南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

DISCRETE MATHEMATICS FOR COMPUTER SCIENCE

Dr. Qi Wang

Department of Computer Science and Engineering
Office: Room903, Nanshan iPark A7 Building
Email: wangqi@sustc.edu.cn

The Chinese Remainder Theorem

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

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\ldots \\ x &\equiv a_n \pmod{m_n} \end{aligned}$$

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

The Chinese Remainder Theorem

- **Proof** Let $M_k = m/m_k$ for $k = 1, 2, \ldots, n$ and $m = m_1 m_2 \cdots m_n$. Since $\gcd(m_k, M_k) = 1$, there is an integer y_k, an inverse of M_k modulo m_k such that $M_k y_k \equiv 1 \pmod{m_k}$. Let

$$x = a_1 M_1 y_1 + a_2 M_2 y_2 + \cdots a_n M_n y_n.$$

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

Back Substitution

- We may also solve systems of linear congruences with pairwise relatively prime moduli by *back substitution*.

Back Substitution

- We may also solve systems of linear congruences with pairwise relatively prime moduli by *back substitution*.

Example

$$\begin{aligned} x &\equiv 2 \pmod 3 \\ x &\equiv 3 \pmod 5 \\ x &\equiv 2 \pmod 7 \end{aligned}$$

Back Substitution

- We may also solve systems of linear congruences with pairwise relatively prime moduli by *back substitution*.

Example

$$\begin{aligned} x &\equiv 2 \pmod 3 \\ x &\equiv 3 \pmod 5 \\ x &\equiv 2 \pmod 7 \end{aligned}$$

$$\begin{aligned} x &\equiv 8 \pmod{15} \\ x &\equiv 2 \pmod{21} \end{aligned}$$

Modular Arithmetic in CS

- Modular arithmetic and congruencies are used in CS:
  - ◇ Pseudorandom number generators
  - ◇ Hash functions
  - ◇ Cryptography

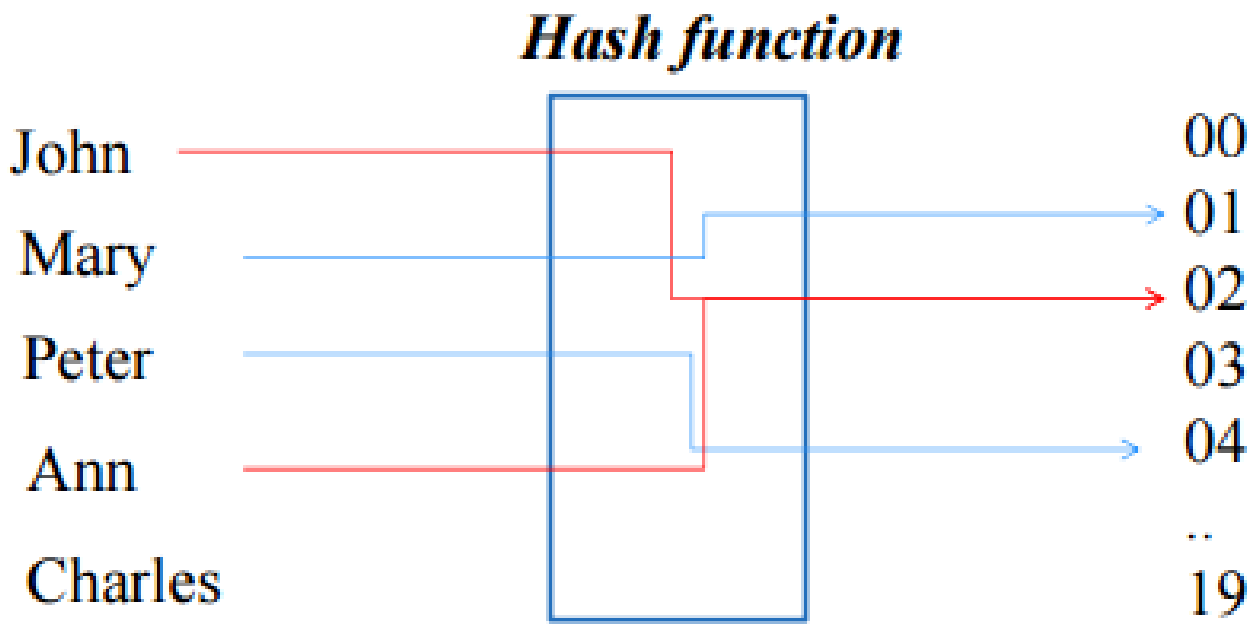
Hash Functions

- 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.

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?

Hash Functions

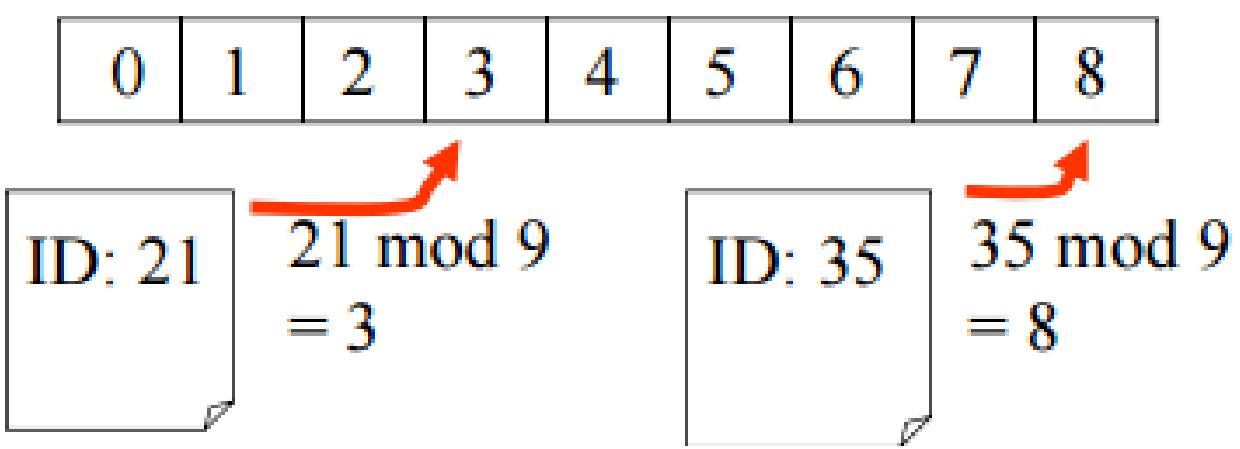
- **Problem**: Given a large collection of records, how can we store and find a record quickly?

  Solution: Use a hash function, calculate the location of the record based on the record's ID.

