# The Jacobi Symbol Problem for Quadratic Congruences and Applications to Cryptography

Ferucio Laurenţiu Ţiplea*

*Department of Computer Science*
*"Alexandru Ioan Cuza" University of Iaşi*
*700505 Iaşi, Romania*

**Abstract**

The hardness of solving the quadratic residuosity problem is the basis for establishing the security of many cryptographic schemes. Two of these are the public key encryption scheme and the identity-based encryption scheme proposed by Cocks. In this paper, we introduce a new computational problem: the problem of distinguishing between the Jacobi symbols of the solutions of a quadratic congruence modulo an RSA integer. We show that the security of the two encryption schemes is equivalent to the hardness of this problem, while the quadratic residuosity problem reduces to this new problem. We then specialize the problem to roots of quadratic residues and establish several computational indistinguishability relationships.

*Keywords:* Jacobi symbol, quadratic congruence, hard problem, identity-based encryption, computational indistinguishability

## 1. Introduction

A problem is called hard if there is no probabilistic algorithm of polynomial time complexity to solve it with non-negligible probability. There is no mathematical proof for the hardness of a mathematical problem. Unsuccessful attempts to efficiently solve certain problems eventually led to the assumption that those problems are hard. Among them are the factorization or discrete logarithm problem. Notice, however, that a hardness assumption is not a mathematical argument and so, some believed-to-be-hard problems might become easy in the future.

The security of a cryptographic construction $\mathcal{S}$ is studied within some *security model* $SM$ that specifies a security goal to be achieved by $\mathcal{S}$, and an attack model against which the security goal is to be achieved. Then, $\mathcal{S}$ is $SM$*-secure* if the problem $SM(S)$ of breaking $\mathcal{S}$'s security goal through the attack model specified by $SM$ is hard. This is where the hardness assumptions and the reduction technique come into play. More exactly, to prove that $\mathcal{S}$ is $SM$-secure we do as follows:

- Choose a problem $H$ for which there is a hardness assumption;
- Reduce $H$ to $SM(\mathcal{S})$ in the sense that if breaking the $SM$-security of $\mathcal{S}$ would be easy, then $H$ becomes easy.

The conclusion then is that $\mathcal{S}$ achieves $SM$-security provided that $H$ is hard.

The *quadratic residuosity problem* (QRP) is one of the seemingly hard problems. This problem involves deciding whether an integer with the Jacobi symbol +1 is a quadratic residue. Since all attempts to solve it efficiently failed, the assumption was adopted that no probabilistic algorithm of polynomial time complexity can distinguish with a non-negligible probability between quadratic residues and quadratic non-residues with the Jacobi symbol +1. This assumption is known as the *quadratic residuosity assumption* (QRA). It and the problem of quadratic residuosity are of great importance in cryptography [18, 11, 12, 3, 8, 6, 2, 7, 13, 10, 9].

*Corresponding author
*Email address:* ferucio.tiplea@uaic.ro (Ferucio Laurenţiu Ţiplea)

*Contribution.* Cocks's public-key encryption (CPKE) and identity-based encryption (CIBE) schemes [8] are two well-known cryptographic schemes that achieve $IND\text{-}CPA$ security, provided that $QRP$ is hard. That is, $QRP$ reduces to the $IND\text{-}CPA$ security of any of the two schemes (in the sense we have already discussed: if breaking the $IND\text{-}CPA$ security of any of the two schemes is easy, then $QRP$ is easy).

The question now is whether the $IND\text{-}CPA$ security of the two Cocks' schemes reduces to $QRP$. In other words, the question is whether the $IND\text{-}CPA$ security of any of these schemes is equivalent to $QRP$. The equivalence between the security of a cryptographic scheme and a hard computational problem can have multiple advantages:

- Computational problems are usually formulated more simply, eliminating details that are not of algorithmic importance (which may appear in the description of a cryptographic scheme);
- Allows easy correlation with other computational problems;
- Provides a clearer picture of the security level of the cryptographic scheme;
- May facilitate security comparisons between cryptographic schemes.

We introduce a new hard computational problem in this paper, called the *Jacobi symbol problem for quadratic congruences* ($JSP(QC)$), and we show that:

1. The $IND\text{-}CPA$ security of any of the two Cocks' schemes is equivalent to $JSP(QC)$;
2. $QRP$ reduces to $JSP(QC)$.

The second item tells that $JSP(QC)$ is at least as hard as $QRP$. We claim that $JSP(QC)$ is, in fact, strictly harder than $QRP$.

We then specialize $JSP(QC)$ to roots of quadratic residues modulo anti-Blum integers. We divide the quadratic residues into two classes according to the Jacobi symbol of their roots, which in turn induces a partition into two classes of the integers with the Jacobi symbol $+1$ but which are not quadratic residues. We then establish computational indistinguishable relationships between these distributions. Thus, we refine the problem of distinguishing between quadratic residues and non-residues depending on the Jacobi symbol of the roots.

*Paper structure.* Our paper is structured into six sections, the first one being an introduction. The second section establishes the basic notation and terminology for the entire paper. Then, we present some results about quadratic congruences. The fourth section is dedicated to the computational problem we propose, namely the Jacobi symbol problem for quadratic congruences. Connections between this problem, the quadratic residuosity problem, and the security of Cocks' schemes are established. The fifth section specializes the Jacobi symbol problem for quadratic congruences to roots of quadratic congruences and establishes several computational indistinguishability results. The conclusions of our work are presented in the sixth section.

## 2. Preliminaries

We recall here the basic notation and terminology used in the paper. For details the reader is referred to [1, 17, 19, 20, 16].

*Number theory.* We use $\mathbb{Z}$ to denote the set of integers and $(a, b)$ for the gcd of the integers $a$ and $b$ (it will be clear from context when $(a, b)$ is of the pair of the two integers and not their gcd). When $(a, b) = 1$, the integers $a$ and $b$ are called *co-prime.* $\mathbb{Z}_n$ stands for $\{0, \ldots, n-1\}$ and $\mathbb{Z}_n^* = \{a \in \mathbb{Z}_n \mid (a, n) = 1\}$, for any positive integer $n$.

Two integers $a$ and $b$ are congruent module an integer $n$, denoted $a \equiv b \bmod n$ or $a \equiv_n b$, if $n$ divides $a - b$. When $n \neq 0$, the remainder of the integer division of $a$ by $n$ is expressed $a \bmod n$ or $(a)_n$.

An *RSA integer*, also called *RSA modulus*, is a product $n = pq$ of two distinct odd primes $p$ and $q$ (as a matter of convention, we allays assume $p < q$).

Given a system of congruences in the non-determinate $x$,

$$x \equiv b_i \bmod m_i \quad \text{for all } 1 \leq i \leq n,$$

the *Chinese Remainder Theorem* (CRT) [17, 19] states that the system has a unique solution modulo $m_1 \cdots m_n$, whenever $m_1, \ldots, m_n$ are pairwise co-prime.

Given two co-prime integers $a$ and $n$, we say that $a$ is a *quadratic residue modulo* $n$ if $a \equiv_n x^2$, for some integer $x$; the integer $x$ is called a *square root* of $a$ modulo $n$.

Given an odd prime integer $p$, the *Legendre symbol* of an integer $a$ modulo $p$, denoted $\left(\frac{a}{p}\right)$, is 1 when $a$ is a quadratic residue modulo $p$, 0 when $p$ divides $a$, and $-1$, otherwise. The extension to odd moduli $n > 0$, called the *Jacobi symbol*, is 1 when $n = 1$ and

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{e_1} \cdots \left(\frac{a}{p_m}\right)^{e_m}$$

if $n = p_1^{e_1} \cdots p_m^{e_m}$ is the prime factorization of $n$. For ease of expression, we will use the term "Jacobi symbol" both in the case of prime and composite modules.

