

Discrete Mathematics for Computer Science

Lecture 9: Cryptography

Dr. Ming Tang

Department of Computer Science and Engineering
Southern University of Science and Technology (SUSTech)
Email: tangm3@sustech.edu.cn



Inverse of a modulo m

Theorem: If a and m are relatively prime integers and $m > 1$, then an inverse of a modulo m exists. The inverse is unique modulo m . That is,

- there is a unique positive integer \bar{a} less than m that is an inverse of a modulo m and
- every other inverse of a modulo m is congruent to \bar{a} modulo m .)

Proof of existence: Since $\gcd(a, m) = 1$, there are integers s and t such that

$$sa + tm = 1.$$

Hence $sa + tm \equiv 1 \pmod{m}$. Since $tm \equiv 0 \pmod{m}$, it follows that $sa \equiv 1 \pmod{m}$. This means that s is an inverse of a modulo m .

Proof of Uniqueness.

How to find inverses?

Using extended Euclidean algorithm.

Number of Solutions to Congruences

Theorem: Let $\gcd(a, m) = d$. Let $m' = m/d$ and $a' = a/d$. The congruence $ax \equiv b \pmod{m}$ has solutions if and only if $d \mid b$.

- If $d \mid b$, then there are exactly d “solutions”, where by “solution” we mean a congruence class mod m .
- If x_0 is a solution, then the other solutions are given by $x_0 + m', x_0 + 2m', \dots, x_0 + (d - 1)m'$.

Proof:

“The number of solutions is d ”: Consider two solutions x_0 and x_1 . $ax_0 \equiv b \pmod{m}$ and $ax_1 \equiv b \pmod{m}$ imply that $m \mid a(x_1 - x_0)$ and $m' \mid a'(x_1 - x_0)$. This implies further that $x_1 = x_0 + km'$.

To finish the proof, observe that as k runs through the values $0, 1, \dots, d - 1$ (the residues mod d), the congruence classes $[x_0 + (m/d)k]_m$ run through all the solutions.

The Chinese Remainder Theorem

Theorem (The Chinese Remainder Theorem): Let m_1, m_2, \dots, m_n be pairwise relatively prime positive integers greater than 1 and a_1, a_2, \dots, a_n arbitrary integers. Then, the system

$$x \equiv a_1 \pmod{m_1}$$

$$x \equiv a_2 \pmod{m_2}$$

...

$$x \equiv a_n \pmod{m_n}$$

has a unique solution modulo $m = m_1 m_2 \dots m_n$.

(That is, there is a solution x with $0 \leq x < m$, and all other solutions are congruent modulo m to this solution.)

The Chinese Remainder Theorem

Proof: To show such a solution exists: Let $M_k = m/m_k$ for $k = 1, 2, \dots, n$ and $m = m_1m_2\dots m_n$. Thus, $M_k = m_1\dots m_{k-1}m_{k+1}\dots m_n$.

Since $\gcd(m_k, M_k) = 1$, there is an integer y_k , an inverse of M_k modulo m_k , such that $M_ky_k \equiv 1 \pmod{m_k}$. Let

$$x = a_1M_1y_1 + a_2M_2y_2 + \dots + a_nM_ny_n.$$

It is checked that x is a solution to the n congruences:

$$x \pmod{m_k} = (a_1M_1y_1 + a_2M_2y_2 + \dots + a_nM_ny_n) \pmod{m_k}$$

Since $M_k = m/m_k$, we have $x \pmod{m_k} = a_kM_ky_k \pmod{m_k}$. Since $M_ky_k \equiv 1 \pmod{m_k}$, we have $a_kM_ky_k \pmod{m_k} = a_k \pmod{m_k}$. Thus,

$$x \equiv a_k \pmod{m_k}.$$

The Chinese Remainder Theorem

How to prove the **uniqueness** of the solution modulo m ?

Proof: Suppose that x and x' are both solutions to all the congruences. As x and x' give the same remainder, when divided by m_k , their difference $x - x'$ is a multiple of each m_k for all $k = 1, 2, \dots, n$.

As m_1, m_2, \dots, m_n be pairwise relatively prime positive integers, their product m divides $x - x'$, and thus x and x' are congruent modulo m , i.e., $x \equiv x' \pmod{m}$.

