Lecture Notes 13

53 Chinese Remainder Theorem

We now return to a basic result of number theory that will be used later.

Let $n_1, n_2, \ldots, n_k$ be positive *pairwise relatively prime* positive integers[1], let $n = \prod_{i=1}^{k} n_i$, and let $a_i \in \mathbf{Z}_i$ for $i = 1, \ldots, k$. Consider the system of congruence equations with unknown x:

$$\begin{array}{rcl} x & \equiv & a_1 \pmod{n_1} \\ x & \equiv & a_2 \pmod{n_2} \\ & \vdots & \\ x & \equiv & a_k \pmod{n_k} \end{array} \tag{1}$$

The *Chinese Remainder Theorem* says that (1) has a unique solution in $\mathbf{Z}_n$.

To solve for x, let

$$N_i = n/n_i = \underbrace{n_1 n_2 \ldots n_{i-1}} \cdot \underbrace{n_{i+1} \ldots n_k},$$

and compute $M_i = N_i^{-1} \bmod n_i$, for $1 \le i \le k$. Note that $N_i^{-1} \pmod{n_i}$ exists since $\gcd(N_i, n_i) = 1$ by the pairwise relatively prime condition. We can compute N_i^{-1} using the methods of section 47 (lecture notes 11). Now let

$$x = (\sum_{i=1}^{k} a_i M_i N_i) \bmod n \tag{2}$$

If $j \neq i$, then $M_j N_j \equiv 0 \pmod{n_i}$ since $n_i | N_j$. On the other hand, $M_i N_i \equiv 1 \pmod{n_i}$ by definition of M_i. Hence,

$$x \equiv \sum_{i=1}^{k} a_i M_i N_i \equiv \underbrace{0a_1 + \ldots + 0a_{i-1}} + 1a_i + \underbrace{0a_{i+1} \ldots 0a_k} \equiv a_i \pmod{n_i} \tag{3}$$

for all $1 \le i \le k$, establishing that (2) is a solution of (1).

To see that the solution is unique in $\mathbf{Z}_n$, let χ be the mapping $x \mapsto (x \bmod n_1, \ldots, x \bmod n_k)$. χ is a surjection[2] from $\mathbf{Z}_n$ to $\mathbf{Z}_{n_1} \times \ldots \times \mathbf{Z}_{n_k}$ since we have just shown for all $(a_1, \ldots, a_k) \in \mathbf{Z}_{n_1} \times \ldots \times \mathbf{Z}_{n_k}$ that there exists $x \in Z_n$ such that $\chi(x) = (a_1, \ldots, a_k)$. Since also $|\mathbf{Z}_n| = |\mathbf{Z}_{n_1} \times \ldots \times \mathbf{Z}_{n_k}|$, χ is a bijection, and (1) has only one solution in $\mathbf{Z}_n$.

A less slick but more direct way of seeing the same thing is to suppose that $x = u$ and $x = v$ are both solutions to (1). Then $u \equiv v \pmod{n_i}$, so $n_i | (u - v)$ for all i. By the pairwise relatively prime condition on the n_i, it follows that $n | (u - v)$, so $u \equiv v \pmod{n}$. Hence, the solution is unique in $\mathbf{Z}_n$.

[1] This means that $\gcd(n_i, n_j) = 1$ for all $1 \le i < j \le k$.

[2] A *surjection* is an onto function.

54 Homomorphic property of χ

The bijection χ is interesting in its own right, for it establishes a one-to-one correspondence between members of $\mathbf{Z}_n$ and k-tuples $(a_1, \ldots, a_k)$ in $\mathbf{Z}_{n_1} \times \ldots \times \mathbf{Z}_{n_k}$. This lets us reason about and compute with k-tuples and then translate the results back to $\mathbf{Z}_n$.

