AN TOÀN VÀ BẢO MẬT THÔNG TIN

Chương 0: Giới thiệu môn học

Nguyễn Duy Phúc
duyphucit@live.com
Vĩnh Long, 02/2014

Tổ chức môn học

- Thời gian học:

  60h = 10 tuần x 6h
- Từ 10/02/2014 đến 19/04/2014
- Kiểm tra thường xuyên: 3
- Thi cuối kỳ: thực hành
- Điều kiện dự thi: dự giảng >=80%, trung bình kiểm tra thường xuyên >=5

Nội dung môn học (1)

- Chương 1: Tổng quan
- Chương 2: Mã hóa khóa bí mật
- Chương 3: DES
- Chương 4: Mã hóa khóa công khai
- Chương 5: Hàm băm
- Chương 6: Mã xác thực thông điệp
- Chương 7: Chữ ký số
- Chương 8: Bảo mật mạng và Internet

Nội dung môn học (2)

- Chương 9: Xâm nhập (Intruder)
- Chương 10: Mã độc (Malware)
- Chương 11: Tường lửa (Firewall)

Tài liệu tham khảo

- Slides bài giảng môn học
- William Stallings: **Cryptography and Network Security** – Prentice Hall, 2011
- Chuck Easttom: **Computer Security Fundamentals** – Pearson, 2012
- Eric Cole, etc. : **Network Security Fundamentals** – Wiley, 2008
- Keyword: computer security, network security, cryptography

Thông tin liên lạc

- Nguyễn Duy Phúc
- Khoa Công nghệ thông tin, Trường Đại học Sư phạm Kỹ thuật Vĩnh Long
- Email: duyphucit@live.com, phucnd@vlute.edu.vn

- Website môn học: sdrv.ms/ZANGIV

AN TOÀN VÀ BẢO MẬT THÔNG TIN

Chương 1: Tổng quan

Nguyễn Duy Phúc
duyphucit@live.com
Vĩnh Long, 02/2014

Khái niệm về bảo mật máy tính

- Bảo mật máy tính (computer security): hoạt động bảo vệ được thiết lập cho một hệ thống thông tin tự động nhằm đảm bảo tính toàn vẹn, sẵn sàng, bí mật của tài nguyên trong hệ thống.

Khái niệm về bảo mật máy tính (2)

- Tính bí mật (Confidentiality)
  - Bí mật dữ liệu (Data confidentiality)
  - Sự riêng tư (Privacy)
- Tính toàn vẹn (Integrity)
  - Toàn vẹn dữ liệu
  - Toàn vẹn hệ thống
- Tính sẵn sàng (Availability)

Khái niệm về bảo mật máy tính (3)

Ngoài ra còn

- Tính xác thực (Authenticity): xác minh được người dùng, nguồn dữ liệu
- Trách nhiệm (Accountability): ghi nhận được hoạt động của một thực thể trong hệ thống. Tránh việc phủ nhận thông tin (nonrepudiation) và phục vụ cho việc phân tích chứng cứ (forensic)

