COM S 687 Introduction to Cryptography	Sep 14, 2006

Lecture 7: Computational Number Theory III

Instructor: Rafael Pass *Scribe: Huijia(Rachel) Lin*

1 Introduction

In the last lecture, we talked about some basics in computational number theory. In specific, we introduced groups Z_p^* and Z_n^*, efficient operations in Z_p^*, discrete logarithm assumption and a collection of One Way Permutation(OWP) F_{exp} based on the discrete logarithm assumption.

However, in cryptography, we are more interested in trap-door functions, which are easy to compute and hard to invert unless given trap-door information. In this class, we will go deeper into computational number theory, in particular, the Chinese Remainder theorem, then look at operations in Z_n^*, and finally introduce RSA and Rabin assumptions which shed lights to the design of trap-door functions.

2 Chinese Remainder

We denote $\mathcal{Z}_n = \{1, \cdots, n\}$. Let $m = m_1 m_2$. Suppose $y \in \mathcal{Z}_m$. Consider the numbers $a_1 \equiv y \bmod m_1 \in \mathcal{Z}_{m_1}$, $a_2 = y \bmod m_2 \in \mathcal{Z}_{m_2}$. The Chinese remainder theorem considers the question of recombining a_1, a_2 back to get y. It says there is a unique way to do this under some conditions, and under these conditions says how.

Theorem 1 *Let $m_1, m_2, \cdots, m_k$ be pairwise relatively prime integers. That is, $\gcd(m_i, m_j) = 1$ for $1 \le i < j \le k$. Let $a_i \in \mathcal{Z}_{m_i}$ for $1 \le i \le k$ and set $m = m_1 m_2 \cdots m_k$. Then there exists a unique $y \in \mathcal{Z}_m$ such that $y \equiv a_i$ mod m_i for $i = 1, \cdots, k$. Furthermore there is an $O(k^2)$ time algorithm to compute y given a_1, a_2, m_1, m_2, where $k = \max(|m_1|, |m_2|)$.*

Proof. : For each i, let $n_i = (m/m_i) \in \mathcal{Z}$. By hypothesis, $\gcd(m_i, n_i) = 1$ and hence $\exists b_i \in Z_{m_i}$ such that $n_i b_i \equiv 1 \bmod m_i$. Let $c_i = b_i n_i$. Then $c_i \equiv 1 \bmod m_i \equiv 0 \bmod m_j$ for $j \neq i$. Set $y \equiv \sum_i c_i a_i \bmod m$. Then $y \equiv a_i \bmod m_i$ for each i.

Further, if $y' \equiv a_i \bmod m_i$ for each i then $y' \equiv y \bmod m_i$ for each i and since the m_is are pairwise relatively prime, it follows that $y \equiv y' \bmod m$, proving uniqueness.

We shall prove the theorem for a special case, i.e. when $k = 2$.

Theorem 2 *Let $n = pq$, with $\gcd(p, q) = 1$. Given $a_1 \in \mathcal{Z}_p$ and $a_2 \in \mathcal{Z}_q$, there is a unique $x \in Z_n$, such that:*

$$x \equiv a_1 \ mod\ p \qquad\qquad x \equiv a_2 \ mod\ q \tag{1}$$

In other words, the map $x \to (x \ mod\ p,\ x \ mod\ q)$ from $\mathcal{Z}_n$ to $\mathcal{Z}_p \times \mathcal{Z}_q$ is 1-1 and onto.

Proof. In this case, using Euclidean Algorithm, we can find $ap + bq \equiv 1 \bmod n$. Let $c = bq$ and $d = ap$. Then $s \equiv ca_1 + da_2 \bmod n$. We can verify that x satisfies (1). ■

A special case is when $a_1 = a_2 = a$, then $s \equiv ca_1 + da_2 = a(c + d) = a \bmod n$. In other words, $a \in \mathcal{Z}_p$ and $a \in \mathcal{Z}_q$ maps to $a \in \mathcal{Z}_n$

3 Operations in Z_n^*

Recall that $Z_n^* = \{x : x \in \mathcal{Z}_n \wedge \gcd(x, n) = 1\}$, where $n = pq$, p, q are primes greater than 2 and have equal length. Last class we discussed operations in Z_p^*, in the following we will look at operations in Z_n^*.