This implies that given a solution x with $0 \leq x < m$, all other solutions are congruent modulo m to this solution.

Back Substitution

We may also solve systems of linear congruences with pairwise relatively prime moduli m_1, m_2, \dots, m_n by back substitution.

Example:

- (1) $x \equiv 1 \pmod{5}$
- (2) $x \equiv 2 \pmod{6}$
- (3) $x \equiv 3 \pmod{7}$

According to (1), $x = 5t + 1$, where t is an integer.

Substituting this expression into (2), we have $5t + 1 \equiv 2 \pmod{6}$, which means that $t \equiv 5 \pmod{6}$. Thus, $t = 6u + 5$, where u is an integer.

Substituting $x = 5t + 1$ and $t = 6u + 5$ into (3), we have

$30u + 26 \equiv 3 \pmod{7}$, which implies that $u \equiv 6 \pmod{7}$. Thus, $u = 7v + 6$, where v is an integer.

Thus, we must have $x = 210v + 206$. Translating this back into a congruence,

$$x \equiv 206 \pmod{210}.$$

The Chinese Remainder Theorem

What if m_1, m_2, \dots, m_n are positive integers greater than 1, but they are **not** pairwise relatively prime?

$$x \equiv a_1 \pmod{m_1}$$

$$x \equiv a_2 \pmod{m_2}$$

...

$$x \equiv a_n \pmod{m_n}$$

Translate these congruence into a set of congruence that together are equivalent to the given congruence:

For $x \equiv a_k \pmod{m_k}$, suppose m_k can be written as $m_k = b_k^1 b_k^2 \cdots b_k^r$, where $b_k^1, b_k^2, \dots, b_k^r$ are all primes. Then, $x \equiv a_k \pmod{m_k}$ is equivalent to the following set of congruence:

$$x \equiv a_k \pmod{b_k^1}$$

$$x \equiv a_k \pmod{b_k^2}$$

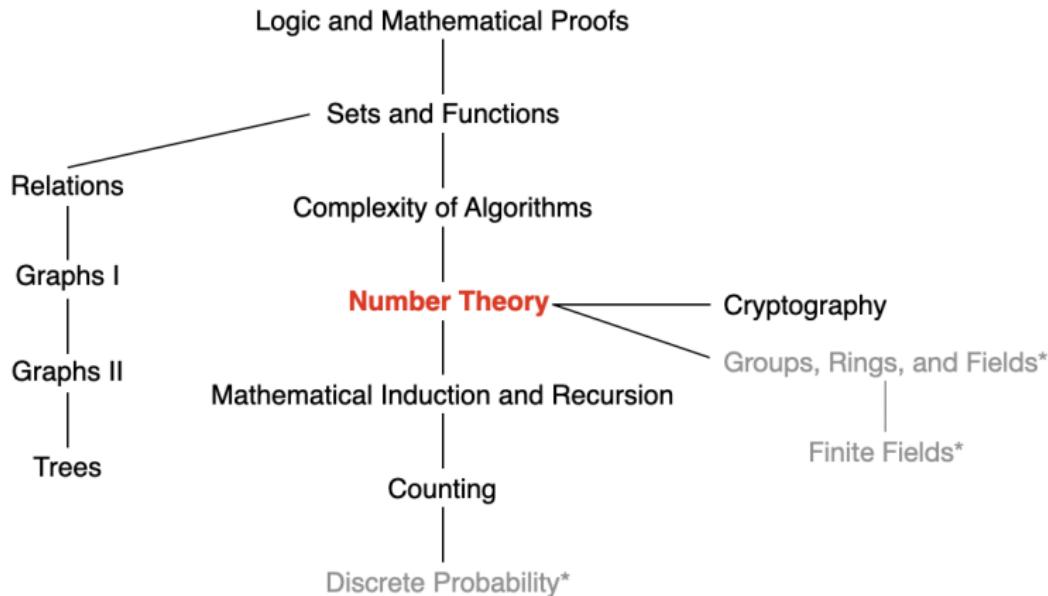
...

