk, KeyGen sets $R_{\mathsf{id}} = \mathsf{H}(\mathsf{id})$. If $R_{\mathsf{id}} \in \mathbb{QR}_N$, KeyGen computes $r_{\mathsf{id}} = R_{\mathsf{id}}^{1/2} \mod N$; otherwise it computes $r_{\mathsf{id}} = (uR_{\mathsf{id}})^{1/2} \mod N$. Finally, KeyGen returns $\mathsf{sk}_{\mathsf{id}} = \{r_{\mathsf{id}}\}$ as user's private key.

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, m)$ On inputting mpk, an identity id and a message $m \in \{0, 1\}$, Enc derives the hash value $R_{\mathsf{id}} = \mathsf{H}(\mathsf{id})$. Enc then chooses at random two polynomials f_1, f_2 of degree 1 from $\mathbb{Z}_N[x]$ and two bits $\beta_1, \beta_2 \xleftarrow{R} \{0, 1\}$. Set

$$\begin{aligned}
g_1^{(0)}(x) &= (-1)^m f_1(x)^2 && \mod (x^2 - R_{\mathsf{id}}) \\
g_1^{(1)}(x) &= (-1)^m x \cdot f_1(x)^2 && \mod (x^2 - R_{\mathsf{id}}) \\
g_2^{(0)}(x) &= (-1)^m f_2(x)^2 && \mod (x^2 - uR_{\mathsf{id}}) \\
g_2^{(1)}(x) &= (-1)^m x \cdot f_2(x)^2 && \mod (x^2 - uR_{\mathsf{id}})
\end{aligned}$$

The returned ciphertext is

$$C = \left(g_1^{(\beta_1)}(x), g_2^{(\beta_2)}(x)\right).$$

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}, C)$ On inputting mpk, a secret key $\mathsf{sk}_{\mathsf{id}} = \{r_{\mathsf{id}}\}$ and a ciphertext polynomial set $C = (C_1(x), C_2(x))$, if $r_{\mathsf{id}}^2 \equiv R_{\mathsf{id}} \pmod N$, Dec sets $h(x) = C_1(x)$ and computes $\sigma = \mathcal{GT}_N(R_{\mathsf{id}}, C_1(x))$; otherwise it sets $h(x) = C_2(x)$ and computes $\sigma = \mathcal{GT}_N(R_{\mathsf{id}}, C_2(x))$. Finally, Dec computes

$$m' = \begin{cases} \left(\frac{h(r_{\mathsf{id}})}{N}\right), & \text{if } \sigma = 1; \\ \left(\frac{r_{\mathsf{id}} h(r_{\mathsf{id}})}{N}\right), & \text{otherwise.} \end{cases}$$

and recovers the message m as $\mathcal{J}_N(m')$.

Correctness. According to the correctness proof of the scheme in §3, it is enough to show that the decryption is correct when $\sigma = -1$ and $r_{\mathsf{id}}^2 \equiv R_{\mathsf{id}} \pmod N$. In this case, we have $C_1(x) = g_1^{(1)}(x)$ and

$$m' = \left(\frac{r_{\mathsf{id}} C_1(r_{\mathsf{id}})}{N}\right) = \left(\frac{(-1)^m r_{\mathsf{id}}^2 f_1(r_{\mathsf{id}})^2}{N}\right) = (-1)^m.$$

Thus, the decryption works correctly.

Remark 3. The computation amount in the decryption is about twice times larger than that of the scheme in §3. However, the efficiency of the encryption and the size of the ciphertext expansion do not change.

It is easy to see that the above scheme is also IND-ID-CPA secure by comparing the ciphertexts between it and the scheme in §3: the ciphertext polynomials for the two schemes differ at most by a polynomial x. Therefore, assuming that there exists an IND-ID-CPA adversary $\mathcal{A}$ against the above scheme, we can construct an adversary $\mathcal{B}$ which can break the IND-ID-CPA security of the scheme in §3; given the ciphertext of the above scheme, $\mathcal{B}$ finds the original two polynomials $f_1(x)$ and $f_2(x)$ using Galbraith's test. Then $\mathcal{B}$ gives the ciphertext $C = (f_1(x), f_2(x))$ to $\mathcal{A}$. Finally, $\mathcal{B}$ returns whatever $\mathcal{A}$ returns.

