Some Connections Between Primitive Roots and Quadratic Non-Residues Modulo a Prime

Sorin Iftene
Department of Computer Science
"Al. I. Cuza" University
Iasi, Romania
Email: siftene@info.uaic.ro

***Abstract*—In this paper we present some interesting connections between primitive roots and quadratic non-residues modulo a prime. Using these correlations, we improve the existing randomized algorithm for generating primitive roots and we propose a polynomial deterministic algorithm for generating primitive roots for primes with special forms (for example, for safe primes). The key point of our improvement is the fact that the evaluation of Legendre-Jacobi symbol is much faster than an exponentiation.**

***Index Terms*—primitive roots, Legendre-Jacobi symbol, quadratic non-residues, square roots.**

I. Introduction

Generating primitive roots modulo a prime is an fundamental problem in number theory, with major applications in cryptography. Diffie-Hellman key establishment scheme [1], ElGamal public-key cryptosystem [2], Schnorr identification scheme [3] and Digital Signature Scheme [4] are only a few examples which rely on generating primitive roots or elements of a certain order. Finding quadratic non-residues modulo a prime is another interesting problem in number theory. Tonelli-Shanks algorithm ([5], [6]) and Cippola-Lehmer algorithm ([7], [8]) for computing square roots modulo a prime and Goldwasser-Micali probabilistic encryption scheme [9] are the most important applications that rely on generating quadratic non-residues.

In this paper we discuss comparatively these two issues and we improve the existing randomized algorithm for generating primitive roots. Moreover, we propose polynomial deterministic algorithms for generating primitive roots for primes with special forms. The paper is structured as follows. Section 2 is dedicated to some mathematical preliminaries on primitive roots and quadratic non-residues. Section 3 presents some connections between primitive roots and quadratic non-residues modulo a prime. Generating primitive roots is discussed in Section 4. The last section concludes the paper.

II. Mathematical Background

In this section we present some basic facts on number theory, focusing on primitive roots and quadratic non-residues. For more details, the reader is referred to [10], [11], [12], [13]. Computational aspects can be found in [14].

A. The order of an element. Primitive roots

Definition 1: Let $m \geq 2$ and $a \in \mathbf{Z}_m^*$. The *order* of a modulo m, denoted by $ord_m(a)$, is defined as

$$ord_m(a) = min(\{l \in \mathbf{N}^* | a^l \equiv 1 \; mod \; m\})$$

The most important properties of the order of an element are summarized in Proposition 1.

Proposition 1: Let $m \geq 2$, $a \in \mathbf{Z}_m^*$, and k, l some integers.

1) If $a^k \equiv 1 \; mod \; m$ then $ord_m(a) | k$. As a particular case, we obtain $ord_m(a) | \phi(m)$, where ϕ denotes *Euler's totient function*;
2) The relation $a^k \equiv a^l \; mod \; m$ is equivalent with
$$k \equiv l \; mod \; ord_m(a);$$
3) The next relation holds true
$$ord_m(a^k \; mod \; m) = \frac{ord_m(a)}{(ord_m(a), k)}.$$

Definition 2: Let $m \geq 2$ and $\alpha \in \mathbf{Z}_m^*$. The element α is called *primitive root modulo* m if $ord_m(\alpha) = \phi(m)$.

Remark 1: In case that α is a primitive root modulo m, every element β from the set $\mathbf{Z}_m^*$ can be uniquely expressed as $\beta = \alpha^i \; mod \; m$, for $i \in \mathbf{Z}_{\phi(m)}$. The value i will be referred to as the *discrete logarithm (modulo* m*) to the base* α *of* β and we will write $i = log_\alpha \beta$. While an expression of form $\beta = \alpha^i \; mod \; m$ can be efficiently computed given α, i, m (see, for example, [14]), the problem of finding the discrete logarithm modulo m to the base α of β, given α, β, m is intractable.

The most important properties of the primitive roots are presented in Proposition 2.

Proposition 2: Let $m \geq 2$, $\alpha \in \mathbf{Z}_m^*$. Then

1) $\mathbf{Z}_m^*$ has primitive roots if and only if $m \in \{2, 4, p^k, 2p^k\}$, where p is an odd prime and $k \geq 1$ (Gauss' Theorem);
2) If $\mathbf{Z}_m^*$ has primitive roots, then there are exactly $\phi(\phi(m))$ primitives roots modulo m;
3) If $\mathbf{Z}_m^*$ has primitive roots, α is a primitive root modulo m if and only if the next relation holds:
$$\alpha^{\frac{\phi(m)}{r}} \not\equiv 1 \; mod \; m,$$
for any prime divisor r of $\phi(m)$.

