Published in M. S. Obaidat, P. Lorenz, and P. Samarati, Eds., *12th International Conference on Security and Cryptography (SECRYPT 2015)*, pp. 398–404, INSTICC Press, 2015.

A Key-Private Cryptosystem From the Quadratic Residuosity

Marc Joye
Technicolor, 175 S San Antonio Rd, Los Altos, CA 94022, USA
marc.joye@technicolor.com

Keywords: Public-Key Encryption, Key Privacy, Quadratic Residuosity.

Abstract: This paper presents a key-private public-key cryptosystem. More specifically, in addition to confidentiality, it provides privacy. Informally, ciphertexts yield no information whatsoever about its recipient (beyond what is publicly known). The presented cryptosystem also features a very fast key generation: the key generation boils down to a mere squaring modulo an RSA modulus. Further, it comes with strong security guarantees: it is proved to be semantically secure and key-private under the standard quadratic residuosity assumption.

1 INTRODUCTION

In numerous scenarios, the recipient's identity in a transmission needs to be kept private. This allows users to maintain some privacy. Protecting communication content may be not enough, as already observed in a couple of papers (e.g., (Barth et al., 2006; Bellare et al., 2001; Kiayias et al., 2007)). For example, by analyzing the traffic between an antenna and a mobile device, one can recover some information about [at least] user's position and some details about the use of her mobile device. This information leaks easily during all day: it is a common habit, indeed, to use a mobile phone every day and to keep it (almost) always switched on.

Key privacy in public-key encryption assumes a "homogeneous" environment. Indeed, if users make use of different cryptosystems or of the same cryptosystem but with keys of different lengths, anonymity is likely to be lost. The notion of anonymity is therefore is restricted to users sharing the same cryptosystem (with different keys) and common parameters. This implicitly defines a group of users.

Kiayias *et al.* introduce and model in (Kiayias et al., 2007) the concept of *group encryption*. This is the analogue for encryption of group signatures (Chaum and van Heyst, 1991). Group encryption allows one to conceal the identity of the recipient of a given ciphertext among a set of legitimate receivers. However, in case of misuse, some authority (the group manager) is capable of recovering the recipient's identity. This paper mostly deals with *full anonymity*: anonymity cannot be revoked.

Furthermore, in addition to security and privacy properties, group encryption offers *verifiability*: a sender can convince a verifier that the formed ciphertext can be decrypted by a group member. In this paper, we relax the requirements for group encryption. In the particular context of media broadcasting or wireless communications, we face a different situation where the sender (the broadcaster or the wireless emitter) can be trusted. This relaxation is justified by the fact that, in practical uses of the infrastructure, the sender has no interest in cheating because of business and reputation aspects. Moreover, it is very unlikely that an attacker can impersonate the sender, due to the particular material infrastructure needed (expensive, powerful, ...). Such an attacker should, indeed, mute the licit signals and substitute them with illicit ones, keeping all existing communications alive and faking the attacked ones.

As aforementioned, *key-private encryption* is a form of encryption which allows one to conceal the identity of the ciphertext's recipient. Known constructions for key-private cryptosystems involve somewhat costly key generations. We present in this paper a key-private cryptosystem enjoying a fast key generation. In our case, the key generation boils down to a mere modular squaring. Furthermore, to our best knowledge, the presented cryptosystem is the sole key-private construction that is provably secure under the standard quadratic residuosity assumption, in the standard model.

Outline of the Paper The rest of this paper is organized as follows. In the next section, we review some background on public-key encryption. We then proceed in Section 3 with the presentation of a key-private cryptosystem. We show its correctness and study its features. In Section 4, we prove that the scheme is semantically secure and key-private. Finally, we conclude in Section 5.

2 PRELIMINARIES

2.1 Public-Key Encryption

In order to better capture the property that users may share some common parameters in a homogeneous environment, the key generation algorithm is divided in two sub-algorithms: the *common-key generation* algorithm and the *key generation* algorithm.

Following the syntax of (Bellare et al., 2001) (see also (Goldwasser and Micali, 1984)), we define a *public-key encryption scheme* as a tuple of four algorithms $(\mathtt{SETUP}, \mathtt{KEYGEN}, \mathtt{ENCRYPT}, \mathtt{DECRYPT})$:

COMMON-KEY GENERATION The common-key generation algorithm $\mathtt{SETUP}$ takes as input a security parameter 1^κ and outputs some common parameters $\mathsf{PP} \overset{R}{\leftarrow} \mathtt{SETUP}(1^\kappa)$.

