Lecture Notes, Week 7

1 QR Probabilistic Cryptosystem

Let $n = pq$, p, q distinct odd primes. We can divide the numbers in $\mathbf{Z}_n^*$ into four classes depending on their membership in QR_p and QR_q.[1] Let Q_n^{11} be those numbers that are quadratic residues mod both p and q; let Q_n^{10} be those numbers that are quadratic residues mod p but not mod q; let Q_n^{01} be those numbers that are quadratic residues mod q but not mod p; and let Q_n^{00} be those numbers that are neither quadratic residues mod p nor mod q. Under these definitions, $Q_n^{11} = \mathrm{QR}_n$ and $Q_n^{00} \cup Q_n^{01} \cup Q_n^{10} = \mathrm{QNR}_n$.

Fact Given $a \in Q_n^{00} \cup Q_n^{11}$, there is no known feasible algorithm for determining whether or not $a \in \mathrm{QR}_n$ that gives the correct answer significantly more than 1/2 the time.

The Goldwasser-Micali cryptosystem is based on this fact. The public key consist of a pair $e = (n, y)$, where $n = pq$ for distinct odd primes p, q, and $y \in Q_n^{00}$. The private key consists of p. The message space is $\mathcal{M} = \{0, 1\}$.

To encrypt $m \in \mathcal{M}$, Alice chooses a random $a \in \mathrm{QR}_n$. She does this by choosing a random member of $\mathbf{Z}_n^*$ and squaring it. If $m = 0$, then $c = a \bmod n$. If $m = 1$, then $c = ay \bmod n$. The ciphertext is c.

It is easily shown that if $m = 0$, then $c \in Q_n^{11}$, and if $m = 1$, then $c \in Q_n^{00}$. One can also show that every $a \in Q_n^{11}$ is equally likely to be chosen as the ciphertext in case $m = 0$, and every $a \in Q_n^{00}$ is equally likely to be chosen as the ciphertext in case $m = 1$. Eve's problem of determining whether c encrypts 0 or 1 is the same as the problem of distinguishing between membership in Q_n^{00} and Q_n^{11}, which by the above fact is believed to be hard. Anyone knowing the private key p, however, can use the Euler Criterion to quickly determine whether or not c is a quadratic residue mod p and hence whether $c \in Q_n^{11}$ or $c \in Q_n^{00}$, thereby determining m.

2 Legendre Symbol

Recall that $\mathrm{QR}_n \subseteq \mathbf{Z}_n^*$ is the set of quadratic residues (perfect squares) modulo n. Let p be an odd prime, $a \in \mathbf{Z}_p$. The *Legendre symbol* $\left(\frac{a}{p}\right)$ is a number in $\{-1, 0, +1\}$, defined as follows:

$$\left(\frac{a}{p}\right) = \begin{cases} +1 & \text{if } a \in \mathrm{QR}_p \\ 0 & \text{if } p \mid a \\ -1 & \text{if } a \in \mathbf{Z}_p^* - \mathrm{QR}_p \end{cases}$$

By the Euler Criterion (see lecture notes week 6, section 6.4), we have

[1] To be strictly formal, we classify $a \in \mathbf{Z}_n^*$ according to whether or not $(a \bmod p) \in \mathrm{QR}_p$ and whether or not $(a \bmod q) \in \mathrm{QR}_q$.

