CS181A Notes #6 Cryptosystems Based on Quadratic Residuosity

We describe some cryptosystems based on the hardness of Quadratic Residuosity.

> QUADRATIC RESIDUOSITY
> *Input*: An integer N and an integer $x \in \mathbb{Z}_N^\star$.
>
> *Output*: True, if x is a quadratic residue modulo N (that is, there is another integer $y \in \mathbb{Z}_N^\star$ so that $y^2 \equiv x \pmod{N}$), or False, otherwise.

First, we develop some useful number-theoretic machinery related to the above problem. For a prime p, we call x a **quadratic residue modulo** p if there is some $y \in \mathbb{Z}_p^\star$ so that $y^2 \equiv x \pmod{p}$. We denote QR_p as the set of all quadratic residues modulo p, and similarly, QNR_p as the set of all quadratic *non*-residues modulo p.

The **Legendre symbol** of x modulo p is defined as

$$\left(\frac{x}{p}\right) = \begin{cases} +1 & \text{if } x \in QR_p \\ -1 & \text{if } x \notin QR_p \end{cases}$$

Claim 1. *(Euler's criterion) For a prime p and $x \in \mathbb{Z}_p^\star$, we have*

$$\left(\frac{x}{p}\right) = x^{(p-1)/2} \pmod{p}.$$

Proof. Since $\mathbb{Z}_p^\star$ is cyclic, let g be its generator. Consider the set $\{g^j : j = 1, 2, \ldots, p-1\}$. We claim that g^j is a quadratic residue modulo p iff j is even. If j is even, it is clear that $(g^{j/2})^2$ is a residue. Suppose that g^j is a residue. So, there is some y so $y^2 = g^j \pmod{p}$. Say, $y = g^i$, for some i, which implies that $j = 2i$ is even. Now, note $(g^j)^{(p-1)/2} = +1$ if j is even, by Fermat's little theorem. If j is odd, then $(g^j)^{(p-1)/2} = -1$ since $g^{(p-1)/2} = -1 \pmod{p}$. □

Claim 2. *For a prime p and integers $x, y \in \mathbb{Z}_p^\star$, we have*

$$\left(\frac{xy}{p}\right) = \left(\frac{x}{p}\right)\left(\frac{y}{p}\right).$$

Proof. This follows immediately from Euler's criterion. □

Given an integer N, the **Jacobi symbol** of x modulo N is defined as

$$\left(\frac{x}{N}\right) = \prod_{i=1}^{k} \left(\frac{x}{p_i}\right)^{e_i},$$

where $N = \prod_{i=1}^{k} p_i^{e_i}$ is the prime factorization of N. We will focus on the case when N is a product of two distinct primes $N = pq$ for which the Jacobi symbol simply becomes

$$\left(\frac{x}{N}\right) = \left(\frac{x}{p}\right)\left(\frac{x}{q}\right).$$

It is known that the Jacobi symbol of x modulo N can be computed efficiently even without knowledge of the prime factorization of N.

The Goldwasser-Micali system Suppose we are given a positive integer k which is the security parameter. As the receiver, Bob prepares his cryptographic keys as follows:

1. Choose two distinct random k-bit prime numbers p and q.
2. Compute $N = pq$.
3. Choose a random integer $y \in \mathbb{Z}_N^\star$ so

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

   We call such a y a *pseudo-square*.

The public keys are (N, y) and the secret keys are (p, q).

To encrypt a m-bit message $x \in \{0,1\}^m$, Alice follows these steps:

1. For each $i = 1, \ldots, m$:
   (a) Choose a random $x_i \in \mathbb{Z}_N^\star$.
   (b) Set $c_i \equiv x_i^2 y^{x_i} \pmod{N}$.
2. The ciphertext is $(c_1, \ldots, c_m)$.

For Bob to decrypt the m-tuple ciphertext $(c_1, \ldots, c_m)$, he computes

$$x_i = \frac{1}{2}\left[1 - \left(\frac{c_i}{p}\right)\right] \in \{0, 1\}.$$

Remark: The Goldwasser-Micali system has an obvious drawback as being impractical due to the large ciphertext-plaintext ratio. But, it has an amazing property of being *semantically secure*. Informally, this means that the distribution of actual ciphertext observed by an eavesdropper is statistically indistinguishable from the same distribution of ciphertext that can be generated by the eavesdropper itself. This strongly implies that the actual ciphertext sent did not leak any non-negligible "information" about the plaintext.

The Blum-Goldwasser system Suppose we are given a positive integer k which is the security parameter. We call N a *Blum integer* if N is the product of two distinct prime numbers p and q satisfying $p \equiv q \equiv 3 \pmod 4$. As the receiver, Bob prepares his cryptographic keys by choosing a random Blum integer $N = pq$. The public key is N and the secret keys are (p, q). To encrypt a m-bit message $x \in \{0, 1\}^m$, Alice follows these steps:

1. Choose a random $x_0 \in \mathbb{Z}_N^\star$.
   This sets the initial seed for the **pseudorandom bit generator** of Blum-Blum-Shub (BBS).

2. For each $i = 1, \ldots, m$:

   (a) Compute $x_i \equiv x_{i-1}^2 \pmod N$.
   This runs the modular squaring step of the BBS generator.

   (b) Set $b_i = 0$, if x_i is even, and $b_i = 1$, if x_i is odd.
   This defines the pseudorandom output bits of the BBS generator.

   (c) Compute $c_i = x_i \oplus b_i$.
   This applies a one-time pad step using the pseudorandom bits.

3. Compute $x_{m+1} \equiv x_m^2 \pmod p$.
   Runs the BBS generator for one more step.

4. The ciphertext is $(c_1, \ldots, c_m)$ and x_{m+1}.

For Bob to decrypt the ciphertext $(c_1, \ldots, c_m)$ and x_{m+1}, he first computes the successive square roots x_{m+1} using the following fact.

Claim 3. *Let p be a prime for which $p \equiv 3 \pmod 4$. If x is a quadratic residue modulo p, then*

$$y \equiv \pm x^{(p+1)/4} \pmod p$$

satisfies $y^2 \equiv x \pmod p$.

Since Bob knows p and q, he can compute the four distinct square roots of any solvable equation of the form $x^2 \equiv a \pmod N$. Since exactly one of these square roots is a quadratic residue, namely the one given by the positive form of the above formula, Bob can recover each x_i used in Alice's BBS sequence. This enables Bob to also recover the pseudorandom bits b_i used and hence the message bits x_i itself. The specific details are as follows:

1. Compute $u \equiv ((p+1)/4)^m \pmod{p-1}$ and $v \equiv ((q+1)/4)^m \pmod{q-1}$. These are the exponents which allows Bob to move backwards through the BBS sequence.

2. Compute $a \equiv x_{m+1}^u \pmod p$ and $b \equiv x_{m+1}^v \pmod q$.
   This produces $x_1 \equiv a \pmod p$ and $x_1 \equiv b \pmod q$.

3. Reconstruct $x_1 \in \mathbb{Z}_N^\star$ from $(a, b) \in \mathbb{Z}_p^\star \times \mathbb{Z}_q^\star$.
   This uses the Chinese Remainder Theorem.

4. Retrace all the BBS squaring steps used by Alice and recover thereby recover the pseudorandom bits b_i and hence the message bits x_i.