$$x \equiv a_k \pmod{b_k^r}$$

Summary: Linear Congruences

- Compute $ax \equiv b \pmod{m}$, with $\gcd(a, m) = 1$
 - ▶ Inverse of a modulo m $a\bar{a} \equiv 1 \pmod{m}$: existence? uniqueness?
 - ▶ Extended Euclidean Algorithm
 - ▶ $x \equiv \bar{a}b \pmod{m}$
- $ax \equiv b \pmod{m}$, with $\gcd(a, m) = d$
- Chinese Remainder Theorem:
 - ▶ $x = a_1M_1y_1 + a_2M_2y_2 + \dots + a_nM_ny_n$.
 - ▶ What if not pairwise relatively prime

This Lecture



Number Theory: divisibility and modular arithmetic, integer representations, primes, greatest common divisors, linear congruences, **application**, ...

Modular Arithmetic in CS

Modular arithmetic and congruencies are used in CS:

- Pseudorandom number generators
- Hash functions
- Cryptography

Pseudorandom Number Generators

Linear congruential method

We choose four numbers:

- the modulus m
- multiplier a
- increment c
- seed x_0

We generate a sequence of numbers $x_1, x_2, \dots, x_n, \dots$ with $0 \leq x_i < m$ by using the congruence

$$x_{n+1} = (ax_n + c) \bmod m$$

Pseudorandom Number Generators

Linear congruential method

$$x_{n+1} = (ax_n + c) \bmod m$$

$m = 9$, $a = 7$, $c = 4$, and $x_0 = 3$:

$$x_1 = 7x_0 + 4 \bmod 9 = 7 \cdot 3 + 4 \bmod 9 = 25 \bmod 9 = 7,$$

$$x_2 = 7x_1 + 4 \bmod 9 = 7 \cdot 7 + 4 \bmod 9 = 53 \bmod 9 = 8,$$

$$x_3 = 7x_2 + 4 \bmod 9 = 7 \cdot 8 + 4 \bmod 9 = 60 \bmod 9 = 6,$$

$$x_4 = 7x_3 + 4 \bmod 9 = 7 \cdot 6 + 4 \bmod 9 = 46 \bmod 9 = 1,$$

$$x_5 = 7x_4 + 4 \bmod 9 = 7 \cdot 1 + 4 \bmod 9 = 11 \bmod 9 = 2,$$

$$x_6 = 7x_5 + 4 \bmod 9 = 7 \cdot 2 + 4 \bmod 9 = 18 \bmod 9 = 0,$$

$$x_7 = 7x_6 + 4 \bmod 9 = 7 \cdot 0 + 4 \bmod 9 = 4 \bmod 9 = 4,$$

$$x_8 = 7x_7 + 4 \bmod 9 = 7 \cdot 4 + 4 \bmod 9 = 32 \bmod 9 = 5,$$

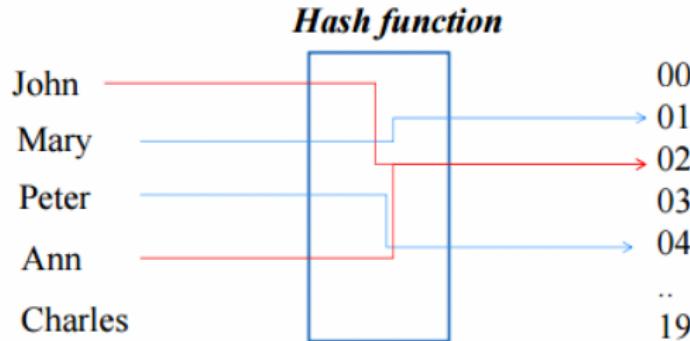
$$x_9 = 7x_8 + 4 \bmod 9 = 7 \cdot 5 + 4 \bmod 9 = 39 \bmod 9 = 3.$$

This sequence contains nine different numbers before repeating.

Hash Functions

A **hash function** is an algorithm that maps data of **arbitrary length** to data of a **fixed length**. The values returned by a hash function are called **hash values** or hash codes.