  Example: A common hash function is

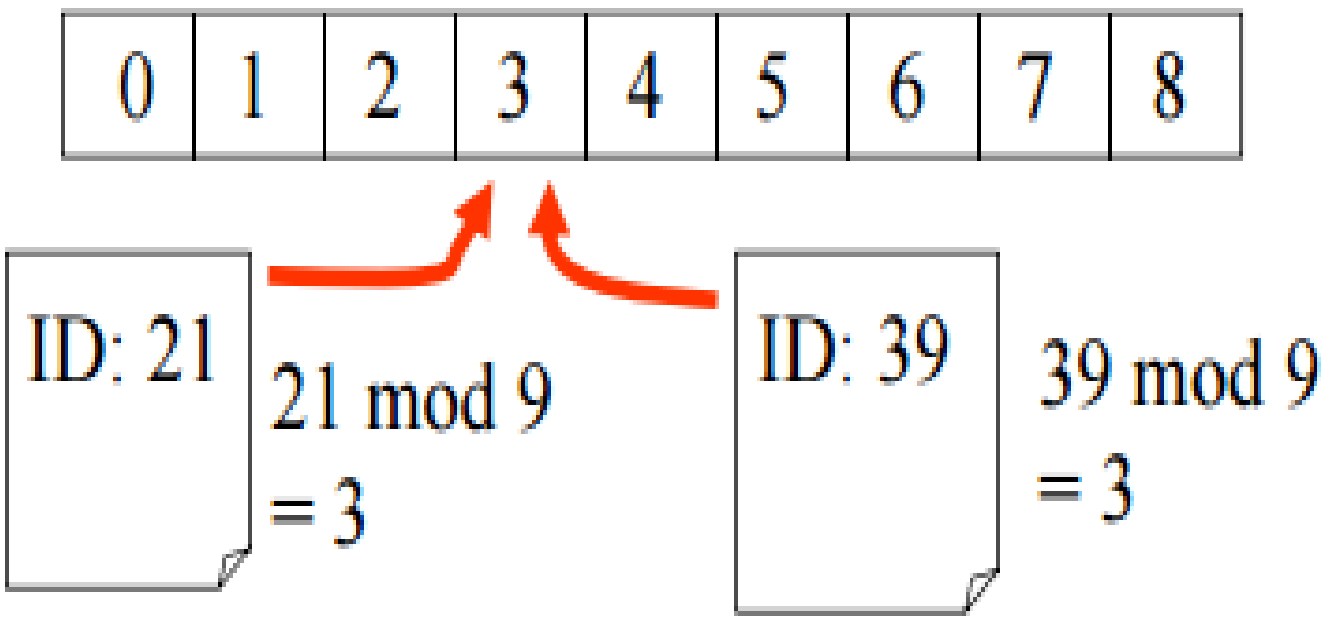
  - $h(k) = k \bmod n$,

  where n is the number of available storage locations.



Hash Functions

- Two records mapped to the same location

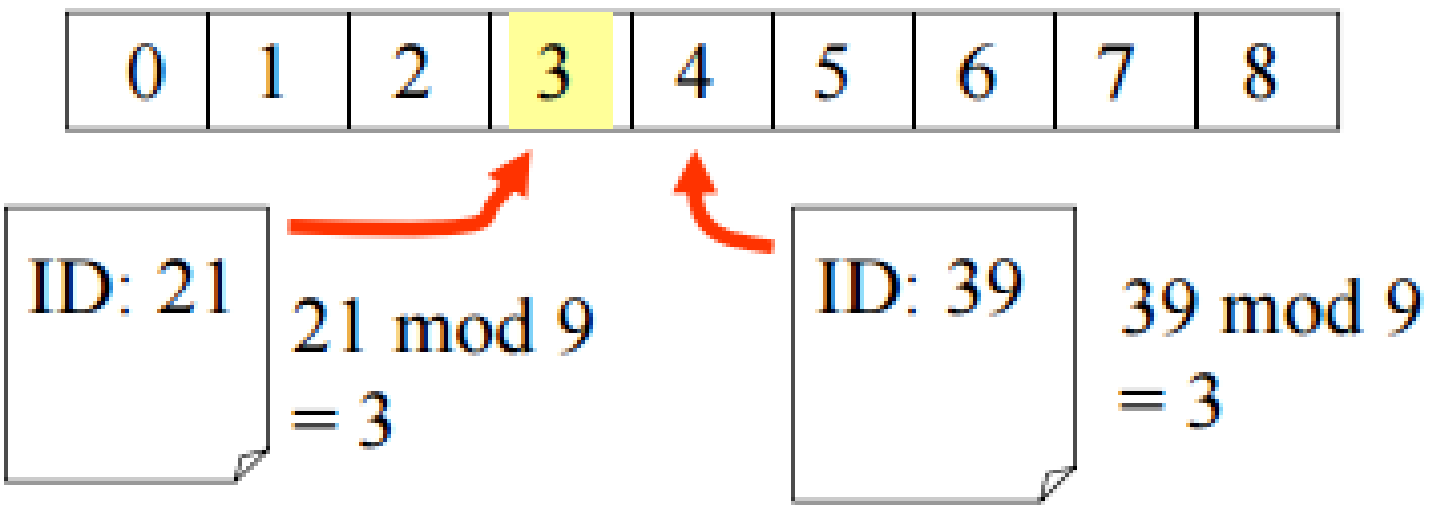


Hash Functions

- **Solution 1**: move to the next available location

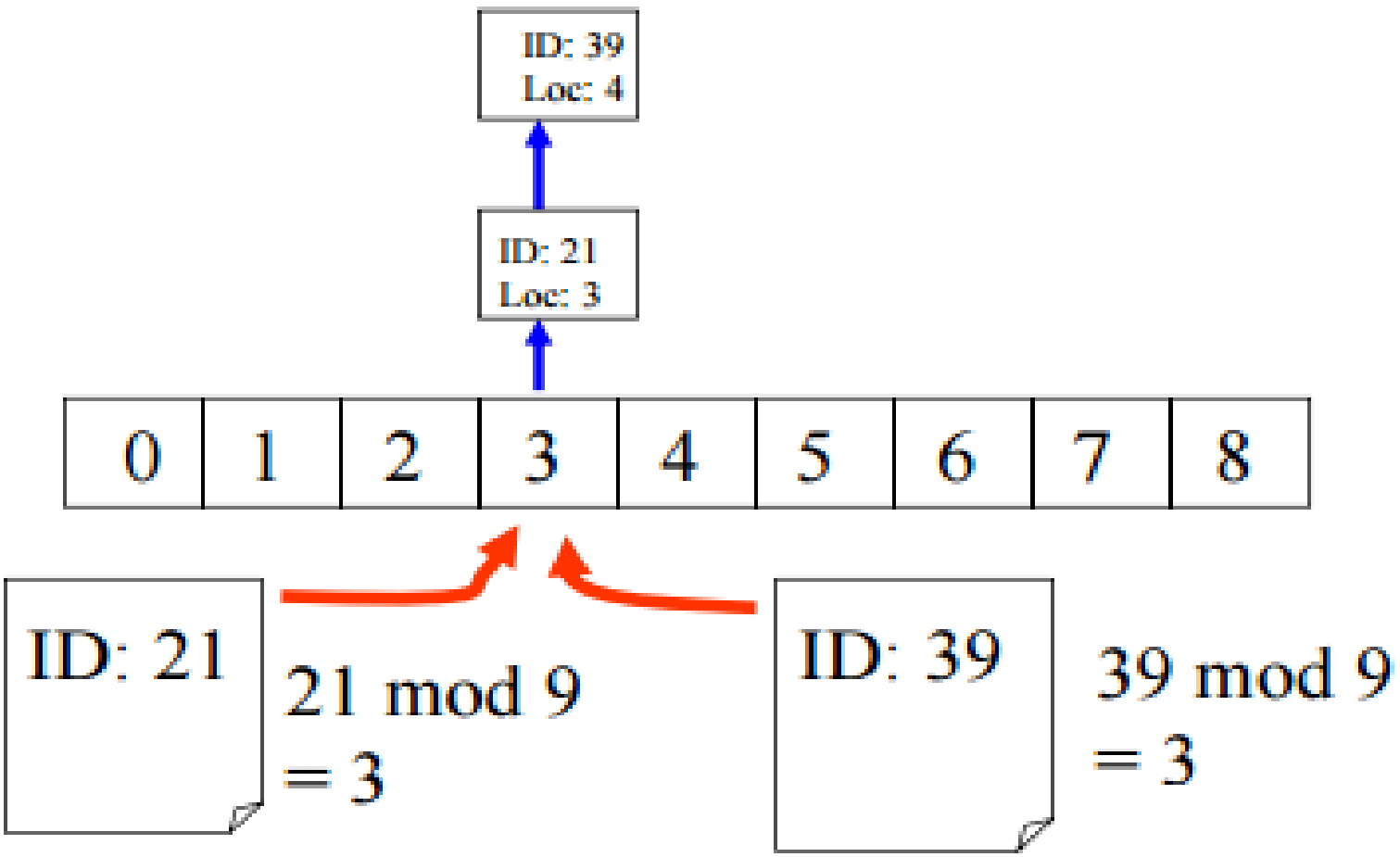
try

$$h_0(k) = k \bmod n$$
$$h_1(k) = (k+1) \bmod n$$
$$\ldots$$
$$h_m(k) = (k+m) \bmod n$$