Theorem 1 *Let p be an odd prime, $a \in \mathbf{Z}_p^*$. Then*

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

The Legendre symbol satisfies the following *multiplicative property*:

Fact Let p be an odd prime, $a_1, a_2 \in \mathbf{Z}_p^*$. Then

$$\left(\frac{a_1 a_2}{p}\right) = \left(\frac{a_1}{p}\right)\left(\frac{a_2}{p}\right)$$

Not surprisingly, if a_1 and a_2 are both quadratic residues, then so is $a_1 a_2$. This shows that the fact is true for the case that

$$\left(\frac{a_1}{p}\right) = \left(\frac{a_2}{p}\right) = 1.$$

More surprising is the case when neither a_1 nor a_2 are quadratic residues, so

$$\left(\frac{a_1}{p}\right) = \left(\frac{a_2}{p}\right) = -1.$$

In this case, the above fact says that the product $a_1 a_2$ *is* a quadratic residue since

$$\left(\frac{a_1 a_2}{p}\right) = (-1)(-1) = 1.$$

Here's a way to see this. Let g be a primitive root of p. Write $a_1 \equiv g^{k_1} \pmod p$ and $a_2 \equiv g^{k_2} \pmod p$. Since a_1 and a_2 are not quadratic residues, it must be the case that k_1 and k_2 are both odd; otherwise $g^{k_1/2}$ would be a square root of a_1, or $g^{k_2/2}$ would be a square root of a_2. But then $k_1 + k_2$ is even since the sum of any two odd numbers is always even. Hence, $g^{(k_1+k_2)/2}$ is a square root of $a_1 a_2 \equiv g^{k_1+k_2} \pmod p$, so $a_1 a_2$ is a quadratic residue.

3 Jacobi Symbol

The *Jacobi symbol* extends the Legendre symbol to the case where the "denominator" is an arbitrary odd positive number n with prime factorization $\prod_{i=1}^{k} {p_i}^{e_i}$.

3.1 Definition

We define

$$\left(\frac{a}{n}\right) = \prod_{i=1}^{k} \left(\frac{a}{p_i}\right)^{e_i}. \tag{1}$$

(By convention, this product is 1 when $k = 0$, so $\left(\frac{a}{1}\right) = 1$.) The symbol on the right side of (1) is the Legendre symbol, and the symbol on the left is the Jacobi symbol. Clearly, when $n = p$ is an odd prime, the Jacobi symbol and Legendre symbols agree, so the Jacobi symbol is a true extension of our earlier notion.

What does the Jacobi symbol mean when n is not prime? If $\left(\frac{a}{n}\right) = -1$ then a is definitely not a quadratic residue modulo n, but if $\left(\frac{a}{n}\right) = 1$, a might or might not be a quadratic residue. Consider the important case of $n = pq$ for p, q distinct odd primes. Then

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p}\right)\left(\frac{a}{q}\right)$$

so there are two possibilities for $\left(\frac{a}{n}\right) = 1$: either $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = +1$ or $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = -1$. In the first case, a is a quadratic residue modulo both p and q, so a is a quadratic residue modulo n. In the second case, a is not a quadratic residue modulo either p or q, and it is *not* a quadratic residue modulo n, either. Such numbers a are sometimes called "pseudo-squares" since they have Jacobi symbol 1 but are not quadratic residues.

3.2 Identities

The Jacobi symbol is easily computed using Equation 1 and Theorem 1 if the factorization of n is known. Similarly, $\gcd(u, v)$ is easily computed if the factorizations of u and v are known. The Euclidean algorithm allows us to compute $\gcd(u, v)$ efficiently even without knowing the factors of n. A similar algorithm allows $\left(\frac{a}{n}\right)$ to be computed efficiently without knowing the factorization of a or n.

The algorithm is based on identities satisfied by the Jacobi symbol:

1. $\left(\frac{0}{1}\right) = 1$; $\left(\frac{0}{n}\right) = 0$ for $n \neq 1$;
2. $\left(\frac{2}{n}\right) = 1$ if $n \equiv \pm 1 \pmod 8$; $\left(\frac{2}{n}\right) = -1$ if $n \equiv \pm 3 \pmod 8$;
3. $\left(\frac{a_1}{n}\right) = \left(\frac{a_2}{n}\right)$ if $a_1 \equiv a_2 \pmod n$;
4. $\left(\frac{2a}{n}\right) = \left(\frac{2}{n}\right)\left(\frac{a}{n}\right)$;
5. $\left(\frac{a}{n}\right) = -\left(\frac{n}{a}\right)$ if $a \equiv n \equiv 3 \pmod 4$.
6. $\left(\frac{a}{n}\right) = \left(\frac{n}{a}\right)$ if $a \equiv 1 \pmod 4$ or ($a \equiv 3 \pmod 4$ and $n \equiv 1 \pmod 4$);