KEY GENERATION The key generation algorithm $\mathtt{KEYGEN}$ is a randomized algorithm that takes on input PP and returns a matching pair of public key and secret key for some user: $(\mathsf{upk}, \mathsf{usk}) \overset{R}{\leftarrow} \mathtt{KEYGEN}(\mathsf{PP})$.

ENCRYPTION Let $\mathcal{M}$ denote the message space. The encryption algorithm $\mathtt{ENCRYPT}$ is a randomized algorithm that takes in a public key upk and a plaintext $m \in \mathcal{M}$, and returns a ciphertext C. We write $C \leftarrow \mathtt{ENCRYPT}_{\mathsf{upk}}(m)$.

DECRYPTION The decryption algorithm $\mathtt{DECRYPT}$ takes in secret key usk (matching upk) and ciphertext C and returns the corresponding plaintext m or a special symbol $\perp$ indicating that the ciphertext is invalid. We write $m \leftarrow \mathtt{DECRYPT}_{\mathsf{usk}}(C)$ if C is a valid ciphertext and $\perp \leftarrow \mathtt{DECRYPT}_{\mathsf{usk}}(C)$ if it is not.

We require that $\mathtt{DECRYPT}_{\mathsf{usk}}(\mathtt{ENCRYPT}_{\mathsf{upk}}(m)) = m$ for all messages $m \in \mathcal{M}$.

2.2 Security Notions

Indistinguishability of Encryptions The notion of *indistinguishability of encryptions* (Goldwasser and Micali, 1984) captures a strong notion of data-privacy: The adversary should not learn any information whatsoever about a plaintext given its encryption beyond the length of the plaintext.

We view an adversary $\mathcal{A}$ as a pair $(\mathcal{A}_1, \mathcal{A}_2)$ of probabilistic algorithms. This corresponds to adversary $\mathcal{A}$ running in two stages. In the "find" stage, algorithm $\mathcal{A}_1$, on input public parameters PP and a public key upk, outputs two (different) equal-size messages m_0 and $m_1 \in \mathcal{M}$ and some state information s. In the "guess" stage, algorithm $\mathcal{A}_2$ receives a challenge ciphertext C which is the encryption of m_b under upk and where b is chosen at random in $\{0,1\}$. The goal of $\mathcal{A}_2$ is to recover the value of b from s and C.

A public-key encryption scheme is said *semantically secure* (or *indistinguishable*) if

$$\Pr\left[\begin{array}{l} \mathsf{PP} \overset{R}{\leftarrow} \mathtt{SETUP}(1^\kappa), \\ (\mathsf{upk}, \mathsf{usk}) \overset{R}{\leftarrow} \mathtt{KEYGEN}(\mathsf{PP}), \\ (m_0, m_1, s) \leftarrow \mathcal{A}_1(\mathsf{PP}, \mathsf{upk}), \\ b \overset{R}{\leftarrow} \{0,1\}, C \leftarrow \mathtt{ENCRYPT}_{\mathsf{upk}}(m_b) \end{array} : \mathcal{A}_2(s, C) = b\right] - \frac{1}{2}$$

is negligible in the security parameter for any polynomial-time adversary $\mathcal{A}$; the probability is taken over the random coins of the experiment according to the distribution induced by $\mathtt{SETUP}$ and $\mathtt{KEYGEN}$ and over the random coins of the adversary.

As we are in the public-key setting, the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is given the public key upk and so can encrypt any message of its choice. In other words, the adversary can mount chosen-plaintext attacks (CPA). Hence, we write IE-CPA the security notion achieved by a semantically secure encryption scheme.[1]

Indistinguishability of Keys Analogously, the notion of *indistinguishability of keys* captures a strong requirement about key privacy: The adversary should not be able to link whatsoever a ciphertext with its underlying encryption key.

As before, we view an adversary $\mathcal{A}$ as a pair $(\mathcal{A}_1, \mathcal{A}_2)$ of probabilistic algorithms. In the "find" stage, algorithm $\mathcal{A}_1$, on input two public keys upk_0 and upk_1, outputs a message m and some state information s. Then in the "guess" stage, algorithm $\mathcal{A}_2$ receives a challenge ciphertext C which is the encryption of m under upk_b where b is chosen at random in $\{0,1\}$. The goal of $\mathcal{A}_2$ is to recover the value of b from s and C.