* Thực tế việc bảo mật gặp rất nhiều khó khăn

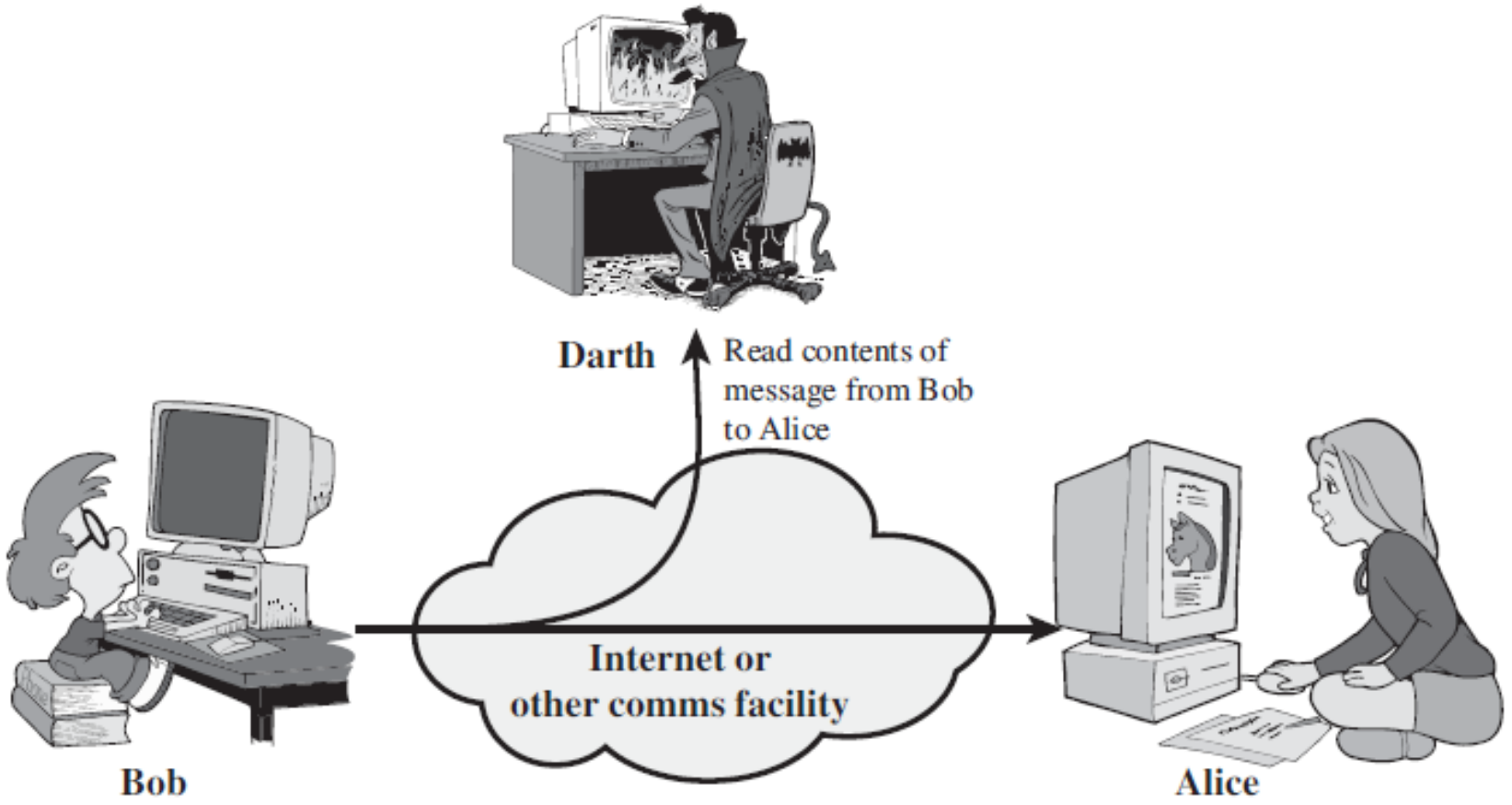
Một số khái niệm khác

- Threat: một yếu tố có thể gây nguy hại cho an ninh của hệ thống
- Tấn công (Attack): hoạt động có chủ ý gây nguy hại đến an ninh của hệ thống
- Cơ chế bảo mật (Security Mechanism): tiến trình/thiết bị được thiết lập để phát hiện, ngăn ngừa, phục hồi đối với tấn công vào hệ thống
- Dịch vụ bảo mật (Security Service): hoạt động sử dụng một hoặc nhiều cơ chế bảo mật để tăng cường tính an ninh cho hệ thống