Proposition 2(3) does not always allow to efficiently generate primitive roots modulo m because computing $\phi(m)$ and factoring $\phi(m)$ are intractable for large integers m.

Elements of order q may be generated via primitive roots. More exactly, if $q|\phi(m)$, α is a primitive root modulo m, and $\beta = \alpha^{\frac{\phi(m)}{q}} \texttt{ mod } m$, then $ord_m(\beta) = q$. Indeed, by the Proposition 1(3), $ord_m(\alpha^{\frac{\phi(m)}{q}} \texttt{ mod } m) = \frac{ord_m(\alpha)}{(ord_m(\alpha), \frac{\phi(m)}{q})} = \frac{\phi(m)}{(\phi(m), \frac{\phi(m)}{q})} = q$.

B. Quadratic (non-)residues. Square roots

Definition 3: Let p be a prime and $a \in \mathbf{Z}_p^*$. We say that a is a *quadratic residue modulo* p if there exists $b \in \mathbf{Z}_p^*$ with the property $a = b^2 \ mod \ p$. Otherwise, a is a *quadratic non-residue modulo* p.

For the simplicity of the notation, from this point forward we will omit the modular reduction modulo p but the reader must be aware that all computations are performed modulo p if not explicitly stated otherwise.

If $b^2 = a$ then b will be referred to as a *square root of* a. We have to remark that if a is a quadratic residue modulo p, p odd prime, then a has exactly two square roots - if b is a square root of a, then $p - b$ is the other one. In particular, 1 has the square roots 1 and (-1) (in this case, (-1) will be regarded as being $p - 1$) or, equivalently, $a^2 = 1 \Leftrightarrow (a = 1 \vee a = (-1))$.

Definition 4: The *Legendre symbol* of a modulo p, denoted as $\left(\frac{a}{p}\right)$, is defined to be equal to ± 1 depending on whether a is a quadratic residue modulo p. More exactly,

$$\left(\frac{a}{p}\right) = \begin{cases} 1, & \text{if } a \text{ is a quadratic residue mod } p; \\ -1, & \text{otherwise.} \end{cases}$$

The *Jacobi symbol* is a generalization of the Legendre symbol to arbitrary moduli and is defined as (the same notation is used):

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{e_1} \left(\frac{a}{p_2}\right)^{e_2} \cdots \left(\frac{a}{p_k}\right)^{e_k},$$

for any odd positive integer n, $a \in \mathbf{Z}_n^*$, where $p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ is the prime factorization of n.

The most important properties of the Legendre-Jacobi symbol are summarized in Proposition 3 (for more details, see [12], [13]).

Proposition 3:
1) (Euler's criterion) For any prime p and $a \in \mathbf{Z}_p^*$, the following relation holds true:
$$a^{\frac{p-1}{2}} = \left(\frac{a}{p}\right);$$
2) $\left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}}$, that implies that (-1) is a quadratic non-residue modulo p if and only if $p \equiv 3 \ mod \ 4$;
3) $\left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}}$, that implies that 2 is a quadratic non-residue modulo p if and only if $p \equiv \pm 3 \ mod \ 8$;
4) $\left(\frac{ab}{n}\right) = \left(\frac{a}{n}\right)\left(\frac{b}{n}\right)$;
5) $\left(\frac{a}{n}\right) = \left(\frac{b}{n}\right)$ if and only if $a \equiv b \ mod \ n$; in particular, $\left(\frac{a}{n}\right) = \left(\frac{a \ mod \ n}{n}\right)$;
6) (*law of quadratic reciprocity*) $\left(\frac{a}{n}\right) = \left(\frac{n}{a}\right)(-1)^{\frac{a-1}{2} \cdot \frac{n-1}{2}}$, for any coprime odd positive integers a, n.