There are many ways to turn these identities into an algorithm. Below is a straightforward recursive approach. Slightly more efficient iterative implementations are also possible.

```
int jacobi(int a, int n)
/* Precondition: a, n >= 0; n is odd */
{
  if (a == 0)                      /* identity 1 */
    return (n==1) ? 1 : 0;
  if (a == 2) {                    /* identity 2 */
    switch (n%8) {
    case 1:
    case 7:
      return 1;
    case 3:
    case 5:
      return -1;
    }
  }
  if ( a >= n )                    /* identity 3 */
    return jacobi(a%n, n);
  if (a%2 == 0)                    /* identity 4 */
    return jacobi(2,n)*jacobi(a/2, n);
  /* a is odd */                   /* identities 5 and 6 */
```

```
    return (a%4 == 3 && n%4 == 3) ? -jacobi(n,a) : jacobi(n,a);
}
```

4 Strassen-Solovay Test of Compositeness

Recall that a test of compositeness for n is a set of predicates $\{\tau_a(n)\}_{a\in\mathbf{Z}_n^*}$ such that if $\tau(n)$ succeeds (is true), then n is composite. The Strassen-Solovay Test is the set of predicates $\{\nu_a(n)\}_{a\in\mathbf{Z}_n^*}$, where

$$\nu_a(n) = \text{true iff } \left(\frac{a}{n}\right) \not\equiv a^{(n-1)/2} \pmod{n}.$$

If n is prime, the test always fails by Theorem 1. Equivalently, if some $\nu_a(n)$ succeeds, then n must be composite. Hence, the test is a valid- test of compositeness.

Let $b = a^{(n-1)/2}$. There are two possible reasons why the test might succeed. One possibility is that $b^2 \equiv a^{n-1} \not\equiv 1 \pmod{n}$ in which case $b \not\equiv \pm 1 \pmod{n}$. This is just the Fermat test $\zeta_a(n)$ from section 10.1 of lecture notes week 5. A second possibility is that $a^{n-1} \equiv 1 \pmod{n}$ but nevertheless, $b \not\equiv \left(\frac{a}{n}\right) \pmod{n}$. In this case, b is a square root of $1 \pmod{n}$, but it might have the opposite sign from $\left(\frac{a}{n}\right)$, or it might not even be ± 1 since 1 has additional square roots when n is composite. We claim without proof that for some constant $c > 0$ and all composite numbers n, the probability that $\nu_a(n)$ succeeds for a randomly-chosen $a \in \mathbf{Z}_n^*$ is at least c. I believe that $c \geq 1/4$, but this fact must be checked.

5 Miller-Rabin Test of Compositeness

The Miller-Rabin Test is more complicated to describe than the Solovay-Strassen Test, but the probability of error (that is, the probability that it fails when n is composite) seems to be lower than for Solovay-Strassen, so that the same degree of confidence can be achieved using fewer iterations of the test. This makes it faster when incorporated into a primality-testing algorithm. It is also closely related to the algorithm presented in lecture notes week 6, section 1.3 for factoring an RSA modulus given the encryption and decryption keys.

5.1 The test

The test $\mu_a(n)$ is based on computing a sequence $b_0, b_1, \ldots, b_k$ of integers in $\mathbf{Z}_n^*$. If n is prime, this sequence ends in 1, and the last non-1 element, if any, is $n-1$ $(\equiv -1 \pmod{n})$. If the observed sequence is *not* of this form, then n is composite, and the Miller-Rabin Test succeeds. Otherwise, the test fails.