Các hình thức tấn công

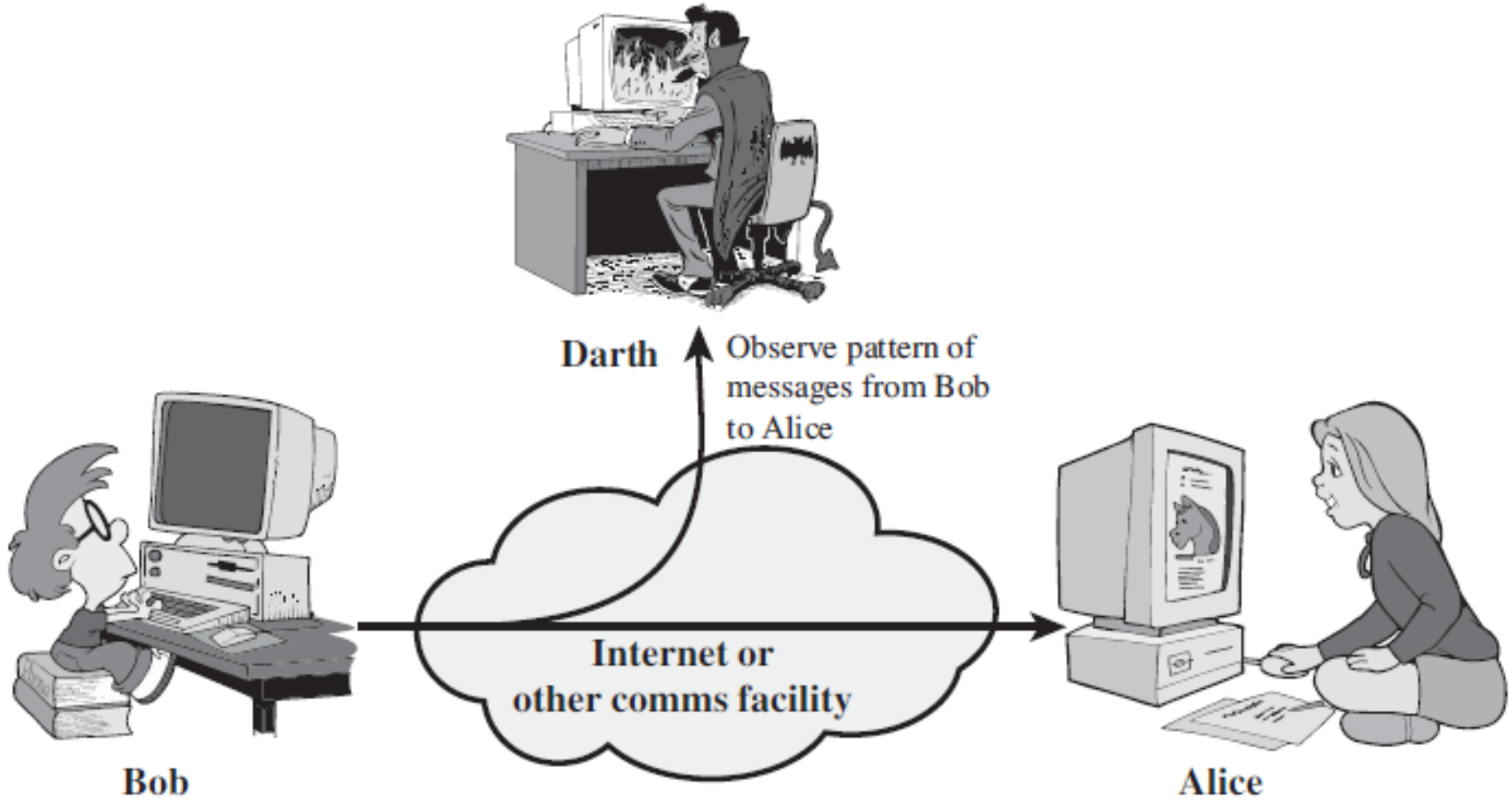
- Phân thành 2 loại:
  - Tấn công bị động (passive attack): lấy hoặc sử dụng thông tin của hệ thống
  - Tấn công chủ động (active attack): thay đổi thông tin hoặc cài đặt thêm các hoạt động không mong muốn vào hệ thống

Các hình thức tấn công (2)