These properties lead to the following algorithm for computing the Legendre-Jacobi symbol ([12, page 113]):

```
LegendreJacobi ($a,n$)

input:    $n$, an odd positive integer, $a \in \mathbf{Z}_n^*$ ;
output:   $\left(\frac{a}{n}\right)$;
begin
   $t := 1$;
   while $a \neq 0$ do
      while $a \ mod \ 2 = 0$ do
         $a := a \ div \ 2$;
         if $n \equiv \pm 3 \ mod \ 8$ then $t := -t$;
      swap($a$,$n$);
      if ($a \equiv 3 \ mod \ 4$ and $n \equiv 3 \ mod \ 4$) then $t := -t$;
      $a := a \ mod \ n$;
   return( $t$)
end.
```

Fig. 1: An algorithm for computing the Legendre-Jacobi symbol

The complexity of this algorithm is $\mathcal{O}((log_2(a))(log_2(n)))$.

Euler's criterion also provides a method of computing the Legendre symbol of a modulo p using an exponentiation modulo p, whose complexity is $\mathcal{O}((log_2(p))^3)$ (see, for example, [14]). There are faster methods for evaluating the Legendre-Jacobi symbol - see, for example, [15], in which are presented algorithms of complexity $\mathcal{O}(\frac{(log_2(p))^2}{log_2(log_2(p))})$ for computing the Legendre-Jacobi symbol modulo p. Thus, is important to remark that the evaluation of the Legendre-Jacobi symbol is much faster than an exponentiation.

It is known (see, for example, [16][Remark 2.151]) that for an odd prime p, half of the elements in $\mathbf{Z}_p^*$ are quadratic non-residues modulo p. Still, no deterministic polynomial algorithm is known for finding a quadratic non-residue modulo a prime. A randomized algorithm for finding a quadratic non-residue modulo a prime p is to simply generate random elements in $\mathbf{Z}_p^*$ until one is found having its Legendre-Jacobi symbol equal with -1. The expected number iterations before a quadratic non-residue modulo p is found is two. An algorithm for generating a quadratic residue is presented next.

```
GenerateQuadraticNonResidue($p$)
input:    $p$, an odd prime;
output:   a quadratic non-residue modulo $p$;
begin
   if $p \equiv 3 \ mod \ 4$ then return($-1$)
   if $p \equiv 5 \ mod \ 8$ then return($2$)
   repeat
      generate randomly $a \in \mathbf{Z}_p^*$
   until LegendreJacobi($a,p$)=$-1$
   return($a$)
end.
```

Fig. 2: An algorithm for generating a quadratic non-residue modulo a prime p

Computing square roots modulo a prime is another fundamental problem in number theory, with major applications as primality testing, factorization or elliptic point compression. According to Bach and Shallit [12, Notes on Chapter 7, page 194] and Lemmermeyer [13, Exercise 1.16, Page 29], Lagrange was the first one who has stated an explicit formula for computing square roots in the case $p \equiv 3 \ mod \ 4$ in 1769. According to the same sources ([12, Exercise 1, page 188] and [13, Exercise 1.17, Page 29]), the case $p \equiv 5 \ mod \ 8$ has been solved by Legendre in 1785. Atkin [17] has also found a simple solution for the case $p \equiv 5 \ mod \ 8$ in 1992. For the rest of the cases, Tonelli-Shanks algorithm ([5], [6]) and Cippola-Lehmer algorithm ([7], [8]) can be used, these algorithms relying on a quadratic non-residue modulo p. An algorithm for finding the set of square roots modulo a prime p, for a given element $a \in \mathbf{Z}_p^*$, is sketched next.

```
SquareRoots($p,a$)
input:    $p$, an odd prime, $a \in \mathbf{Z}_p^*$ ;
output:   the set of the square roots modulo $p$ of $a$;
begin
   if LegendreJacobi($a,p$) $= -1$ then return($\emptyset$)
   if $a = 1$ then return($\{1, -1\}$)
   if $p \equiv 3 \ mod \ 4$ then return($\{a^{\frac{p+1}{4}}, p - a^{\frac{p+1}{4}}\}$)
   if $p \equiv 5 \ mod \ 8$ then
      begin
         $u := (2a)^{\frac{p-5}{8}}$;
         $v := 2au^2$;
         $b = ua(v-1)$;
         return($\{b, p-b\}$)
      end
   call Tonelli-Shanks or Cippola-Lehmer
end.
```

Fig. 3: An algorithm for finding the set of square roots

This algorithm can be recursively called for obtaining the set of the solutions of the equation $x^{2^s} = 1$, for a given $s \geq 2$.

III. Primitive Roots versus Quadratic Non-Residues (Modulo a Prime)

In this section we discuss some interesting connections between these two topics. Some of these correlations will be exploited in the next section in order to develop more efficient algorithms for generating primitive roots modulo a prime.