Hash Functions

- **Solution 2**: remember the exact location in a secondary structure that is searched sequentially



Applications of Number Theory in Cryptography

- Introduction
- Symmetric cryptography
- Asymmetric cryptography
- RSA Cryptosystem
- DLP and El Gamal cryptography
- Diffie-Hellman key exchange protocol
- Cryptocurrency, e.g., bitcoin

Fermat's Little Theorem

- **Theorem (Fermat's little theorem)** : Let p be a prime, and let x be an integer such that $x \not\equiv 0 \bmod p$. Then

$$x^{p-1} \equiv 1 \pmod{p}.$$

Fermat's Little Theorem

- **Theorem (Fermat's little theorem)** : Let p be a prime, and let x be an integer such that $x \not\equiv 0 \bmod p$. Then

$$x^{p-1} \equiv 1 \pmod{p}.$$

Example: Find $7^{222} \pmod{11}$

Fermat's Little Theorem

- **Theorem (Fermat's little theorem)** : Let p be a prime, and let x be an integer such that $x \not\equiv 0 \bmod p$. Then

$$x^{p-1} \equiv 1 \pmod{p}.$$

Example: Find $7^{222} \pmod{11}$

$$7^{222} = 7^{22 \cdot 10+2} = (7^{10})^{22}7^2 = 1^{22} \cdot 49 \equiv 5 \pmod{11}$$

Fermat's Little Theorem

- **Theorem (Fermat's little theorem)** : Let p be a prime, and let x be an integer such that $x \not\equiv 0 \bmod p$. Then

$$x^{p-1} \equiv 1 \pmod{p}.$$

Example: Find $7^{222} \pmod{11}$

$$7^{222} = 7^{22 \cdot 10+2} = (7^{10})^{22}7^2 = 1^{22} \cdot 49 \equiv 5 \pmod{11}$$

$\mathcal{Q}$: How to prove Fermat's little theorem?

Fermat's Little Theorem

- **Theorem (Fermat's little theorem)** : Let p be a prime, and let x be an integer such that $x \not\equiv 0 \bmod p$. Then

$$x^{p-1} \equiv 1 \pmod{p}.$$

Example: Find $7^{222} \pmod{11}$

$$7^{222} = 7^{22 \cdot 10+2} = (7^{10})^{22}7^2 = 1^{22} \cdot 49 \equiv 5 \pmod{11}$$

$\mathcal{Q}$: How to prove Fermat's little theorem?

$$\{1, 2, \ldots, p-1\} = \{x, 2x, \ldots, x(p-1) \pmod{p}\}$$

Euler's Theorem

- Euler's *totient* function: $\phi(n)$

  the number of positive integers coprime to n in $\mathbb{Z}_n$

Euler's Theorem

- Euler's *totient* function: $\phi(n)$

  the number of positive integers coprime to n in $\mathbb{Z}_n$

$$\phi(p) = p - 1$$
$$\phi(pq) = (p-1)(q-1)$$
$$\phi(p^i) = p^i - p^{i-1}$$

Euler's Theorem

- Euler's *totient* function: $\phi(n)$

  the number of positive integers coprime to n in $\mathbb{Z}_n$

$$\phi(p) = p - 1$$
$$\phi(pq) = (p-1)(q-1)$$
$$\phi(p^i) = p^i - p^{i-1}$$

- **Theorem (Euler's theorem)** : Let n be a positive integer, and let x be an integer such that $\gcd(x, n) = 1$. Then

$$x^{\phi(n)} \equiv 1 \pmod{n}.$$

Euler's Theorem

- Euler's *totient* function: $\phi(n)$

  the number of positive integers coprime to n in $\mathbb{Z}_n$

$$\phi(p) = p - 1$$
$$\phi(pq) = (p-1)(q-1)$$
$$\phi(p^i) = p^i - p^{i-1}$$

- **Theorem (Euler's theorem)** : Let n be a positive integer, and let x be an integer such that $\gcd(x, n) = 1$. Then

$$x^{\phi(n)} \equiv 1 \pmod{n}.$$

$\mathcal{Q}$: How to prove Euler's theorem?

The Multiplicative Order

- Let $a \in \mathbb{Z}$ and $n \in \mathbb{N}$. If $\gcd(a, n) = 1$, the least $\ell \in \mathbb{N}$ such that $a^{\ell} \equiv 1 \pmod{n}$ is called the *order of a modulo n*, denoted by $\mathrm{ord}_n(a)$.

The Multiplicative Order

- Let $a \in \mathbb{Z}$ and $n \in \mathbb{N}$. If $\gcd(a, n) = 1$, the least $\ell \in \mathbb{N}$ such that $a^{\ell} \equiv 1 \pmod{n}$ is called the *order of a modulo n*, denoted by $\mathrm{ord}_n(a)$.

  Theorem Let $a \in \mathbb{Z}$ and $n \in \mathbb{N}$ with $\gcd(a, n) = 1$. Then $\mathrm{ord}_n(a)$ exists and divides $\phi(n)$.

The Multiplicative Order

- Let $a \in \mathbb{Z}$ and $n \in \mathbb{N}$. If $\gcd(a, n) = 1$, the least $\ell \in \mathbb{N}$ such that $a^{\ell} \equiv 1 \pmod{n}$ is called the *order of a modulo n*, denoted by $\mathrm{ord}_n(a)$.

Theorem Let $a \in \mathbb{Z}$ and $n \in \mathbb{N}$ with $\gcd(a, n) = 1$. Then $\mathrm{ord}_n(a)$ exists and divides $\phi(n)$.

Example $n = 7$, then

$$\mathrm{ord}_7(1) = 1,\ \mathrm{ord}_7(2) = 3,\ \mathrm{ord}_7(3) = 6$$
$$\mathrm{ord}_7(4) = 3,\ \mathrm{ord}_7(5) = 6,\ \mathrm{ord}_7(6) = 2$$

The Multiplicative Order

- Let $a \in \mathbb{Z}$ and $n \in \mathbb{N}$. If $\gcd(a, n) = 1$, the least $\ell \in \mathbb{N}$ such that $a^{\ell} \equiv 1 \pmod{n}$ is called the *order of a modulo n*, denoted by $\mathrm{ord}_n(a)$.

Theorem Let $a \in \mathbb{Z}$ and $n \in \mathbb{N}$ with $\gcd(a, n) = 1$. Then $\mathrm{ord}_n(a)$ exists and divides $\phi(n)$.

Example $n = 7$, then

$$\mathrm{ord}_7(1) = 1,\ \mathrm{ord}_7(2) = 3,\ \mathrm{ord}_7(3) = 6$$
$$\mathrm{ord}_7(4) = 3,\ \mathrm{ord}_7(5) = 6,\ \mathrm{ord}_7(6) = 2$$

$\mathcal{Q}$: How to prove this theorem?

The Multiplicative Order

- Let $a \in \mathbb{Z}$ and $n \in \mathbb{N}$. If $\gcd(a, n) = 1$, the least $\ell \in \mathbb{N}$ such that $a^{\ell} \equiv 1 \pmod{n}$ is called the *order of a modulo n*, denoted by $\mathrm{ord}_n(a)$.