Let $QR_n$ ($QNR_n$, $J_n^+$, $J_n^-$) be the set of quadratic residues (quadratic non-residues, integers with the Jacobi symbol $+1$, integers with the Jacobi symbol $-1$, respectively) from $\mathbb{Z}_n^*$. The following facts are well-known [1, 16, 17]:

1. $|QR_n| = |QNR_n|$ when $n$ is an odd prime integer;
2. For any integer $n > 0$, $a \in \mathbb{Z}_n^*$ is a quadratic residue modulo $n$ if and only if is is a quadratic residue modulo any prime factor of $n$;
3. For any RSA modulus $n = pq$, $|J_n^+| = |J_n^-|$ and $|QR_n| = \frac{|J_n^+|}{2}$;
4. For any RSA modulus $n = pq$, if we split $J_n^-$ into two subsets $J_n^\pm = \{a \in J_n^- \mid \left(\frac{a}{p}\right) = 1 \text{ and } \left(\frac{a}{q}\right) = -1\}$ and $J_n^\mp = \{a \in J_n^- \mid \left(\frac{a}{p}\right) = -1 \text{ and } \left(\frac{a}{q}\right) = 1\}$, then $QR_n$, $J_n^+ \setminus QR_n$, $J_n^\pm$, and $J_n^\mp$ partition $\mathbb{Z}_n^*$ into four subsets of equal size. These subsets are called the *quadrants* of $\mathbb{Z}_n^*$.

*Probabilistic algorithms. Probabilistic polynomial time* (PPT) algorithms [20] play an important role in cryptography. For such an algorithm $\mathcal{A}$, $b \leftarrow \mathcal{A}(D)$ means that $b$ is an output of $\mathcal{A}$ on some input from $D$, and $P(b \leftarrow \mathcal{A}(D))$ stands for the probability with which $\mathcal{A}$ outputs $b$. An oracle for $\mathcal{A}$ can be viewed as a black box $f$ that can perform a particular computation whenever it is queried by $\mathcal{A}$. We do not care about $f$'s implementation or how it works. We only assume that $f$ returns the computation result in $\mathcal{O}(1)$ time complexity. The notation $\mathcal{A}^f$ is used to specify that $\mathcal{A}$ may query the oracle $f$.

A positive function $f(\lambda)$ is *negligible* if for any polynomial function $poly(\lambda)$ there is $\lambda_0$ such that $f(\lambda) < 1/poly(\lambda)$, for any $\lambda \geq \lambda_0$. If $1 - f(\lambda)$ is negligible, then $f(\lambda)$ is called *overwhelming*.

When a problem cannot be solved by any PPT algorithm, except with negligible probability, we will say that it is *hard*; otherwise, it will be called *easy*. The problem $A$ *reduces* to the problem $B$, denoted $A \preceq B$, if $A$'s hardness implies $B$'s hardness (equivalent to say, assuming $B$ easy implies $A$ easy). If $A \preceq B$ and $B \preceq A$, then $A$ and $B$ are called *equivalent*, denoted $A \sim B$.

*Probability distributions and indistinguishability.* A PPT algorithm $\mathcal{A}$ that on inputs from a probability distribution $D$ outputs a bit $b \in \{0, 1\}$ is called a *distinguisher*. The *advantage* of of a PPT $\mathcal{A}$ on two families of probability distributions $X = (X_\lambda)_\lambda$ and $Y = (Y_\lambda)_\lambda$, denoted $Adv_{\mathcal{A},X,Y}(\lambda)$, is the function

$$Adv_{\mathcal{A},X,Y}(\lambda) = |P(1 \leftarrow \mathcal{A}(X_\lambda)) - P(1 \leftarrow \mathcal{A}(Y_\lambda))|$$

When $Adv_{\mathcal{A},X,Y}$ is negligible for any distinguisher $\mathcal{A}$, $X$ and $Y$ are called *computationally indistinguishable*, denoted $\overset{c}{\approx}$.

*Publc-key encryption.* A *public-key encryption* (PKE) scheme is a triple of algorithms $\mathcal{S} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$, where:

- $\mathcal{G}$ is a PPT algorithm that takes as input a security parameter $\lambda$ and outputs a pair $(pk, sk)$ consisting of a public key $pk$ and a symmetric key $sk$;
- $\mathcal{E}$ is a PPT algorithm that takes as input a public key $pk$ and a message $m$ and outputs a ciphertext;

- $\mathcal{D}$ is a *deterministic polynomial-time* (DPT) algorithm that takes as input a private key $sk$ and a ciphertext $c$ and outputs a message $m$ or a special symbol $\perp$ denoting failure. It is required that $\mathcal{D}(sk, \mathcal{E}(pk, m)) = m$, for all pairs $(pk, sk)$ output by $\mathcal{G}$ and any message $m$.

To define the $IND\text{-}CPA$ security of a PKE scheme $\mathcal{S}$, consider the following probabilistic experiment, where $\mathcal{A}$ is a PPT algorithm and $b \in \{0, 1\}$.

$IND\text{-}CPA$ experiment $PKE_{\mathcal{A},\mathcal{S}}^{cpa\text{-}b}(\lambda)$

1. $(pk, sk) \leftarrow \mathcal{G}(\lambda)$;
2. $(m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{E}}(\lambda, pk)$ with $m_0 \neq m_1$ and $|m_0| = |m_1|$;
3. $c \leftarrow \mathcal{E}(pk, m_b)$;
4. $b' \leftarrow \mathcal{A}^{\mathcal{E}}(c, \sigma)$;
5. Return $b'$.

($\sigma$ denotes state information).

We say that $\mathcal{S}$ has *indistinguishable encryptions under chosen plaintext attack* or that it is $IND\text{-}CPA$ *secure* if the advantage of $\mathcal{A}$ is negligible for any PPT $\mathcal{A}$, where

$$Adv_{\mathcal{A},\mathcal{S}}(\lambda) = |P(1 \leftarrow PKE_{\mathcal{A},\mathcal{S}}^{cpa\text{-}0}(\lambda)) - P(1 \leftarrow PKE_{\mathcal{A},\mathcal{S}}^{cpa\text{-}1}(\lambda))|.$$

We will denote by $IND\text{-}CPA(\mathcal{S})$ the problem of breaking the $IND\text{-}CPA$ security of $\mathcal{S}$ (in fact, this is the problem to distinguish between two oracles).

*Identity-based encryption* (IBE) is a form of PKE, where the public key can be computed by sender, while the corresponding private key has to be computed by a dedicated key generator. So, an IBE scheme consists of four PPT algorithms $\mathcal{S} = (Setup, \mathcal{G}, \mathcal{E}, \mathcal{D})$ as follows:

1. *Setup* is a PPT algorithm that takes as input a security parameter $\lambda$ and outputs the system public parameters $PP$ together with a master key $Msk$;
2. $\mathcal{G}$ is a PPT algorithm that takes as input an identity $ID$ together with the master key $Msk$ and outputs a private key associated to $ID$;
3. $\mathcal{E}$ is a PPT algorithm that, starting with the public parameter $PP$, an identity $ID$, and a message $m$, encrypts $m$ into some ciphertext $c$ (the encryption key is $ID$ or some binary string derived from $ID$);
4. $\mathcal{D}$ is a DPT algorithm that a ciphertext $c$ into a message or a special symbol $\perp$ (denoting failure) by using the private key associated to $ID$ (and delivered by $\mathcal{G}$).

The concept of $IND\text{-}CPA$ security can be extended to IBE schemes as well by means of the following experiment.

$IND\text{-}ID\text{-}CPA$ experiment $IBE_{\mathcal{A},\mathcal{S}}^{cpa\text{-}b}(\lambda)$

1. $(PP, Msk) \leftarrow Setup(\lambda)$;
2. $(m_0, m_1, ID) \leftarrow \mathcal{A}^{\mathcal{E},\mathcal{G}}(\lambda, PP)$ with $m_0 \neq m_1$ and $|m_0| = |m_1|$;
3. $c \leftarrow \mathcal{E}(PP, ID, m_b)$;
4. $b' \leftarrow \mathcal{A}^{\mathcal{E},\mathcal{G}}(c, \sigma)$;
5. Return $b'$.

($\sigma$ denotes state information. It is assumed that the identity $ID$ in step 3 was never queried for private key extraction in steps 2 and 4).

We say that $\mathcal{S}$ has *indistinguishable encryptions under chosen plaintext attack* or that it is $IND\text{-}ID\text{-}CPA$ *secure* if the advantage of $\mathcal{A}$ is negligible for any PPT $\mathcal{A}$, where

$$Adv_{\mathcal{A},\mathcal{S}}(\lambda) = |P(1 \leftarrow IBE_{\mathcal{A},\mathcal{S}}^{cpa\text{-}0}(\lambda)) - P(1 \leftarrow IBE_{\mathcal{A},\mathcal{S}}^{cpa\text{-}1}(\lambda))|$$

We will denote by $IND\text{-}ID\text{-}CPA(\mathcal{S})$ the problem of breaking the $IND\text{-}ID\text{-}CPA$ security of $\mathcal{S}$ (in fact, this is the problem to distinguish between two oracles).