The following theorem estimates the size of the first component of the scheme's ciphertext space when its encryption selects $\beta_1 = 0$.

Theorem 3. *With the notations in the above scheme, if we fix N and m, and assume without loss that $R_{\mathsf{id}} = \mathsf{H}(\mathsf{id}) \in \mathbb{QR}_N$, then the set*

$$Z_{N,m,R_{\mathsf{id}}} = \left\{ C_{a,b}(x) = (-1)^m (ax+b)^2 \bmod (x^2 - R_{\mathsf{id}}) : a, b \xleftarrow{R} \mathbb{Z}_N^* \;\middle|\; ar_{\mathsf{id}} \pm b \in \mathbb{Z}_N^* \right\}$$

has size at least $\frac{\varphi(N)(p-3)(q-3)}{16}$ (φ denotes the Euler's totient function). Therefore, the set of the first component of the scheme's ciphertext has size at least $\frac{\varphi(N)(p-3)(q-3)}{8}$ when its encryption selects $\beta_1 = 0$.

Proof. We have by a simple calculation that

$$C_{a,b}(x) = (-1)^m (ax+b)^2 \equiv (-1)^m \left(a^2 R_{\mathsf{id}} + b^2 + 2abx\right) \pmod{x^2 - R_{\mathsf{id}}}.$$

Suppose that $C_{a_1,b_1}(x) = C_{a_2,b_2}(x)$, we have

$$\begin{aligned} a_1^2 R_{\mathsf{id}} + b_1^2 &\equiv a_2^2 R_{\mathsf{id}} + b_2^2 \pmod N \\ 2a_1 b_1 &\equiv 2a_2 b_2 \pmod N \end{aligned}$$

This is equivalent to

$$\begin{aligned} (a_1 r_{\mathsf{id}} + b_1)^2 &\equiv (a_2 r_{\mathsf{id}} + b_2)^2 \pmod N \\ (a_1 r_{\mathsf{id}} - b_1)^2 &\equiv (a_2 r_{\mathsf{id}} - b_2)^2 \pmod N \end{aligned}$$

Fixing a_1 and b_1, if $a_1 r_{\mathsf{id}} + b_1 \in \mathbb{Z}_N^*$ and $a_1 r_{\mathsf{id}} - b_1 \in \mathbb{Z}_N^*$ (the latter means that $\mathcal{GT}_N(R_{\mathsf{id}}, C_{a_1,b_1}(x)) = 1$), then there are at most 16 choices of $a_2 \in \mathbb{Z}_N^*$ and $b_2 \in \mathbb{Z}_N^*$ for which $C_{a_1,b_1}(x) = C_{a_2,b_2}(x)$. The number of cases of $a_1 r_{\mathsf{id}} \pm b_1 \in \mathbb{Z}_N^*$ for $a_1, b_1 \in \mathbb{Z}_N^*$ is exactly $\varphi(N)(p-3)(q-3)$. This proves the first assertion. It is then clear that $Z_{N,0,R_{\mathsf{id}}} \cap Z_{N,1,R_{\mathsf{id}}} = \emptyset$ since the decryption algorithm can recover the original message. This proves the remaining assertion. □

Given an RSA modulus $N = pq$ and $\Delta \in \mathbb{Z}_N^*$, define the following sets:

- $\mathbb{S}_{N,\Delta} = \left\{ u \in \mathbb{Z}_N^* \;\middle|\; \gcd(u^2 - \Delta, N) = 1 \right\}$
- $\mathbb{S}_{N,\Delta}^{[-1]} = \left\{ u \in \mathbb{Z}_N^* \;\middle|\; \left(\frac{u^2-\Delta}{N}\right) = -1 \right\}$
- $\mathbb{S}_{N,\Delta}^{[+1]} = \left\{ u \in \mathbb{Z}_N^* \;\middle|\; \left(\frac{u^2-\Delta}{N}\right) = 1 \right\}$