Theorem Let $a \in \mathbb{Z}$ and $n \in \mathbb{N}$ with $\gcd(a, n) = 1$. Then $\mathrm{ord}_n(a)$ exists and divides $\phi(n)$.

Example $n = 7$, then

$$\mathrm{ord}_7(1) = 1,\ \mathrm{ord}_7(2) = 3,\ \mathrm{ord}_7(3) = 6$$
$$\mathrm{ord}_7(4) = 3,\ \mathrm{ord}_7(5) = 6,\ \mathrm{ord}_7(6) = 2$$

$\mathcal{Q}$: How to prove this theorem?

Using Euler's theorem and $\phi(n) = q \cdot \mathrm{ord}_n(a) + r$

Primitive Roots

- A *primitive root* modulo a prime p is an integer $r \in \mathbb{Z}_p$ such that every nonzero element of $\mathbb{Z}_p$ is a power of r.

Primitive Roots

- A *primitive root* modulo a prime p is an integer $r \in \mathbb{Z}_p$ such that every nonzero element of $\mathbb{Z}_p$ is a power of r.

  Example: 3 is a primitive root of $\mathbb{Z}_7$. 2 is not a primitive root of $\mathbb{Z}_7$.

Primitive Roots

- A *primitive root* modulo a prime p is an integer $r \in \mathbb{Z}_p$ such that every nonzero element of $\mathbb{Z}_p$ is a power of r.

  Example: 3 is a primitive root of $\mathbb{Z}_7$. 2 is not a primitive root of $\mathbb{Z}_7$.

  Theorem There is a primitive root modulo n if and only if $n = 2, 4, p^e$ or $2p^e$, where p is an odd prime.

Primitive Roots

- A *primitive root* modulo a prime p is an integer $r \in \mathbb{Z}_p$ such that every nonzero element of $\mathbb{Z}_p$ is a power of r.

  Example: 3 is a primitive root of $\mathbb{Z}_7$. 2 is not a primitive root of $\mathbb{Z}_7$.

  Theorem There is a primitive root modulo n if and only if $n = 2, 4, p^e$ or $2p^e$, where p is an odd prime.

  $\mathcal{Q}$: proof? The number of primitive roots? $*$

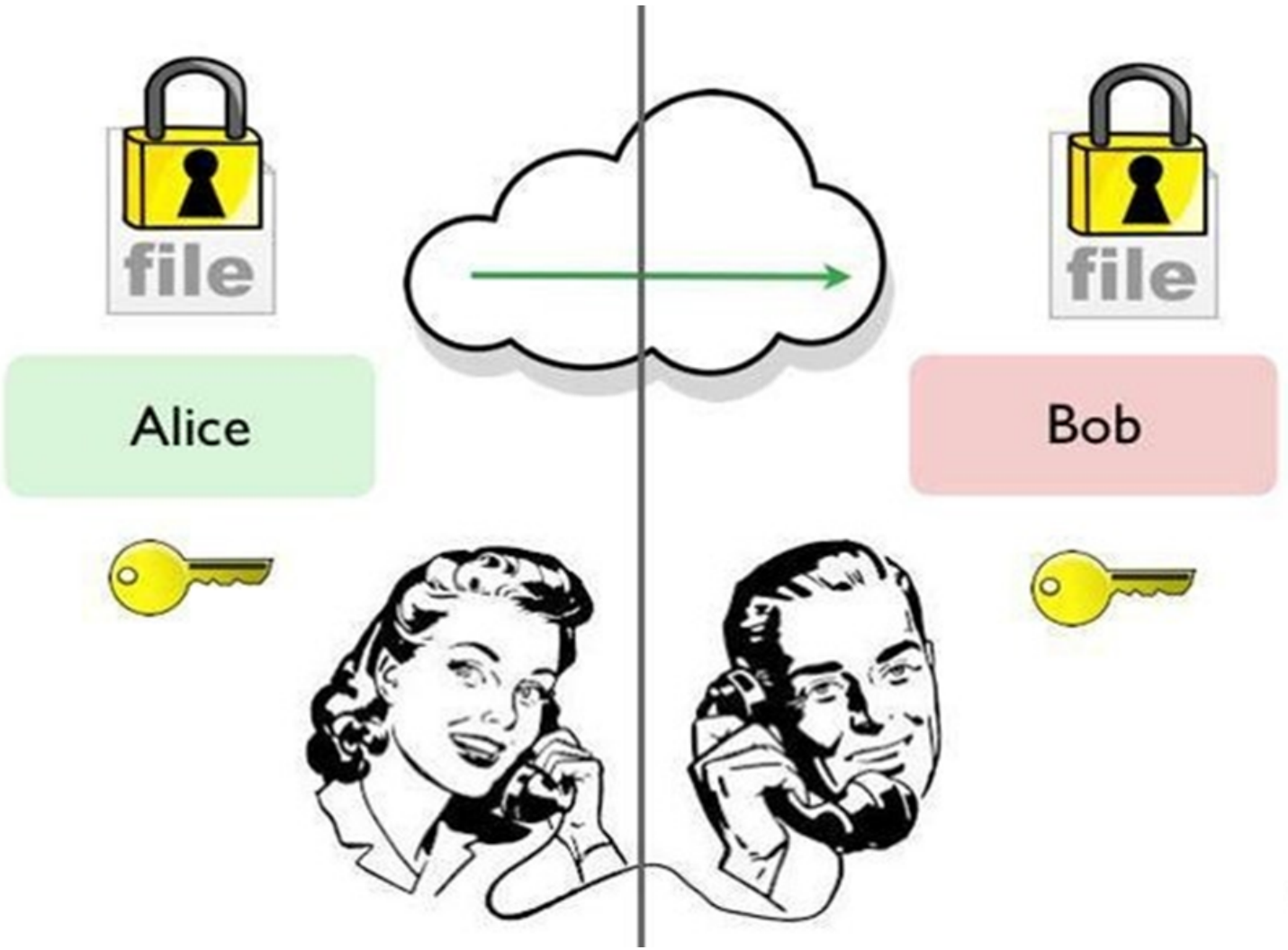
Classical One-Key Ciphers

- A 5-tuple $(\mathcal{M}, \mathcal{C}, \mathcal{K}, E_k, D_k)$, where
  - $\mathcal{M}$, $\mathcal{C}$ and $\mathcal{K}$ denote the plaintext space, ciphertext space, and key space.
  - Any $k \in \mathcal{K}$ could be the encryption and decryption key;
  - E_k and D_k are encryption and decryption transformatons with $D_k(E_k(m)) = m$ for each $m \in \mathcal{M}$.

Classical One-Key Ciphers

- Give a cipher $(\mathcal{M}, \mathcal{C}, \mathcal{K}, E_k, D_k)$:
  - **Encryption:** $c = E_k(m)$, where E_k is usually applied to blocks of the plaintext m.
  - **Decryption:** $m = D_k(c)$, where D_k is usually applied to blocks of the ciphertext c.

Symmetric Cryptography



Attacks on Classical Cryptography

- **Ciphertext-only attack**

  determine the decryption transformation D_k, or k, or m from the ciphertext c.

- **Known-plaintext attack**

  determine the decryption transformation D_k, or k, from a ciphertext-plaintext pair (c, m).