The *homomorphic property* of χ means that performing an arithmetic operation on $x \in \mathbf{Z}_n$ corresponds to performing the similar operation on each of the components of $\chi(x)$. More precisely, let $\odot$ be one of the arithmetic operations $+$, $-$, or $\times$. If $\chi(x) = (a_1, \ldots, a_k)$ and $\chi(y) = (b_1, \ldots, b_k)$, then

$$\chi((x \odot y) \bmod n) = ((a_1 \odot b_1) \bmod n_1,\ \ldots,\ (a_k \odot b_k) \bmod n_k). \tag{4}$$

In other words, if one first performs $z = (x \odot y) \bmod n$ and then computes $z \bmod n_i$, the result is the same as if one instead first computed $a_i = (x \bmod n_i)$ and $b_i = (y \bmod n_i)$ and then performed $(a_i \odot b_i) \bmod n_i$. This relies on the fact that $(z \bmod n) \bmod n_i = z \bmod n_i$, which holds because $n_i \,|\, n$.

55 RSA Decryption Works for All of Z_n

In section 44 (lecture notes 10), we showed that RSA decryption works when $m, c \in \mathbf{Z}_n^*$ but omitted the proof that it actually works for all $m, c \in \mathbf{Z}_n$. We now use the Chinese Remainder Theorem to supply this missing piece.

Let $n = pq$ be an RSA modulus, p, q distinct primes, and let e and d be the RSA encryption and decryption exponents, respectively. We show $m^{ed} \equiv m \pmod{n}$ for all $m \in \mathbf{Z}_n$.

Define $a = (m \bmod p)$ and $b = (m \bmod q)$, so

$$\begin{aligned} m &\equiv a \pmod{p} \\ m &\equiv b \pmod{q} \end{aligned} \tag{5}$$

Raising both sides to the power ed gives

$$\begin{aligned} m^{ed} &\equiv a^{ed} \pmod{p} \\ m^{ed} &\equiv b^{ed} \pmod{q} \end{aligned} \tag{6}$$

We now argue that $a^{ed} \equiv a \pmod{p}$. If $a \equiv 0 \pmod{p}$, then obviously $a^{ed} \equiv 0 \equiv a \pmod{p}$. If $a \not\equiv 0 \pmod{p}$, then $\gcd(a, p) = 1$ since p is prime, so $a \in \mathbf{Z}_p^*$. By Euler's theorem,

$$a^{\phi(p)} \equiv 1 \pmod{p}$$

Since $ed \equiv 1 \pmod{\phi(n)}$, we have $ed = 1 + u\phi(n) = 1 + u\phi(p)\phi(q)$ for some integer u. Hence,

$$a^{ed} \equiv a^{1+u\phi(p)\phi(q)} \equiv a \cdot \left(a^{\phi(p)}\right)^{u\phi(q)} \equiv a \cdot 1^{u\phi(q)} \equiv a \pmod{p} \tag{7}$$

Similarly,

$$b^{ed} \equiv b \pmod{q} \tag{8}$$

Combining the pair (6) with (7) and (8) yields

$$\begin{aligned} m^{ed} &\equiv a \pmod{p} \\ m^{ed} &\equiv b \pmod{q} \end{aligned}$$

Thus, m^{ed} is a solution to the system of equations

$$\begin{aligned} x &\equiv a \pmod{p} \\ x &\equiv b \pmod{q} \end{aligned} \tag{9}$$

From (5), m is also a solution of (9). By the Chinese Remainder Theorem, the solution to (9) is unique modulo n, so $m^{ed} \equiv m \pmod{n}$ as desired.

56 RSA Security

Several possible attacks on RSA are discussed below and their relative computational difficulties discussed.

56.1 Factoring n

The security of RSA depends on the computational difficulty of several different problems, corresponding to different ways that Eve might attempt to break the system. The first of these is the *RSA factoring problem*: Given a number n that is known to be the product of two primes p and q, find p and q. Clearly if Eve can find p and q, then she can compute the decryption key d from the public encryption key e (in the same way that Alice did when generating the key) and subsequently decrypt all ciphertexts.

56.2 Computing $\phi(n)$

Eve doesn't really need to know the factors of n in order to compute d. It is enough for her to know $\phi(n)$. Computing $\phi(n)$ is no harder than factoring n since $\phi(n)$ is easily computed given the factors of n, but is it perhaps easier? If it were, then Eve would have a possible attack on RSA different from factoring n. It turns out that it is not easier, for if Even knows $\phi(n)$, then she can factor n. She simply sets up the system of quadratic equations

$$\begin{aligned} n &= pq \\ \phi(n) &= (p-1)(q-1) \end{aligned}$$

in two unknowns p and q and solves for p and q using standard methods of algebra.