Example:



Hash Functions

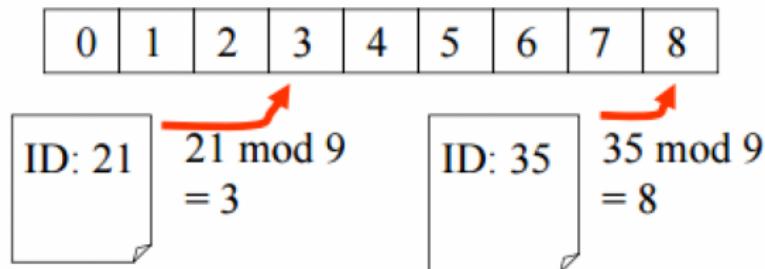
Problem: Given a large collection of records, how can we store and find a record quickly?

Solution: Use a hash function, calculate the [location of the record](#) based on the record's ID.

A common function is

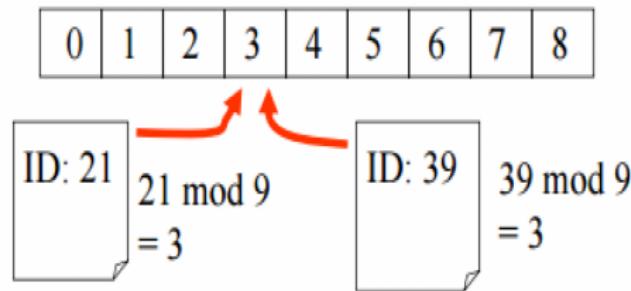
$$h(k) = k \bmod m,$$

where m is the number of available storage locations.



Hash Functions

Two records mapped to the same location



How to address this?

Hash Functions

One way is to assign the **first free location** following the occupied memory location assigned by the hashing function.

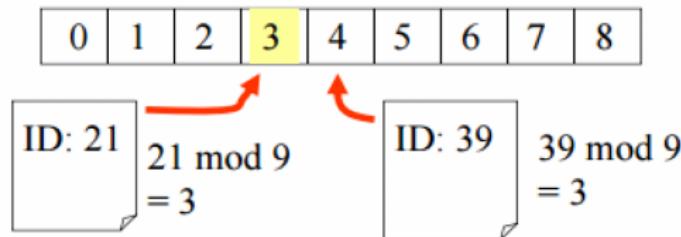
try

$$h_0(k) = k \bmod n$$

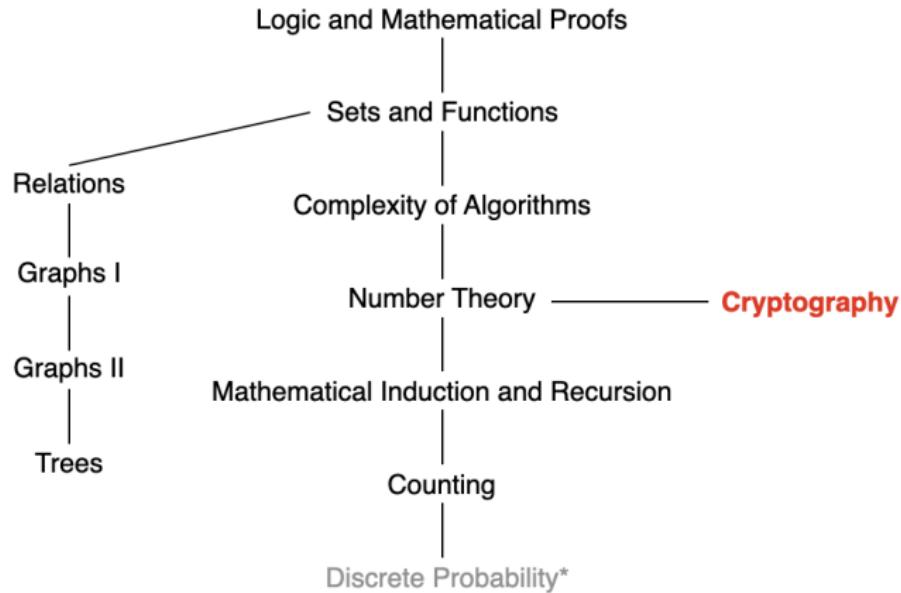
$$h_1(k) = (k+1) \bmod n$$

...

$$h_m(k) = (k+m) \bmod n$$



This Lecture



Cryptography: classical cryptography, RAS cryptosystem



Southern University
of Science and
Technology

Cryptography

History of almost 4000 years (from 1900 B.C.)