More formally, a public-key encryption scheme is

[1]We deviate from the usual notation of IND-CPA to emphasize the fact that indistinguishability is about encryptions.

said *anonymous* (or *key-private*) if

$$\Pr\left[\begin{array}{l}\mathsf{PP} \overset{R}{\leftarrow} \mathtt{SETUP}(1^\kappa)\\ (\mathsf{upk}_0,\mathsf{usk}_0) \overset{R}{\leftarrow} \mathtt{KEYGEN}(\mathsf{PP}),\\ (\mathsf{upk}_1,\mathsf{usk}_1) \overset{R}{\leftarrow} \mathtt{KEYGEN}(\mathsf{PP}),\\ (m,s) \leftarrow \mathcal{A}_1(\mathsf{PP},\mathsf{upk}_0,\mathsf{upk}_1),\\ b \overset{R}{\leftarrow} \{0,1\}, C \leftarrow \mathtt{ENCRYPT}_{\mathsf{upk}_b}(m)\end{array} : \mathcal{A}_2(s,C) = b\right] - \frac{1}{2}$$

is negligible in the security parameter for any polynomial-time adversary $\mathcal{A}$; the probability is taken over the random coins of the experiment according to the distribution induced by SETUP and KEYGEN and over the random coins of the adversary.

This definition of anonymity gives rise to the security notion of IK-CPA or *indistinguishability of keys under chosen-plaintext attacks*.

Of course, the goals of data-privacy and key-privacy can be combined to define extended security notions. A public-key encryption scheme achieves IND-CPA *security* (indistinguishability under chosen-plaintext attacks) if it is both IE-CPA and IK-CPA.

2.3 Complexity Assumptions

It is useful to introduce some notation. Let $N = pq$ be the product of two (odd) primes p and q. The Jacobi symbol modulo N of an integer a is denoted by $\left(\frac{a}{N}\right)$. The set of integers whose Jacobi symbol is 1 is denoted by $\mathbb{J}_N$, $\mathbb{J}_N = \left\{a \in (\mathbb{Z}/N\mathbb{Z})^\times \mid \left(\frac{a}{N}\right) = 1\right\}$; the set of quadratic residues is denoted by $\mathbb{QR}_N$, $\mathbb{QR}_N = \left\{a \in (\mathbb{Z}/N\mathbb{Z})^\times \mid \left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = 1\right\}$. Note that $\mathbb{QR}_N$ is a subset of $\mathbb{J}_N$.

Definition 1 (Quadratic Residuosity Assumption)**.** *Let* RSAGen *be a probabilistic algorithm which, given a security parameter* 1^κ*, outputs primes* p *and* q *and their product* $N = pq$*. The* Quadratic Residuosity *(*QR*) assumption asserts that the success probability defined as the distance*

$$\Big|\Pr[\mathcal{D}(x,N) = 1 \mid x \overset{R}{\leftarrow} \mathbb{QR}_N] - \Pr[\mathcal{D}(x,N) = 1 \mid x \overset{R}{\leftarrow} \mathbb{J}_N \setminus \mathbb{QR}_N]\Big|$$

is negligible for any probabilistic polynomial-time distinguisher $\mathcal{D}$*; the probabilities are taken over the experiment of running* $(N,p,q) \leftarrow \mathsf{RSAGen}(1^\kappa)$ *and choosing at random* $x \in \mathbb{QR}_N$ *and* $x \in \mathbb{J}_N \setminus \mathbb{QR}_N$*.*

3 A KEY-PRIVATE CRYPTOSYSTEM

3.1 Description

Using the syntax introduced in Section 2.1, the cryptosystem is defined as follows.

$\mathtt{SETUP}(1^\kappa)$ Given as input security parameter 1^κ, SETUP generates an RSA modulus $N = pq$ where p and q are prime and $p \equiv -q \pmod 4$. The factorization of N is erased. The public parameters are $\mathsf{PP} = \{N\}$.

$\mathtt{KEYGEN}(\mathsf{PP})$ For user i, the key generation algorithm KEYGEN picks a random element $r_i \in_R \mathbb{Z}/N\mathbb{Z}$ and sets $R_i = {r_i}^2 \bmod N$. It outputs the public key $\mathsf{upk}_i = \{R_i\}$ and matching private key $\mathsf{usk}_i = \{r_i\}$.

$\mathtt{ENCRYPT}_{\mathsf{upk}_i}(m)$ To encrypt a message $m \in \{0,1\}$ for user i, ENCRYPT chooses at random $t \in_R \mathbb{Z}/N\mathbb{Z}$ and $\beta \in_R \{0,1\}$, and sets

$$\tau = (-1)^m \left(\frac{2t}{N}\right)$$

and

$$c = \begin{cases} \dfrac{t^2 + R_i}{2t} \bmod N & \text{if } \beta = 0\\[2ex] \dfrac{2R_i t}{t^2 + R_i} \bmod N & \text{if } \beta = 1\end{cases} .$$

The returned ciphertext is $C = \{\tau, c\}$.