First, consider the square operation in Z_n^*. Let $QR_n = \{x^2 \bmod n : x \in Z_n^*\}$. Then we claim:

Theorem 3

$$x \in QR_n \qquad \Leftrightarrow \qquad x \ mod\ p \in QR_p \ \wedge \ x \ mod\ q \in QR_q$$

Proof. If direction: Let $y \equiv x \bmod p$ and $z \equiv x \bmod q$. As $y \in QR_p$ and $z \in QR_q$, $\exists a \in Z_p^*,\ y \equiv a^2 \bmod p$ and $\exists b \in Z_q^*,\ z \equiv b^2 \bmod q$. From Chinese Remainder Theorem, $\exists s \in Z_n^*, s \equiv a \bmod p$ and $s \equiv b \bmod q$. We show that s is one square root of x in Z_n^*:

$$s^2 \bmod p \equiv a^2 \bmod p \equiv y \bmod p$$
$$s^2 \bmod q \equiv b^2 \bmod q \equiv z \bmod q$$

Therefore s^2 is congruent modulo x, i.e. $x \equiv s^2 \bmod n$. Hence s is x's square root, and $x \in QR_n$

Only if direction: $x \in QR_n$, then $\exists a \in Z_n^*,\ x \equiv a^2 \bmod n$

$$x \bmod p \equiv a^2 \bmod p \equiv (a \bmod p)^2 \bmod p \Rightarrow x \bmod p \in Z_p^*$$
$$x \bmod q \equiv a^2 \bmod q \equiv (a \bmod q)^2 \bmod q \Rightarrow x \bmod q \in Z_q^*$$

Also we claim that:

Theorem 4 *$|QR_n| = |Z_n^*|/4$ and the mapping $x \to x^2$, $x \in Z_n^*$, is 4 to 1.*

Proof. We only need to show that the mapping $x \to x^2$ is 4 to 1. Then $|QR_n| = |Z_n^*|/4$ automatically follows.

From theorem 3, we know that given $x \in QR_n$, $y \equiv x \bmod p \in QR_p$ and $z \equiv x \bmod q \in QR_q$.

In last class, we proved that $|QR_p| = |Z_p^*|/2$ and the mapping $x \to x^2$, $x \in Z_p^*$ is 2 to 1. So we have unique square roots a_1, a_2 for y and b_1, b_2 for z. Take any two of them a_i, b_j, by Chinese remainder theorem $\exists s \in Z_n^*$, such that $s \equiv a_i \bmod p$ and $s \equiv b_j \bmod q$. And by the same argument in proof of theorem 3, s is a square root of x in Z_n^*. There are in total 4 combinations of a_i and b_j, thus we have 4 such s that are x's square roots.

Remark that given p and q, it is easy to compute the square roots for elements in Z_n^*. The proof above shows that the square root of $x \in Z_n^*$ can be combined from square roots of $x \bmod p$ and $x \bmod q$ in Z_p^* and Z_q^* respectively. And from last class, we have shown that square root operation is efficient in Z_p^*. So given p and q, we can simply calculates the square roots of $x \bmod p$ and $x \bmod q$, and combine the result to get square roots of x. However, if without p and q, it is not known that whether it can be done efficiently.

In brief, we have the following operations in Z_n^*:

1) Finding a random element in Z_n^* is efficient. As it can be done by picking a random number r from $\mathcal{Z}_n$ and check whether $\gcd(r, n) = 1$. We will hit an element in Z_n^* with high probability as with probability $|Z_n^*|/|\mathcal{Z}_n|$ the randomly sampled r would fall inside Z_n^*, which is $(p-1)(q-1)/pq$

2) Multiplication, division and exponentiation are efficient as that in Z_p^*.

3) Square root can be computed efficiently with knowledge of p and q. It is conjectured to be hard if without p and q.

4) ith root is the same to the case of square root: efficient to compute with $p,\ q$ and not known otherwise.

4 RSA collection

Based on the conjecture that the ith root in Z_n^* cannot be computed efficiently without knowing $p,\ q$, we have:

RSA collection: $I = \{n,\ e\ :\ n = pq,$ where p, q are prime and $|p| = |q|,\ e \in Z_{\phi(n)}^*\}$
$Gen(1^k) \to n,\ e$ where $n = pq$ and $p,\ q$ are random k-bit primes, e is a random element in

$Z^*_{\phi(n)}$. $f_{n,e} \ : \ Z^*_n \to Z^*_n$ defined by $f_{n,e}(x) \equiv x^e \bmod n$.

RSA assumption: RSA is a collection of OWF.

First we observe the fact that RSA is a collection of permutations.
Proof. Since $e \in Z^*_{\phi(n)}$, $\exists\ d$ such that $ed \equiv 1 \bmod \phi(n)$, i.e. d is e's inverse in $Z^*_{\phi(n)}$. We can define the inverse map of RSA as: $z \equiv y^d \bmod n$, that is, given $y = f_{n,e}(x)$:

$$z \equiv y^d \equiv f_{n,e}(x)^d \equiv (x^e)^d \bmod n \equiv x^{(ed \bmod \phi(n))} \bmod n \equiv x \bmod n$$

Hence, RSA is a permutation.

As RSA is a permutation, the RSA assumption equals that RSA is a collection of OWP, which is tightly related with the factoring assumption in the way that:

Theorem 5 *RSA is a one way permutation(OWP) only if factoring assumption, i.e., the RSA assumption implies the factoring assumption holds.*

Proof. We prove by contrapositive: if factoring is possible in polynomial time, then we can break RSA in polynomial time. Formally, assume $\exists A$ and polynomial function $p(k)$ so that A can factor $n = pq$ with probability $1/p(k)$, where p and q are random k-bits primes. Then $\exists A'$, which can invert $f_{n,e}$ with probability $1/p(k)$, where $n = pq$, $p,\ q \leftarrow \{0,1\}^k$ primes, and $e \leftarrow Z^*_{\phi(n)}$.