## 3. Quadratic congruences

We present in this section some results on solving quadratic congruences modulo a prime and an RSA integer. For the completeness of the presentation, some known results are recalled and accompanied by brief proof sketches.

*3.1. Quadratic congruences modulo a prime integer*

We will focus on solving quadratic congruences

$$a_2x^2 + a_1x + a_0 \equiv 0 \; mod \; p, \tag{1}$$

where $p$ is an odd prime integer and $a_0, a_1, a_2 \in \mathbb{Z}_p$. For the congruence not to degenerate into a linear one, we will ask for $(a_2, p) = 1$. Under this requirement, we may multiply the quadratic congruence by $a_2^{-1} \bmod p$ without changing its solutions. So, we may consider the quadratic congruence in the equivalent form $x^2 + cx + a \equiv 0 \bmod p$. For technical reasons, we write the congruence in the form:

$$x^2 - cx + a \equiv 0 \bmod p, \tag{2}$$

where $a, c \in \mathbb{Z}_p$. If $a = 0$, the congruence becomes $x(x - c) \equiv 0 \bmod p$, which trivially leads to the solutions 0 and $c$ in $\mathbb{Z}_p$. As a result, we will avoid this case and, in what follows, we assume $a \in \mathbb{Z}_p^*$.

Although not presented in this form, the following result is part of any standard textbook on number theory, such as [1, 17].

**Proposition 1** (Solving quadratic congruences)**.** *Let $p$ be an odd prime integer, $a \in \mathbb{Z}_p^*$, $c \in \mathbb{Z}_p$, and $\Delta = (c^2 - 4a) \bmod p$.*

1. *If $\Delta \in QR_p$, then:*
   - *(a) The congruence (2) has two distinct solutions in $\mathbb{Z}_p^*$, namely $(c + \sqrt{\Delta})/2 \bmod p$ and $(c - \sqrt{\Delta})/2 \bmod p$, where $\sqrt{\Delta}$ is an arbitrary root modulo $p$ of $\Delta$;*
   - *(b) If $t \in \mathbb{Z}_p^*$ is one of the solutions for (2), then the other solution in $\mathbb{Z}_p^*$ is $at^{-1} \bmod p$;*
   - *(c) The two solutions in $\mathbb{Z}_p^*$ for (2), $t$ and $at^{-1} \bmod p$, satisfy*

     $$\left(\frac{at^{-1}}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{t}{p}\right).$$

     *Therefore, they have the same Jacobi symbol modulo $p$ if and only if $a \in QR_p$;*
2. *If $\Delta = 0$, then:*
   - *(a) $a \equiv (c/2)^2 \bmod p$, and so $a \in QR_p$ and $c \in \mathbb{Z}_p^*$;*
   - *(b) The congruence (2) has a (double) solution in $\mathbb{Z}_p^*$, namely $t = c/2 \bmod p$, which is also one of the two roots in $\mathbb{Z}_p^*$ of $a$;*
3. *If none of the above occurs, the congruence (2) has no solution.*

*Proof.* According to the hypothesis, the congruence (2) is equivalent to

$$4x^2 - 4cx + 4a \equiv 0 \bmod p, \tag{3}$$

which in turn can be re-written as

$$(2x - c)^2 \equiv \Delta \bmod p. \tag{4}$$

It is now clear that the congruence (2) has solutions only if $\Delta = 0$ or $\Delta \in QR_p$. This answers the last item of Proposition 1.

1. Let us assume that $\Delta$ is a quadratic residue modulo $p$. Then, (4) leads to

$$p | (2x - c - \sqrt{\Delta})(2x - c + \sqrt{\Delta}),$$

from which follows that $(c + \sqrt{\Delta})/2 \bmod p$ and $(c - \sqrt{\Delta})/2 \bmod p$ are solutions in $\mathbb{Z}_p$ for (2). It is straightforward to check that they are non-congruent modulo $p$. If we assume that $p$ divides one of them, then $p$ divides their product and so, $p|a$, which is a contradiction. Therefore, both solutions are in $\mathbb{Z}_p^*$, and thus 1(a) is proved.

1(b) requires only a simple check, and 1(c) follows from the basic properties of the Jacobi symbol.

2. If $\Delta = 0$, then $a \equiv (c/2)^2 \bmod p$, and so $a \in QR_p$ (remark that $a \in \mathbb{Z}_p^*$ by the hypothesis). Moreover, $c \in \mathbb{Z}_p^*$. Therefore, 2(a) is proved.

To prove 2(b), remark that (4) becomes $(2x - c)^2 \equiv 0 \bmod p$, which leads to the (double) solution $c/2 \bmod p$ in $\mathbb{Z}_p^*$. □

Each solvable congruence $x^2 - cx + a \equiv 0 \bmod p$ is precisely defined by:

1. The odd prime integer p;
2. $a \in \mathbb{Z}_p^*$, which is the product modulo $p$ of the solutions in $\mathbb{Z}_p^*$, including the case of a double solution;
3. $c \in \mathbb{Z}_p$, which is the sum modulo $p$ of the solutions in $\mathbb{Z}_p$, including the case of a double solution.

Therefore, we can count the solvable quadratic congruences by counting the subsets $\{t, at^{-1}\}$ with $a, t \in \mathbb{Z}_p^*$ (remark that $t \equiv at^{-1} \bmod p$ if and only if $a \in QR_p$ and $t$ is a square root of $a$ modulo $p$.

Given an odd prime $p$, $a \in \mathbb{Z}_p^*$, and $s \in \{-, +\}$, define the set

$$QC^s_{p,a} = \{c \in \mathbb{Z}_p \mid x^2 - cx + a \equiv 0 \bmod p \text{ is solvable and all its solutions have the Jacobi symbol } s\}.$$

**Proposition 2.** *Let $p$ be an odd prime integer and $a \in QR_p$. Then,*

$$||QC^+_{p,a}| - |QC^-_{p,a}|| = \begin{cases} 1, & \text{if } p \equiv 1 \bmod 4 \\ 0, & \text{otherwise.} \end{cases}$$

*Proof.* Let $a \in QR_p$. In this case, $a$ has two non-congruent roots in $\mathbb{Z}_p^*$.

Any $t_1 \in \mathbb{Z}_p^*$ that is not a root of $a$ defines uniquely a solvable quadratic congruence with the distinct solutions $t_1$ and $at_1^{-1} \bmod p$. If $t_2 \in \mathbb{Z}_p^*$ is not congruent to $t_1$ and $at_1^{-1} \bmod p$, then $at_2^{-1} \bmod p$ is not congruent to $t_1$ and $at_1^{-1} \bmod p$. So, $\{t_1, at_1^{-1}\}$ and $\{t_2, at_2^{-1}\}$ are disjoint sets that define two distinct solvable quadratic congruences.

According to Proposition 1(1c), $t_1 \in QR_p$ if and only if $at_1^{-1} \in QR_p$ and so, their sum modulo $p$ is in $QC^+_{p,a}$. Likewise, $t_1 \in QNR_p$ if and only if $at_1^{-1} \in QNR_p$ and so, their sum modulo $p$ is in $QC^-_{p,a}$.

Therefore, pairs in $QR_p$ define values $c$ in $QC^+_{p,a}$, while pairs in $QNR_p$ define values $c$ in $QC^+_{p,a}$. Moreover, $|QR_p| = |QNR_p|$.

To end the proof, it remains for us to clarify the situation of the two roots in $\mathbb{Z}_p^*$ of $a$. Each such root defines a value $c$. If $p \equiv 1 \bmod 4$, both roots have the same Jacobi symbol and, therefore, both are either in $QR_p$ or $QNR_p$. So one of the sets $QC^+_{p,a}$ or $QC^-_{p,a}$ will have an extra value $c$. If $p \equiv 3 \bmod 4$, both roots of $a$ have opposite Jacobi symbols. In this case, $QC^+_{p,a}$ or $QC^-_{p,a}$ will have the same cardinal. □

**Remark 1.** *If we do not include in $QC^+_{p,a}$ and $QC^-_{p,a}$ the values $c$ obtained from the roots of $a$, then $|QC^+_{p,a}| = |QC^-_{p,a}|$ no matter of the odd prime integer $p$ (please see the proof of Proposition 2).*

A brief discussion on the complexity of computing the solutions of a quadratic congruence modulo a prime integer concludes the section.

