Lecture 10: Public Key Cryptography

Lecturer: Kurt Mehlhorn & He Sun

Today's lecture is about the Public Key Cryptography. We first discuss general ideas in designing cryptography protocols. Typically, the encryption scheme is a pair of algorithms, encryption algorithm and decryption algorithm. When sending a message, the sender fist uses the encryption algorithm to encode the message, and send the encoded messages, called *ciphertext*, over the channel. Upon receiving a ciphertext, the receiver applies the decryption algorithm to decode the message, and receive the original message.

Some possible approaches in designing cryptographic protocols:

- Use a dictionary as the key
- Use a random sequence as the key
- Use a pseudorandom generator and a random seed as the key

1 Modular Arithmetic

Definition 1. *The number a is equivalent (congruent) to the number b modulo n, expressed by $a \equiv b \pmod{n}$, if a differs from b by an exact multiple of n.*

Lemma 2. *The following statements hold:*

- *If $a \equiv b \pmod{n}$ and $c \equiv d \pmod{n}$, then $a + c \equiv b + d \pmod{n}$.*
- *If $a \equiv b \pmod{n}$ and $c \equiv d \pmod{n}$, then $ac \equiv bd \pmod{n}$.*

Example. $321 \times 741 \equiv 1 \times 1 \equiv 1 \pmod{5}$.

Example. $715^{10000} \equiv 1 \pmod{7}$.

Example. $321^3 \equiv 6^3 \pmod{7} = 36 \times 6 \pmod{7} \equiv 6 \pmod{7}$.

Example. $320^{984} \equiv 1 \pmod{7}$

Let us look at the solutions of the example above. We first write down the binary expression of 984, i.e.

$$\begin{aligned} 984 &= 512 + 256 + 128 + 64 + 16 + 8 \\ &= 2^9 + 2^8 + 2^7 + 2^6 + 2^4 + 2^3. \end{aligned}$$

Note that $320^9 84 \equiv 5^{984} \pmod{7}$. Moreover, we have the following:

- $5^2 = 25 \equiv 4 \pmod{7}$
- $5^4 = 4 \times 4 \pmod{7} \equiv 2 \pmod{7}$

- $5^8 = 2 \times 2 \pmod 7 \equiv 4 \pmod 7$
- $5^{16} = 4 \times 4 \pmod 7 = 2 \pmod 7$
- $5^{32} \equiv 4 \pmod 7$
- $5^{64} \equiv 2 \pmod 7$
- $5^{128} \equiv 4 \pmod 7$
- $5^{256} \equiv 2 \pmod 7$
- $5^{512} \equiv 4 \pmod 7$

Hence

$$\begin{aligned} 5^{984} &= 5^{512+256+128+64+16+8} \pmod 7 \\ &\equiv 4 \times 2 \times 4 \times 2 \times 2 \times 2 \times 4 \pmod 7 \\ &\equiv 1 \pmod 7 \end{aligned}$$

2 Fermat's Little Theorem

We call that n is divisible by m if $n = km$.

Theorem 3 (Fermat's Little Theorem)**.** *If p is a prime number, then $a^p \equiv a \pmod p$ for all a.*

An alternative formulation of the Fermat's Little Theorem is as follows: If p is a prime number and a is any integer not divisible by p, then $a^{p-1} \equiv 1 \pmod p$.

Definition 4. *If $ab \equiv 1 \pmod m$, then b is called the multiplicative inverse of a modulo m.*

3 The Euclidean Algorithm

Given two integers r_0 and r_1, the Euclidean Algorithm finds the greatest common divisor of r_0 and r_1, denoted by $\gcd(r_0, r_1)$.

Before present the algorithm, we first look at the following lemma.

Lemma 5. $\gcd(r_0, r_1) = \gcd(r_1, r_0 \mod r_1)$

Theorem 6 (The Euclidean Algorithm)**.** *Given two integers $0 < b < a$, we make a repeated application of the division algorithm to obtain a series of division equations, which eventually terminate with a zero remainder:*

$$\begin{aligned} a &= bq_1 + r_1, 0 < r_1 < b \\ b &= r_1 q_2 + r_2, 0 < r_2 < r_1 \\ &\cdots \\ r_{j-2} &= r_{j-1} q_j + r_j, 0 < r_j < r_{j-1} \\ r_{j-1} &= r_j q_{j+1} \end{aligned}$$

The greatest common divisor $\gcd(a, b)$ of a and b is r_j, the last nonzero remainder in the division process.

Now we show that the Euclidean Algorithm can be used to compute a multiplicative inverse.

Definition 7. *If* $ab \equiv 1 \pmod{p}$*, then* b *is called the multiplicative inverse of* a *module* p*.*

Theorem 8 (Multiplicative Inverse Algorithm)**.** *Given two integers* $0 < b < a$*, consider the Euclidean Algorithm equations which yield* $\gcd(a, b) = r_j$*. Rewrite all of these equations except the last one, by solving for the remainders:*

$$\begin{aligned} r_1 &= a - bq_1 \\ r_2 &= b - r_1 q_2 \\ r_3 &= r_1 - r_2 q_3 \\ &\cdots \\ r_{j-1} &= r_{j-3} - r_{j-2} q_{j-1} \\ r_j &= r_{j-2} - r_{j-1} q_j \end{aligned}$$

Then, in the last of these equations, $r_j = r_{j-2} - r_{j-1}q_j$*, replace* r_{j-1} *with its expression in terms of* r_{j-3} *and* r_{j-2} *from the equation immediately above it. Continue this process successively, replacing* $r_{j-2}, r_{j-3}, \cdots$*, until we obtain the final equation*

$$r_j = ax + by,$$

where x *and* y *are integers. In the special case that* $\gcd(a, b) = 1$*, the integer equation reads*

$$1 = ax + by.$$

Therefore we deduce

$$1 \equiv by \mod a$$

so that the residue of y *is the multiplicative inverse of* b*, mod* a*.*

4 The RSA Algorithm

- p, q are two prime numbers
- Let $n = p \cdot q$
- Pick a positive integer r that has no common factor with $(p-1) \cdot (q-1)$
- Find a multiplicative inverse of r modulo $(p-1) \cdot (q-1)$, i.e. we find a number s such that $rs \equiv 1 \pmod{(p-1) \cdot (q-1)}$

Encryption: We need to know n, r. Assume that the message is $x \leq n$. Let

$$y \triangleq x^r \pmod{n}.$$

The pair of values n, r are **public encryption key**. This information is publicly available, and anyone can compute y if they are given x.

Decryption: To decrypt, you need to know s, the **private decryption key**. With the value of s, you simply compute

$$z \triangleq y^s \pmod{n} \equiv x^{rs} \equiv x \pmod{n}.$$

That is, you need to know s to decrypt. Now s is the multiplicative inverse of r modulo $(p-1)(q-1)$. The outsiders know r, and if they knew $(p-1)(q-1)$, then it would be easy (with the Euclidean Algorithm) to compute s. But they do not know $(p-1)(q-1)$. They know n, which is equal to pq, but they do not have n factored into p and q. To find $(p-1)(q-1)$, they need to know the prime factors p and q of n, and factoring large numbers is not easy.