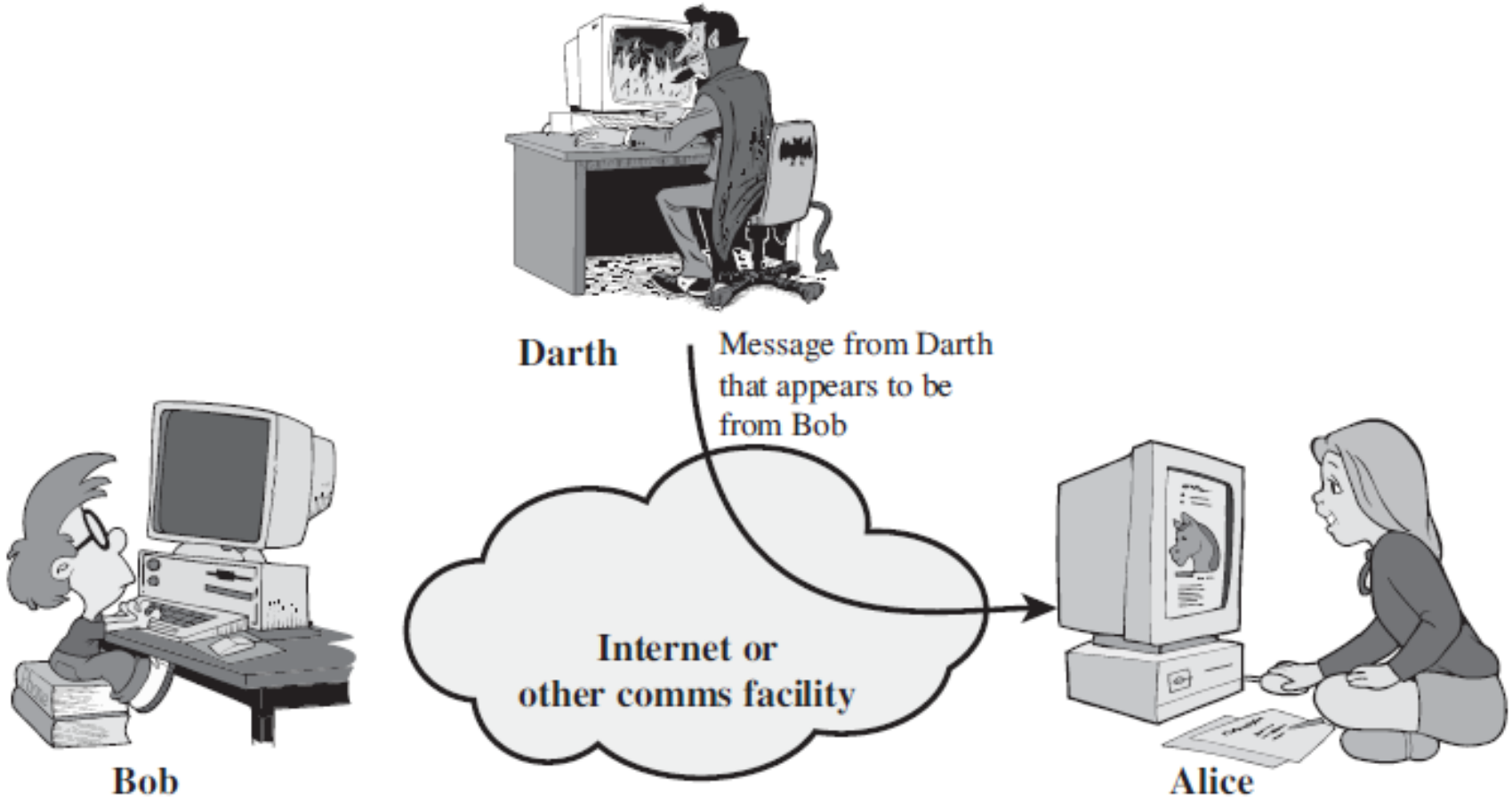
(a) Release of message contents

Các hình thức tấn công (3)