Proposition 4: Let p be an odd prime and α a primitive root modulo p. Then α is a quadratic non-residue modulo p.

Proof: Using Proposition 2(3), we obtain that $\alpha^{\frac{p-1}{r}} \neq 1$ for any prime divisor r of $(p-1)$. For $r = 2$ ($(p-1)$ is even) we have that $\alpha^{\frac{p-1}{2}} \neq 1$ (this is equivalent with $\alpha^{\frac{p-1}{2}} = -1$ because $(\alpha^{\frac{p-1}{2}})^2 = 1$ by Fermat's Theorem and 1 has the only square roots 1 and -1 modulo a prime) which, in adition with Euler's criterion (Prop 3(1)) leads to the fact that α is a quadratic non-residue modulo p. ■

Proposition 5: Let p be an odd prime and α a quadratic non-residue modulo p. Then α is a primitive root modulo p if and only if $\alpha^{\frac{p-1}{r}} \neq 1$ for any odd prime divisor r of $(p-1)$.

Proof: Directly from Proposition 2(3) and Euler's criterion (Prop 3(1)). ■

Directly from Proposition 4 and Proposition 5 we obtain:

Proposition 6: Let p be an odd prime such that there is $s \geq 1$ so that $p - 1 = 2^s$ and $\alpha \in \mathbf{Z}_p^*$. Then α is a primitive root modulo p if and only if α is a quadratic non-residue modulo p.

Proposition 7: Let p be an odd prime and $a, \alpha \in \mathbf{Z}_p^*$. The following relations hold true:

1) If a is a quadratic non-residue modulo p and $k \in \mathbf{Z}$ then a^k is a quadratic non-residue modulo p if and only if k is odd;
2) If α is a primitive root modulo p and $k \in \mathbf{Z}$ then α^k is a primitive root modulo p if and only if $(k, p-1) = 1$.

Proof:

1) It follows directly from the property $\left(\frac{a^k}{p}\right) = \left(\frac{a}{p}\right)^k = \left(\frac{a}{p}\right)^{k \ mod \ 2}$;
2) From Proposition 1(3) we obtain that $ord_p(\alpha^k) = \frac{ord_p(\alpha)}{(ord_p(\alpha),k)} = \frac{p-1}{(k,p-1)}$ (if α is a primitive root then its order is $\phi(p) = p-1$). Thus, α^k is a primitive root if and only if $ord_p(\alpha^k) = p-1$, or, equivalently, $(k, p-1) = 1$. ■

Proposition 8: Let p be an odd prime. The following relations hold true:

1) If a and b are quadratic non-residues modulo p then ab is a quadratic residue modulo p;
2) If a is a quadratic non-residue modulo p and b is a quadratic residue modulo p then ab is a quadratic non-residue modulo p;
3) If a and b are quadratic residues modulo p then ab is a quadratic residue modulo p;
4) If α and β are primitive roots modulo p then $\alpha\beta$ is a non-primitive root modulo p.

Proof:

(1), (2), (3) follow directly from Proposition 3(4);

(4) From Proposition 4 we obtain that α and β are quadratic non-residues modulo p which leads to the fact that $\alpha\beta$ is a quadratic residue modulo p and, thus, $\alpha\beta$ is not a primitive root modulo p (also by Proposition 4); ∎

Remark 2:
1) It is interesting that a result similar to the property presented in Proposition 8(2) does not hold for primitive roots. More exactly, if α is a primitive root modulo p and β is a non-primitive root modulo p then $\alpha\beta$ may be or not a primitive root modulo p. Indeed, in this case β can be expressed as α^k, with $(k, p-1) \neq 1$ (from Proposition 7(2)). We obtain that $\alpha\beta = \alpha^{k+1}$. We may have $(k+1, p-1) = 1$ (for example, in case $p = 7$, $k = 4$) and in this case $\alpha\beta$ is a primitive root modulo p or we may have $(k+1, p-1) \neq 1$ (for example, in case $p = 7$, $k = 2$) which implies that $\alpha\beta$ is a non-primitive root modulo p.
2) Also, a result similar to the property presented in Proposition 8(3) does not hold for primitive roots. More exactly, if α is a non-primitive root modulo p and β is a non-primitive root modulo p then $\alpha\beta$ may be or not a primitive root modulo p. Indeed, in this case, if we consider γ, a primitive root modulo p then α can be expressed as γ^{k_1}, with $(k_1, p-1) \neq 1$ and β can be expressed as γ^{k_2}, with $(k_2, p-1) \neq 1$ (from Proposition 7(2)). We obtain that $\alpha\beta = \gamma^{k_1+k_2}$. We may have $(k_1 + k_2, p-1) = 1$ (for example, in case $p = 7$, $k_1 = 2$, $k_2 = 3$) and in this case $\alpha\beta$ is a primitive root modulo p or we may have $(k_1 + k_2, p-1) \neq 1$ (for example, in case $p = 7$, $k_1 = 2$, $k_2 = 4$) which implies that $\alpha\beta$ is a non-primitive root modulo p.