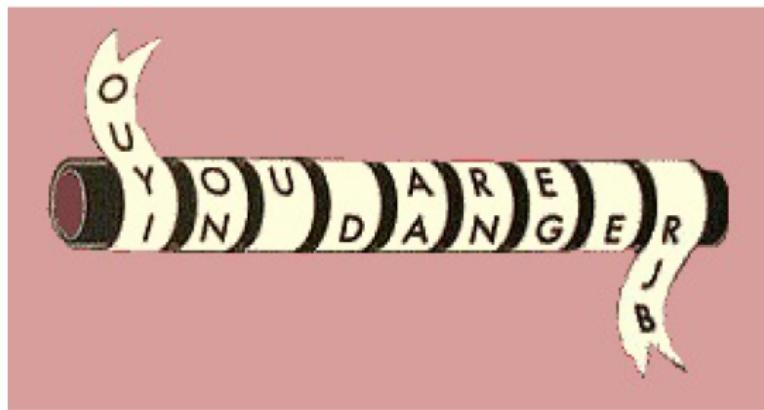
Cryptography = kryptos + graphos

- kryptos: secret
- graphos: writing

One-sentence definition: “[Cryptography](#) is the practice and study of techniques for [secure communication](#) in the presence of third parties called [adversaries](#).” – Ronald L. Rivest

Some Examples

In 405 B.C., the Greek general LYSANDER OF SPARTA was sent a coded message written on the inside of a servant's belt.



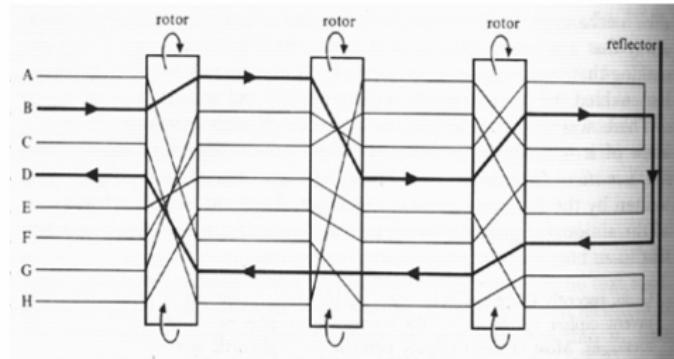
Some Examples

The Greeks also invented a cipher which changed letters to numbers. A form of this code was still being used during World War I.

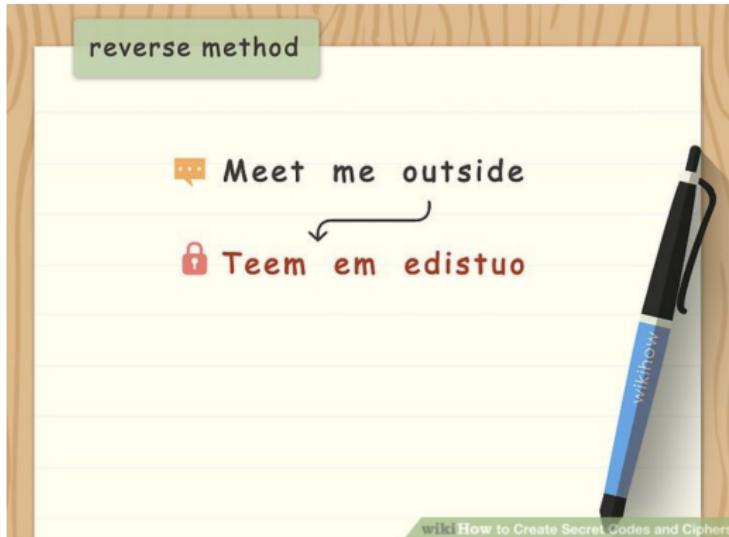
	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I/J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Some Examples

Enigma, Germany coding machine in World War II.



Some Examples



Shift Ciphers

Shift Ciphers: Make messages secret by **shifting** each letter **several letters forward** in the alphabet.