**Remark 2.** *The calculation of solutions for the congruence* (2) *requires first to decide whether the discriminant $\Delta$ is a quadratic residue modulo p. This can be decided in polynomial time $\mathcal{O}(\log^2 p)$ by computing the Jacobi symbol of $\Delta$ modulo p [19]. If $\Delta \in QR_p$, its roots can be computed in polynomial time $\mathcal{O}(\log^3 p + h(\log h)(\log^2 p))$, where $p - 1 = 2^h m$ for some odd $m$ [19]. This gives also the final complexity to compute the solutions.*

### *3.2. Quadratic congruences modulo a composite integer*

Solving quadratic congruences in which the modulus is a composite integer appeals to the Chinese remainder theorem (CRT) and Hensel's lifting lemma [1, 17]. In the following, we will refer only to RSA moduli that are integers of the form $n = pq$, where $p$ and $q$ are distinct odd prime integers. In addition, to be consistent with the assumption in the previous section, the free (constant) coefficient will always be co-prime with $n$. As a result, the only fundamental tool we need is CRT. According to it, solving

$$x^2 - cx + a \equiv 0 \bmod pq, \tag{5}$$

where $a \in \mathbb{Z}_{pq}^*$, is reduced to solving the congruences (2) and

$$x^2 - cx + a \equiv 0 \bmod q \tag{6}$$

and then combining their solutions by CRT. As a result, if (2) and (6) are solvable, they may have each one or two solutions in $\mathbb{Z}_p^*$ and $\mathbb{Z}_q^*$, respectively, which implies that (5) may have one, two, or four solutions in $\mathbb{Z}_{pq}^*$. Thus, if $u \in \mathbb{Z}_p^*$ is a solution for (2) and $v \in \mathbb{Z}_q^*$ is a solution for (6), then the unique modulo $pq$ solution of the system

$$\begin{cases} x \equiv u \bmod p \\ x \equiv v \bmod q \end{cases} \tag{7}$$

is a solution for (5). In addition, distinct pairs $(u, v) \in \mathbb{Z}_p^* \times \mathbb{Z}_q^*$ as above give rise to distinct solutions modulo $pq$ for (5), and all solutions modulo $pq$ for (5) are obtained in this way [1, 17].

The system (7) has exactly one solution modulo $pq$, whose form is shown below.

**Lemma 1.** *Let $p$ and $q$ be two odd and distinct prime integers, $u \in \mathbb{Z}_p$, and $v \in \mathbb{Z}_q$. Then, the unique modulo $pq$ solution for the system (7) has the form*

$$x = (ue_1 + ve_2) \bmod pq \tag{8}$$

*where $e_1 = (q^{-1} \bmod p)q$ and $e_2 = (p^{-1} \bmod q)p$. Moreover,*

$$\left(\frac{x}{p}\right) = \left(\frac{u}{p}\right), \ \left(\frac{x}{q}\right) = \left(\frac{v}{q}\right), \ \textit{and} \ \left(\frac{x}{pq}\right) = \left(\frac{u}{p}\right)\left(\frac{v}{q}\right). \tag{9}$$

*Proof.* The first part of this lemma simply follows from the Chinese remainder theorem [1, 17]. For the second part, remark that

$$e_1 \equiv \begin{cases} 1 \bmod p \\ 0 \bmod q \end{cases} \tag{10}$$

and

$$e_2 \equiv \begin{cases} 1 \bmod q \\ 0 \bmod p \end{cases} \tag{11}$$

Then, apply basic computation rules for the Jacobi symbol. □

To decide if a quadratic congruence has one, two, or four solutions, modulus factorization is not necessary.

**Lemma 2.** *Let $n = pq$ be an RSA modulus. If the congruence (5) is solvable, we can efficiently decide whether it has one, two, or four solutions in $\mathbb{Z}_n^*$ without knowing the factorization of $n$.*

*Proof.* Let $\Delta = (c^2 - 4a) \bmod n$. One can easily check that:

- The congruence (5) has exactly one solution in $\mathbb{Z}_n^*$ when $\Delta = 0$;
- The congruence (5) has exactly two solution in $\mathbb{Z}_n^*$ when $\Delta \neq 0$ but $\left(\frac{\Delta}{n}\right) = 0$;
- The congruence (5) has exactly four solution in $\mathbb{Z}_n^*$ when the first two cases are not met (remark that our hypothesis stipulates that the congruence is solvable).

The proof ends by observing that we can efficiently compute the Jacobi symbol without knowing the factorization of $n$. □

The following two propositions make beneficial connections between $a$'s residuosity and the Jacobi symbol of the solutions for (5).

**Proposition 3.** *Let $n = pq$ be an RSA modulus. Assume that the quadratic congruence* (5) *is solvable and $a \in \mathbb{Z}_n^*$. Then, $a \in QR_n$ if and only if all solutions in $\mathbb{Z}_n^*$ for* (5) *have the same Jacobi symbol.*

*Proof.* If (5) is solvable, then both $x^2 - cx + a \equiv 0 \bmod p$ and $x^2 - cx + a \equiv 0 \bmod q$ are solvable (each of them having one or two solution in $\mathbb{Z}_p^*$ and $\mathbb{Z}_q^*$, respectively). So, (5) may have one, two, or four solutions in $\mathbb{Z}_n^*$.

The solutions for (5) have the same Jacobi symbol if and only if the solutions for the congruence modulo $p$ have the same symbol Jacobi and the solutions for the congruence modulo $q$ have the same symbol Jacobi. According to Proposition 1(1c), this is equivalent to saying that $(a)_p \in QR_p$ and $(a)_q \in QR_q$, which in turn is equivalent to $a \in QR_n$. □

**Proposition 4.** *Let $n = pq$ be an RSA modulus. Assume that the quadratic congruence* (5) *is solvable and $a \in \mathbb{Z}_n^*$. Then:*

1. *$a \in QNR_n$ if and only if the congruence* (5) *has two or four non-congruent solutions in $\mathbb{Z}_n^*$, half of them having the Jacobi symbol $+1$ and the other half, $-1$.*

2. *$a \in J_n^+ \setminus QR_n$ if and only if the congruence* (5) *has four non-congruent solutions in $\mathbb{Z}_n^*$, distributed one by one in the four quadrants of $\mathbb{Z}_n^*$.*

*Proof.* If (5) is solvable, both (2) and (6) are solvable.

1. Assume that $a \in QNR_n$. Then, $(a)_p \in QNR_p$ or $(a)_q \in QNR_q$. Therefore, at least one of the two congruences (2) and (6) have two non-congruent solutions (in $\mathbb{Z}_p^*$ or $\mathbb{Z}_q^*$) of opposite Jacobi symbols (Proposition 1(1c)). The other congruence may have two solutions of opposite or the same Jacobi symbols, or it may have one solution (Proposition 1(2)). So, (5) has two or four solutions in $\mathbb{Z}_n^*$, having the distribution of Jacobi symbols as specified in the proposition.

Conversely, the hypothesis shows that at least one of the two congruences modulo $p$ and $q$ has two non-congruent solutions of opposite Jacobi symbols. Suppose that this is the congruence modulo $p$. Then $(a)_p \in QNR_p$ (Proposition 1(1c)). As with respect to $(a)_q$, this may be in $QR_q$ or $QNR_q$. As a result, $a \in QNR_n$.

To prove 2 we do a similar reasoning to that above. Remark first that each of the congruences modulo $p$ and $q$ has two non-congruent solutions (in $\mathbb{Z}_p^*$ and $\mathbb{Z}_q^*$, correspondingly) of opposite Jacobi symbols (because $(a)_p \in QNR_p$ and $(a)_q \in QNR_q$). It is then straightforward to see that CRT will generate solutions to (5) with the distribution specified in the proposition.