IV. Generating Primitive Roots Modulo a Prime

Proposition 2(3) from Section II gives a method of generating a primitive root modulo an odd prime p, knowing the prime decomposition of $\phi(p) = p-1$, as presented in Figure 4 (which presents the classic randomized algorithm for generating a primitive root modulo an odd prime).

```
PrimitiveRoot1($p$)
input:      $p$, an odd prime;
additional input:  the prime factorization of $(p-1)$;
output:     a primitive root modulo $p$;
begin
   generate randomly $\alpha \in \mathbf{Z}_p^*$;
   $ok := 1$;
   for each prime $r$, $r|(p-1)$ do
      if $\alpha^{\frac{p-1}{r}} = 1$ then $ok := 0$
   if $ok = 1$ then return($\alpha$)
end.
```

Fig. 4: The classic algorithm for generating a primitive root modulo an odd prime p, knowing the prime factorization of $(p-1)$

The probability of success, denoted by $p_{success}$, of the classic randomized algorithm (which is of Las Vegas type) presented above is given by the ratio between the number of primitive roots and the total number of elements in $\mathbf{Z}_p^*$, i.e., $p_{success} = \frac{\phi(\phi(p))}{\phi(p)}$ (by Proposition 2(2)).

Using Proposition 5 from Section III we obtain the following improved algorithm:

```
PrimitiveRoot2($p$)
input:      $p$, an odd prime;
additional input:  the prime factorization of $(p-1)$;
output:     a primitive root modulo $p$;
begin
   $\alpha$ :=GenerateQuadraticNonResidue($p$);
   $ok := 1$;
   for each odd prime $r$, $r|(p-1)$ do
      if $\alpha^{\frac{p-1}{r}} = 1$ then $ok := 0$
   if $ok = 1$ then return($\alpha$)
end.
```

Fig. 5: An improved algorithm for generating a primitive root modulo an odd prime p, knowing the prime factorization of $(p-1)$

In fact, the main difference between PrimitiveRoot1 and PrimitiveRoot2 is that the exponentiation $\alpha^{\frac{p-1}{2}}$ (which is the operation with the greatest exponent among those involved in Algorithm PrimitiveRoot1) is replaced with the evaluation of the Legendre-Jacobi symbol (in order to verify that α is a quadratic non-residue modulo p). The improvement is given by the fact that the evaluation of Legendre-Jacobi symbol is much faster than an exponentiation (see, e.g., [12, page 113] or [15]). Moreover, for primes p with certain forms, a quadratic non-residue modulo p can be generated without effectively evaluating the Legendre-Jacobi symbol (see Algorithm GenerateQuadraticNonResidue from Section II) and this can lead to further improvements.

The biggest impact of our replacement is obtained in the case of the primes p of special form $p = 2^s q + 1$, where $s \geq 1$ and q is an odd prime (in this case Algorithm PrimitiveRoot1 implies only two exponentiations, with exponents $\frac{p-1}{2}$ and $\frac{p-1}{q}$ and in Algorithm PrimitiveRoot2 the operation with exponent $\frac{p-1}{2}$ is replaced with the evaluation of the Legendre-Jacobi symbol).

For such primes p, an element $\alpha \in \mathbf{Z}_p^*$ is a primitive root if and only if (α quadratic non-residue modulo $p \wedge \alpha^{2^s} \neq 1$).