– $(\mathbb{S}_{N,\Delta})^2 = \left\{u \in \mathbb{Z}_N^* \;\middle|\; \left(\frac{u^2-\Delta}{p}\right) = \left(\frac{u^2-\Delta}{q}\right) = 1\right\}$

For a prime p, let $\mathbb{QR}_p$ denotes the set of quadratic residues modulo p containing 0[4]. Perron [18] proved that any r relatively prime to p the set $r + \mathbb{QR}_p$ contains k quadratic residues and k quadratic non-residues when $p = 4k - 1$, or $k + 1$ quadratic residues and k quadratic non-residues when $p = 4k + 1$ and $r \in \mathbb{QR}_p$. Now, we take $r = -\Delta = -R_{\mathsf{id}}$ and assume without loss that $p \equiv 3 \pmod 4$, $q \equiv 1 \pmod 4$ and $R_{\mathsf{id}} \in \mathbb{QR}_N$. There are $\left(\frac{p+1}{4} - 1\right) \times 2 = \frac{p-3}{2}$ elements $u \in \mathbb{Z}_p^*$ for which $\left(\frac{u^2-\Delta}{p}\right) = 1$. Similarly, there are $\left(\frac{q+3}{4} - 2\right) \times 2 = \frac{q-5}{2}$ elements $u \in \mathbb{Z}_q^*$ for which $\left(\frac{u^2-\Delta}{q}\right) = 1$. Thus the size of $(\mathbb{S}_{N,\Delta})^2$ equals $\frac{(p-3)(q-5)}{4}$ and the size of $\mathbb{S}_{N,\Delta}^{[+1]}$ equals $\frac{(p-3)(q-5)}{4} + \frac{(p-3)(q-1)}{4} = \frac{(p-3)(q-3)}{2}$ (See also [20, Corollary 3.4]). Consequently, the set

$$S_{N,\Delta}^{[+1]} = \left\{a + bx : a, b \xleftarrow{R} \mathbb{Z}_N^* \;\middle|\; \frac{a}{b} \in \mathbb{S}_{N,\Delta}^{[+1]}\right\}$$

has size $\frac{\varphi(N)(p-3)(q-3)}{2}$. We have proved that the set of the first component of the scheme's ciphertext has size at least $\frac{\varphi(N)(p-3)(q-3)}{8}$ when $\beta_1 = 0$. Since this set can not cover the set $S_{N,\Delta}^{[+1]}$, to prove that the scheme achieves anonymity, we need to make the following complexity assumption:

Assumption 1 *Given an identity* id*, the set* $\left\{(f, g) \;\middle|\; f \in S_{N,R_{\mathsf{id}}}^{[+1]},\ g \in S_{N,uR_{\mathsf{id}}}^{[+1]}\right\}$ *is computationally equivalent to the scheme's ciphertext space when the identity of the recipient is* id*, and* $\mathsf{Enc}(\mathsf{PP}, \mathsf{id}, \cdot)$ *selects* $\beta_1 = \beta_2 = 0$*.*

When $\mathsf{Enc}(\mathsf{PP}, \mathsf{id}, \cdot)$ selects $\beta_1 = \beta_2 = 1$, it is clear that each component of the ciphertext space has size at least $\frac{\varphi(N)(p-3)(q-3)}{8}$. However, the set

$$S_{N,\Delta}^{[-1]} = \left\{c + dx : c, d \xleftarrow{R} \mathbb{Z}_N^* \;\middle|\; \frac{c}{d} \in \mathbb{S}_{N,\Delta}^{[-1]}\right\}$$

also has size $\frac{\varphi(N)(p-3)(q-3)}{2}$. Again, we shall make another assumption:

