Cryptography

Cunsheng DING, HKUST

COMP3631

Lecture 10: Introduction to Public-Key Cryptography

Objectives of This Lecture

1. Introduce the idea of public-key cryptography.
2. Outline the history of public-key cryptography.
3. Introduce some elementary number theory.

A Disadvantage of One-Key Ciphers

A Disadvantage of One-Key Ciphers

One-key cipher: $(\mathcal{M}, \mathcal{C}, \mathcal{K}, E_k, D_k)$ or $(\mathcal{M}, \mathcal{C}, \mathcal{K}, E_k, D_k, u)$, where the encryption and decryption key are the same.

- The sender and receiver must share the same secret key. Key generation and distribution are a must.
- If 10000 people want to communicate (two and two, in all possible ways), each must keep 9999 secret keys, and the system requires a total of

$$9999 \cdot 10000/2 = 4995000$$

secret keys. This makes key management difficult.

Question: Is there any cipher whose key management is easy when it is used by a large group of people?

Answer: Yes, public-key ciphers.

The Idea of Public-Key Cryptography

Two-key Ciphers

A six-tuple $(\mathcal{M}, \mathcal{C}, \mathcal{K}_e, \mathcal{K}_d, E_{k_e}, D_{k_d})$, where

- $\mathcal{M}$, $\mathcal{C}$, $\mathcal{K}_e, \mathcal{K}_d$ are respectively the plaintext space, ciphertext space, encryption key space, and decryption key space;
- $k_e \in \mathcal{K}_e$ and $k_d \in \mathcal{K}_d$ are corresponding encryption and decryption keys respectively, and $k_e \neq k_d$ **in general**;
- E_{k_e} and D_{k_d} are the encryption and decryption transformations, and

$$D_{k_d}(E_{k_e}(m)) = m,$$

for all $m \in \mathcal{M}$ (unique and correct decryption).

A usage of two-key ciphers: A group of people generate and share a pair (k_e, k_d) for encryption and decryption. Both keys are kept confidential.

The Idea of Public-Key Cryptography

Suppose that our university has a two-key cipher $(\mathcal{M}, \mathcal{C}, \mathcal{K}_e, \mathcal{K}_d, E_{k_e}, D_{k_d})$, which is in the public domain.

I generate my encryption and decryption key pair (k_e, k_d), and then publicize my encryption key k_e in the public domain, in order for anybody else to encrypt a message and send it to me. But I keep my decryption key k_d confidential. **Everyone in our university does the same.**

A two-key cipher used in this special way is called a **public-key cipher**.

Comment: The encryption key k_e is called the **public key**, and the decryption key k_d is called the **private key**, which must be kept confidential by its holder.

The Security of Public-Key Ciphers

A public-key cipher $(\mathcal{M}, \mathcal{C}, \mathcal{K}_e, \mathcal{K}_d, E_{k_e}, D_{k_d})$ is **computationally-secure** if the following two conditions are satisfied:

C1: it is "computationally infeasible" to derive the private key k_d from the public key k_e; and

C2: it is "computationally infeasible" to derive the plaintext m if the corresponding ciphertext c is known.

- In theory, a public key k_e may contain all information about the private key k_d. But it should be computationally infeasible to derive k_d from k_e.
- E_{k_e} is known to everyone. So, its inverse function D_{k_d} is also known in theory. But it should be computationally infeasible to derive the decryption function D_{k_d} from the known E_{k_e} and k_e.

A Public-key Cipher not Satisfying C1 & C2

Matrix: An $n \times m$ matrix $A = [a[i,j]]$ over $\mathbf{Z}_2 = \{0,1\}$ is a 2-dimensional array of the form:

$$A = \begin{bmatrix} a[1,1] & a[1,2] & \cdots & a[1,m-1] & a[1,m] \\ a[2,1] & a[2,2] & \cdots & a[2,m-1] & a[2,m] \\ \vdots & \vdots & & \vdots & \vdots \\ a[n,1] & a[n,2] & \cdots & a[n,m-1] & a[n,m] \end{bmatrix},$$

which has n rows and m columns, and each $a[i,j] \in \{0,1\}$.