We obtain the following algorithm:

```
PrimitiveRoot3($p$)
input:    $p$ prime, $p = 2^s q + 1$, $s \geq 1$, $q$ odd prime;
output:   a primitive root modulo $p$;
begin
  $\alpha$ :=GenerateQuadraticNonResidue($p$);
  if $\alpha^{2^s} \neq 1$ then return($\alpha$)
end.
```

Fig. 6: An algorithm for generating a primitive root modulo a prime p, $p = 2^s q + 1$, $s \geq 1$, q odd prime

For p prime, $p = 2^s q + 1$, $s \geq 1$, q is an odd prime, we obtain that

$$p_{success} = \frac{\phi(\phi(p))}{\phi(p)} = \frac{(2^s - 2^{s-1})(q-1)}{2^s q} = \frac{q-1}{2q} \approx \frac{1}{2}$$

Evidently, this algorithm may be iterated until a primitive root is finally outputted. The expected number iterations before a primitive root is found is two.

A. Finding Deterministic Algorithms for Generating Primitive Roots Modulo Special Primes

In this part of the paper we try to find efficient deterministic algorithms for generating primitive roots modulo p, $p = 2^s q + 1$, $s \geq 1$, q odd prime, for certain small values of s ($s \in \{1, 2\}$). In these cases, quadratic non-residues are known: (-1) for $s = 1$ (in this case, $p \equiv 3 \ mod\ 4$) and 2 for $s = 2$ (in this case, $p \equiv 5 \ mod\ 8$). The main idea is to find, with a single random generation, an element that is simultaneously a quadratic non-residue and a non-solution for the equation $x^{2^s} = 1$. We may use the following tricks (see Proposition 8):

1) If α is a quadratic residue then, by multiplying it with a quadratic non-residue, we will obtain a quadratic non-residue;
2) If α is a quadratic non-residue then, by multiplying it with a quadratic residue, we will obtain a quadratic non-residue;
3) If α is a solution of the equation $x^{2^s} = 1$ then, by multiplying it with a non-solution of the same equation, we will obtain a non-solution of the same equation;
4) If α is a non-solution of the equation $x^{2^s} = 1$ then, by multiplying it with a solution of the same equation, we will obtain a non-solution of the same equation.

Thus, in case that a quadratic non-residue β is somehow provided, we need to consider two cases:

1) If $\beta^{2^s} \neq 1$ - then $\alpha = \beta$ is a primitive root modulo p;
2) If $\beta^{2^s} = 1$ - then, by multiplying β with a non-solution of the equation $x^{2^s} = 1$ which is simultaneously a quadratic residue modulo p, we will obtain a primitive root modulo p - more exactly, any transformation of type $\alpha = \beta\gamma^2$ will work as long $(\gamma^2)^{2^s} \neq 1$ (we have to remark that there exist quadratic residues that are non-solutions for the equation $x^{2^s} = 1$ because this equation has at most 2^s solutions in $\mathbf{Z}_p^*$ whereas there are $\frac{p-1}{2} = 2^{s-1}q$ quadratic residues modulo p). The simplest option is to choose $\gamma = 2$ (when $p \not| 4^{2^s} - 1$), which translates in multiplying β with 4 modulo p.

We obtain the following algorithm:

```
PrimitiveRoot4($p$)
input:    $p$ prime, $p = 2^s q + 1$, $s \geq 1$, $q$ odd prime;
additional input:  $\beta$, a quadratic non-residue mod $p$;
output:   a primitive root modulo $p$;
begin
  if $\beta^{2^s} \neq 1$ then return($\beta$)
  $\gamma := 2$;
  while $(\gamma^2)^{2^s} = 1$ do $\gamma := \gamma + 1$;
  $\alpha := \beta\gamma^2$;
  return($\alpha$)
end.
```

Fig. 7: An algorithm for generating a primitive root modulo a prime p, $p = 2^s q + 1$, $s \geq 1$, q odd prime, in case that a quadratic non-residue β is provided

A different strategy (but less applicable and, moreover, less efficient) is to start with α as a non-solution of the equation $x^{2^s} = 1$. There are two approaches for achieving this:

- generating a random α and testing if $\alpha^{2^s} \neq 1$;
- computing all the solutions of the equation $x^{2^s} = 1$ (using recursively the algorithm SquareRoots presented in Section II) and generating α outside this set; this approach is suitable only for very small s ($s = 1$ or $s = 2$).

Then, if α is a quadratic non-residue modulo p then α is a primitive root modulo p; else, by multiplying α with β we will obtain a primitive root modulo p providing that β is a solution of the equation $x^{2^s} = 1$.