Simple Substitution Ciphers

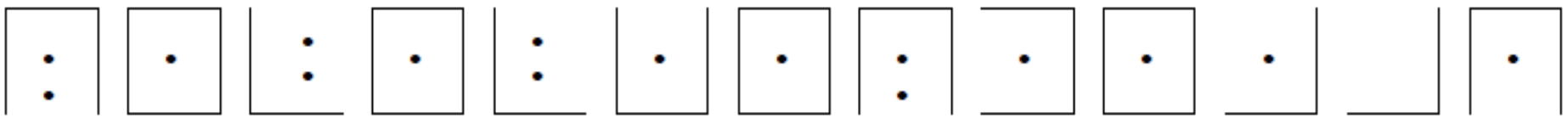
- Let f be a 1-to-1 function from A to B. It is a 5-tuple $(\mathcal{M}, \mathcal{C}, \mathcal{K}, E_k, D_k)$, where
  - $\mathcal{M} = A^*$ and $\mathcal{C} = B^*$
  - $\mathcal{K}$ is the set of all possible f
  - $k = f \in \mathcal{K}$ is the encryption and decryption key
  - For a message $m = m_0 m_1 m_2 \cdots$,

    $$c = E_k(m) = f(m_0) f(m_1) f(m_2) \cdots$$

  - For a ciphertext $c = c_0 c_1 c_2 \cdots$,

    $$m = D_k(c) = f^{-1}(c_0) f^{-1}(c_1) f^{-1}(c_2) \cdots$$

Simple Substitution Ciphers: Examples



a	b	c
d	e	f
g	h	i

k	l	m
n	o	p
q	r	s

t	u	v
w	x	y
z	j	

Simple Substitution Ciphers: Caesar Cipher

a	b	c	$\cdots$	y	z
0	1	2	$\cdots$	24	25

$$f(x) = (x + 3) \bmod 26$$

Simple Substitution Ciphers: Caesar Cipher

a	b	c	$\cdots$	y	z
0	1	2	$\cdots$	24	25

$$f(x) = (x + 3) \bmod 26$$

VENI, VIDI, VICI

YHQL YLGL YLFL

Simple Substitution Ciphers: Examples

a	b	c	$\cdots$	y	z
0	1	2	$\cdots$	24	25

Affine cipher: Take any (k_0, k_1) with $\gcd(k_0, 26) = 1$ and $0 \leq k_0 \leq 25$, define the 1-to-1 mapping f by

$$f(a) = (ak_0 + k_1) \bmod 26.$$

Simple Substitution Ciphers: Examples

a	b	c	$\cdots$	y	z
0	1	2	$\cdots$	24	25

Affine cipher: Take any (k_0, k_1) with $\gcd(k_0, 26) = 1$ and $0 \leq k_0 \leq 25$, define the 1-to-1 mapping f by

$$f(a) = (ak_0 + k_1) \bmod 26.$$

Caesar cipher: $(k_0, k_1) = (1, 3)$

Simple Substitution Ciphers: Examples

a	b	c	$\cdots$	y	z
0	1	2	$\cdots$	24	25

Affine cipher: Take any (k_0, k_1) with $\gcd(k_0, 26) = 1$ and $0 \leq k_0 \leq 25$, define the 1-to-1 mapping f by

$$f(a) = (ak_0 + k_1) \bmod 26.$$

Caesar cipher: $(k_0, k_1) = (1, 3)$

$\mathcal{Q}$: Why $\gcd(k_0, 26) = 1$?

Simple Substitution Ciphers: Examples

Cryptanalysis: Suppose that we know two pairs of plaintext-ciphertext characters (m_1, c_1) and (m_2, c_2), i.e.,

$$c_1 = (m_1 k_0 + k_1) \bmod 26$$

$$c_2 = (m_2 k_0 + k_1) \bmod 26$$

It is possible to solve the equations to obtain k_0 and k_1.

Simple Substitution Ciphers: Examples

Cryptanalysis: Suppose that we know two pairs of plaintext-ciphertext characters (m_1, c_1) and (m_2, c_2), i.e.,

$$c_1 = (m_1 k_0 + k_1) \bmod 26$$

$$c_2 = (m_2 k_0 + k_1) \bmod 26$$

It is possible to solve the equations to obtain k_0 and k_1.

$\mathcal{Q}$: When the set of two equations above has a unique solution (k_0, k_1)?

Other Symmetric Ciphers

DES

3DES

IDEA

Blowfish

Rijndael (AES)

...

Other Symmetric Ciphers

DES

3DES

IDEA

Blowfish

Rijndael (AES)

...

"Confusion" & "Diffusion"

Claude E. Shannon, "Communication Theory of Secrecy Systems", Bell System Technical Journal, vol. 28-4, pages 656-715, 1949.

Other Symmetric Ciphers

DES

3DES

IDEA

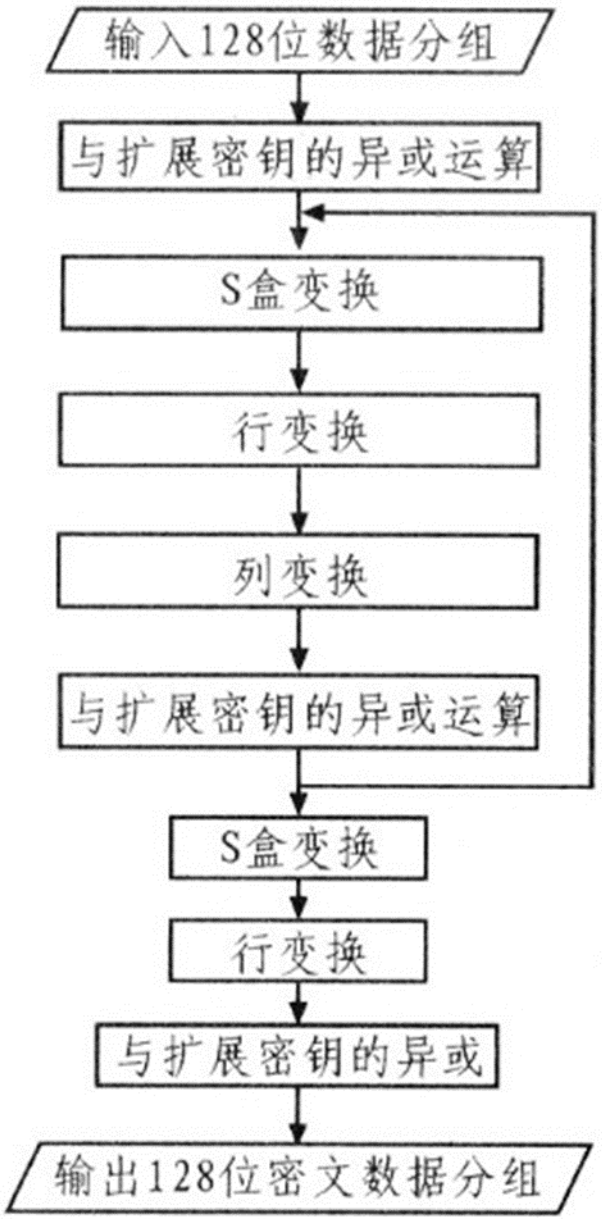
Blowfish

Rijndael (AES)

...

"Confusion" & '

Claude E. Shannon, "C
Systems", Bell System
656-715, 1949.