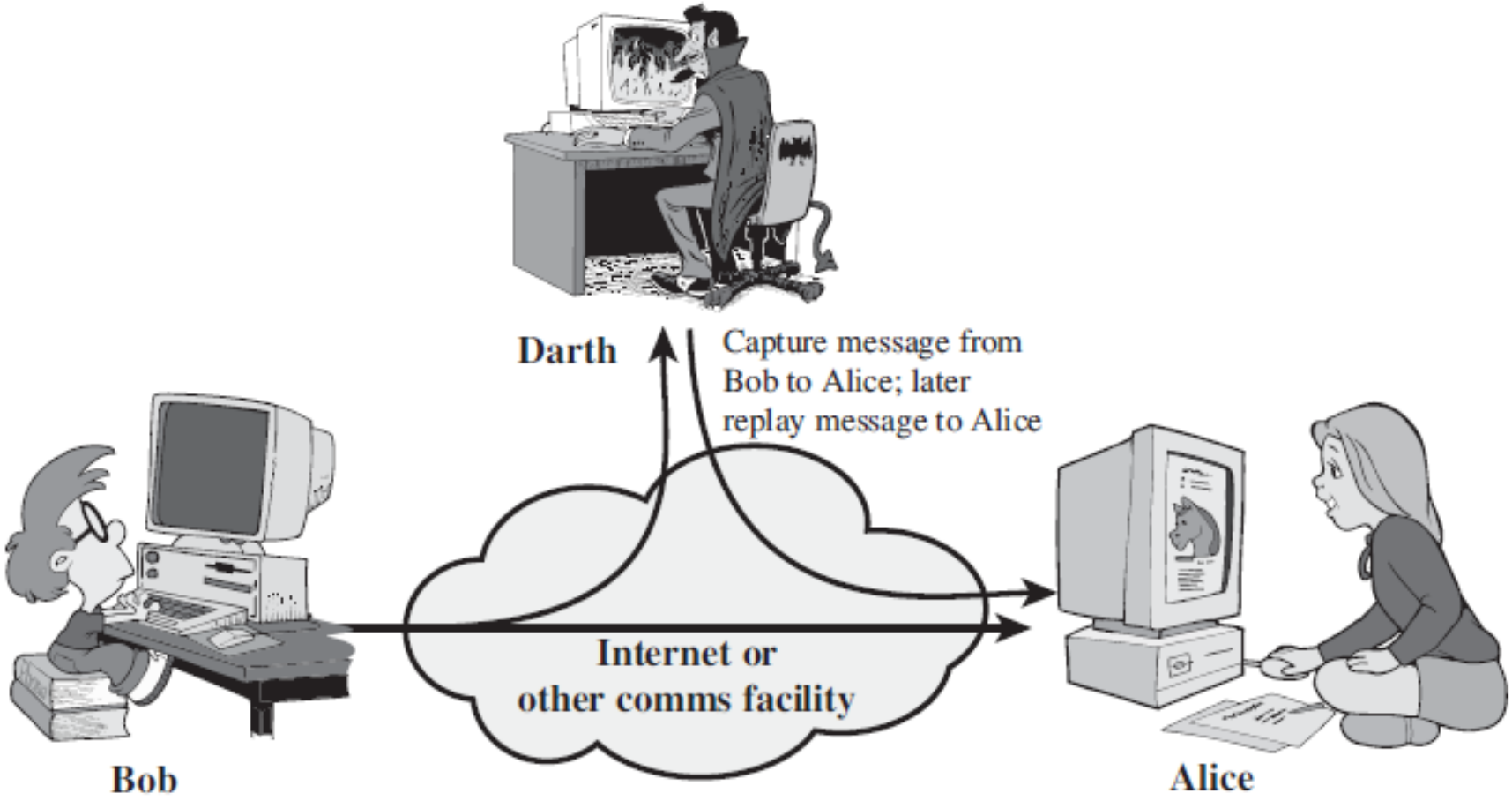
(b) Traffic analysis

Các hình thức tấn công (4)