We will propose next some efficient deterministic algorithms for generating primitive roots in the cases $s = 1$, $s = 2$.

1) The Case $s = 1$: This is exactly the case of *safe primes*, i.e, primes p of form $p = 2q + 1$, q odd prime. These primes satisfy $p \equiv 3 \ mod\ 4$ and thus (-1) is a quadratic non-residue modulo p. The element (-1) is also a solution for the equation $x^{2^1} = 1$. Because 1 and (-1) are the only solutions of this equation, it is sufficient to find a quadratic non-residue in the set $\{2, 3, \ldots, p-2\}$. The second strategy presented above leads to the following deterministic algorithm:

```
PrimitiveRootSafePrime1($p$)
input:    $p$ prime, $p = 2q + 1$, $q$ odd prime;
output:   a primitive root modulo $p$;
begin
  generate randomly $\alpha \in \{2, 3, \ldots, p-2\}$;
  if LegendreJacobi($\alpha$,$p$)=$-1$ then return($\alpha$)
                                   else return($-\alpha$)
end.
```

Fig. 8: An initial deterministic algorithm for generating a primitive root modulo a prime p, $p = 2q + 1$, q odd prime

Indeed, if α is a quadratic residue modulo p then $(-\alpha)$ is a quadratic non-residue modulo p (because (-1) is a quadratic non-residue modulo p). Moreover, if $\alpha \in \{2, 3, \ldots, p-2\}$ then also $(-\alpha) \in \{2, 3, \ldots, p-2\}$. The complexity of the above algorithm is $\mathcal{O}((log_2(p))^2)$ if we use the Algorithm `LegendreJacobi` presented in Section II but if we use faster methods for evaluating the Legendre-Jacobi symbol (see, for example, [15]), we can obtain an algorithm of complexity $\mathcal{O}(\frac{(log_2(p))^2}{log_2(log_2(p))})$.

The algorithm `PrimitiveRoot4`, for primes p of form $p = 2q+1$, q odd prime (in this case, we may choose $\beta = -1$), reduces to the following very simple algorithm:

```
PrimitiveRootSafePrime2($p$)

input:    $p$ prime, $p = 2q+1$, $q$ odd prime;
output:   a primitive root modulo $p$;
begin
   generate randomly $\gamma \in \{2, 3, \ldots, p-2\}$;
   $\alpha := -\gamma^2$;
   return($\alpha$)
end.
```

Fig. 9: An alternative deterministic algorithm for generating a primitive root modulo a prime p, $p = 2q+1$, q odd prime

Indeed, in this case, $\beta^{2^1} = 1$ and $(\gamma^2)^{2^1} = 1 \Leftrightarrow \gamma = \pm 1$.

Algorithm `PrimitiveRootSafePrime2` involves only one modular squaring modulo p and, therefore, its complexity is $\mathcal{O}((log_2(p))^2)$ (see, for example, [14]).

Example 1 illustrates the application of the algorithms `PrimitiveRootSafePrime1` and `PrimitiveRootSafePrime2`.

Example 1: (*with artificially small parameters*) Let $p = 11$ ($q = 5$). In this case 2 is a primitive root modulo 11 because $2^{\frac{11-1}{2}} = 10 \neq 1$ and $2^{\frac{11-1}{5}} = 4 \neq 1$. The rest of the primitive roots can be obtained as 2^3, 2^7, 2^9 (see Proposition 7(2)) which leads to the elements 8, 7, and, respectively, 6.

- `PrimitiveRootSafePrime1` - suppose that $\alpha = 5$ is generated in the first step of the algorithm; because $\left(\frac{5}{11}\right) = 1$ ($5 = 4^2 \ mod\ 11$) then $(-5) = 6$ is returned, which is indeed a primitive root modulo 11;
- `PrimitiveRootSafePrime2` - by considering all the values $\gamma \in \{2, 3, 4, 5\}$ we may obtain all the primitive roots modulo 11; these are: $(-2^2) = 7$, $(-3^2) = 2$, $(-4^2) = 6$, and, finally, $(-5^2) = 8$.