56.3 Finding d

Another possibility is that Eve might somehow be able to compute d from e and n even without the ability to factor n or compute $\phi(n)$. That would represent yet another attack that couldn't be ruled out by the assumption that the RSA factoring problem is hard. However, that too is not possible, as we now show.

Suppose Eve knows n and e and is somehow able to obtain d. Then Eve can factor n by a probabilistic algorithm. The algorithm is presented in Figure56.1.

We begin by finding unique integers s and t such that $2^s t = ed - 1$ and t is odd. This is always possible since $ed - 1 \neq 0$. One way to find s and t is to express $ed - 1$ in binary. Then s is the number of trailing zeros and t is the value of the binary number that remains after the trailing zeros are removed. Since $ed - 1 \equiv 0 \pmod{\phi(n)}$ and $4 \mid \phi(n)$ (since both $p - 1$ and $q - 1$ are even), it follows that $s \geq 2$.

To factor n:
  /* Find s, t such that $ed - 1 = 2^s t$ and t is odd */
  $s = 0;\ t = ed - 1;$
  while (t is even) $\{s{+}{+};\ t/{=}2;\}$
  /* Search for non-trivial square root of $1 \pmod n$ */
  do {
    /* Find a random square root b of $1 \pmod n$ */
    choose $a \in \mathbf{Z}_n^*$ at random;
    $b = a^t \bmod n$;
    while ($b^2 \not\equiv 1 \pmod n$) $b = b^2 \bmod n$;
  } while ($b \equiv \pm 1 \pmod n$);
  /* Factor n */
  $p = \gcd(b - 1, n)$;
  $q = n/p$;
  return (p, q);
}

Figure 56.1: Algorithm for factoring n given d.

Now, for each a chosen at random from $\mathbf{Z}_n^*$, define a sequence $b_0, b_1, \ldots, b_s$, where $b_i = a^{2^i t} \bmod n$, $0 \le i \le s$. Each number in the sequence is the square of the number preceding it modulo n. The last number in the sequence is 1 by Euler's theorem and the fact that $\phi(n) \,|\, (ed-1)$. Since $1^2 \bmod n = 1$, every element of the sequence following the first 1 is also 1. Hence, the sequence consists of a (possibly empty) block of non-1 elements, following by a block of 1's.

It is easily verified that b_0 is the value of b when the innermost while loop is first entered, and b_k is the value of b after the k^{th} iteration of that loop. The loop executes at most $s - 1$ times since it terminates just before the first 1 is encounter, that is, when $b^2 \equiv 1 \pmod n$. At this time, $b = b_k$ is a square root of $1 \pmod n$.

Over the reals, we know that each positive number has two square roots, one positive and one negative, and no negative numbers have real square roots. Over $\mathbf{Z}_n^*$ for $n = pq$, it turns out that $1/4$ of the numbers have square roots, and each number that has a square root actually has four. Since 1 does have a square root modulo n (itself), there are four possibilities for b: $\pm 1 \bmod n$ and $\pm r \bmod n$ for some $r \in \mathbf{Z}_n^*$, $r \not\equiv \pm 1 \pmod n$.

The do loop terminates if and only if $b \not\equiv \pm 1 \pmod n$. At that point we can factor n. Since $b^2 \equiv 1 \pmod n$, we have $n \,|\, (b^2 - 1) = (b + 1)(b - 1)$. But since $b \not\equiv \pm 1 \pmod n$, n does not divide $b + 1$ and n does not divide $b - 1$. Therefore, one of the factors of n divides $b + 1$ and the other divides $b - 1$. Hence, $\gcd(b - 1, n)$ is a non-trivial factor of n.

It can be shown that there is at least a 0.5 probability that $b \not\equiv \pm 1 \pmod n$ for the b computed by this algorithm in response to a randomly chosen $a \in \mathbf{Z}_n^*$. Hence, the expected number of iterations of the do loop is at most 2. (See Evangelos Kranakis, *Primality and Cryptography*, Theorem 5.1 for details.)