$\mathtt{DECRYPT}_{\mathsf{usk}_i}(C)$ Given a ciphertext $C = \{\tau, c\}$, the decryption algorithm DECRYPT first computes $\sigma = \left(\frac{c^2 - R_i}{N}\right)$. If $\sigma = -1$, it updates c as $c \leftarrow \frac{R_i}{c} \bmod N$. It then returns plaintext m as

$$m = \frac{1 - \tau \cdot \left(\frac{c + r_i}{N}\right)}{2} .$$

3.2 Correctness

We show that a correctly generated ciphertext $C = \{\tau, c\}$ decrypts to the matching plaintext m.

First observe that the condition $p \equiv -q \pmod 4$ implies that $\left(\frac{-1}{N}\right) = \left(\frac{-1}{p}\right)\left(\frac{-1}{q}\right) = -1$. This in turn implies that $\sigma = (-1)^\beta$. Indeed, there are two cases:

1. $[\beta = 0]$ Then $c = \frac{t^2 + R_i}{2t} \bmod N$. Consequently, we get

$$\begin{aligned} c^2 - R_i &\equiv \frac{t^4 + {R_i}^2 + 2t^2 R_i}{4t^2} - R_i\\ &\equiv \left[\frac{t^2 - R_i}{2t}\right]^2 \pmod N .\end{aligned}$$

This yields $\left(\frac{c^2 - R_i}{N}\right) = 1$.

2. [$\beta = 1$] Then $c = \frac{2R_i t}{t^2+R_i} \bmod N$. Hence, we have

$$\begin{aligned} c^2 - R_i &\equiv \frac{4{R_i}^2 t^2}{t^4 + {R_i}^2 + 2t^2 R_i} - R_i \\ &\equiv -R_i \left[\frac{-4R_i t^2}{t^4 + {R_i}^2 + 2t^2 R_i} + 1\right] \\ &\equiv -R_i \left[\frac{t^2 - R_i}{t^2 + R_i}\right]^2 \pmod N \end{aligned}$$

and thus $\left(\frac{c^2-R_i}{N}\right) = \left(\frac{-R_i}{N}\right) = -1$.

When $\sigma = -1$ (or equivalently, $\beta = 1$), the decryption algorithm updates c as $c \leftarrow \frac{R_i}{c} \bmod N$, which gives $c = \frac{t^2+R_i}{2t} \bmod N$. In all cases, we then have

$$\tau \cdot \left(\frac{c + r_i}{N}\right) = \tau \cdot \left(\frac{2t}{N}\right) = (-1)^m$$

by noting that $c + r_i \equiv \frac{t^2+R_i}{2t} + r_i \equiv \frac{(t+r_i)^2}{2t} \equiv 2t \left[\frac{t+r_i}{2t}\right]^2 \pmod N$ since $R_i = {r_i}^2 \bmod N$; and thereby

$$\frac{1 - \tau \cdot \left(\frac{c+r_i}{N}\right)}{2} = \frac{1-(-1)^m}{2} = m \ .$$

3.3 Comparison

For the sake of comparison, we review below the celebrated Goldwasser-Micali cryptosystem (Goldwasser and Micali, 1984) and the single-bit variant of the BGH cryptosystem (Boneh et al., 2007) —-both relying on the quadratic residuosity, without random oracles.

Goldwasser-Micali Cryptosystem This cryptosystem does not allow multiple users to use the same RSA modulus. There is no SETUP algorithm.

KEYGEN(1^κ) Given as input security parameter 1^κ, KEYGEN generates an RSA modulus $N_i = p_i q_i$ where p_i and q_i are prime. It also chooses a random element $y_i \in \mathbb{J}_{N_i} \setminus \mathbb{QR}_{N_i}$. It outputs the public key for user i, $\mathsf{upk}_i = \{N_i, y_i\}$, and the matching private key $\mathsf{usk}_i = \{p_i\}$.

ENCRYPT$_{\mathsf{upk}_i}(m)$ To encrypt a message $m \in \{0,1\}$ for user i, ENCRYPT chooses at random $t \in_R \mathbb{Z}/N_i\mathbb{Z}$ and sets

$$C = {y_i}^m t^2 \bmod N_i \ .$$

The returned ciphertext is C.

