SQUARES IN $\mathbb{Z}/N\mathbb{Z}$

We study squares in the ring $\mathbb{Z}/N\mathbb{Z}$ from a theoretical and computational point of view. We present two related cryptographic schemes.

1. SQUARES IN $\mathbb{Z}/p\mathbb{Z}$

Consider for example the prime $p = 13$. Write the liste of squares in $(\mathbb{Z}/p\mathbb{Z})^*$. How many of them ? Why ?

Let p be an odd prime. For any integer x one defines the **Legendre symbol** $\left(\frac{x}{p}\right)$ in the following way :

- $\left(\frac{x}{p}\right) = 0$ if p divides x,
- $\left(\frac{x}{p}\right) = 1$ if x is a non-zero square modulo p,
- $\left(\frac{x}{p}\right) = -1$ if x is not a square modulo p.

The map $x \mapsto \left(\frac{x}{p}\right)$ induces a group homomorphism from $\mathbb{F}_p^*$ onto $\{1, -1\}$. It is a *character*. Actually $\left(\frac{x}{p}\right) = x^{\frac{p-1}{2}} \bmod p$. This provides a first method to compute this symbol efficiently. The famous quadratic reciprocity law states that

Theorem 1.1. *If p and q are odd (positive) prime integers then*

$$\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = (-1)^{\frac{(p-1)(q-1)}{4}}.$$

One can check this theorem on a few small examples e.g. $p = 5$ and $q = 7$ or $p = 5$ and $q = 3$.

2. COMPUTING SQUARE ROOTS IN $\mathbb{Z}/p\mathbb{Z}$

Assume $p = 103$ and $x = 46$. One checks that $46^{51} \equiv 1 \bmod 103$ so 46 is a square modulo 103. We observe that p is congruent to 3 modulo 4. So $\frac{p-1}{2} = 51$ is odd. The inverse of 2 modulo 51 is 26. Indeed

$$26 \times 2 = 1 + 51.$$

Set $y = x^{26} \bmod 103$. One has $y^2 = x^{1+51} = x$ so y is a square root of x. Computing $y = x^{26}$ is easy using fast exponentiation. On finds $y = 46 \bmod 103$. We thus have a very efficient method to compute square roots modulo a prime integer congruent to 3 modulo 4.

Assume now that $p = 101$ and $x = 13 \bmod 101$. One checks that $x^{50} \equiv 1 \bmod 101$ so x is a square modulo 101. The order of x in the group $(\mathbb{Z}/101\mathbb{Z})^*$ divides de 50. The largest odd divisor of 50 is 25 but $x^{25} \equiv -1 \bmod 101$. So the order of x is even. We cannot compute a square root of x as in the first example.

To overcome this difficulty we assume that we know a non-quadratic residue z modulo 101. There are many of them since half of the non-zero residues are not squares. We pick a random integer z between 2 and $p-1$ and we compute $z^{50} \bmod 101$. With probability $\frac{1}{2}$ the result is -1 and we are done. For example $z = 46$ is fine since $z^{50} = -1 \bmod 101$.

Let us multiply x by $z^2 = 96 \bmod 101$. We get $X = xz^2 = 36 \bmod 101$. And this time $X^{25} = x^{25}z^{50} = 1 \bmod 101$ so X has odd order. One easily computes a square root of X by inverting 2 modulo 25. Indeed $2\times 13 = 1+25$ so X^{13} is a square root of X. Set now $y = X^{13}z^{-1}$. One checks that y is a square root of x.

Attention : this is a probabilistic method. One does not know a general deterministic polynomial time algorithm to compute square roots modulo a prime.

3. Squares in $\mathbb{Z}/N\mathbb{Z}$

Let $N \geqslant 2$ be an integer with prime decomposition $N = \prod_i p_i^{e_i}$. From Chinese remainder theorem we know that an integer x is a square modulo N if and only if it is a square modulo every $p_i^{e_i}$. One can compute a square root of x by first computing square roots module every $p_i^{e_i}$ then glue these square roots thanks to Chinese remainder theorem. This is efficient if we know the factorization of N.