Here's a simple example of the use of this algorithm. Suppose $n = 5 \times 11 = 55$, $e = 3$, and $d = 27$. (These are possible RSA values since $\phi(n) = 40$ and $ed = 81 \equiv 1 \pmod{40}$.) Then $ed - 1 = 80 = (1010000)_2$, so $s = 4$ and $t = 5$.

Now, suppose we take $a = 2$. We compute the sequence of b's as follows:

$$\begin{aligned}
b_0 &= a^t \bmod n = 2^5 \bmod 55 = 32 \\
b_1 &= (b_0)^2 \bmod n = (32)^2 \bmod 55 = 1024 \bmod 55 = 34 \\
b_2 &= (b_1)^2 \bmod n = (34)^2 \bmod 55 = 1156 \bmod 55 = 1 \\
b_3 &= (b_2)^2 \bmod n = (1)^2 \bmod 55 = 1 \\
b_4 &= (b_3)^2 \bmod n = (1)^2 \bmod 55 = 1
\end{aligned}$$

Since the last $b_i \neq 1$ in this sequence is 34, and $34 \not\equiv -1 \pmod{55}$, then 34 is a non-trivial square root of 1 modulo 55. It follows that $\gcd(34 - 1, 55) = 11$ is a prime divisor of n.

56.4 Finding plaintext

Eve isn't really interested in factoring n, computing $\phi(n)$, or finding d, except as a means to read Alice's secret messages. Hence, the problem we would like to be hard is the problem of computing the plaintext m given an RSA public key (n, e) and a ciphertext c. The above shows that this problem is no harder than factoring n, computing $\phi(n)$, or finding d, but it does not rule out the possibility of some clever way of decrypting messages without actually finding the decryption key. Perhaps there is some feasible probabilistic algorithm that finds m with non-negligible probability, maybe not even for all ciphertexts c but for some non-negligible fraction of them. Such a method would "break" RSA and render it useless in practice. No such algorithm has been found, but neither has the possibility been ruled out, even under the assumption that the factoring problem itself is hard.

57 Primitive Roots

We have completed our discussion of RSA. We turn next to other number-theoretic techniques with important cryptographic applications. We begin by looking in greater detail at the structure of $\mathbf{Z}_n^*$, the set of integers relatively prime to n.

Let $g \in \mathbf{Z}_n^*$ and consider the successive powers $g, g^2, g^3, \ldots$, all taken modulo n. Because $\mathbf{Z}_n^*$ is finite, this sequence must eventually repeat. By Euler's theorem, this sequence contains 1 since $g^k = 1$ (in $\mathbf{Z}_n$) for $k = \phi(n)$. Let k be the smallest positive integer such that $g^k = 1$. We call k the *order of* g and write $\mathrm{ord}(g) = k$. The elements $\{g, g^2, \ldots, g^k = 1\}$ form a subgroup S of $\mathbf{Z}_n^*$. The order of S (number of elements in S) is $\mathrm{ord}(g)$; hence $\mathrm{ord}(g) \,|\, \phi(n)$. We say that g *generates* S and that S is *cyclic*.

We say g is a *primitive root* of n if g generates all of $\mathbf{Z}_n^*$, that is, every element of $\mathbf{Z}_n^*$ can be written as g raised to some power modulo n. By definition, this holds if and only if $\mathrm{ord}(g) = \phi(n)$. Not every integer n has primitive roots. By Gauss's theorem, the numbers having primitive roots are $1, 2, 4, p^k, 2p^k$, where p is an odd prime and $k \geq 1$. In particular, every prime has primitive roots.

The number of primitive roots of p is $\phi(\phi(p))$. This is because if g is a primitive root of p and $x \in \mathbf{Z}_{\phi(p)}^*$, then g^x is also a primitive root of p. Why? We need to argue that every element h in $\mathbf{Z}_p^*$ can be expressed as $(g^x)^y$ for some y. Since g is a primitive root, we know that $h \equiv g^\ell \pmod{p}$ for some ℓ. We wish to find y such that $g^{xy} \equiv g^\ell \pmod{n}$. By Euler's theorem, this is possible if the congruence equation $xy \equiv \ell \pmod{\phi(n)}$ has a solution y. We know that a solution exists if $\gcd(x, \phi(n)) = 1$. But this is the case since x was chosen to be in $\mathbf{Z}_{\phi(p)}^*$.