INTRODUCTION TO CRYPTOGRAPHY (PUBLIC KEY CRYPTOGRAPHY)

Preliminaries:

Euler's function: Recall that for any integer $n \geq 1$, Euler's function $\phi(n)$ denotes the number of positive integers not exceeding n and relatively prime to it. (We count 1 as relatively prime to all numbers).

Let's look at the properties of Euler's function:

Problem 1. Show that

(1) $\phi(p) = p - 1$ for p prime;
(2) $\phi(p^k) = p^k - p^{k-1}$, where p is a prime, and $k \geq 1$ is an integer;

More generally, one can show that $\phi(m \cdot n) = \phi(m) \cdot \phi(n)$, i.e., $\phi(n)$ is a multiplicative function (more complicated). This implies the following formula for the Euler's function $\phi(n)$ of the number $n = p_1^{k_1} p_2^{k_2} \dots p_r^{k_r}$:

$$\begin{aligned} \phi(n) &= (p_1^{k_1} - p_1^{k_1 - 1}) \cdot \dots \cdot (p_r^{k_r} - p_r^{k_r - 1}) = \\ &= n \left(1 - \frac{1}{p_1}\right) \cdot \dots \left(1 - \frac{1}{p_r}\right) \end{aligned}$$

One of the main uses of this number-theoretic function comes from **Euler's theorem:**

$$a^{\varphi(n)} \equiv 1 \bmod n,$$

where a and n are relatively prime.

This is a generalization of Fermat's little theorem:

$$a^{p-1} \equiv 1 \bmod p,$$

where p is prime, and a is relatively prime to a.

Public key cryptography

***RSA system* (A.Rivest, A. Shamir, L. Adleman, 1977). Main idea:** it's hard (very time-consuming) to factor large composite numbers.

How it works:

Preparation:

- Two users select a pair of distinct primes, p and q. The numbers should be very large so that factoring their product $n = pq$ is beyong current computational capabilities. This number n is called the *enryption modulus.*

- Choose an integer k called the *encryption exponent,* so that $gcd(k, \phi(n)) = 1$. (This necessity of this condition is explained later, in the decryption section. Here, $\phi(k)$ is the Euler's function). In particular, any prime larger then both p and q works.
- The information (n, k) is publicly available. However, the factors of n (the numbers p and q are not.

ENCRYPTION:
Convert the plaintext into a string of numbers by assigning letters the numerical value of their place in the alphabet, and to punctuation signs some agreed upon numbers. (Plaintext is assumed to be shorter then the encryption modulus). If the message is too long, it can be broken into blocks of digits of appropriate size.
If P is the plaintext, the encryption C is given by

$$C \equiv P^k \bmod n$$

DECRYPTION:
It would be great to have a number j such that $C^j \equiv P \bmod n$. In other words, j should be such that

$$C^j \equiv (P^k)^j = P^{kj} \equiv P \bmod n.$$

Recall that if P is relatively prime to n, Euler's theorem states that

$$P^{\phi(n)} \equiv P \bmod n.$$

Thus the condition above would be satisfied if

$$kj \equiv 1 \bmod \phi(n),$$

i.e., j should be the inverse to k modulo $\phi(n)$. Such a j exists of we assume k and $\phi(n)$ to be relatively prime. This gives rise to the following decryption procedure:

- First, find the *recovery exponent* j, which is the number such that

  $$kj \equiv 1 \bmod \phi(n).$$

  Since $\gcd(k, \phi(n)) = 1$, this linear congruence has a unique solution modulo $\phi(n) = (p-1) \cdot (q-1)$. Thus, you need to know the prime factors of n to find the recovery exponent. This property means that $kj = 1 + t \cdot \phi(n)$ for some integer t.
- Now get P from C by simply computing $C^j \bmod n$. This works because

  $$C^j \equiv (P^k)^j \equiv P^{1+\phi(n)t} \equiv P \cdot (P^{\phi(n)})^t \equiv P \bmod n$$

  whenever $\gcd(P, n) = 1$. (Simply speaking, to recover the plaintext, raise the ciphertext to the jth power and then reduce modulo n. Notice also that in the last step we have used Euler's theorem: $P^{\phi(n)} = 1 \bmod n$).

Problem 2. Let $p = 29$ and $q = 53$. Then the encryption modulus is $n = 29 \cdot 53 = 1537$ and $\phi(n) = 28 \cdot 52 = 1456$. Let $k = 47$ be the encryption

exponent.
(a) Find the recovery exponent j by solving the congruence

$$kj \equiv 1 \ (\mathrm{mod}\phi(n)).$$

(b) The message NO WAY corresponds to the following plaintext number:

$$P = 131499220024.$$

Since each plaintext block should be an integer less than 1537, let's split P into blocks of three digits each.

Find the corresponding ciphertext number.

The Knapsack cryptosystem. The **Kanpsack problem** is the following problem: given a knapsack of volume V and n items of various volumes $a_1, a_2, \ldots, a_n$, can a subset of these itesm be found that will completely fill the knapsack?