In case $N = 11 \times 13$ and $x = 3$ we find four square roots in this way. In general, if N is odd and has I prime factors one finds 2^k square roots for each x in $(\mathbb{Z}/N\mathbb{Z})^*$.

Conversely, if we have a black box that returns a square root for any residue modulo N then we can find non trivial factors of N. Why ?

We say that there is a probabilistic **reduction** of the problem of factoring integers to the problem of computing square roots modulo integers. There is also a probabilistic reduction of the problem of computing square roots modulo integers to the problem of factoring integers. So these two problems have similar complexity (at least in the probabilistic world).

4. The Jacobi symbol

Assume $N \geqslant 3$ is an odd integer and les $N = \prod_i p_i^{e_i}$ be its prime decomposition. One defines the Jacobi symbol as

$$\left(\frac{x}{N}\right) = \prod_i \left(\frac{x}{p_i}\right)^{e_i}.$$

The symbol $\left(\frac{x}{N}\right)$ only depends on the class of x modulo N. It has many evident multiplicative properties inherited from the Legendre symbol. For example $\left(\frac{a}{b}\right) = 0$ if and only if a and b are not coprime. The *quadratic reciprocity law* can be extended to Jacobi symbols.

Theorem 4.1 (Gauss). *Let $M \geqslant 3$ and $N \geqslant 3$ be odd coprime integers. One has*

$$\left(\frac{-1}{M}\right) = (-1)^{\frac{M-1}{2}} \text{ and } \left(\frac{2}{M}\right) = (-1)^{\frac{M^2-1}{8}} \text{ and } \left(\frac{M}{N}\right)\left(\frac{N}{M}\right) = (-1)^{\frac{(M-1)(N-1)}{4}}.$$

Using this theorem one can quickly compute Jacobi symbols. The algorithm is very similar to Euclidean algorithm.

When N is not a prime, the Jacobi symbol does not suffice to distinguish quadratic residues from non-quadratic residues. For example if $N = pq$ is a product of two distinct odd primes and x is prime to N then $\left(\frac{x}{N}\right) = 1$ if and only if x is either a square modulo p and modulo q or x is neither a square modulo p nore modulo q. In this last case one says that x is a *false square*. It is considered to be difficult to distinguish true and false squares.

5. A ZERO-KNOWLEDGE IDENTIFICATION PROTOCOL

Identification is a crucial need to secure communications. One needs to check the identity of one's contacts. Assume Carole belongs to a secret organization. She needs to contact James. She never met him. In order to avoid any infiltration of the organization, each of them must be able to check that he his dealing with the other. Here is a possible protocol. James gets close to Carole and tells her "BELOTE". Carole answers "REBELOTE". If everything goes well Carole and James know that they are in contact with the right person. We assume of course that the two passwords (BELOTE and REBELOTE) have been provided by the organization to each of them. These passwords should be used only once. Otherwise some enemy may intercept and re-use them. Another concern is the *man-in-the-middle* attack. A false Carole (say Karole) may contact James. James would give her the password BELOTE to prove himself. Then Karole would quickly contact Carole and send her the password BELOTE. Carole would accept this password and answer REBELOTE to Karole. Now Karole would be able to convince James that she is Carole.

All these difficulties are consequences of the following two self contradictory principles

(1) The identity is defined by the knowledge of some information (e.g. a password) that must be kept secret.
(2) Proving oneself is achieved by unveiling this information (e.g. sending the password).

Biometric identification techniques (fingerprints, iris recognition, voice identification, ...) may be a solution, but this is not the scope of this text.

The following identification scheme is due to Feige, Fiat and Shamir. It is a **Zero-Knowledge** identification scheme. James can prove that he knows a secret without unveiling it.