Encryption:

- Assign each letter an integer $p \in \mathbf{Z}_{26} = \{0, 1, \dots, 25\}$ based on the location of the letter in the alphabet.
- Replace p with $f(p)$:

$$f(p) = (p + k) \bmod 26.$$

- Maps $f(p)$ back to the alphabet.

Shift Ciphers

Example: What is the secret message produced from the message “MEET YOU IN THE PARK” using the Shift cipher with $k = 3$?

Solution: First replace the letters in the message with numbers. This produces

12 4 4 19 24 14 20 8 13 19 7 4 15 0 17 10.

Now replace each of these numbers p by $f(p) = (p + 3) \bmod 26$. This gives

15 7 7 22 1 17 23 11 16 22 10 7 18 3 20 13.

Translating this back to letters produces the encrypted message “PHHW BRX LQ WKH SDUN.”

Shift Ciphers

Shift Ciphers: Make messages secret by **shifting** each letter several letters forward in the alphabet.

Decryption:

- Assign each letter an integer $p \in \mathbf{Z}_{26} = \{0, 1, \dots, 25\}$ based on the location of the letter in the alphabet.
- Replace p with $f^{-1}(p)$:

$$f^{-1}(p) = (p - k) \bmod 26.$$

- Maps $f^{-1}(p)$ back to the alphabet.

Shift Ciphers

We can generalize shift ciphers further to slightly enhance security by using a function of the form

$$f(p) = (ap + b) \bmod 26.$$

How about the decryption? Suppose $\gcd(a, 26) = 1$.

Suppose that $c = (ap + b) \bmod 26$ with $\gcd(a, 26) = 1$. To decrypt, we need to show how to express p in terms of c . That is, we solve the congruence for p :

$$c \equiv ap + b \pmod{26}.$$

Subtract b from both sides, we have $ap \equiv c - b \pmod{26}$. Since $\gcd(a, 26) = 1$, we know that there is an inverse \bar{a} of a modulo 26:

$$p \equiv \bar{a}(c - b) \pmod{26}.$$

Cryptanalysis

The process of recovering plaintext from ciphertext **without** knowledge of both the encryption method and the key is known as **cryptanalysis** or breaking codes.

How to break messages that were encrypted using a **shift cipher**?

Solution 1: Try each 26 possible shifts.

Solution 2: Try different values of k based on the **frequency of letters** in the ciphertext. The nine most common letters in English text: E: 13%, T: 9%, A: 8%, O: 8%, I: 7%, N: 7%, S: 7%, H: 6%, and R: 6%.

Private Key Cryptosystem

In a **private key cryptosystem**, once you know an encryption key, you can **quickly find** the decryption key.

When a private key cryptosystem is used, two parties who wish to communicate in secret must share a secret key.



Any problems?

- Two people who want to communicate **securely** need to securely exchange this key.
- New key is used for each communication session between two parties.

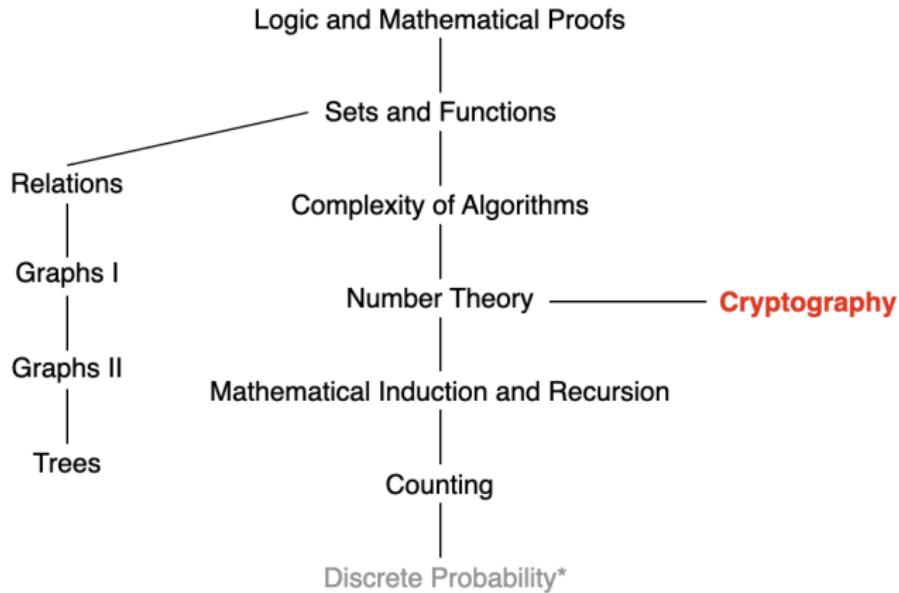
Public Key Cryptosystem

In **public key cryptosystems**, knowing how to send an encrypted message **does not** help decrypt messages.