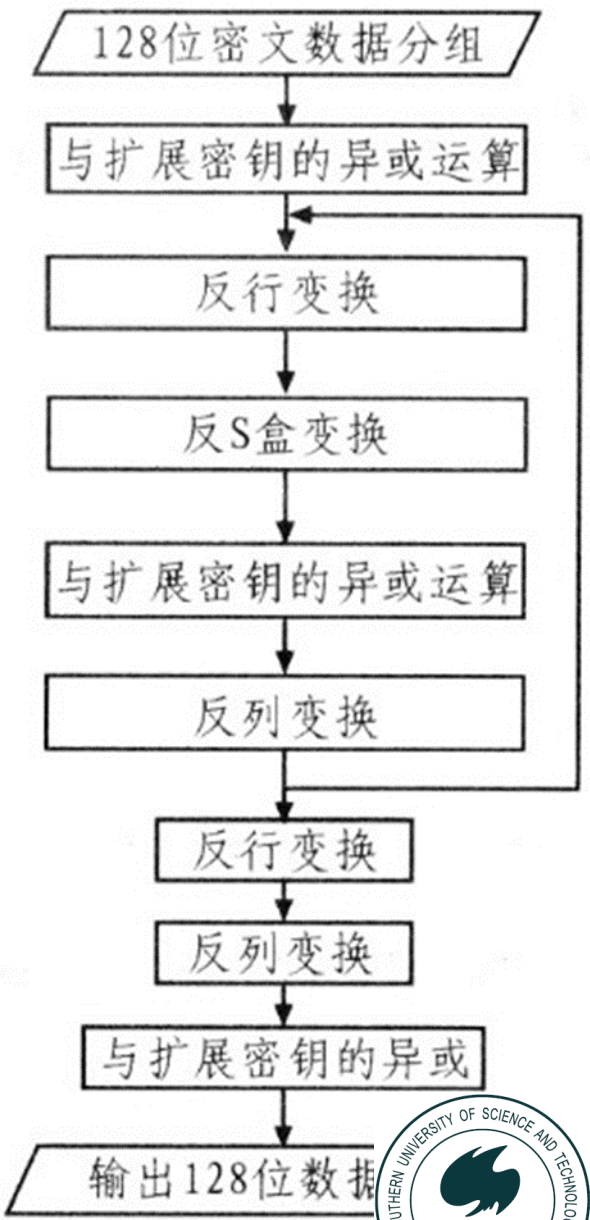


图 1 AES 加密/解密框图

Public-Key Cryptosystems

W. Diffie, M. E. Hellman, “New directions in cryptography”, IEEE Transactions on Information Theory, vol. 22-6, pages 644-654, 1976.

Public-Key Cryptosystems

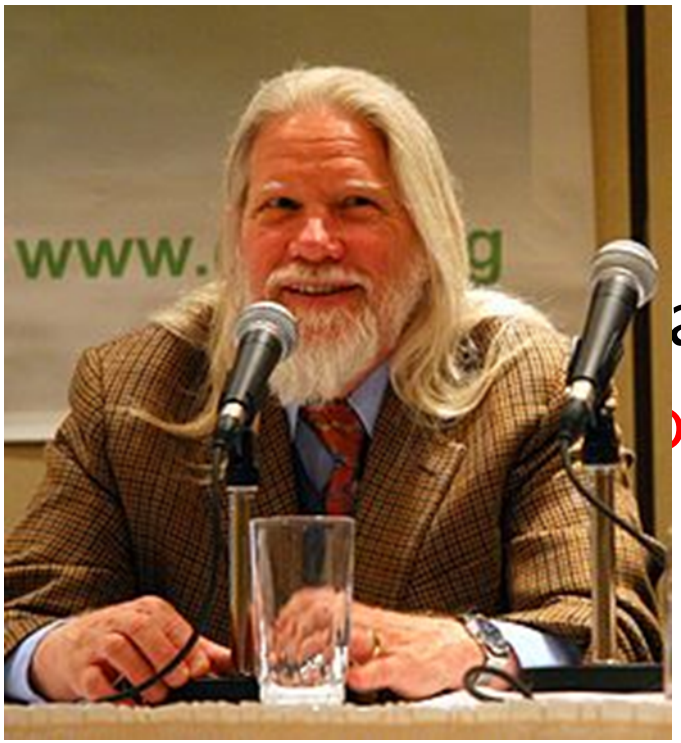
W. Diffie, M. E. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory, vol. 22-6, pages 644-654, 1976.

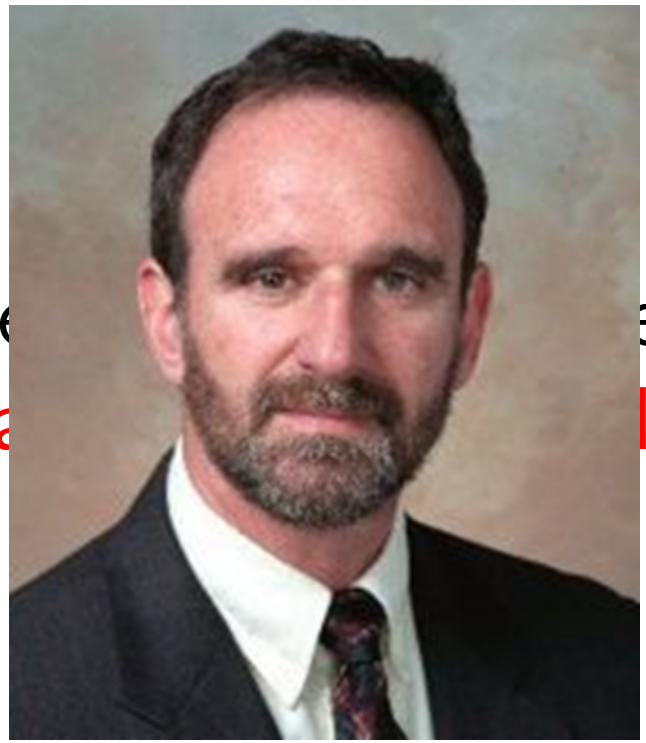
Four properties:

1. $D(E(M)) = M$
2. Both E and D are easy to compute
3. It is computationally infeasible to derive D from E
4. $E(D(M)) = M$

