DISCRETE MATHEMATICS

W W L CHEN

Chapter 12

PUBLIC KEY CRYPTOGRAPHY

12.1. Basic Number Theory

A hugely successful public key cryptosystem is based on two simple results in number theory and the currently very low computer speed. In this section, we shall discuss the two results in elementary number theory.

PROPOSITION 12A. *Suppose that $a, m \in \mathbb{N}$ and $(a, m) = 1$. Then there exists $d \in \mathbb{Z}$ such that $ad \equiv 1 \pmod{m}$.*

Proof. Since $(a, m) = 1$, it follows from Proposition 4J that there exist $d, v \in \mathbb{Z}$ such that $ad + mv = 1$. Hence $ad \equiv 1 \pmod{m}$. ○

Definition. The Euler function $\phi : \mathbb{N} \to \mathbb{N}$ is defined for every $n \in \mathbb{N}$ by letting $\phi(n)$ denote the number of elements of the set

$$S_n = \{x \in \{1, 2, \ldots, n\} : (x, n) = 1\};$$

in other words, $\phi(n)$ denotes the number of integers among $1, 2, \ldots, n$ that are coprime to n.

Example 12.1.1. We have $\phi(4) = 2$, $\phi(5) = 4$ and $\phi(6) = 2$.

Example 12.1.2. We have $\phi(p) = p - 1$ for every prime p.

Example 12.1.3. Suppose that p and q are distinct primes. Consider the number pq. To calculate $\phi(pq)$, note that we start with the numbers $1, 2, \ldots, pq$ and eliminate all the multiples of p and q. Now among these pq numbers, there are clearly q multiples of p and p multiples of q, and the only common multiple of both p and q is pq. Hence $\phi(pq) = pq - p - q + 1 = (p - 1)(q - 1)$.

PROPOSITION 12B. *Suppose that $a, n \in \mathbb{N}$ and $(a, n) = 1$. Then $a^{\phi(n)} \equiv 1 \pmod{n}$.*

PROOF. Suppose that

$$S_n = \{r_1, r_2, \ldots, r_{\phi(n)}\}$$

is the set of the $\phi(n)$ distinct numbers among $1, 2, \ldots, n$ which are coprime to n. Since $(a, n) = 1$, the $\phi(n)$ numbers

$$ar_1, ar_2, \ldots, ar_{\phi(n)} \tag{1}$$

are also coprime to n.

We shall first of all show that the $\phi(n)$ numbers in (1) are pairwise incongruent modulo n. Suppose on the contrary that $1 \le i < j \le \phi(n)$ and

$$ar_i \equiv ar_j \pmod{n}.$$

Since $(a, n) = 1$, it follows from Proposition 12A that there exists $d \in \mathbb{Z}$ such that $ad \equiv 1 \pmod{n}$. Hence

$$r_i \equiv (ad)r_i \equiv d(ar_i) \equiv d(ar_j) \equiv (ad)r_j \equiv r_j \pmod{n},$$

clearly a contradiction.

It now follows that each of the $\phi(n)$ numbers in (1) is congruent modulo n to precisely one number in S_n, and vice versa. Hence

$$r_1 r_2 \ldots r_{\phi(n)} \equiv (ar_1)(ar_2) \ldots (ar_{\phi(n)}) \equiv a^{\phi(n)} r_1 r_2 \ldots r_{\phi(n)} \pmod{n}. \tag{2}$$

Note now that $(r_1 r_2 \ldots r_{\phi(n)}, n) = 1$, so it follows from Proposition 12A that there exists $s \in \mathbb{Z}$ such that $r_1 r_2 \ldots r_{\phi(n)} s \equiv 1 \pmod{n}$. Combining this with (2), we obtain

$$1 \equiv r_1 r_2 \ldots r_{\phi(n)} s \equiv a^{\phi(n)} r_1 r_2 \ldots r_{\phi(n)} s \equiv a^{\phi(n)} \pmod{n}$$

as required. ◯

12.2. The RSA Code

The RSA code, developed by Rivest, Shamir and Adleman, exploits Proposition 12B above and the fact that computers currently take too much time to factorize numbers of around 200 digits.

The idea of the RSA code is very simple. Suppose that p and q are two very large primes, each of perhaps about 100 digits. The values of these two primes will be kept secret, apart from the code manager who knows everything. However, their product

$$n = pq$$

will be public knowledge, and is called the modulus of the code. It is well known that

$$\phi(n) = (p-1)(q-1),$$

but the value of $\phi(n)$ is again kept secret. The security of the code is based on the fact that one needs to know p and q in order to crack it. Factorizing n to obtain p and q in any systematic way when n is of some 200 digits will take many years of computer time!

REMARK. To evaluate $\phi(n)$ is a task that is as hard as finding p and q. However, it is crucial to keep the value of $\phi(n)$ secret, although the value of n is public knowledge. To see this, note that

$$\phi(n) = pq - (p+q) + 1 = n - (p+q) + 1,$$

so that

$$p + q = n - \phi(n) + 1.$$

But then

$$(p-q)^2 = (p+q)^2 - 4pq = (n - \phi(n) + 1)^2 - 4n.$$

It follows that if both n and $\phi(n)$ are known, it will be very easy to calculate $p+q$ and $p-q$, and hence p and q also.