Assumption 2 *Given an identity* id*, the set* $\left\{(f, g) \;\middle|\; f \in S_{N,R_{\mathsf{id}}}^{[+1]},\ g \in S_{N,uR_{\mathsf{id}}}^{[-1]}\right\}$ *is computationally equivalent to the scheme's ciphertext space when the identity of the recipient is* id*, and* $\mathsf{Enc}(\mathsf{PP}, \mathsf{id}, \cdot)$ *selects* $\beta_1 = \beta_2 = 1$*.*

Theorem 4. *If Assumption 1 and 2 hold, the above scheme is anonymous.*

Proof. Let id_0^* and id_1^* be two distinct challenge identities. Without loss of generality, we assume that both $\mathsf{H}(\mathsf{id}_0^*)$ and $\mathsf{H}(\mathsf{id}_1^*)$ are in $\mathbb{QR}_N$. Letting $\Delta = R_{\mathsf{id}_r^*} = \mathsf{H}(\mathsf{id}_r^*)$ for some $r \in \{0, 1\}$, consider the following two distributions:

$$D_{0,r} = \left\{\{g_1^{(\beta_1)}(x), g_2^{(\beta_2)}(x)\} \;\leftarrow \mathsf{Enc}\left(\mathsf{mpk}, \mathsf{id}_r^*, m\right) :\ m, \beta_1, \beta_2 \in \{0, 1\}\right\}$$

$$D_{1,r} = \left\{\{a + bx, c + dx\} \;:\; a, b, c, d \xleftarrow{R} \mathbb{Z}_N^*,\ \frac{a}{b} \in \mathbb{S}_{N,\Delta},\ \frac{c}{d} \in \mathbb{S}_{N,\Delta}\right\}$$

[4] Perron considered the integer 0 as a quadratic residue. We should deal with it carefully.

We claim that $D_{0,r}$ and $D_{1,r}$ are computationally indistinguishable with overwhelming probability. The first component of an element in $D_{0,r}$ can be written as

$$\begin{cases} a_1 + b_1 x \ : \ \frac{a_1}{b_1} \in (\mathbb{S}_{N,\Delta})^2, & \text{if } \beta_1 = 0; \\ a_2 + b_2 x \ : \ \frac{a_2}{b_2} \in \mathbb{S}_{N,\Delta}^{[-1]}, & \text{otherwise.} \end{cases}$$

If Assumption 1 holds, since $S_{N,\Delta}^{[+1]} \cup S_{N,\Delta}^{[-1]} = \left\{a + bx : a, b \in \mathbb{Z}_N^* \ \middle| \ \frac{a}{b} \in \mathbb{S}_{N,\Delta}\right\}$ and β_1 is chosen at random, we deduce that the first component of an element in $D_{0,r}$ are computationally indistinguishable from that in $D_{1,r}$. If Assumption 2 holds, the similar arguments are valid for the second component, and hence we have proved the claim. Since $D_{1,0}$ and $D_{1,1}$ are also computationally indistinguishable with overwhelming probability, this proves that $D_{0,0}$ and $D_{0,1}$ are computationally indistinguishable with overwhelming probability, and hence the scheme is anonymous. □

5 Conclusion

The encryptions in known variants of Cocks' scheme are much slower than the corresponding decryptions, i.e., the scheme by Clear *et al.* [12] needs about 79 ms and 27 ms for encrypting a 128-bit message with a 1024-bit RSA modulus N. Our second variant features both anonymity and the best encryption compared with other variants (i.e., nearly 10 times faster than those in the same setting according to the running times in [6, Table 1]). Furthermore, they inherit the homomorphic property. These make schemes from quadratic residuosity more competitive in the fields of IBE.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China (Grant No.61632012 and 61672239), in part by the Peng Cheng Laboratory Project of Guangdong Province (Grant No. PCL2018KP004), and in part by the "Fundamental Research Funds for the Central Universities".