Public-Key Cryptosystems

W. Diffie, M. E. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory, vol. 22-6, pages 644-654, 1976.

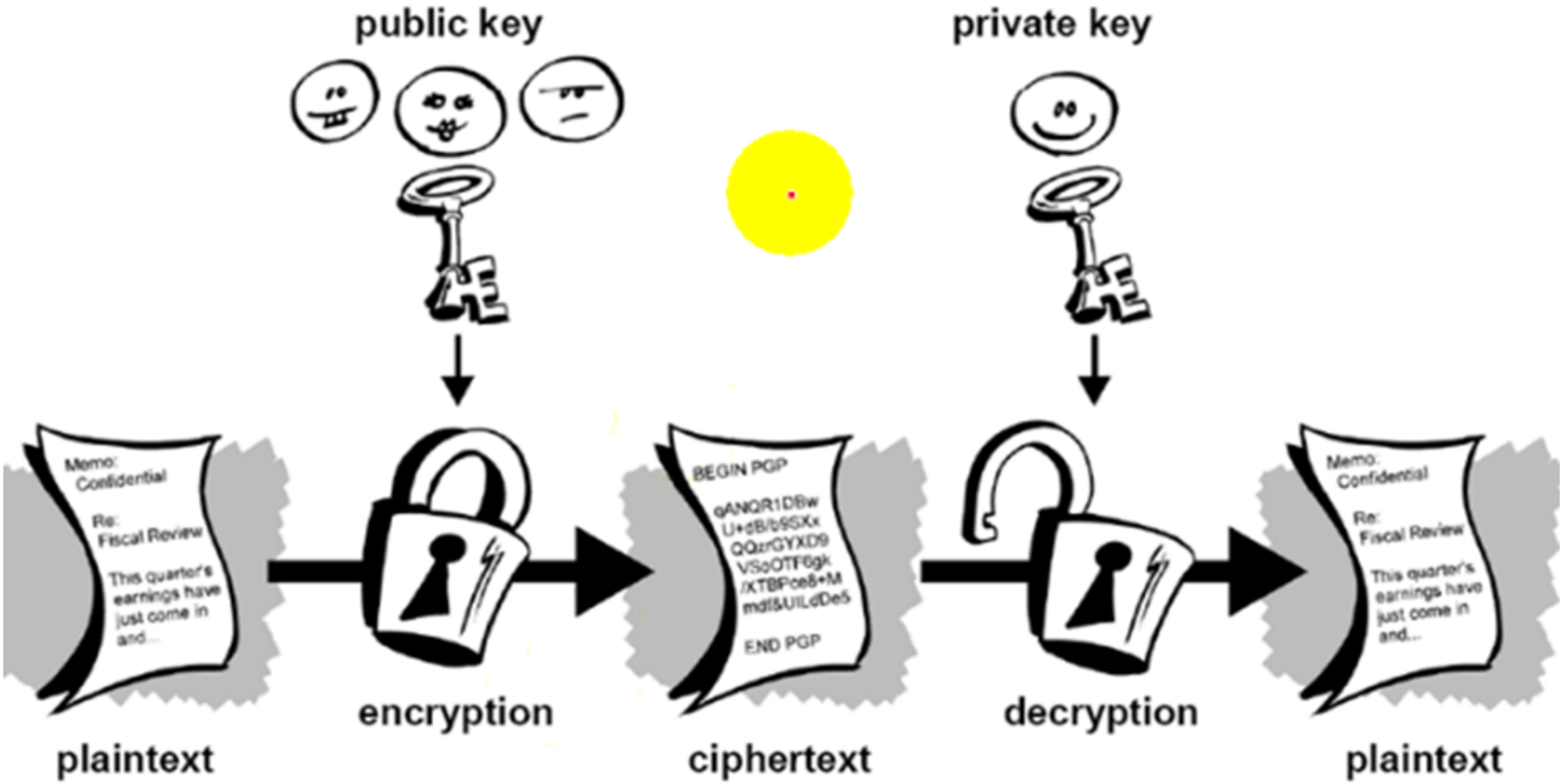




are e

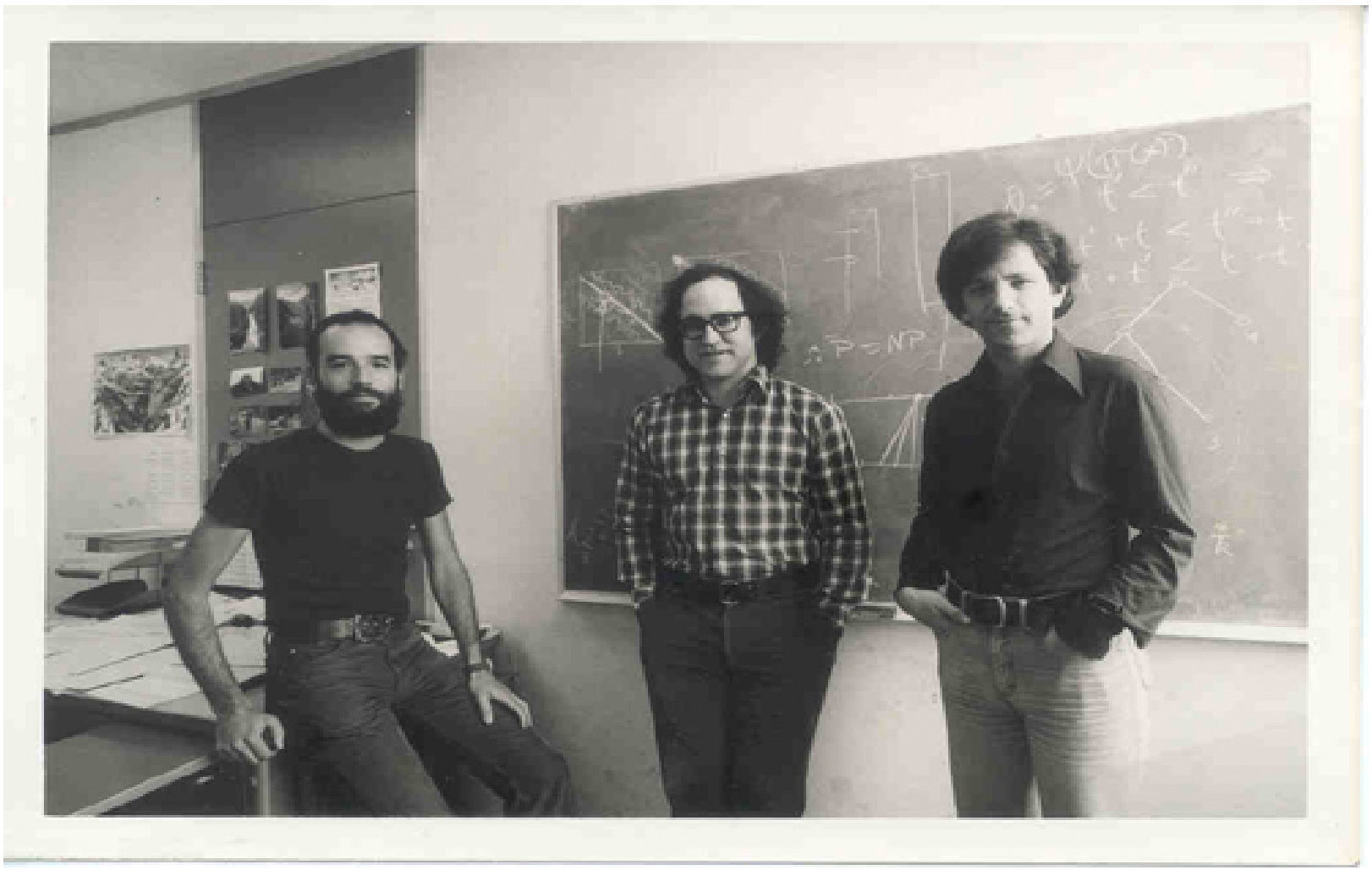
ona lerive D

Asymmetric Cryptography



Public-Key Cryptosystems

R. Rivest, A. Shamir, L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, vol. 21-2, pages 120-126, 1978.



RSA Public-Key Cryptosystem

Pick two large primes, p and q. Let $n = pq$, then $\phi(n) = (p-1)(q-1)$. Encryption and decryption keys e and d are selected such that

- $\gcd(e, \phi(n)) = 1$
- $ed \equiv 1 \pmod{\phi(n)}$

RSA Public-Key Cryptosystem

Pick two large primes, p and q. Let $n = pq$, then $\phi(n) = (p-1)(q-1)$. Encryption and decryption keys e and d are selected such that

- $\gcd(e, \phi(n)) = 1$
- $ed \equiv 1 \pmod{\phi(n)}$

- $C = M^e \bmod n$ (RSA **encryption**)
- $M = C^d \bmod n$ (RSA **decryption**)

RSA Public-Key Cryptosystem

$$C = M^e \bmod n \text{ (RSA } \textbf{encryption)}$$

$$M = C^d \bmod n \text{ (RSA } \textbf{decryption)}$$

- **Theorem 12** (Correctness) : Let p and q be two odd primes, and define $n = pq$. Let e be relatively prime to $\phi(n)$ and let d be the multiplicative inverse of e modulo $\phi(n)$. For each integer x such that $0 < x < n$,

$$x^{ed} \equiv x \pmod{n}.$$

RSA Public-Key Cryptosystem

$$C = M^e \bmod n \text{ (RSA } \textbf{encryption)}$$

$$M = C^d \bmod n \text{ (RSA } \textbf{decryption)}$$

- **Theorem 12** (Correctness) : Let p and q be two odd primes, and define $n = pq$. Let e be relatively prime to $\phi(n)$ and let d be the multiplicative inverse of e modulo $\phi(n)$. For each integer x such that $0 < x < n$,

$$x^{ed} \equiv x \pmod{n}.$$

$\mathcal{Q}$: How to prove this?