Vice versa, the fact that the congruence (5) has four solutions in $\mathbb{Z}_n^*$ distributed one by one in the four quadrant of $\mathbb{Z}_n^*$ shows that each of the two congruences modulo $p$ and $q$ must have two non-congruent solutions of opposite Jacobi symbols. As a result, $a$ must be in $J_n^+ \setminus QR_n$. □

Let $n = pq$ be an RSA modulus, $a \in \mathbb{Z}_n^*$, and $s \in \{-, +\}$. Extending the notation from the previous section to RSA moduli, denote by $QC_{n,a}^s$ the set

$$QC_{n,a}^s = \{c \in \mathbb{Z}_p \mid x^2 - cx + a \equiv 0 \bmod n \text{ is solvable and all its solutions have the Jacobi symbol } s\}.$$

**Proposition 5.** *Let $n = pq$ be an RSA modulus and $a \in QR_n$. Then,*

$$||QC_{n,a}^+| - |QC_{n,a}^-|| \leq 1.$$

*Proof.* Each pair of integers

$$(c_1, c_2) \in QC_{p,(a)_p}^+ \times QC_{q,(a)_q}^+ \cup QC_{p,(a)_p}^- \times QC_{q,(a)_q}^-$$

produces a unique integer $c \in QC_{n,a}^+$, and each integer $c \in QC_{n,a}^+$ comes from a single pair of integers $(c_1, c_2)$ as above (Lemma 1).

Likewise, each pair of integers

$$(c_1, c_2) \in QC_{p,(a)_p}^+ \times QC_{q,(a)_q}^- \cup QC_{p,(a)_p}^- \times QC_{q,(a)_q}^+$$

produces a unique integer $c \in QC_{n,a}^-$, and each integer $c \in QC_{n,a}^-$ comes from a single pair of integers $(c_1, c_2)$ as above.

From Proposition 2, by a simple computation, we arrive at the proposition's conclusion. □

**Remark 3.** *If we do not include in $QC^+_{p,a}$, $QC^-_{p,a}$, $QC^+_{q,a}$, and $QC^-_{q,a}$ the values $c$ obtained from the roots of $a$ (modulo $p$ and $q$, correspondingly), then $|QC^+_{n,a}| = |QC^-_{n,a}|$ (please see Remark 1).*

A brief discussion on computing the solutions for a quadratic congruence modulo an RSA integer concludes the section.

**Remark 4.** *The calculation of solutions for the congruence* (5) *requires the factorization of $n = pq$. If it can be done in polynomial time, then the solutions can be computed in polynomial time (we compute the solutions for* (2) *and* (6) *and then combine them with the CRT). However, factorization of large RSA moduli is a hard problem and no other method that avoids it is known to compute solutions for* (5).

## 4. The Jacobi symbol problem for quadratic congruences

The Jacobi symbol problem for quadratic congruences, abbreviated $JSP(QC)$, is the problem to compute the Jacobi symbol of the solutions to a solvable quadratic congruence whose free coefficient is a quadratic residue with respect to an RSA modulus. $JSP(QC)$ appears to be a hard problem in the sense that no PPT algorithm can solve it with non-negligible probability.

We formalize below $JSP(QC)$ as a distinguishing problem between two probability distributions. Let $RSA\_Gen$ be an RSA moduli generator, that is, on some input $\lambda$, it outputs $(n, p, q)$, where $p$ and $q$ are two odd distinct primes of the same size $\lambda$ and $n = pq$. In what follows, we will simple write $n \leftarrow RSA\_Gen(\lambda)$ instead of $(n, p, q) \leftarrow RSA\_Gen(\lambda)$, whenever it is not necessary to emphasize the prime integers $p$ and $q$.

We define now four families of probability distributions $\mathcal{QC}^s = (\mathcal{QC}^s_\lambda)_\lambda$ and $\mathcal{QNC}^s = (\mathcal{QNC}^s_\lambda)_\lambda$, where $s \in \{-, +\}$, as follows:

$$\mathcal{QC}^s_\lambda = \{(n, a, c) \mid n \leftarrow RSA\_Gen(\lambda),\ a \leftarrow QR_n,\ t \leftarrow J^s_n,\ c = t + at^{-1} \bmod n\}$$

$$\mathcal{QNC}^s_\lambda = \{(n, a, c) \mid n \leftarrow RSA\_Gen(\lambda),\ a \leftarrow J^+_n \setminus QR_n,\ t \leftarrow J^s_n,\ c = t + at^{-1} \bmod n\}$$

We may say that $\mathcal{QC}^s_\lambda$ is the probability distribution of solvable quadratic congruences (5) whose solutions have the same Jacobi symbol $s$ (see also Proposition 3). So, $JSP(QC)$ is the problem to distinguish between $\mathcal{QC}^+$ and $\mathcal{QC}^-$.

The probability distributions $\mathcal{QNC}^+_\lambda$ and $\mathcal{QNC}^-_\lambda$ will be technically necessary. According to Proposition 4, they are identical.

### *4.1. $JSP(QC)$ and $QRP$*

We prove here that the quadratic residuosity problem reduces to $JSP(QC)$.

Let $RSA\_Gen$ be an RSA moduli generator. This generator gives rise to two probability distributions $\mathcal{QR} = (\mathcal{QR}_\lambda)_\lambda$ and $\mathcal{QNR} = (\mathcal{QNR}_\lambda)_\lambda$ of quadratic residues and non-residues, as follows:

$$\mathcal{QR}_\lambda = \{(n, a) \mid n \leftarrow RSA\_Gen(\lambda),\ a \leftarrow QR_n\}$$

$$\mathcal{QNR}_\lambda = \{(n, a) \mid n \leftarrow RSA\_Gen(\lambda),\ a \leftarrow J^+_n \setminus QR_n\}$$

The *quadratic residuosity problem* ($QRP$) is the problem to distinguish between $\mathcal{QR}$ and $\mathcal{QNR}$ [15]. This is considered a hard problem. More precisely, the following assumption is adopted.

**Definition 1.** *We say that the* quadratic residuosity asumption *(QRA) holds for a generator $RSA\_Gen$ if the distributions $\mathcal{QR}$ and $\mathcal{QNR}$, defined by means of $RSA\_Gen$, are computationally indistinguishable.*

The following result shows that $JSP(QC)$ is harder than $QRP$.

**Theorem 1.** *$QRP \preceq JSP(QC)$.*

*Proof.* Assume that $QRA$ holds for a generator $RSA\_Gen$. Then, the following relationships hold:

$$\begin{array}{rcl}
\mathcal{QC}^+_\lambda & = & \{(n, a, c) \mid n \leftarrow RSA\_Gen(\lambda),\ a \leftarrow QR_n,\ t \leftarrow J^+_n,\ c = t + at^{-1} \bmod n\} \\
& \overset{c}{\approx} & \{(n, a, c) \mid n \leftarrow RSA\_Gen(\lambda),\ a \leftarrow J^+_n \setminus QR_n,\ t \leftarrow J^+_n,\ c = t + at^{-1} \bmod n\} \\
& = & \mathcal{QNC}^+_\lambda \\
& \equiv & \mathcal{QNC}^-_\lambda \\
& = & \{(n, a, c) \mid n \leftarrow RSA\_Gen(\lambda),\ a \leftarrow J^+_n \setminus QR_n,\ t \leftarrow J^-_n,\ c = t + at^{-1} \bmod n\} \\
& \overset{c}{\approx} & \{(n, a, c) \mid n \leftarrow RSA\_Gen(\lambda),\ a \leftarrow QR_n,\ t \leftarrow J^-_n,\ c = t + at^{-1} \bmod n\} \\
& = & \mathcal{QC}^-_\lambda
\end{array}$$

So, $\mathcal{QC}^+$ and $\mathcal{QC}^-$ are computationally indistinguishable. □

### *4.2. JSP(QC) and Cocks' PKE scheme*

In the following, we will connect $JSP(QC)$ and the *IND-CPA* security of Cocks' PKE (CPKE) scheme [8].

The CPKE scheme encrypts bits in $\{-1,+1\}$. It uses quadratic residues as public keys, while their roots are the secret keys. The scheme is presented in Figure 1. Its correctness follows easily from the congruence $c + 2r \equiv t(1 + rt^{-1})^2 \bmod n$ (please see the scheme for the meaning of the parameters).

```
$\mathcal{G}(\lambda)$:        public key: $pk = (n, a)$, where
                   $(n, p, q) \leftarrow RSA\_gen(\lambda)$
                   $a = r^2 \bmod n$ with $r \leftarrow \mathbb{Z}_n^*$
               private key: $sk = (p, q, r)$

$\mathcal{E}(pk, m)$:      $t \leftarrow \mathbb{Z}_n^*$ with $\left(\frac{t}{n}\right) = m$
               output $c = t + at^{-1} \bmod n$

$\mathcal{D}(sk, c)$:      output $m = \left(\frac{c+2r}{n}\right)$
```

Figure 1: Cocks' PKE scheme

A straightforward analysis of the scheme shows that its *IND-CPA* security is equivalent to the indistinguishability of the distributions $\mathcal{QC}^+$ and $\mathcal{QC}^-$.