- Public key is known to the public.
- Private key is kept secret: only the intended recipient of a message can decrypt it.

This Lecture



Cryptography: classical cryptography, RAS cryptosystem



Southern University
of Science and
Technology

Overview

- RSA as Public Key System
 - ▶ Only target recipient can decrypt the message:



- RSA as Digital Signature
- Diffie-Hellman Key Exchange Protocol

RAS Cryptosystem

Rivest-Shamir-Adleman

2002 Turing Award

2002

[Ronald L. Rivest](#),
[Adi Shamir](#) and
[Leonard M. Adleman](#)

For [their ingenious contribution](#) for making [public-key cryptography](#) useful in practice.

Pick two large primes p and q . Let $n = pq$. **Encryption key** (n, e) and **decryption key** (n, d) are selected such that

- $\gcd(e, (p-1)(q-1)) = 1$
- $ed \equiv 1 \pmod{(p-1)(q-1)}$

RSA encryption: $C = M^e \pmod{n}$

RSA decryption: $M = C^d \pmod{n}$

RSA Encryption

- 1 Translate a plaintext message into integers, each with **two digits**, e.g., A is translated into 00, B into 01, . . . , and Z into 25.
- 2 Divide this string into **equally sized blocks** of $2N$ digits
 - ▶ $2N$ is the largest even number such that the number 2525...25 with $2N$ digits does not exceed n .
- 3 For each block, transform it into a ciphertext block:

$$C = M^e \bmod n$$

RSA Encryption: Example

Encrypt the message “STOP” with key ($n = 2537$, $e = 13$). Note that $2537 = 43 \cdot 59$, where $p = 43$ and $q = 59$ are primes, and $\gcd(e, (p - 1)(q - 1)) = 1$.

Solution:

- 1 Translate into integers: 18191415
- 2 Divide this into blocks of 4 digits (because $2525 < 2537 < 252525$):
1819 1415
- 3 Encrypt each block using the mapping

$$C = M^{13} \pmod{2537}.$$

We have $1819^{13} \pmod{2537} = 2081$ and $1415^{13} \pmod{2537} = 2182$.
The encrypted message is 2081 2182.

RSA Decryption

For each block, transform the ciphertext into plaintext message:

$$M = C^d \pmod{n}$$

Example: What is the decrypted message of 0981 0461 with $e = 13$, $p = 43$, $q = 59$?

Solution: Recall that $ed \equiv 1 \pmod{(p-1)(q-1)}$. Thus, $d = 937$ is an inverse of 13 modulo $42 \cdot 58 = 2436$.

For each block, transform it into plaintext message:

$$M = C^{937} \pmod{2537}.$$

Since $0981^{937} \pmod{2537} = 0704$ and $0461^{937} \pmod{2537} = 1115$, the plaintext message is 0704 1115, which is “HELP”.

RAS Cryptosystem

Pick two large primes p and q . Let $n = pq$. **Encryption key** (n, e) and **decryption key** (n, d) are selected such that

- (1) $\gcd(e, (p-1)(q-1)) = 1$
- (2) $ed \equiv 1 \pmod{(p-1)(q-1)}$

RSA encryption: $C = M^e \pmod{n}$;

RSA decryption: $M = C^d \pmod{n}$. Why?

According to (1), the inverse d exists. According to (2), there exists an integer k such that

$$de = 1 + k(p-1)(q-1).$$

It follows that $C^d \equiv (M^e)^d \equiv M^{de} \equiv M^{1+k(p-1)(q-1)} \pmod{n}$.