Each user j of the code will be assigned a public key e_j. This number e_j is a positive integer that satisfies

$$(e_j, \phi(n)) = 1,$$

and will be public knowledge. Suppose that another user wishes to send user j the message $x \in \mathbb{N}$, where $x < n$. This will be achieved by first looking up the public key e_j of user j, and then enciphering the message x by using the enciphering function

$$E(x, e_j) = x^{e_j} \equiv c \pmod{n} \qquad \text{and} \qquad 0 < c < n,$$

where c is now the encoded message. Note that for different users, the coded message c corresponding to the same message x will be different due to the use of the personalized public key e_j in the enciphering process.

Each user j of the code will also be assigned a private key d_j. This number d_j is an integer that satisfies

$$e_j d_j \equiv 1 \pmod{\phi(n)},$$

and is known only to user j. Because of the secrecy of the number $\phi(n)$, it again takes many years of computer time to calculate d_j from e_j, so the code manager who knows everything has to tell each user j the value of d_j. When user j receives the encoded message c, this message is deciphered by using the deciphering function

$$D(c, d_j) = c^{d_j} \equiv y \pmod{n} \qquad \text{and} \qquad 0 < y < n,$$

where y is the decoded message.

Observe that the condition $e_j d_j \equiv 1 \pmod{\phi(n)}$ ensures the existence of an integer $k_j \in \mathbb{Z}$ such that $e_j d_j = k_j \phi(n) + 1$. It follows that

$$y \equiv c^{d_j} \equiv x^{e_j d_j} = x^{k_j \phi(n) + 1} = (x^{\phi(n)})^{k_j} x \equiv x \pmod{n},$$

in view of Proposition 12B. It follows that user j gets the intended message.

We summarize below the various parts of the code. Here $j = 1, 2, \ldots, k$ denote all the users of the system.

Public knowledge	Secret to user j	Known only to code manager
n; $e_1, \ldots, e_k$	d_j	p, q; $\phi(n)$

Note that the code manager knows everything, and is therefore usually a spy!

REMARK. It is important to ensure that $x^{e_j} > n$, so that c is obtained from x by exponentiation and then reduction modulo n. If $x^{e_j} < n$, then since e_j is public knowledge, recovering x is simply a task of taking e_j-th roots. We should therefore ensure that $2^{e_j} > n$ for every j. Only a fool would encipher the number 1 using this scheme.

We conclude this chapter by giving an example. For obvious reasons, we shall use small primes instead of large ones.

EXAMPLE 12.2.1. Suppose that $p = 5$ and $q = 11$, so that $n = 55$ and $\phi(n) = 40$. Suppose further that we have the following:

User 1	$e_1 = 23$	$d_1 = 7$
User 2	$e_2 = 9$	$d_2 = 9$
User 3	$e_3 = 37$	$d_3 = 13$

Note that $23 \cdot 7 \equiv 9 \cdot 9 \equiv 37 \cdot 13 \equiv 1 \pmod{40}$.

- Suppose first that $x = 2$. Then we have the following:

$$\begin{aligned} \text{User 1 :} \quad & c \equiv 2^{23} = 8388608 \equiv 8 \pmod{55} \\ & y \equiv 8^7 = 2097152 \equiv 2 \pmod{55} \\ \text{User 2 :} \quad & c \equiv 2^9 = 512 \equiv 17 \pmod{55} \\ & y \equiv 17^9 = 118587876497 \equiv 2 \pmod{55} \\ \text{User 3 :} \quad & c \equiv 2^{37} = 137438953472 \equiv 7 \pmod{55} \\ & y \equiv 7^{13} = 96889010407 \equiv 2 \pmod{55} \end{aligned}$$

- Suppose next that $x = 3$. Then we have the following:

$$\begin{aligned} \text{User 1 :} \quad & c \equiv 3^{23} = 94143178827 \equiv 27 \pmod{55} \\ & y \equiv 27^7 = 10460353203 \equiv 3 \pmod{55} \\ \text{User 2 :} \quad & c \equiv 3^9 = 19683 \equiv 48 \pmod{55} \\ & y \equiv 48^9 \equiv (-7)^9 = -40353607 \equiv 3 \pmod{55} \\ \text{User 3 :} \quad & c \equiv 3^{37} \equiv 3^{37}(3 \cdot 37)^3 \equiv 3^{40}37^3 \equiv 37^3 = 50603 \equiv 53 \pmod{55} \\ & y \equiv 53^{13} \equiv (-2)^{13} = -8192 \equiv 3 \pmod{55} \end{aligned}$$

REMARK. We have used a few simple tricks about congruences to simplify the calculations somewhat. These are of no great importance here. In practice, all the numbers will be very large, and all calculations will be carried out by computers.

Problems for Chapter 12

1. Consider an RSA code with primes 11 and 13. It is important to ensure that each public key e_j satisfies $2^{e_j} > n$. How many users can there be with no two of them sharing the same private key?

2. Suppose that you are required to choose two primes to create an RSA code for 50 users, with the two primes p and q differing by exactly 2. How small can you choose p and q and still ensure that each public key e_j satisfies $2^{e_j} > n$ and no two of users sharing the same private key?