RSA Public-Key Cryptosystem: Example

Parameters:	p	q	n	$\phi(n)$	e	d
	5	11	55	40	7	23

RSA Public-Key Cryptosystem: Example

Parameters:	p	q	n	$\phi(n)$	e	d
	5	11	55	40	7	23

Public key: $(7, 55)$

Private key: 23

RSA Public-Key Cryptosystem: Example

Parameters:

p	q	n	$\phi(n)$	e	d
5	11	55	40	7	23

Public key: $(7, 55)$

Private key: 23

Encryption: $M = 28, C = M^7 \bmod 55 = 52$

Decryption: $M = C^{23} \bmod 55 = 28$

RSA Public-Key Cryptosystem: Parameters

Parameters: p q n $\phi(n)$ e d

Public key: (e, n)

Private key: d

p, q, $\phi(n)$ must be kept secret!

RSA Public-Key Cryptosystem: Parameters

Parameters: p q n $\phi(n)$ e d

Public key: (e, n)

Private key: d

p, q, $\phi(n)$ must be kept secret!

$\mathcal{Q}$: Why?

The Security of the RSA

Brute-force attack:

Trying all possible private keys.

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

The Security of the RSA

Brute-force attack:

Trying all possible private keys.

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

Attack: Factor n into pq.

Attack: Determine $\phi(n)$ directly.

Attack: Determine d directly.

The Security of the RSA

Brute-force attack:

Trying all possible private keys.

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

Attack: Factor n into pq.

Attack: Determine $\phi(n)$ directly.

Attack: Determine d directly.

Comment: It is believed that determining $\phi(n)$ is equivalent to factoring n. Meanwhile, determining d given e and n, appears to be at least as time-consuming as the integer factoring problem.

The Security of the RSA

In practice, RSA keys are typically 1024 to 2048 bits long.

The Security of the RSA

In practice, RSA keys are typically 1024 to 2048 bits long.

Remark: There are some suggestions for choosing p and q.

A. Salomaa, *Public-Key Cryptography*, 2nd Edition, Springer, 1996, pp. 134-136.

The Security of the RSA

In practice, RSA keys are typically 1024 to 2048 bits long.

Remark: There are some suggestions for choosing p and q.

A. Salomaa, *Public-Key Cryptography*, 2nd Edition, Springer, 1996, pp. 134-136.

$\mathcal{Q}$: Consider the RSA system, where $n = pq$ is the modulus. Let (e, d) be a key pair for the RSA. Define

$$\lambda(n) = \text{lcm}(p-1, q-1)$$

and compute $d' = e^{-1} \bmod \lambda(n)$. Will decryption using d' instead of d still work?

Using RSA for Digital Signature

$S = M^d \bmod n$ (RSA **signature)**

$M = S^e \bmod n$ (RSA **verification)**

Why?

The Discrete Logrithm

- **The discrete logarithm** of an integer y to the base b is an integer x, such that

$$b^x \equiv y \bmod n.$$

The Discrete Logrithm

- **The discrete logarithm** of an integer y to the base b is an integer x, such that

$$b^x \equiv y \bmod n.$$

Discrete Logarithm Problem:

Given n, b and y, find x.

The Discrete Logrithm

- **The discrete logarithm** of an integer y to the base b is an integer x, such that

$$b^x \equiv y \bmod n.$$

Discrete Logarithm Problem:

Given n, b and y, find x.

This is very hard!!!

El Gamal Encryption

- **Setup** Let p be a prime, and g be a generator of $\mathbb{Z}_p$. The private key x is an integer with $1 < x < p - 2$. Let $y = g^x \bmod p$. The public key for *El Gamal encryption* is (p, g, y).

El Gamal Encryption

- **Setup** Let p be a prime, and g be a generator of $\mathbb{Z}_p$. The private key x is an integer with $1 < x < p-2$. Let $y = g^x \bmod p$. The public key for *El Gamal encryption* is (p, g, y).

  El Gamal Encryption: Pick a random integer k from $\mathbb{Z}_{p-1}$,

  $$a = g^k \bmod p$$
  $$b = My^k \bmod p$$

  The ciphertext C consists of the pair (a, b).

  El Gamal Decryption:

  $$M = b(a^x)^{-1} \bmod p$$

Using El Gamal for Digital Signature

$$a = g^k \bmod p$$
$$b = k^{-1}(M - xa) \bmod (p-1)$$

(El Gamal **signature)**

$$y^a a^b \equiv g^M \pmod{p}$$

(El Gamal **verification)**

Using El Gamal for Digital Signature

$$a = g^k \bmod p$$
$$b = k^{-1}(M - xa) \bmod (p-1)$$

(El Gamal **signature)**

$$y^a a^b \equiv g^M \pmod{p}$$

(El Gamal **verification)**

$\mathcal{Q}$: How to verify it?

An Example

Choose $p = 2579$, $g = 2$, and $x = 765$. Hence $y = 2^{765} \bmod 2579 = 949$.

An Example

Choose $p = 2579$, $g = 2$, and $x = 765$. Hence $y = 2^{765} \bmod 2579 = 949$.

- **(Public key)** $k_e = (p, g, y) = (2579, 2, 949)$
- **(Private key)** $k_d = x = 765$

An Example

Choose $p = 2579$, $g = 2$, and $x = 765$. Hence $y = 2^{765} \bmod 2579 = 949$.

- **(Public key)** $k_e = (p, g, y) = (2579, 2, 949)$
- **(Private key)** $k_d = x = 765$

Encryption: Let $M = 1299$ and choose a random $k = 853$,

$$\begin{aligned}(a, b) &= (g^k \bmod p, My^k \bmod p) \\ &= (2^{853} \bmod 2579, 1299 \cdot 949^{853} \bmod 2579) \\ &= (435, 2396).\end{aligned}$$

Decryption:

$$M = b(a^x)^{-1} \bmod p = 2396 \times (435^{765})^{-1} \bmod 2579 = 1299.$$

Security of the El Gamal Cryptosystem

Question 1: Is it feasible to derive x from (p, g, y)?

Security of the El Gamal Cryptosystem

Question 1: Is it feasible to derive x from (p, g, y)?

It is equivalent to solving the DLP. It is believed that there is NO polynomial-time algorithm. p should be large enough, typically 160 bits.

Security of the El Gamal Cryptosystem

Question 1: Is it feasible to derive x from (p, g, y)?

It is equivalent to solving the DLP. It is believed that there is NO polynomial-time algorithm. p should be large enough, typically 160 bits.

Question 2: Given a ciphertext (a, b), is it feasible to derive the plaintext M?

Security of the El Gamal Cryptosystem

Question 1: Is it feasible to derive x from (p, g, y)?

It is equivalent to solving the DLP. It is believed that there is NO polynomial-time algorithm. p should be large enough, typically 160 bits.

Question 2: Given a ciphertext (a, b), is it feasible to derive the plaintext M?

Attack 1: Use $M = by^{-k}$. However, k is randomly picked.

Attack 2: Use $M = b(a^x)^{-1} \bmod p$, but x is secret.

Diffie-Hellman Key Exchange Protocol

User A		User B
Generate random $X_A < p$		Generate random $X_B < p$
calculate $Y_A = \alpha^{X_A} \bmod p$	$Y_A \rightarrow$	Calculate $Y_B = \alpha^{X_B} \bmod p$
Calculate $k = (Y_B)^{X_A} \bmod p$	$\leftarrow Y_B$	Calculate $k = (Y_A)^{X_B} \bmod p$

Announcements

- **Homework assignment** 3

  ◇ P245 Ex. 37, 38, 39, P255 Ex. 2, 26, P272 Ex. 11*, 12, P273 Ex. 42, P274 Ex. 50, 55, P284 Ex. 7*, P285 Ex. 22, P286 Ex. 39, P305 Ex. 23, 28, 30

  ◇ Due on *Nov. 7th, 2017 at the beginning of class*

  ◇ Please try you best to slove problems marked with $*$

  ◇ Please write your homeowrk **neatly**, as a courtesy to graders.

Next Lecture

- induction, ...