A Public-key Cipher not Satisfying C1 & C2

Given an $n \times m$ matrix A and an $m \times l$ matrix B, the multiplication $C = AB$ over $\mathbf{Z}_2$ is an $n \times l$ matrix given by

$$c[i,j] = \sum_{k=1}^{m} a[i,k]b[k,j]$$

for $1 \leq i \leq n$ and $1 \leq j \leq l$, where operations in the sum are mudulo-2 additions and mudulo-2 multiplications.

A Public-key Cipher not Satisfying C1 & C2

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix},$$

then

$$C = AB = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

A Public-key Cipher not Satisfying C1 & C2

Definition: Let A be an $n \times n$ matrix over $\mathbf{Z}_2$. If there exists an $n \times n$ matrix B over $\mathbf{Z}_2$ such that $AB = I_n$, i.e., the $n \times n$ **identity matrix**, then A is said to be **invertible**, and B is the **inverse matrix** of A.

Remark: In general, $A \neq A^{-1}$. But it may happen that $A = A^{-1}$.

Example: A is the inverse of itself:

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad A^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

A Public-key Cipher not Satisfying C1 & C2

Let $\mathcal{M} = \mathcal{C} = \{0,1\}^*$, all the finite binary strings, and let $\mathcal{K}_e = \mathcal{K}_d := \mathcal{K}$ be the set of all invertible 512×512 matrices k over $\mathbf{Z}_2 = \{0,1\}$ with $k \neq k^{-1}$. Public-private key pair $(k_e, k_d) = (k, k^{-1})$. Each plaintext is broken into blocks of length 512 bits. The encryption function and decryption function work on blocks.

Encryption and decryption function: For a 512-bit plaintext block x and the ciphertext block y,

$$y := E_{k_e}(x) = xk, \quad D_{k_d}(y) = yk^{-1},$$

where all the arithmetic operations involved in computing xk are the modulo-2 additions and modulo-2 multiplications.

Comment: C1 and C2 are not statisfied. Why?

Design Requirements for Public-Key Ciphers

The security requirements C1 and C2 described before plus the following efficiency requirements:

1. It is "computationally easy" for a party B to generate a pair $\left(k_e^{(B)}, k_d^{(B)}\right)$.
2. It is "computationally easy" for a sender A, knowing the public key and a plaintext m to be encrypted, to generate the corresponding ciphertext $c = E_{k_e^{(B)}}(m)$.
3. It is "computationally easy" for the receiver B to recover the plaintext $m = D_{k_d^{(B)}}(c)$.

Existence and Construction Problems

Question: Is there any public-key cipher meeting the five requirements described in the previous page?

Answer: Several public-key ciphers in the literature are believed to meet these requirements. But there is no rigorous proof for the belief.

How does one construct a public-key cipher?
Use a problem that is believed to be hard to solve, e.g., the "discrete logarithm problem" (see Lecture 12).

Advantages and Disadvantages

- With a public-key cipher, a user does not need to share many keys with others. This is an advantage of public-key ciphers over private-key ciphers.
- The **disadvantage** of public-key ciphers is their performance in hardware and software, as there is **no** efficient and secure public-key cipher in the public domain.

  —- It is very hard to design an efficient and secure public-key cipher, as the public key is publicized in the public domain.

  —- To satisfy Condition C1, the relation between k_e and k_d must be complicated.

  —- To make it hard to derive D_{k_d} from E_{k_e}, the time complexity of the encryption algorithm E_{k_e} cannot be small.

History of Public-Key Cryptography

History of Public-Key Cryptography (I)

- The idea of public-key cryptography was published by W. Diffie and M. Hellman, and independently by R. Merkle in 1976. It is regarded as a REVOLUTION in the history of cryptography!
- Admiral Bobby Inman, while director of the NSA, claimed that public-key cryptography had been discovered at NSA in the mid-1960s.
- The first (???) *documented* introduction of these concepts was given in 1970 by the Communications-Electronics Security Group, Britain's counterpart of NSA, in a classified report by James Ellis.

History of Public-Key Cryptography (II)

- The Knapsack public-key cipher was developed by Ralph Merkle and Martin Hellman in 1978, but was broken in 1982 by Shamir and Zipple.
- In the same year (1978), another public-key block cipher was invented by Ron **R**ivest, Adi **S**hamir, and Leonard **A**dleman. It is known as RSA. It is easy to understand and to implement, and is one of a few that are still regarded as secure. It is widely used in real-world security systems.
- Many other public-key ciphers have been proposed. Most of them have been broken.