**Theorem 2.** *$JSP(QC) \sim$ IND-CPA$(CPKE)$.*

### *4.3. JSP(QC) and Cocks' IBE scheme*

Cocks' IBE (CIBE) scheme [8] has a setup phase where an RSA modulus $n$, a random integer $e \in J_n^+ \setminus QR_n$, and a hash function $h$ are published. The function $h$ returns elements in $J_n^+$, whenever it is applied to identities. As $a = h(ID)$ is either a quadratic residue or an element in $J_n^+ \setminus QR_n$, exactly one of $a$ and $ea$ is a quadratic residue. So, the CIBE scheme encrypts as CPKE does, but with both "public keys", $a$ and $ea$ (Figure 2).

```
$Setup(\lambda)$:            $PP = (n, e, h)$, where
                        $(n, p, q) \leftarrow RSA\_gen(\lambda)$
                        $e \leftarrow J_n^+ \setminus QR_n$
                        $h$ hash function mapping identities to $J_n^+$
                    $Msk = (p, q)$
$\mathcal{G}(Msk, ID)$:        $a = h(ID)$
                    private key: random square root $r$ of $a$ or $ea$
$\mathcal{E}(PP, ID, m)$:      $a = h(ID)$
                    $t_0, t_1 \leftarrow \mathbb{Z}_n^*$ with $\left(\frac{t_0}{n}\right) = m = \left(\frac{t_1}{n}\right)$
                    output $c_0 = t_0 + at_0^{-1} \bmod n$ and $c_1 = t_1 + eat_1^{-1} \bmod n$
$\mathcal{D}(r, (c_0, c_1))$:     set $b \in \{0, 1\}$ such that $e^b a \equiv_n r^2$
                    output $m = \left(\frac{c_b+2r}{n}\right)$
```

Figure 2: Cocks' IBE scheme

A simple analysis of the CIBE scheme shows that its *IND-ID-CPA* security is equivalent to the indistinguishability of the distributions $\mathcal{CIBE}^+ = (\mathcal{CIBE}_\lambda^+)_\lambda$ and $\mathcal{CIBE}^- = (\mathcal{CIBE}_\lambda^-)_\lambda$ given by:

$$\begin{aligned} \mathcal{CIBE}_\lambda^s = \{(n, e, a, c_1, c_2) \mid\ & n \leftarrow RSA\_Gen(\lambda,\ e \leftarrow J_n^+ \setminus QR_n,\ a \leftarrow J_n^+, \\ & t_1, t_2 \leftarrow J_n^s,\ c_1 = t_1 + at_1^{-1} \bmod n,\ c_2 = t_2 + uat_2^{-1} \bmod n\}, \end{aligned} \tag{12}$$

where $s \in \{-, +\}$, by adversaries that are allowed to query the hash function and the private key generator.

Now, we are ready to prove the following theorem.

**Theorem 3.** *$IND$-$ID$-$CPA(CIBE) \preceq JSP(QC)$. Under the assumption that the hash function in $CIBE$ is implemented as a random oracle, the converse reduction also holds.*

*Proof.* First, assume that $JSP(QC)$ is easy and prove that $IND$-$ID$-$CPA(CIBE)$ is easy. As $QRP \preceq JSP(QC)$, the hypothesis shows that QRP is easy. So, there exists an adversary $\mathcal{A}$ that has a non-negligible advantage against $QRP$ and an adversary $\mathcal{B}$ that has a non-negligible advantage against $JSP(QC)$.

Define a distinguisher $\mathcal{D}$ that on an $IND$-$ID$-$CPA(CIBE)$ instance $(n, e, a, c_1, c_2)$, where $n \leftarrow RSA\_Gen(\lambda)$ for some $\lambda$, does as follows:

1. Run $\mathcal{A}$ to decide with non-negligible probability whether $a$ or $ea$ is a quadratic residue;
2. Run $\mathcal{B}$ on $(n, a, c_1)$ if the answer of $\mathcal{A}$ is 1 (that is, $a$ is a quadratic residue), and on $(n, ea, c_2)$, otherwise;
3. $\mathcal{D}$ outputs what $\mathcal{B}$ outputs.

(remark that $\mathcal{D}$ does not need to query any oracle for $h$ or private key generation).

Clearly, $\mathcal{D}$ has a non-negligible advantage to distinguish from which of the two distributions $\mathcal{CIBE}_\lambda^+$ or $\mathcal{CIBE}_\lambda^-$ the instance $(n, e, a, c_1, c_2)$ comes. So, $IND$-$ID$-$CPA(CIBE)$ is easy.

Vice versa, assume that $IND$-$ID$-$CPA(CIBE)$ is easy and let $\mathcal{A}$ be an adversary that has non-negligible advantage against it. Moreover, assume that the hash function used to compute public keys from identities is a random oracle.

Let $(n, a, c)$ be a $JSP(QC)$ instance, where $n \leftarrow RSA\_Gen(\lambda)$ for some $\lambda$. Recall that $a \in QR_n$. Define a distinguisher $\mathcal{B}$ that on $(n, a, c)$ does as follows:

1. $e \leftarrow J_n^+$;
2. $\bar{t} \leftarrow \mathbb{Z}_n^*$;
3. Compute $\bar{c} = \bar{t} + ea\bar{t}^{-1} \bmod n$;
4. Run $\mathcal{A}$ on $(n, e, a, c, \bar{c})$, simulating for it a random oracle for hash function $h$ and an oracle for private key calculation as follows:
   - When $\mathcal{A}$ queries $h$ on the identity $ID$ for the first time, $\mathcal{B}$ randomly generates $v \leftarrow J_n^+$ and a bit $b \leftarrow \{0, 1\}$, returns $h(ID) = e^b v^2 \bmod n$ to $\mathcal{A}$ and also stores $(ID, v, b)$ in its internal database.

     For any other $ID$ query, $\mathcal{B}$ will return the same value;
   - When $\mathcal{A}$ queries a private key for the identity $ID$ and $(ID, v, b)$ is in its database for some $v$ and $b$, $\mathcal{B}$ will return $v$, if $b = 0$, and $ev$, otherwise.

     If the $ID$ private key query is for the first time, $\mathcal{B}$ first computes $h(ID)$ as above and then answers to the private key query.

   It is quite clear that $h$ implemented in this way is a random oracle.
5. $\mathcal{B}$ returns what $\mathcal{A}$ returns.

Two cases are to be analyzed.

*Case 1:* $e \in J_n^+ \setminus QR_n$. Then, $\mathcal{B}$ has the same probability $\mathcal{A}$ has to guess the Jacobi symbol of the solutions.

*Case 2:* $e \in QR_n$. Then, $\mathcal{B}$ has the probability 1/2 to guess the Jacobi symbol of the solutions because each of them is equally probable.

Therefore,

$$\begin{aligned}
Adv_{\mathcal{B},\mathcal{QC}^+,\mathcal{QC}^-}(\lambda) &= 2\left|P(s \leftarrow \mathcal{B}(\mathcal{QC}_\lambda^s) \mid s \leftarrow \{-,+\}) - \tfrac{1}{2}\right| \\
&= 2\,|P(s \leftarrow \mathcal{B}(\mathcal{QC}_\lambda^s) \mid s \leftarrow \{-,+\},\ \text{Case\_1})P(\text{Case\_1}) \\
&\quad + P(s \leftarrow \mathcal{B}(\mathcal{QC}_\lambda^s) \mid s \leftarrow \{-,+\},\ \text{Case\_2})P(\text{Case\_2}) - \tfrac{1}{2}| \\
&= 2\left|\tfrac{1}{2}P(s \leftarrow \mathcal{A}(\mathcal{CIBE}_\lambda^s) \mid s \leftarrow \{-,+\}) + \tfrac{1}{2}\cdot\tfrac{1}{2} - \tfrac{1}{2}\right| \\
&= \left|P(s \leftarrow \mathcal{A}(\mathcal{CIBE}_\lambda^s) \mid s \leftarrow \{-,+\}) - \tfrac{1}{2}\right| \\
&= \tfrac{1}{2}Adv_{\mathcal{A},\mathcal{CIBE}^+,\mathcal{CIBE}^-}(\lambda).
\end{aligned}$$

So, $\mathcal{B}$ has a non-negligible advantage against $JSP(QC)$, showing that this problem is easy. □

## 5. The Jacobi symbol problem for square roots

We specialize the results from the previous section to square roots of $a \in QR_n$ or, equivalently, solutions to the congruence

$$x^2 - a \equiv 0 \bmod n \tag{13}$$

But for that, we need a little discussion on the integer -1.