References

1. Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., Shi, H.: Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. J. Cryptology **21**(3), 350–391 (2008). https://doi.org/10.1007/s00145-007-9006-6
2. Ateniese, G., Gasti, P.: Universally anonymous IBE based on the quadratic residuosity assumption. In: Fischlin, M. (ed.) Topics in Cryptology - CT-RSA 2009. LNCS, vol. 5473, pp. 32–47. Springer (2009). https://doi.org/10.1007/978-3-642-00862-7_3
3. Bellare, M., Boldyreva, A., Desai, A., Pointcheval, D.: Key-privacy in public-key encryption. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 566–582. Springer (2001). https://doi.org/10.1007/3-540-45682-1_33

4. Boneh, D., Crescenzo, G.D., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer (2004). https://doi.org/10.1007/978-3-540-24676-3_30
5. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer (2001). https://doi.org/10.1007/3-540-44647-8_13
6. Boneh, D., Gentry, C., Hamburg, M.: Space-efficient identity based encryption-without pairings. In: 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07). pp. 647–657. IEEE (2007)
7. Boneh, D., LaVigne, R., Sabin, M.: Identity-based encryption with e^{th} residuosity and its incompressibility. In: Autumn 2013 TRUST Conference. Washington DC (Oct 9-10, 2013), poster presentation (2013)
8. Brent, R.P., Zimmermann, P.: An $O(M(n)\ \log n)$ algorithm for the Jacobi symbol. In: Hanrot, G., Morain, F., Thomé, E. (eds.) Algorithmic Number Theory, 9th International Symposium, 2010. LNCS, vol. 6197, pp. 83–95. Springer (2010). https://doi.org/10.1007/978-3-642-14518-6_10
9. Brent, R.P., Zimmermann, P.: Modern computer arithmetic, vol. 18. Cambridge University Press (2010)
10. Clear, M., Hughes, A., Tewari, H.: Homomorphic encryption with access policies: Characterization and new constructions. In: Youssef, A.M., Nitaj, A., Hassanien, A.E. (eds.) AFRICACRYPT 2013. LNCS, vol. 7918, pp. 61–87. Springer (2013). https://doi.org/10.1007/978-3-642-38553-7_4
11. Clear, M., McGoldrick, C.: Additively homomorphic IBE from higher residuosity. In: Lin, D., Sako, K. (eds.) PKC 2019. LNCS, vol. 11442, pp. 496–515. Springer (2019). https://doi.org/10.1007/978-3-030-17253-4_17
12. Clear, M., Tewari, H., McGoldrick, C.: Anonymous IBE from quadratic residuosity with improved performance. In: Pointcheval, D., Vergnaud, D. (eds.) AFRICACRYPT 2014. LNCS, vol. 8469, pp. 377–397. Springer (2014). https://doi.org/10.1007/978-3-319-06734-6_23
13. Cocks, C.C.: An identity based encryption scheme based on quadratic residues. In: Honary, B. (ed.) Cryptography and Coding, 8th IMA International Conference, 2001, Proceedings. LNCS, vol. 2260, pp. 360–363. Springer (2001). https://doi.org/10.1007/3-540-45325-3_32
14. Crescenzo, G.D., Saraswat, V.: Public key encryption with searchable keywords based on Jacobi symbols. In: Srinathan, K., Rangan, C.P., Yung, M. (eds.) INDOCRYPT 2007. LNCS, vol. 4859, pp. 282–296. Springer (2007). https://doi.org/10.1007/978-3-540-77026-8_21
15. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Dwork, C. (ed.) Proceedings of the 40th Annual ACM Symposium on Theory of Computing, 2008. pp. 197–206. ACM (2008). https://doi.org/10.1145/1374376.1374407
16. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of computer and system sciences **28**(2), 270–299 (1984)
17. Joye, M.: Identity-based cryptosystems and quadratic residuosity. In: Cheng, C., Chung, K., Persiano, G., Yang, B. (eds.) Public-Key Cryptography - PKC 2016. LNCS, vol. 9614, pp. 225–254. Springer (2016). https://doi.org/10.1007/978-3-662-49384-7_9
18. Perron, O.: Bemerkungen über die verteilung der quadratischen reste. Mathematische Zeitschrift **56**(2), 122–130 (1952)

19. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) Advances in Cryptology, Proceedings of CRYPTO '84. LNCS, vol. 196, pp. 47–53. Springer (1984). https://doi.org/10.1007/3-540-39568-7_5
20. Tiplea, F.L., Iftene, S., Teseleanu, G., Nica, A.: On the distribution of quadratic residues and non-residues modulo composite integers and applications to cryptography. Appl. Math. Comput. **372** (2020). https://doi.org/10.1016/j.amc.2019.124993

A A Public-Key Encryption with Keyword Search Scheme from Quadratic Residuosity

Boneh *et al.* introduced the notion of *public-key encryption with keyword search* (PEKS) and gave a proper security model and a construction methodology in [4]. PEKS is a form of "searchable encryption" that performs a keyword search on data encrypted using a public-key system. A promising application of PEKS is that of intelligent email routing. One may consider that mails come through a gateway which tests whether a keyword (e.g., "urgent") exists in an email. Of course, any other information about the email can not be revealed. A PEKS scheme consists of four PPT algorithms $(\mathsf{KeyGen}, \mathsf{PEKS}, \mathsf{Trapdoor}, \mathsf{Test})$.

$\mathsf{KeyGen}(1^\kappa)$ The key generation algorithm KeyGen is a randomized algorithm that takes as input a security parameter 1^κ and generates a public/private key pair $(\mathsf{pk}, \mathsf{sk})$.

$\mathsf{PEKS}(\mathsf{pk}, W)$ Given a public key pk and a keyword W, PEKS returns a searchable ciphertext S for W.

$\mathsf{Trapdoor}(\mathsf{sk}, W)$ Given a private key sk and a keyword W, the trapdoor algorithm Trapdoor produces a trapdoor T_W for keyword W.

$\mathsf{Test}(\mathsf{pk}, S, T_W)$ Given a public key pk, a searchable ciphertext $S \leftarrow \mathsf{PEKS}(\mathsf{pk}, W')$ and a trapdoor $T_W \leftarrow \mathsf{Trapdoor}(\mathsf{sk}, W)$, the test algorithm Test returns a bit b with 1 meaning "accept" or "yes" and 0 meaning "reject" or "no". It is required that $b = 1$ when $W = W'$.

In [1], the authors presented a new transform called new-ibe-2-peks that transforms any IND-ID-CPA-secure and anonymous IBE scheme into a PEKS-IND-CPA-secure and *computationally consistent* PEKS scheme. The resulting PEKS-encryption algorithm picks and encrypts a random message X and appends X to the ciphertext. We can naturally apply new-ibe-2-peks to the scheme of §4 and obtain the following PEKS scheme from quadratic residuosity.

$\mathsf{KeyGen}(1^\kappa)$ Given a security parameter κ, KeyGen defines a parameter k and generates two RSA primes p and q such that $p \equiv -q \mod 4$ and their product $N = pq$. KeyGen also samples an element $u \xleftarrow{R} \mathbb{J}_N \setminus \mathbb{QR}_N$. The public key is $\mathsf{pk} = \{N, k, u, \mathsf{H}\}$ where H is a publicly available cryptographic hash function mapping an arbitrary binary string to $\mathbb{J}_N$. The secret key is $\mathsf{sk} = \{p, q\}$.