2) The Case $s = 2$: This is the case of primes $p = 2^2q+1$, q odd prime, that satisfy $p \equiv 5 \ mod\ 8$. Thus, 2 is a quadratic non-residue modulo p. If 2 is also a non-solution for the equation $x^{2^2} = 1$ then 2 is a primitive root modulo p. The relation $2^{2^2} = 1 \ mod\ p$ is equivalent with $p|15$. Because p is a prime and $p \geq 13$ (because $q \geq 3$) then we obtain that 2 cannot be a solution for the equation $x^{2^2} = 1 \ mod\ p$ and, thus, 2 is a primitive root modulo p. If we are interested in finding another primitive roots, we may raise 2 at any odd power that is not divisible by q.

V. Conclusions and Future Work

In this paper we have analysed some interesting connections between primitive roots and quadratic non-residues modulo a prime. Using some correlations between the mentioned topics, we have proposed an improvement of the classic randomized algorithm for generating primitive roots modulo a prime, by replacing an exponentiation with the evaluation of the Legendre-Jacobi symbol, which is much faster. Then we have focused on finding efficient deterministic algorithms for generating primitive roots modulo p, $p = 2^sq+1$, $s \geq 1$, q odd prime, for certain small values of s ($s \in \{1, 2\}$). Because in these cases, quadratic non-residues are known, some simple algorithms can be developed, as presented in Section IV-A.

It is interesting to investigate more complex cases, as $p = 2^sq^r + 1$ or even $p = 2^sq_1^{r_1}q_2^{r_2} + 1$. We will consider these issues in our future work.

References

[1] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, no. 6, pp. 644–654, 1976.

[2] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, pp. 469–472, 1985, (a preliminary version appeared in "Advances in Cryptology – Crypto '84", G. R. Blakley and D. Chaum, eds., Lecture Notes in Computer Science 196 (1985), 10-18).

[3] C.-P. Schnorr, "Efficient identification and signatures for smart cards," in *Advances in Cryptology - CRYPTO '89*, ser. Lecture Notes in Computer Science, G. Brassard, Ed., vol. 435. Springer, 1989, pp. 239–252.

[4] FIPS 186-3, "Digital Signature Standard", Federal Information Processing Standards Publication 186, June 2009, http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf.

[5] A. Tonelli, "Bemerkung über die Auflösung quadratischer Congruenzen," *Göttinger Nachrichten*, pp. 344–346, 1891.

[6] D. Shanks, "Five number-theoretic algorithms," in *Proceedings of the second Manitoba conference on numerical mathematics*, ser. Congressus Numerantium, R. Thomas and H. Williams, Eds., vol. 7. Utilitas Mathematica, 1973, pp. 51–70.

[7] M. Cipolla, "Un metodo per la risolutione della congruenza di secondo grado," *RendicontodellAccademia Scienze Fisiche e Matematiche, Napoli*, vol. 9, pp. 154–163, 1903.

[8] D. Lehmer, "Computer technology applied to the theory of numbers," in *Studies in number theory*, ser. MAA Studies in Mathematics, W. Leveque, Ed., vol. 6. Prentice-Hall, 1969, pp. 117–151.

[9] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Sciences*, vol. 28, no. 2, pp. 270–299, 1984.

[10] H. Cohen, *A Course in Computational Algebraic Number Theory*, 4th ed., ser. Graduate Texts in Mathematics. Springer-Verlag, 2000.

[11] F. L. Ţiplea, *Algebraic Foundations of Computer Science*. Polirom, 2006, (in Romanian).

[12] E. Bach and J. Shallit, *Algorithmic Number Theory, Volume I: Efficient Algorithms*. MIT Press, 1996.

[13] F. Lemmermeyer, *Reciprocity Laws. From Euler to Eisenstein*. Springer-Verlag, 2000.

[14] F. L. Ţiplea, S. Iftene, C. Hriţcu, I. Goriac, R. Gordân, and E. Erbiceanu, "MpNT: A multi-precision number theory package. Number-theoretic algorithms (I)," "Al.I.Cuza" University of Iaşi, Faculty of Computer Science, Tech. Rep. TR 03-02, 2003, (available at http://www.infoiasi.ro/~tr/tr.pl.cgi).

[15] S. Eikenberry and J. Sorenson, "Efficient algorithms for computing the Jacobi symbol," *Journal of Symbolic Computation*, vol. 26, no. 4, pp. 509–523, 1998.

[16] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, ser. Discrete Mathematics and Its Applications. CRC Press, vol. 6.

[17] A. Atkin, "Probabilistic primality testing (summary by F. Morain)," INRIA, Tech. Rep. 1779, 1992, URL:http://algo.inria.fr/seminars/sem91-92/atkin.pdf.