DECRYPT$_{\mathsf{usk}_i}(C)$ Given a ciphertext C, the decryption algorithm DECRYPT first computes $\sigma = \left(\frac{C}{p_i}\right)$. It then returns plaintext m as

$$m = \frac{1-\sigma}{2} \ .$$

Single-bit BGH Cryptosystem In the public-key setting, the BGH cryptosystem requires a publicly available oracle Q taking as input an RSA modulus N and two quadratic residues $R, S \in \mathbb{QR}_N$ and outputting two polynomials $f, g \in (\mathbb{Z}/N\mathbb{Z})[X]$ such that

- $f(r)\,g(s) \in \mathbb{QR}_N$ for all square roots r of R and s of S;
- $f(r)f(-r) \in \mathbb{QR}_N$ for all square roots r of R.

This can be achieved by deterministically constructing a solution $(x,y) \in (\mathbb{Z}/N\mathbb{Z})^2$ to the equation

$$Rx^2 + Sy^2 = 1$$

and returning

$$f(X) = xX + 1 \quad \text{and} \quad g(X) = 2yX + 2 \ .$$

Following (Boneh et al., 2007), it is readily verified that $f(r)g(s) = 2(xr+1)(ys+1) = (Rx^2 + Sy^2 - 1) + 2(xr+1)(ys+1) = (rx + sy + 1)^2 \in \mathbb{QR}_N$ and $f(r)f(-r) = (xr+1)(-xr+1) = -Rx^2 + 1 = Sy^2 \in \mathbb{QR}_N$.

SETUP(1^κ) Given as input security parameter 1^κ, SETUP generates an RSA modulus $N = pq$ where p and q are prime. The factorization of N is erased. The public parameters are $\mathsf{PP} = \{N\}$.

KEYGEN(PP) For user i, the key generation algorithm KEYGEN picks a random element $r_i \in_R \mathbb{Z}/N\mathbb{Z}$ and sets $R_i = {r_i}^2 \bmod N$. It outputs the public key $\mathsf{upk}_i = \{R_i\}$ and matching private key $\mathsf{usk}_i = \{r_i\}$.

ENCRYPT$_{\mathsf{upk}_i}(m)$ To encrypt a message $m \in \{0,1\}$ for user i, ENCRYPT chooses at random $s \in_R \mathbb{Z}/N\mathbb{Z}$ and sets $S = s^2 \bmod N$. It calls oracle Q,

$$(f,g) \leftarrow Q(N, R_i, S) \,,$$

and computes

$$c = (-1)^m \left(\frac{g(s)}{N}\right) \ .$$

The returned ciphertext is $C = \{S, c\}$.

DECRYPT$_{\mathsf{usk}_i}(C)$ Given a ciphertext $C = \{S, c\}$, the decryption algorithm DECRYPT first calls Q to obtain

$$(f,g) \leftarrow Q(N, R_i, S)$$

and computes $\sigma = \left(\frac{f(r_i)}{N}\right)$. It then returns plaintext m as

$$m = \frac{1-\sigma}{2} \ .$$

The BGH cryptosystem requires finding a solution $(x,y) \in (\mathbb{Z}/N\mathbb{Z})^2$ to the equation $R_i x^2 + Sy^2 = 1$, which constitutes a real bottleneck. Indeed, the best method currently available is of quartic complexity.

This in turn incurs rather long encryption and decryption times. The main advantage of the BGH system resides in the bandwidth saved when large ciphertexts (i.e., multi-bit ciphertexts) are processed. As this paper is concerned with speed-efficient (and not bandwidth-efficient) key-private cryptosystems, the BGH cryptosystem will not be included in the comparison.

Key Privacy The Goldwasser-Micali cryptosystem is semantically secure under the quadratic residuosity assumption in the standard model. Unfortunately, it is *not* key-private. As already noticed for the RSA cryptosystem in (Bellare et al., 2001), one problem is that the value of the ciphertext leaks some information about the modulus. If $C > N_j$ then we know for sure that user j (i.e., the user with public key $\{N_j, y_j\}$) is not the recipient of the ciphertext C.

This issue is easily mitigated in the case of RSA by adding a carefully chosen multiple of the modulus to the ciphertext. But this simple fix does not apply here. Indeed, given a ciphertext C for an unknown recipient, we can always compute

$$\tau_j = \left(\frac{C}{N_j}\right) \ .$$

If $\tau_j \neq 1$ then we can deduce that user j is not the recipient of the ciphertext C.

Performance In addition of being key-private, the scheme of Section 3.1 has a much more efficient key generation than the Goldwasser-Micali scheme. It simply requires evaluating a square modulo N whereas the Goldwasser-Micali scheme requires generating two large primes and a pseudo-square. This means that a device with limited computing capabilities can generate keys for the scheme of Section 3.1.