$\mathsf{PEKS}(\mathsf{pk}, W)$ Given a public key pk and a keyword W, PEKS selects a k-bit message $X = [x_{k-1}, x_{k-2}, \ldots, x_0]$ (with $x_i \in \{0, 1\}$) and computes $R = \mathsf{H}(W)$. For each $i = 0, 1, \ldots k-1$, it chooses at random two polynomials $f_{i,1}, f_{i,2}$ of degree 1 from

$\mathbb{Z}_N[x]$, and two bits $\beta_{i,1}$, $\beta_{i,2} \xleftarrow{R} \{0,1\}$. Set

$$\begin{aligned} g_{i,1}^{(0)}(x) &= (-1)^{x_i} f_{i,1}(x)^2 && \bmod (x^2 - R) \\ g_{i,1}^{(1)}(x) &= (-1)^{x_i} x \cdot f_{i,1}(x)^2 && \bmod (x^2 - R) \\ g_{i,2}^{(0)}(x) &= (-1)^{x_i} f_{i,2}(x)^2 && \bmod (x^2 - uR) \\ g_{i,2}^{(1)}(x) &= (-1)^{x_i} x \cdot f_{i,2}(x)^2 && \bmod (x^2 - uR) \end{aligned}$$

PEKS returns the searchable ciphertext

$$S = \left(g_{0,1}^{(\beta_{0,1})}(x), g_{0,2}^{(\beta_{0,2})}(x), g_{1,1}^{(\beta_{1,1})}(x), g_{1,2}^{(\beta_{1,2})}(x), \ldots, g_{k-1,1}^{(\beta_{k-1,1})}(x), g_{k-1,2}^{(\beta_{k-1,2})}(x), X \right).$$

$\mathsf{Trapdoor}(\mathsf{sk}, W)$ Given a private key sk and a keyword W, the trapdoor algorithm $\mathsf{Trapdoor}$ computes $R = \mathsf{H}(W)$. If $R \in \mathbb{QR}_N$, it computes $T_W = R^{1/2} \bmod N$; otherwise it computes $T_W = (uR)^{1/2} \bmod N$. $\mathsf{Trapdoor}$ returns T_W.

$\mathsf{Test}(\mathsf{pk}, S, T_W)$ Given a public key pk, a searchable ciphertext

$$S = (C_{0,1}(x), C_{0,2}(x), C_{1,1}(x), C_{1,2}(x), \ldots, C_{k-1,1}(x), C_{k-1,2}(x), X)$$

where $C_{i,j}(x) = c_{i,j,0} + c_{i,j,1}x, \forall 0 \leq i < k, \forall 1 \leq j \leq 2$, and a trapdoor $T_W \leftarrow \mathsf{Trapdoor}(\mathsf{sk}, W)$, the test algorithm Test computes $R = \mathsf{H}(W)$. If $T_W^2 \equiv R \pmod N$, Test computes $\sigma_i = \left(\frac{c_{i,1,0}^2 - c_{i,1,1}^2 R}{N} \right)$ and sets $h_i(x) = C_{i,1}(x), \forall 0 \leq i < k$; otherwise it computes $\sigma_i = \left(\frac{c_{i,2,0}^2 - c_{i,2,1}^2 uR}{N} \right)$ and sets $h_i(x) = C_{i,2}(x), \forall 0 \leq i < k$. Finally, Test computes

$$x_i' = \begin{cases} \left(\frac{h_i(T_W)}{N} \right), & \text{if } \sigma_i = 1; \\ \left(\frac{T_W h_i(T_W)}{N} \right), & \text{otherwise.} \end{cases}$$

and recovers $X' = [\mathcal{J}_N(x_{k-1}'), \mathcal{J}_N(x_{k-2}'), \ldots, \mathcal{J}_N(x_0')]$. Test returns 1 if $X = X'$; and 0 otherwise.

For encrypting a message m with n keywords $W_1, W_2, \ldots, W_n$ with user's public key upk, Boneh *et al.* in [4] suggested that the sender computes and sends the ciphertext

$$C = (\mathsf{Enc}(\mathsf{upk}, m), \mathsf{PEKS}(\mathsf{upk}, W_1), \mathsf{PEKS}(\mathsf{upk}, W_2), \ldots, \mathsf{PEKS}(\mathsf{upk}, W_n))$$

to a proxy given the trapdoor T_{W_i} for each keyword W_i. Then the proxy can test whether m contains some keyword W_i, but it learns nothing more about any other information about m.