Assuming that $\gcd(M, p) = \gcd(M, q) = 1$, we have $M^{p-1} \equiv 1 \pmod{p}$ and $M^{q-1} \equiv 1 \pmod{q}$. (see Theorem 3 in Section 4.4)



SUSTech

Southern University
of Science and
Technology

RAS Cryptosystem

According to (1), the inverse d exists. According to (2), there exists an integer k such that

$$de = 1 + k(p-1)(q-1).$$

It follows that $C^d \equiv (M^e)^d = M^{de} = M^{1+k(p-1)(q-1)} \pmod{n}$.

Assuming that $\gcd(M, p) = \gcd(M, q) = 1$, we have $M^{p-1} \equiv 1 \pmod{p}$ and $M^{q-1} \equiv 1 \pmod{q}$.

$$C^d \equiv M \cdot (M^{p-1})^{k(q-1)} \equiv M \cdot 1 = M \pmod{p}$$

$$C^d \equiv M \cdot (M^{q-1})^{k(p-1)} \equiv M \cdot 1 = M \pmod{q}.$$

Because $\gcd(p, q) = 1$, we have

$$C^d \equiv M \pmod{pq}.$$

This basically implies that

$$M = C^d \pmod{n}$$

RSA as Public Key System

Pick two large primes p and q . Let $n = pq$. **Encryption key** (n, e) and **decryption key** (n, d) are selected such that

- (1) $\gcd(e, (p-1)(q-1)) = 1$
- (2) $ed \equiv 1 \pmod{(p-1)(q-1)}$

RSA encryption: $C = M^e \pmod{n}$;

RSA decryption: $M = C^d \pmod{n}$.

RSA as a Public Key System

- Public key: (n, e)
- Private key: d
- p, q must be kept **secret**!

Why is the RSA cryptosystem suitable for public key cryptography?

RSA as Public Key System

RSA as a Public Key System

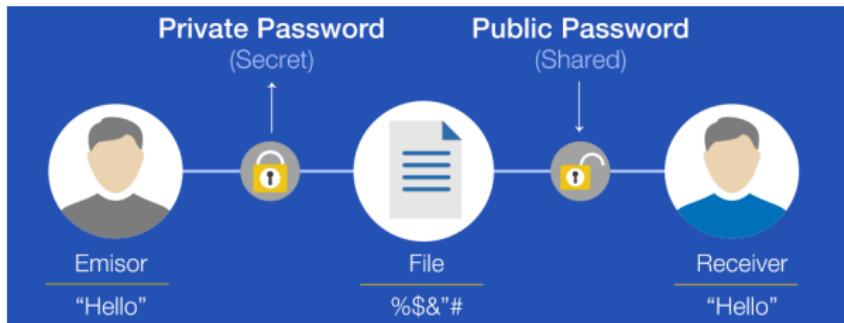
- Public key: (n, e) ; Private key: (n, d)
- p, q must be kept **secret**!

Why is the RSA cryptosystem suitable for public key cryptography?

- It is possible to **rapidly construct** a public key by finding two large primes p and q , each with more than 200 digits.
- When we know p and q , we can **quickly find** an inverse d .
- However, **no method** is known to decrypt messages that is not based on finding a factorization of n .
 - ▶ **Factorization** is believed to be a **difficult problem**.
 - ▶ The most efficient factorization methods known (as of 2010) require billions of years to factor 400-digit integers.

Overview

- RSA as Public Key System
- RSA as Digital Signature
 - ▶ The recipient of the message knows that it came from the person they think it came from.



- Diffie-Hellman Key Exchange Protocol

RSA as Digital Signature

Alice's RSA public key is (n, e) and her private key is d .

Alice splits the plaintext message into blocks and applies her decryption function:

$$S = M^d \bmod n \quad (\text{RSA signature})$$

When a recipient receives her message, they apply Alice's encryption function:

$$M = S^e \bmod n \quad (\text{RSA verification})$$

Alice can send her message to as many people as she wants and by signing it in this way, **every recipient can be sure it came from Alice.**

Overview

- RSA as a Public Key System
- RSA as Digital Signature
- Diffie-Hellman Key Exchange Protocol
 - ▶ Exchange a secret key over an insecure communications channel