The key generation can even be made easier —at the expense of a larger public key— by defining R_i as $R_i = {r_i}^2 + \alpha N$ (instead of $R_i = {r_i}^2 \bmod N$) for some random $\alpha \in_R \{0,1\}^{|N|+\kappa}$.

4 SECURITY ANALYSIS

The two next propositions assess the security of the scheme under the quadratic residuosity assumption.

Proposition 1. *The scheme is* IE-CPA *under the quadratic residuosity assumption.*

Proof. Assume there exists an IE-CPA adversary $\mathcal{A}$ that can break the scheme with probability ε. We will use $\mathcal{A}$ to decide whether a random element w in $\mathbb{J}_N$ is a quadratic residue modulo N or not.

Consider the following distinguisher $\mathcal{D}(w,N)$ for solving the QR problem:

1. Define $R_i = w$, set $\mathsf{upk}_i = \{R_i\}$, and give (N, upk_i) to $\mathcal{A}$;
2. Choose a random bit $b \in_R \{0,1\}$ and compute the encryption of b under public key upk_i as $C_b = \{\tau_b, c_b\}$ where $\tau_b = (-1)^b \left(\frac{2t}{N}\right)$ and
$$c_b = \begin{cases} \frac{t^2+R_i}{2t} \bmod N & \text{if } \beta = 0 \\ \frac{2R_i t}{t^2+R_i} \bmod N & \text{if } \beta = 1 \end{cases}$$
some random element $t \in_R \mathbb{Z}/N\mathbb{Z}$ and bit $\beta \in_R \{0,1\}$;
3. Give $C_b = \{\tau_b, c_b\}$ to $\mathcal{A}$ and obtain its guess b';
4. If $b' = b$ return 1; otherwise return 0.

There are two cases to distinguish.

Case 1 Suppose first that w is a quadratic residue modulo N. Clearly, $\mathcal{D}$ returns 1 exactly when $\mathcal{A}$ wins in the IE-CPA game. We thus have $\Pr[\mathcal{D}(w,N) = 1 \mid w \in \mathbb{QR}_N] = \varepsilon$.

Case 2 Suppose now that $w \in \mathbb{J}_N \setminus \mathbb{QR}_N$. It is important to see that if $t \pmod{\{p,q\}}$ is replaced with $\frac{R_i}{t} \pmod{\{p,q\}}$ in the computation of c_b, the value of c_b is unchanged:

$$\frac{\left[\frac{R_i}{t}\right]^2 + R_i}{2\frac{R_i}{t}} \equiv \frac{t^2+R_i}{2t} \pmod{\{p,q\}}$$

and

$$\frac{2R_i\frac{R_i}{t}}{\left[\frac{R_i}{t}\right]^2 + R_i} \equiv \frac{2R_i t}{t^2+R_i} \pmod{\{p,q\}} \ .$$

Hence, consider $t_1, t_2, t_3 \in (\mathbb{Z}/N\mathbb{Z})^\times$ such that

- $t_1 \equiv t \pmod p$, $t_1 \equiv R_i/t \pmod q$;
- $t_2 \equiv R_i/t \pmod p$, $t_2 \equiv t \pmod q$;
- $t_3 \equiv R_i/t \pmod p$, $t_3 \equiv R_i/t \pmod q$.

In $\mathcal{A}$'s view, from c_b, the four possible values t, t_1, t_2, and t_3 are equally likely. At the same time, since $R_i \in \mathbb{J}_N \setminus \mathbb{QR}_N$ we also have $\left(\frac{t}{N}\right) = \left(\frac{t_3}{N}\right) \neq \left(\frac{t_1}{N}\right) = \left(\frac{t_2}{N}\right)$.

The probability that $\mathcal{A}$ recovers $\left(\frac{2t}{N}\right)$ from c_b is therefore $\frac{1}{2}$ —note that τ_b carries no information on $\left(\frac{2t}{N}\right)$ since b is random in $\{0,1\}$. As a result, $\mathcal{D}$ will return 1 with probability $\frac{1}{2}$.