The sequence is computed as follows:

1. Write $n - 1 = 2^k m$, where m is an odd positive integer. Computationally, k is the number of 0's at the right (low-order) end of the binary expansion of n, and m is the number that results from n when the k low-order 0's are removed.

2. Let $b_0 = a^m \bmod n$.

3. For $i = 1, 2, \ldots, k$, let $b_i = (b_{i-1})^2 \bmod n$.

An easy inductive proof shows that $b_i = a^{2^i m} \bmod n$ for all i, $0 \leq i \leq k$. In particular, $b_k \equiv a^{2^k m} = a^{n-1} \pmod{n}$.

5.2 Validity

To see that the test is valid, we must show that $\mu_a(p)$ fails for all $a \in \mathbf{Z}_p^*$ when p is prime. By Euler's theorem[2], $a^{p-1} \equiv 1 \pmod{p}$, so we see that $b_k = 1$. Since 1 has only two square roots, 1 and -1, modulo p, and b_{i-1} is a square root of b_i modulo p, the last non-1 element in the sequence (if any) must be $-1 \bmod p$. This is exactly the condition for which the Miller-Rabin test fails. Hence, it fails whenever n is prime, so if it succeeds, n is indeed composite.

5.3 Accuracy

How likely is it to succeed when n is composite? It succeeds whenever $a^{n-1} \not\equiv 1 \pmod{n}$, so it succeeds whenever the Fermat test $\zeta_a(n)$ would succeed. (See lecture notes week 5, section 10.1.) But even when $a^{n-1} \equiv 1 \pmod{n}$ and the Fermat test fails, the Miller-Rabin test will succeed if the last non-1 element in the sequence of b's is one of the square roots of 1 other than ± 1. It can be proved that $\mu_a(n)$ succeeds for at least $3/4$ of the possible values of a. Empirically, the test almost always succeeds when n is composite, and one has to work to find a such that $\mu_a(n)$ fails.

5.4 Example

For example, take $n = 561 = 3 \cdot 11 \cdot 17$. This number is interesting because it is the first Carmichael number. A *Carmichael number* is an odd composite number n that satisfies $a^{n-1} \equiv 1 \pmod{n}$ for all $a \in \mathbf{Z}_n^*$. (See `http://mathworld.wolfram.com/CarmichaelNumber.html`.) These are the numbers that I have been calling "pseudoprimes". Let's go through the steps of computing $\mu_{37}(561)$.

We begin by finding m and k. 561 in binary is 1000110001 (a palindrome!). Then $n - 1 = 560 = (1000110000)_2$, so $k = 4$ and $m = (100011)_2 = 35$. We compute $b_0 = a^m = 37^{35} \bmod 561 = 265$ with the help of the computer. We now compute the sequence of b's, also with the help of the computer. The results are shown in the table below:

i	b_i
0	265
1	100
2	463
3	67
4	1

This sequence ends in 1, but the last non-1 element $b_3 \not\equiv -1 \pmod{561}$, so the test $\mu_{37}(561)$ succeeds. In fact, the test succeeds for every $a \in \mathbf{Z}_{561}^*$ except for $a = 1, 103, 256, 460, 511$. For each of those values, $b_0 = a^m \equiv 1 \pmod{561}$.

5.5 Optimization

In practice, one only wants to compute as many of the b's as necessary to determine whether or not the test succeeds. In particular, one can stop after computing b_i if $b_i \equiv \pm 1 \pmod{n}$. If $b_i \equiv -1 \pmod{n}$ and $i < k$, the test fails. If $b_i \equiv 1 \pmod{n}$ and $i \geq 1$, the test succeeds. This is because we know in this case that $b_{i-1} \not\equiv -1 \pmod{n}$, for if it were, the algorithm would have stopped after computing b_{i-1}.

[2] This is also called Fermat's little theorem.