We construct $A'(n, e, y \equiv x^e \bmod n)$ as follows:

1) Let algorithm A try to factor n, which returns p and q.

2) Check whether $n = qp$, if not then abort.

3) Else we have found $p,\ q$ so that we can compute $\phi(n) = (p-1)(q-1)$.

4) Compute the inverse of e in $Z^*_{\phi(n)}$, $ed \equiv 1 \bmod \phi(n)$.

5) Output $y^d \bmod n$

The algorithm feeds factoring algorithm A with the product of two random primes with exactly the same distribution as that in factoring assumption. Hence in the first step A will return the correct prime factors with probability $1/p(k)$. Provided that the factors are correct, then we can compute the inverse of y in the same way as we construct the inverse map of $f_{n,e}$. And this always succeeds with probability 1. Overall A' succeeds with probability $1/p(k)$.

Remark that the converse is still unknown, i.e., whether factoring assumption implies RSA assumption is still open.

5 Rabin collection

Similar to ith root, the conjecture that taking square root in Z_n^* is hard without knowing p and q suggests Rabin collection:
Rabin Collection $I = \{n : n = pq,$ where p, q are prime $|p| = |q|\}$, $Gen(1^k) \to n$, where $n = pq$ and $p,\ q$ are random k-bit primes, $f_n : Z_N^* \to Z_n^*$ is defined by $f_n(x) \equiv x^2 \bmod n$.

Note that Rabin is not a special case of RSA since $\gcd(2, \phi(n)) \neq 1$.

What's more, stronger than the RSA collection, the assumption that Rabin is OWF actually implies factoring assumption. We claim the following:

Theorem 6 *Rabin is a OWF iff factoring assumption holds.*

Proof. If direction: If factoring is hard then Rabin is a OWF. We prove by contrapositive: if Rabin is breakable then factoring can be done efficiently. Formally, if $\exists A$ and polynomial function $p(k)$ such that A can inverse f_n with probability $1/p(k)$ for sufficiently large k, that is:

$$\Pr[p,\ q \leftarrow \{0,1\}^k \text{ prime }, \ n = pq,\ x \leftarrow Z_n^*,\ y = f_n(x),\ z = A(n, f_n(x)) \ : \ z^2 \equiv y \bmod n] > 1/p(k)$$

Then there exists A' able to factor the product of two random k-bits primes with probability $1/2p(k)$ for sufficiently large k.

$$\Pr[p,\ q \leftarrow \{0,1\}^k \text{ prime }, \ n = pq,\ z = A'(n) \ : \ z \in \{p, q\}] > 1/2p(k)$$

Given A, we construct $A'(n)$ as follows:

1) Sample a random x from Z_n^*.

2) Calculate the square of x as $y \equiv x^2 \bmod n$

3) Let A calculates the square root of y, $A(n, y)$, which outputs z.

4) Check whether $z^2 \equiv y \bmod n$. If not, then abort.

5) If z is a correct square root, output $\gcd(x - z, n)$

First, because the input to A' are the product of two random k-bits primes and A' randomly sample x from Z_n^*, the inputs to A in step 3 have exactly the same distribution as that to Rabin. Then algorithm A will return a correct square root of x with probability $1/p(k)$. And because A does not know about x, and it would output one of the four square roots with equal probability, with probability $1/2p(k)$, $z \not\equiv x$ and $z \not\equiv -x$. x and z are both square roots of y, hence:

$$x^2 \equiv z^2 \bmod n \quad \Rightarrow \quad (x - z)(x + z) \equiv 0 \bmod n$$

When $z \not\equiv x$ and $z \not\equiv -x$, It can only be that they are congruent modulo p and q. Then the $\gcd(x - z, n)$ would be p or q with probability $1/2p(k)$. Thus factoring assumption breaks.

Only if direction: We want to show if Rabin is hard then factoring is also hard. Similar to the proof of that RSA is OWP implies factoring is hard, we show by contrapositive, that is, if $\exists A$ and polynomial function $p(k)$, A can factor $n = pq$ with probability $1/p(k)$, where p and q are random k-bits primes. Then $\exists A'$, which can invert f_n with probability $1/p(k)$.

The construction of $A'(n, y \equiv x^2 \bmod n)$ is also similar to that in RSA case: first feed A with n, which returns $p,\ q$. Check whether $n = pq$, if not then abort, else compute square root of $y \bmod p$ and $y \bmod q$ in Z_p^* and Z_q^* respectively. Pick one pair of square roots $a,\ b$ and compute $s = ac + bd$. Output s.

Since A receives n with the same distribution as in the factoring assumption, A will return the correct factoring with probability $1/p(k)$ and from the correct p, q we can compute the square root of u with probability 1. Overall A' succeeds with probability $1/p(k)$.