We so obtain:

$$\Big|\Pr\big[\mathcal{D}(w,N)=1 \mid w \in \mathbb{QR}_N\big] - \Pr\big[\mathcal{D}(w,N)=1 \mid w \in \mathbb{J}_N \setminus \mathbb{QR}_N\big]\Big| = \big|\varepsilon - \tfrac{1}{2}\big|$$

which must be negligible by the QR assumption. Hence, the scheme is IE-CPA secure under the QR assumption. □

In order to prove that the scheme is key-private, we need useful lemma adapted from (Ateniese and Gasti, 2009, Lemma 2). It considers the two following sets:

$$\mathbb{X}_0 = \Big\{u \in \mathbb{Z}/N\mathbb{Z} \mid \Big(\tfrac{u^2-R_i}{p}\Big) = \Big(\tfrac{u^2-R_i}{q}\Big) = 1\Big\}$$

and

$$\mathbb{X}_1 = \Big\{u \in \mathbb{Z}/N\mathbb{Z} \mid \Big(\tfrac{u^2-R_i}{p}\Big) = \Big(\tfrac{u^2-R_i}{q}\Big) = -1\Big\}\ .$$

Lemma 1. *Let* RSAGen *be a probabilistic algorithm which, given a security parameter* 1^κ*, outputs primes* p *and* q *and their product* $N = pq$*. Let also* r_i *be a random element in* $\mathbb{Z}/N\mathbb{Z}$ *and* $R_i = {r_i}^2 \bmod N$*. Then, under the quadratic residuosity assumption, the statistical distance*

$$\Big|\Pr\big[\mathcal{D}(x,R_i,N)=1 \mid x \overset{R}{\leftarrow} \mathbb{X}_0\big] - \Pr\big[\mathcal{D}(x,R_i,N)=1 \mid x \overset{R}{\leftarrow} \mathbb{X}_1\big]\Big|$$

is negligible for any probabilistic polynomial-time distinguisher $\mathcal{D}$*; the probabilities are taken over the experiment of running* $(N,p,q) \leftarrow \mathsf{RSAGen}(1^\kappa)$*, sampling* $r_i \overset{R}{\leftarrow} (\mathbb{Z}/N\mathbb{Z})^\times$*, and choosing at random* $x \in \mathbb{X}_0$ *and* $x \in \mathbb{X}_1$*.* □

Proposition 2. *The scheme is* IK-CPA *under the quadratic residuosity assumption.*

Proof. Since the scheme is already known to be IE-CPA, Halevi's sufficient condition for key-privacy (Halevi, 2005) teaches us that the scheme meets the IK-CPA notion if the statistical distance between the two distributions

$$D_0 = \big\{(\mathsf{upk}_0,\mathsf{upk}_1,\mathtt{ENCRYPT}_{\mathsf{upk}_0}(m)) \mid (\mathsf{upk}_0,\mathsf{usk}_0),(\mathsf{upk}_1,\mathsf{usk}_1) \overset{R}{\leftarrow} \mathtt{KEYGEN}(\mathsf{PP}),\ m \overset{R}{\leftarrow} \mathcal{M}\big\}$$

and

$$D_1 = \big\{(\mathsf{upk}_0,\mathsf{upk}_1,\mathtt{ENCRYPT}_{\mathsf{upk}_1}(m)) \mid (\mathsf{upk}_0,\mathsf{usk}_0),(\mathsf{upk}_1,\mathsf{usk}_1) \overset{R}{\leftarrow} \mathtt{KEYGEN}(\mathsf{PP}),\ m \overset{R}{\leftarrow} \mathcal{M}\big\}$$

is negligible. In our case, a ciphertext encrypted under key upk_b (with $b \in \{0,1\}$) is of the form $C_b = (\tau_b, c_b)$ where $\tau_b = (-1)^m \big(\frac{2t}{N}\big)$ for some t. If message m is unknown and uniformly drawn in $\{0,1\}$, τ_b does not help in distinguishing between D_0 and D_1. Only the c_b-component of C_b needs to be considered. When the public key is $\mathsf{upk}_b = R_{i_b}$ then

$$c_b^{(\beta)} := \begin{cases} c_b^{(0)} = \frac{t^2+R_{i_b}}{2t} \bmod N\text{, or} \\ c_b^{(1)} = \frac{2R_{i_b}t}{t^2+R_{i_b}} \bmod N \end{cases}$$

for some random $t \in \mathbb{Z}/N\mathbb{Z}$.

Omitting $\mathsf{upk}_0, \mathsf{upk}_1$ to ease the reading, Halevi's criterion requires that the distributions $D_0 = \{c_0^{(\beta)} \mid \beta \overset{R}{\leftarrow} \{0,1\}\}$ and $D_1 = \{c_1^{(\beta)} \mid \beta \overset{R}{\leftarrow} \{0,1\}\}$ are indistinguishable with overwhelming probability.