**Remark 5.** *It is well-known that, given an odd prime $p$, $-1 \in QR_p$ if and only if $p \equiv 1 \bmod 4$ [17]. Based on this, the following equivalences can easily be established:*

1. *For any odd positive integer $n > 2$, $-1 \in QR_n$ if and only if $p \equiv 1 \bmod 4$, for any prime factor $p$ of $n$.*

   *Therefore, if at least one prime factor of $n$ is congruent to 3 modulo 4, $-1$ is not a quadratic residue modulo $n$.*

2. *For any RSA modulus $n = pq$, $-1 \in J_n^+ \setminus QR_n$ if and only if $p, q \equiv 3 \bmod 4$.*

*RSA moduli $n = pq$ with the property $p, q \equiv 3 \bmod 4$ are called* Blum integers *[4, 5, 14]. To have appropriate terminology for the opposite case, we refer to the RSA moduli $n = pq$ with $p, q \equiv 1 \bmod 4$, as* anti-Blum integers*.*

**Remark 6.** *Let $n = pq$ be an RSA modulus and $a \in QR_n$. Then, from Remark 5 we obtain the following properties:*

1. *$-a \in QR_n$ if and only if $n$ is an anti-Blum integer;*
2. *$-a \in J_n^+ \setminus QR_n$ if and only if $n$ is a Blum integer.*

Now, from Propositions 3 and 4, and Remark 6 we obtain the following result.

**Corollary 1.** *Let $n = pq$ be an RSA modulus and $a \in QR_n$.*

1. *All four roots of $a$ modulo $n$ have the same Jacobi symbol if and only if $n$ is an anti-Blum integer.*
2. *The four roots of $a$ modulo $n$ are distributed one by one in the four quadrants of $\mathbb{Z}_n^*$ if and only if $n$ is a Blum integer.*

Given $n$ an anti-Blum integer and $s \in \{-, +\}$, define the following set of quadratic residues modulo $n$:

$$QR_n^s = \{a \in QR_n | (\exists t \in J_n^s)(a \equiv t^2 \bmod n)\}.$$

As $n$ is an anti-Blum integer, all roots of $a \in QR_n^s$ have the same Jacobi symbol $s$.

**Proposition 6.** *Let $n$ be an anti-Blum integer. Then, the following properties hold:*

1. *If $a, b \in QR_n^+$ or $a, b \in QR_n^-$, then $(ab)_n \in QR_n^+$;*
2. *If $a \in QR_n^+$ and $b \in QR_n^-$, then $(ab)_n \in QR_n^-$;*
3. *If $a \in QR_n^s$, then $(a^{-1})_n \in QR_n^s$, for any $s \in \{-, +\}$;*
4. *$QR_n^+$ and $QR_n^-$ are disjoint, have the same cardinality, and their union is $QR_n$.*

*Proof.* 1 and 2 follow easily from the definition of the sets $QR_n^+$ and $QR_n^-$.

3. Let $a \in QR_n^s$ and $s \in \{-, +\}$. If $t \in J_n^s$ is a root of $a$ modulo $n$, $(t^{-1})_n$ is a root of $a^{-1}$ modulo $n$. Moreover, $\left(\frac{t^{-1}}{n}\right) = \left(\frac{t}{n}\right) = s$. So, $(a^{-1})_n \in QR_n^s$.

4. Directly from the definition follows that $QR_n^+$ and $QR_n^-$ are disjoint, and their union is $QR_n$. To prove that they have the same cardinality, remark that $|J_n^+| = |J_n^-|$ and exactly four integers from $J_n^s$ define a distinguished integer in $QR_n^s$, for any $s \in \{-, +\}$. □

Given $b \in \mathbb{Z}_n^*$ and $s \in \{-, +\}$, define the set $b \cdot QR_n^s$ by

$$b \cdot QR_n^s = \{(ba)_n | a \in QR_n^s\}.$$

**Proposition 7.** *Let $n$ be an anti-Blum integer. Then, the following properties hold:*

1. *The sets $b \cdot QR_n^+$ and $b \cdot QR_n^-$ are disjoint, have the same cardinality, and their union is $J_n^+ \setminus QR_n$, for any $b \in J_n^+ \setminus QR_n$.*
2. *$b_1 \cdot QR_n^s = b_2 \cdot QR_n^s$, for any $b_1, b_2 \in J_n^+ \setminus QR_n$ with $(b_1 b_2)_n \in QR_n^+$ and any $s \in \{-,+\}$.*
3. *$b_1 \cdot QR_n^+ = b_2 \cdot QR_n^-$, for any $b_1, b_2 \in J_n^+ \setminus QR_n$ with $(b_1 b_2)_n \in QR_n^-$.*

*Proof.* 1. It is trivial to check that the two sets are disjoint and their union is $J_n^+ \setminus QR_n$, for any $b \in J_n^+ \setminus QR_n$. It is also immediately verified that $|b \cdot QR_n^s| = |QR_n^s|$, for any $s \in \{-,+\}$. As $|QR_n^+| = |QR_n^-|$ (Proposition 6(4)), it follows that $|b \cdot QR_n^+| = |b \cdot QR_n^-|$.

2. Let $b_1, b_2 \in J_n^+ \setminus QR_n$ with $(b_1 b_2)_n \in QR_n^+$ and $s \in \{-,+\}$. We show that for any $a_1 \in QR_n^s$ there exists $a_2 \in QR_n^s$ such that $b_1 a_1 \equiv_n b_2 a_2$. This will prove that $b_1 \cdot QR_n^s \subseteq b_2 \cdot QR_n^s$, and the converse inclusion would follow a similar proof line.

Indeed, if we take $a_2 = b_1 b_2^{-1} a_1 \bmod n$ we obtain $b_1 a_1 \equiv_n b_2 a_2$. Therefore, we only need to prove that $a_2 \in QR_n^s$. But that comes down to showing that $(b_1 b_2^{-1})_n \in QR_n^+$. The congruence

$$b_1 b_2^{-1} \equiv b_1 b_2 (b_2^{-1})^2 \bmod n$$

shows that $t b_2^{-1} \bmod n$ is a root of $b_1 b_2^{-1}$ modulo $n$, for any root $t$ of $b_1 b_2$ modulo $n$. As

$$\left(\frac{t b_2^{-1}}{n}\right) = \left(\frac{t}{n}\right)\left(\frac{b_2^{-1}}{n}\right) = 1 \cdot 1 = 1,$$

it follows that $(b_1 b_2^{-1})_n \in QR_n^+$.

3. The proof is similar to that in item 2, except that this time we will prove that $(b_1 b_2^{-1})_n \in QR_n^-$. □

**Example 1.** *Let $p = 5$ and $q = 13$. Then, $n = 65$ is an anti-Blum integer. The set $QR_n$ has 12 integers, distributed as follows:*

$$\begin{aligned} QR_n^+ &= \{1, 4, 16, 49, 61, 64\} \\ QR_n^- &= \{9, 14, 29, 36, 51, 56\} \end{aligned}$$

*If we take $b_1 = 7 \in J_n^+ \setminus QR_n$, we obtain:*

$$\begin{aligned} 7 \cdot QR_n^+ &= \{7, 18, 28, 37, 47, 58\} \\ 7 \cdot QR_n^- &= \{2, 8, 32, 33, 57, 63\} \end{aligned}$$

*As $b_2 = 8 \in J_n^+ \setminus QR_n$ and $(b_1 b_2)_n \in QR_n^-$, $8 \cdot QR_n^+ = 7 \cdot QR_n^-$ and $8 \cdot QR_n^- = 7 \cdot QR_n^+$.*

Given $n$ an anti-Blum integer, the set $J_n^+$ is partitioned into 4 equally sized subsets as shown in Figure 3. The subsets $b \cdot QR_n^+$ and $b \cdot QR_n^-$ can change each other depending on $b$ and the source from where they come ($QR_n^+$ or $QR_n^-$), but not as content (Proposition 7(3)).