In other words: solve the equation

$$\sum_{i=1}^{n} a_i x_i = V$$

for given $0 < a_1 < \cdots < a_n$ and V with respect to x_i's, where the allowed values of x_i's are 0 and 1.

We will denote such a problem by $(a_1, \ldots, a_n; V)$ for brevity.

Example 3. The knapsack problem

$$22 = 3x_1 + 7x_2 + 9x_3 + 11x_4 + 20x_5$$

has no solutions.

The problem

$$27 = 3x_1 + 7x_2 + 9x_3 + 11x_4 + 20x_5$$

has two distinct solutions:

$$x_2 = x_3 = x_4 = 1, \quad x_1 = x_5 = 0$$

and

$$x_2 = x_5 = 1, \quad x_1 = x_3 = x_4 = 0.$$

Finding solution to a randomly chosen knapsack problem is difficult.

Problem 4. (1) How many choices (possibilities) do you have to try to solve a knapsack problem with n items?

(2) Invent a problem that has at least two distinct solutions.

A knapsack problem is called *superincreasing* if the coeeficients satisfy the condition

$$a_i > a_1 + \cdots + a_{i-1}, \qquad i = 2, 3, \ldots, n.$$

Problem 5. Solve the following *superincreasing* knapsack problem:

$$3x_1 + 5x_2 + 11x_3 + 20x_4 + 41x_5.$$

asdfa

Problem 6. Consider the knapsack problem of the form

$$V = x_1 + 2x_2 + 4x_3 + \ldots 2^n x_n,$$

where $a_k = 2^k$ for all k, and $V < 2^{n+1}$.

Solve this system. What does x_k represent?

Problem 7. Describe a procedure of solving a general superincreasing knapsack problem.

Knapsack cryptosystem. Idea: Multiplying coefficients of a knapsack problem by a constant factor and then taking the remainder modulo fixed modulus can change a superincreasing problem (easy to solve) into general one (hard to solve)

PRELIMINARY DATA:

- Select superincreasing sequence $a_1, \ldots, a_n$; an encryption modulus m and a multiplier $k \in (0, m)$ such that $m > 2a_n$ and $\gcd(k, m) = 1$. (The last condition guarantees that there k has an inverse, j, with respect to modulus m);
- Multiply each element of $(a_1, \ldots, a_n)$ by k and take the remainder modulo m to get a new knapsack problem with coefficients

$$b_i \equiv ka_i \mod m.$$

ENCRYPTION (PUBLIC KEY=$(b_1, \ldots, b_n)$)

- Convert the plaintext message into a string P of 0's and $1's$ using the binary equivalent of letters.
- Split P into blocks of n digits (with the last block being filled out by 1s if n ecessary).
- Use the public encrypting system $(b_1, \ldots, b_n)$ to transform a given plaintext block $p_1 \ldots p_n$into the sum

$$S = b_1x_1 + \cdots + b_nx_n.$$

  The numbers S can be communicated through an insecure communication channel.
- Because a general knapsack problem is hard to solve, decoding (without knowing $(a_1, \ldots, a_n)$) is very hard.

DECRYPTION (PRIVATE KEY=$(a_1, \ldots, a_n), \quad m, k$)

- Conver the hard knapsack problem $(S, ; b, \ldots, b_n)$ into a superincreasing one as follows. Let

$$S' \equiv j \cdot S \mod m,$$

  where $j \equiv k^{-1} \mod m$. Since $m > 2a_n > a_1 + \ldots a_n$, it follows that

$$S' = a_1x_1 + \cdots + a_na_n,$$

  and $0 \leq S' < m$.
- The solution to the above superincreasing problem give the solutions to the difficult problem. The plaintext block $x_1 \ldots x_n$ of n digits is recovered from S.

Problem 8. Suppose that $(a_1, \ldots, a_5) = (3, 5, 11, 20, 41)$; $m = 85$ and $k = 44$. Then

$$(b_1, \ldots, b_5) = (47, 50, 59, 30, 190)$$

is the public enryption key.

(1) Solve the congruence $44x = 1 \bmod 85$ to get $j = k^{-1} \bmod 85$.

(2) Convert the message HELP US into a string of 0s and 1s:

(3) Encrypt the message using the encryption key above:

(4) After that, to decode one needs to multiply each ciphertext number by 29 and reduce modulo 85 to produce a superincreasing knapsack problem. Perform this operation for the first block of numbers.

(5) Recover the first block of the binary equivalent of the plaintext. Did you get what you expected to get?

This cryptosystem (intriduced by Merkle and Hellman in 1978) was later found not very secure. In 1982, A. Shamir found a fast algorithm for solving knapsack problems with coefficients obtained by multiplying coefficients of a superincreasing sequence by a constant factor and then reducing modulo a given modulus. The system can be made more secure by iteratnig the modular multiplication method with different values of (a, m). Some versions of this system are still in use today.