Write

$$\mathbb{X}_{0,b} = \Big\{u \in \mathbb{Z}/N\mathbb{Z} \mid \Big(\frac{u^2-R_{i_b}}{p}\Big) = \Big(\frac{u^2-R_{i_b}}{q}\Big) = 1\Big\}$$
$$\mathbb{X}_{1,b} = \Big\{u \in \mathbb{Z}/N\mathbb{Z} \mid \Big(\frac{u^2-R_{i_b}}{p}\Big) = \Big(\frac{u^2-R_{i_b}}{q}\Big) = -1\Big\}$$
$$\mathbb{Y}_b = \Big\{u \in \mathbb{Z}/N\mathbb{Z} \mid \Big(\frac{u^2-R_{i_b}}{p}\Big) = -\Big(\frac{u^2-R_{i_b}}{q}\Big)\Big\}$$

As shown in Section 3.2, we have $c_b^{(0)} \in \mathbb{X}_{0,b}$ and $c_b^{(1)} \in \mathbb{Y}_b$ since

$$\Big(c_b^{(0)}\Big)^2 - R_{i_b} = \left[\frac{t^2-R_{i_b}}{2t}\right]^2$$

and

$$\Big(c_b^{(1)}\Big)^2 - R_{i_b} = -R_{i_b}\left[\frac{t^2-R_{i_b}}{t^2+R_{i_b}}\right]^2$$

where $t \overset{R}{\leftarrow} \mathbb{Z}/N\mathbb{Z}$. Hence, we can see that

$$D_b \overset{c}{\equiv} \begin{cases} \{u \mid u \overset{R}{\leftarrow} \mathbb{X}_{0,b} \cup \mathbb{X}_{1,b}\} & \text{when } \beta = 0 \\ \{u \mid u \overset{R}{\leftarrow} \mathbb{Y}_b\} & \text{when } \beta = 1 \end{cases}\ .$$

The first assertion (when $\beta = 0$) follows from Lemma 1 (the notation $\overset{c}{\equiv}$ means computationally equivalent —under the QR assumption in this case). Indeed, we have $\{u \mid u \overset{R}{\leftarrow} \mathbb{X}_{0,b}\} \overset{c}{\equiv} \{u \mid u \overset{R}{\leftarrow} \mathbb{X}_{1,b}\}$. The second assertion (when $\beta = 1$) follows by noting that the Jacobi symbol $\Big(\frac{-R_{i_b}}{N}\Big) = -1$.

As a consequence, assuming the QR assumption, the distribution D_b appears indistinguishable from the uniform distribution over $\mathbb{Z}/N\mathbb{Z}$. This concludes the proof by noting that D_0 and D_1 are essentially the same sets: any random element is D_0 is also an element in D_1, and vice-versa. □

5 CONCLUSION

The cryptosystem presented in this paper can be used in any application requiring efficient public-key encryption provably secure in the standard model with strong privacy guarantees. Remarkably, it offers both data privacy *and* key privacy under the standard quadratic residuosity assumption. In addition, it features a very fast key generation and is so well suited to constrained devices requiring an on-board key generation.

REFERENCES

Ateniese, G. and Gasti, P. (2009). Universally anonymous IBE based on the quadratic residuosity assumption. In Fischlin, M., editor, *Topics in Cryptology – CT-RSA 2009*, volume 5473 of *Lecture Notes in Computer Science*, pages 32–47. Springer.

Barth, A., Boneh, D., and Waters, B. (2006). Privacy in encrypted content distribution using private broadcast encryption. In Di Crescenzo, G. and Rubin, A., editors, *Financial Cryptography and Data Security*, volume 4107 of *Lecture Notes in Computer Science*, pages 52–64. Springer.

Bellare, M., Boldyreva, A., Desai, A., and Pointcheval, D. (2001). Key-privacy in public-key encryption. In Boyd, C., editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 566–582. Springer.

Boneh, D., Gentry, C., and Hamburg, M. (2007). Space-efficient identity based encryption without pairings. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007)*, pages 647–657. IEEE Computer Society. Full version available as Cryptology ePrint Archive, Report 2007/177.

Chaum, D. and van Heyst, E. (1991). Group signatures. In Davies, D. W., editor, *Advances in Cryptology – EUROCRYPT '91*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265. Springer.

Goldwasser, S. and Micali, S. (1984). Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299.

Halevi, S. (2005). A sufficient condition for key-privacy. IACR Cryptology ePrint Archive, Report 2005/005.

Kiayias, A., Tsiounis, Y., and Yung, M. (2007). Group encryption. In Kurosawa, K., editor, *Advances in Cryptology – ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 181–199. Springer.