To start with, each member X of the organization chooses two large prime integers p_X and q_X and computes their product $N_X = p_X q_X$. Then X chooses a random quadratic residue r_X modulo N_X with uniform probability and a square root f_X of r_X. So $f_X^2 = r_X \bmod N_X$. Indeed it is simpler to choose f_X first.

The collection of all triples (X, N_X, r_X) is published in the phonebook of the organization. The prime factors p_X and q_X and the square root f_X are only known to X. It is the knowledge of f_X that distinguishes X.

Before contacting James, Carole looks for the triple (J, N_J, r_J) in the phonebook. She contacts the alleged James and try to make sure that he knows a square root of r_J modulo N. They interact as follows.

(1) James picks a random quadratic residue $z = u^2 \bmod N_J$ (with uniform probability) and computes $t = zr_J \bmod N_J$. He sends t to Carole.
(2) Carole chooses a random element ϵ (with uniform probability) in $\{1, -1\}$ and send it to James.
(3) If $\epsilon = 1$ James sends u to Carole. Otherwise he sends a square root s of t modulo N_J (he knows such a square root $s = uf_J$ because he knows a square root of r_J and a square root of z.)
(4) If $\epsilon = 1$, Carole computes $z = t/r_J \bmod N_J$ and checks that $z = u^2 \bmod N_J$.
(5) If $\epsilon = 0$, Carole checks that $s^2 = t \bmod N_J$.

This protocol runs in time polynomial in $\log N_J$. It can and should be repeated many (e.g. 1000) times. Carole accepts if the condition checked at the last step has been always satisfied. Otherwise she rejects.

If James wants to prove himself, he can answer Carole's questions. And Carole learns nothing about James' secret because she only gets a series of random quadratic residues modulo N_J. She could produce such a series herself without the support of James. So she receives no information from him.

Assume an enemy (say Octopus) claims to be James. He doesn't know any square root of r_J. So he cannot know at the same time an s and a u such that $s^2 = t$ and $u^2 = z$. So he fails to answer correctly with probability $\geqslant \frac{1}{2}$. When the protocol is iterated n times the probability for Octopus to cheat Carole is $\leqslant 1/2^n$. The security of the scheme relies on the difficulty of computing square roots modulo a composite integer. This is a hard problem as long as factoring integers is hard.

6. Flipping coins with a telephone

Alice and Bob want to flip a coin to decide who is going to dust-sweep tomorrow. However they are not in the same room and they only can interact with a telephone or by mail. They need to pick a random element in $\{\text{heads}, \text{tails}\}$ with uniform probability. None of them should be able to influence on the result.

The following protocol provides a solution under the assumption that false squares cannot be distinguished from true ones efficiently.

(1) Bob chooses two large numbers p and q. He computes the product $N = pq$ and sends it to Alice (he does not send p and q). Bob chooses a residue x modulo N such that $\left(\frac{x}{N}\right) = 1$ (with uniform distribution)) and he sends x to Alice.
(2) Alice receives N and x but she doesn't know p ans q. So she doesn't know if x is a true or a false square. She pick a random element ϵ (with uniform distribution) in $\{1, -1\}$ and she sends it to Bob.

(3) Bob compares ϵ and $\left(\frac{x}{p}\right)$. If they are equal then the result of the protocol is heads and if they are different the result is tails. Bob sends his conclusion to Alice. He justifies it by transmitting p and q.
(4) Alice checks that p and q are prime integers and $N = pq$. She computes the Legendre symbols $\left(\frac{x}{p}\right)$ and $\left(\frac{x}{q}\right)$ and checks Bob's claim.

This protocol runs in time polynomial in $\log N$. The probability distribution on the output is uniform as soon as one of the players is honest (meaning he executes the protocol correctly).

If Bob cheats but Alice is honest then the distribution is uniform because the group composition of two random variables, one of which is uniform, is uniform too.

If Alice wants to influence on the result she must guess some information about whether x is a true or a false square. One assume that it is impossible to get any information about this in polynomial time.