Some Elementary Number Theory

Euler's Totient Function $\phi(n)$

$\phi(n)$**:** The number of positive integers less than n that is relative prime to n. That is,

$$\phi(n) = |\{1 \leq i \leq n-1 : \gcd(i, n) = 1\}|.$$

Example: $\phi(7) = 6$ because

$$\{x : 1 \leq x < 7, \gcd(x, 7) = 1\} = \{1, 2, 3, 4, 5, 6\}.$$

Example: $\phi(6) = 2$ because

$$\{x : 1 \leq x < 6, \gcd(x, 6) = 1\} = \{1, 5\}.$$

Question: What is $\phi(8)$?

Formula for Euler's Totient Function ϕ

Theorem:

- $\phi(p) = p - 1$ for any prime number p.
- $\phi(pq) = (p-1)(q-1)$ for any two distinct primes p and q.

Proof: The first conclusion is straightforward. We now prove the second. Note that pq has only divisors $1, p, q, pq$. The following is the set of integers a such that $1 \leq a < pq$ and $\gcd(a, pq) \neq 1$:

$$\{1p, 2p, \ldots, (q-1)p, \quad 1q, 2q, \ldots, (p-1)q\}$$

which has $(q-1) + (p-1)$ elements. Hence,
$\phi(pq) = pq - 1 - (q-1) - (p-1) = (p-1)(q-1)$.

Fermat's and Euler's Theorem

Euler's Theorem: For every integer a and $n \geq 2$ that are relatively prime,

$$a^{\phi(n)} \bmod n = 1.$$

If $n = p$ is prime, we have **Fermat's Theorem**:

$$a^{p-1} \bmod p = 1.$$

Proof: See, the next slide.

Example: Let $a = 3$ and $n = 10$. Then
$\phi(10) = \phi(2 \times 5) = (2-1)(5-1) = 4$ and

$$a^{\phi(n)} \bmod 10 = 3^4 \bmod 10 = 81 \bmod 10 = 1.$$

Proof of Euler's Theorem

Euler's Theorem: For every integer a and $n \geq 2$ that are relatively prime,

$$a^{\phi(n)} \bmod n = 1.$$

Proof: Define $R = \{1 \leq i < n \mid \gcd(i, n) = 1\}$. By definition, $|R| = \phi(n)$. Since $\gcd(a, n) = 1$, the sets $aR := \{ar \bmod n \mid r \in R\}$ and R are equal. It then follows that

$$\left(\prod_{r \in R} (ar \bmod n)\right) \bmod n = \left(a^{\phi(n)} \prod_{r \in R} r\right) \bmod n = \left(\prod_{r \in R} r\right) \bmod n.$$

Note that the integer $\prod_{r \in R} r$ is relatively prime to n. Multiplying the multiplicative inverse of $\prod_{r \in R} r$ modulo n on both sides of the last equality above yields the desired equality.

Proof of Euler's Theorem: Example

Let $n = 6$. Then $\phi(n) = 2$ and $R := \{1 \le i < n \mid \gcd(i, n) = 1\} = \{1, 5\}$. If a is an integer with $\gcd(a, n) = 1$, then $a \bmod n \in R$. It can be verified that

$$aR := \{ar \bmod n \mid r \in R\} = \{a \times 1 \bmod n, a \times 5 \bmod n\} = R.$$

We have then $\prod_{r \in R} r = 5$ and

$$\left(\prod_{r \in R} (ar \bmod n)\right) \bmod n = \left(a^{\phi(n)} \prod_{r \in R} r\right) \bmod n = \left(a^{\phi(n)} 5\right) \bmod n = 5 \bmod n.$$

Note that the integer $\prod_{r \in R} r = 5$ is relatively prime to n. Multiplying the multiplicative inverse of $\prod_{r \in R} r$ modulo n (i.e., 5) on both sides of the last equality above yields the desired equality:

$$a^{\phi(n)} \bmod n = a^2 \bmod 6 = 1.$$