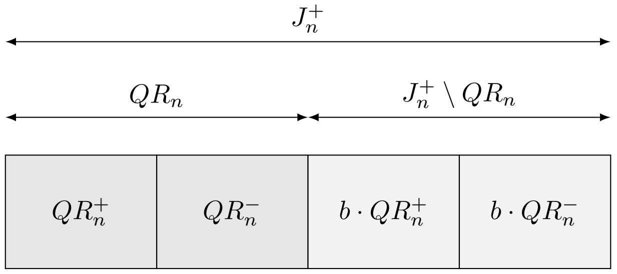

Figure 3: Partition of $J_n^+$, for some $b \in J_n^+ \setminus QR_n$

We now introduce the *Jacobi symbol problem for square roots*, abbreviated $JSP(SR)$, as the problem to compute the Jacobi symbol of the square roots of a quadratic residue modulo an anti-Blum integer. The problem can be formalized as a distinguishing problem between two probability distributions.

Let *aBlum_Gen* be an anti-Blum integer generator. Define two families of probability distributions $\mathcal{QR}^s = (\mathcal{QR}^s_\lambda)_\lambda$, where $s \in \{-,+\}$, as follows:

$$\mathcal{QR}^s_\lambda = \{(n,a) \mid n \leftarrow aBlum\_Gen(\lambda),\, a \leftarrow QR^s_n\}$$

So, $JSP(SR)$ is the problem to distinguish between $\mathcal{QR}^+$ and $\mathcal{QR}^-$.

It is believed that $QRP$ is hard even for Blum integers. There is no argument that $QRP$ would be easy for anti-Blum integers. As a result, we will postulate that this sub-problem of $QRP$, abbreviated $aBQRP$, is also hard. Similar assumptions to $QRA$ (Definition 1) can be formulated for Blum and anti-Blum generators.

The partition $J^+_n$ in Figure 3 allows us to refine the problem of distinguishing between quadratic residues and non-residues depending on the Jacobi symbol of the roots.

Let $\mathfrak{b} = (b_n)_n$ be a sequence of integers with the property $b_n \in J^+_n \setminus QR_n$, whenever $n$ is an anti-Blum integer. Define another two families of probability distributions $\mathfrak{b} \cdot \mathcal{QR}^s = (\mathfrak{b} \cdot \mathcal{QR}^s_\lambda)_\lambda$, where $s \in \{-,+\}$, as follows:

$$\mathfrak{b} \cdot \mathcal{QR}^s_\lambda = \{(n,(b_n a)_n) \mid n \leftarrow aBlum\_Gen(\lambda),\, a \leftarrow QR^s_n\}.$$

Then, the following results follow immediately.

**Proposition 8.** *Let $\mathfrak{b} = (b_n)_n$ be a sequence of integers as above and $s, s' \in \{-,1\}$. Then, the following properties hold:*

1. $\mathcal{QR}^+ \overset{c}{\approx} \mathcal{QR}^-$ *if and only if* $\mathfrak{b} \cdot \mathcal{QR}^+ \overset{c}{\approx} \mathfrak{b} \cdot \mathcal{QR}^-$*;*
2. *If* $\mathcal{QR}^s \overset{c}{\approx} \mathfrak{b} \cdot \mathcal{QR}^{s'}$ *then* $\mathcal{QR} \overset{c}{\approx} \mathcal{QNR}$.

We believe that the converse of Proposition 8(2) also holds.

## 6. Conclusions

When the free coefficient of a quadratic congruence modulo an RSA integer is a quadratic residue, all solutions of the congruence have the same Jacobi symbol. Distinguishing between the two possible Jacobi symbols without knowing the factorization of the modulus appears to be a hard problem. We called it the Jacobi symbol problem of quadratic congruences ($JSP(QC)$) and showed that the following hold:

$$\begin{aligned} QRP &\preceq JSP(QC) \\ &\sim IND\text{-}CPA(CPKE) \\ &\sim IND\text{-}ID\text{-}CPA(CIBE) \end{aligned}$$

(the random oracle model is needed for the last equivalence).

We believe that $JSP(QC)$ is strictly harder than $QRP$.

Specializing $JSP(QC)$ to congruences $x^2 - a \equiv_n 0$, where $n$ is an anti-Blum integer and $a$ is a quadratic residue, we obtain the Jacobi symbol problem for quadratic residues ($JSP(QR)$). $QR_n$ is then partitioned into two subsets of quadratic residues whose roots have the Jacobi symbol +1 ($QR^+_n$) and quadratic residues whose roots have the Jacobi symbol -1 ($QR^-_n$). This partition induces a corresponding partition on $J^+_n \setminus QR_n$. $QRP$ can then be nuanced, taking into account the Jacobi of the roots.

## References

[1] Apostol, T. M. (1976). *Introduction to Analytic Number Theory.* Undergraduate Texts in Mathematics. Springer-Verlag, New York.

[2] Ateniese, G. and Gasti, P. (2009). Universally anonymous ibe based on the quadratic residuosity assumption. In *Proceedings of the The Cryptographers' Track at the RSA Conference 2009 on Topics in Cryptology*, CT-RSA '09, pages 32–47, Berlin, Heidelberg. Springer-Verlag.

[3] Blum, L., Blum, M., and Shub, M. (1986). A simple unpredictable pseudo-random number generator. *SIAM J. Comput.*, 15(2):364–383.

[4] Blum, M. (1981). Coin flipping by telephone. In Gersho, A., editor, *Advances in Cryptology: A Report on CRYPTO 81, CRYPTO 81, IEEE Workshop on Communications Security, Santa Barbara, California, USA, August 24-26, 1981*, pages 11–15. U. C. Santa Barbara, Dept. of Elec. and Computer Eng., ECE Report No 82-04.

[5] Blum, M. (1983). Coin flipping by telephone a protocol for solving impossible problems. *SIGACT News*, 15(1):23–27.

[6] Boneh, D., Gentry, C., and Hamburg, M. (2007). Space-efficient identity based encryptionwithout pairings. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 647–657, Washington, DC, USA. IEEE Computer Society.

[7] Clear, M., Tewari, H., and McGoldrick, C. (2014). Anonymous IBE from quadratic residuosity with improved performance. In *AFRICACRYPT 2014*, volume 8469 of *Lecture Notes in Computer Science*, pages 377–397. Springer.

[8] Cocks, C. (2001). An identity based encryption scheme based on quadratic residues. In *Proceedings of the 8th IMA International Conference on Cryptography and Coding*, pages 360–363, London, UK, UK. Springer-Verlag.

[9] Ţiplea, F. L. (2021). A brief introduction to quadratic residuosity based cryptography. *Rev. Roumaine Math. Pures Appl.*, 66:793–811.

[10] Ţiplea, F. L., Iftene, S., Teşeleanu, G., and Nica, A.-M. (2020). On the distribution of quadratic residues and non-residues modulo composite integers and applications to cryptography. *Applied Mathematics and Computation*, 372:124993.

[11] Goldwasser, S. and Micali, S. (1982). Probabilistic encryption and how to play mental poker keeping secret all partial information. In *STOC 1982*, pages 365–377. ACM.

[12] Goldwasser, S. and Micali, S. (1984). Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299.

[13] Joye, M. (2016). Identity-based cryptosystems and quadratic residuosity. In *PKC 2016*, volume 9614 of *Lecture Notes in Computer Science*, pages 225–254. Springer.

[14] Jr. Kaliski, B. (2011). *Blum Integer*, pages 159–160. Springer US, Boston, MA.

[15] Kaliski, B. (2011). *Quadratic Residuosity Problem*, pages 1003–1003. Springer US, Boston, MA.

[16] Katz, J. and Lindell, Y. (2021). *Introduction to Modern Cryptography*. CRC Press, New York, 3rd edition.

[17] Nathanson, M. B. (2000). *Elementary Methods in Number Theory*, volume 195 of *Graduate Texts in Mathematics*. Springer-Verlag, New York.

[18] Rabin, M. O. (1979). Digitalized signatures and public-key functions as intractable as factorization. Technical report, MIT.

[19] Shoup, V. (2008). *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press.

[20] Sipser, M. (2012). *Introduction to the Theory of Computation*. Cengage Learning.