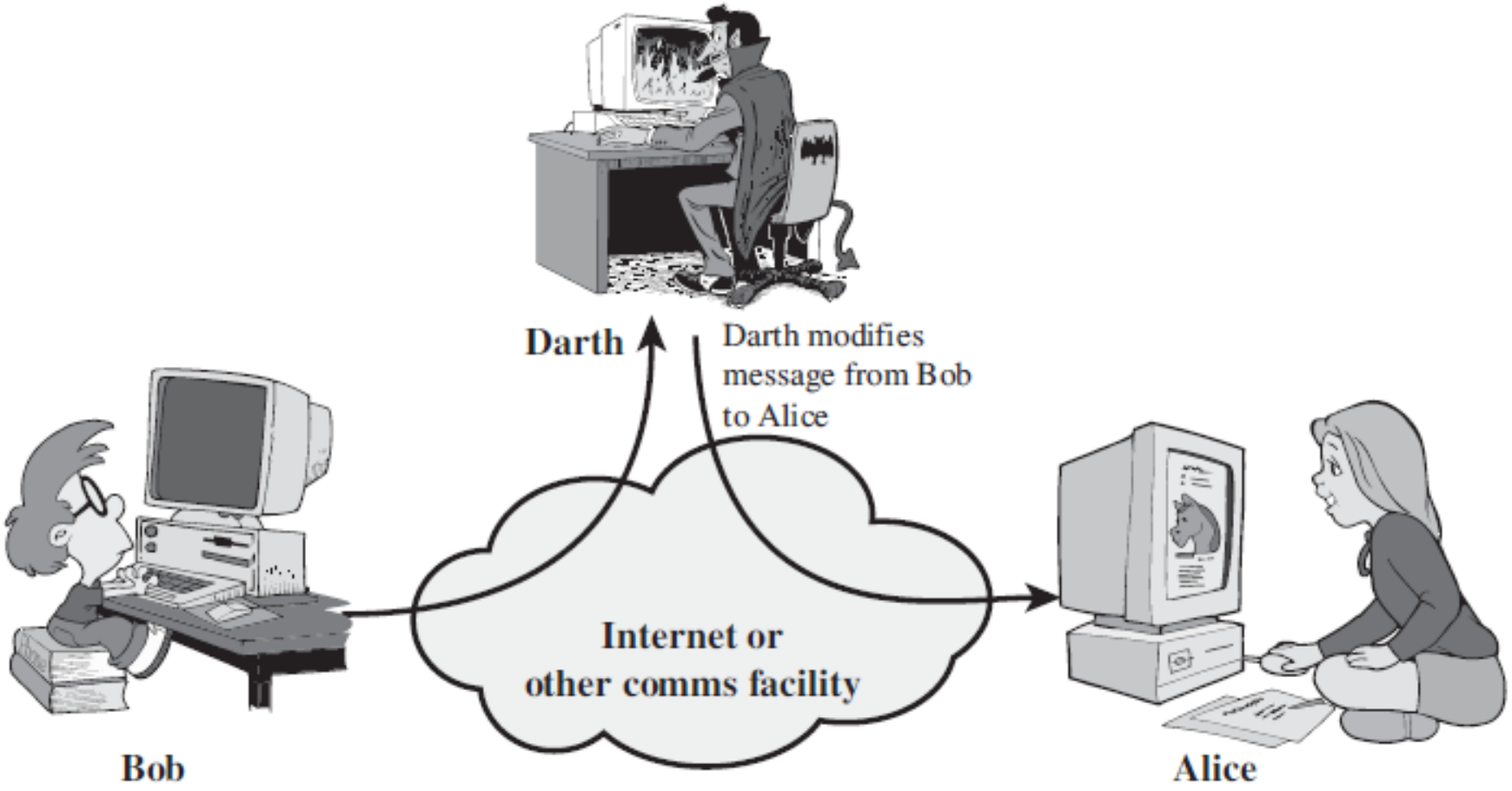
(a) Masquerade

Các hình thức tấn công (5)



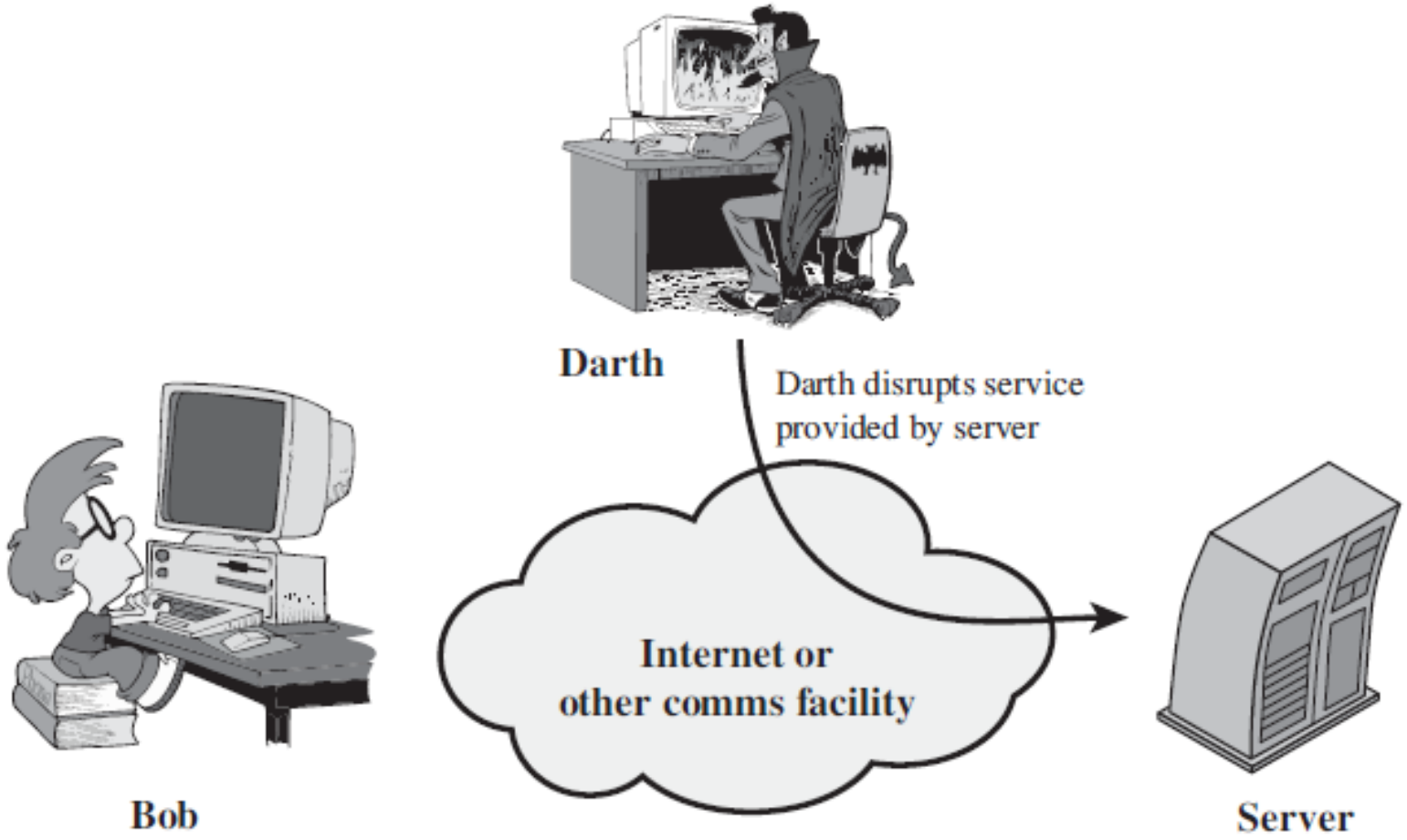
(b) Replay

Các hình thức tấn công (6)



(c) Modification of messages

Các hình thức tấn công (7)



(d) Denial of service

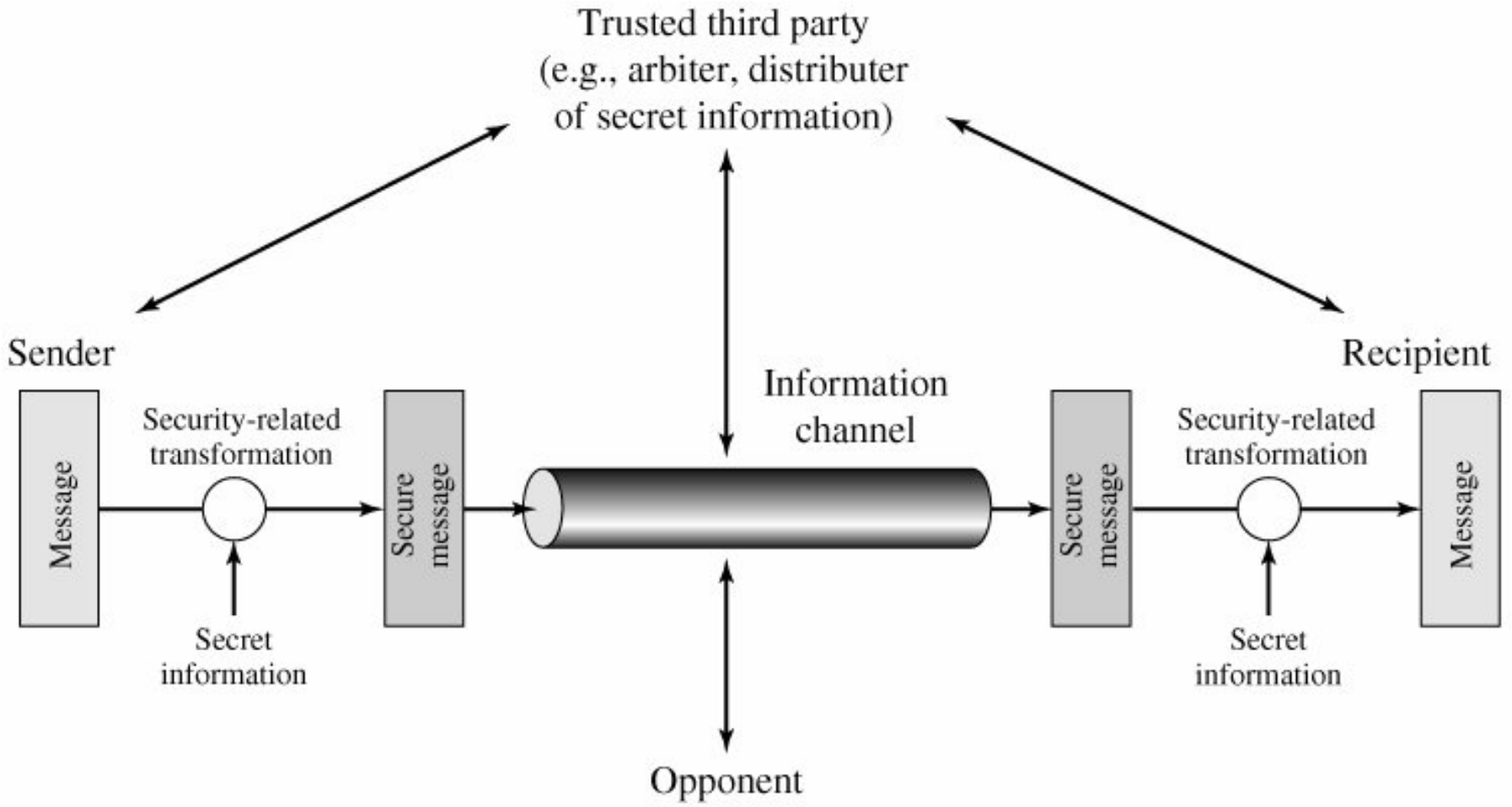
Các dịch vụ bảo mật

- Authentication – chứng thực nguồn gốc của các bên tham gia hoặc của dữ liệu khi truyền
- Access control – ngăn cản truy xuất tài nguyên bất hợp pháp
- Data confidentiality – bảo vệ dữ liệu không bị đọc trộm
- Data integrity – đảm bảo dữ liệu được nhận đúng như đã gửi
- Nonrepudiation – đảm bảo các bên tham gia không chối cãi được khi đã gởi/nhận thông tin

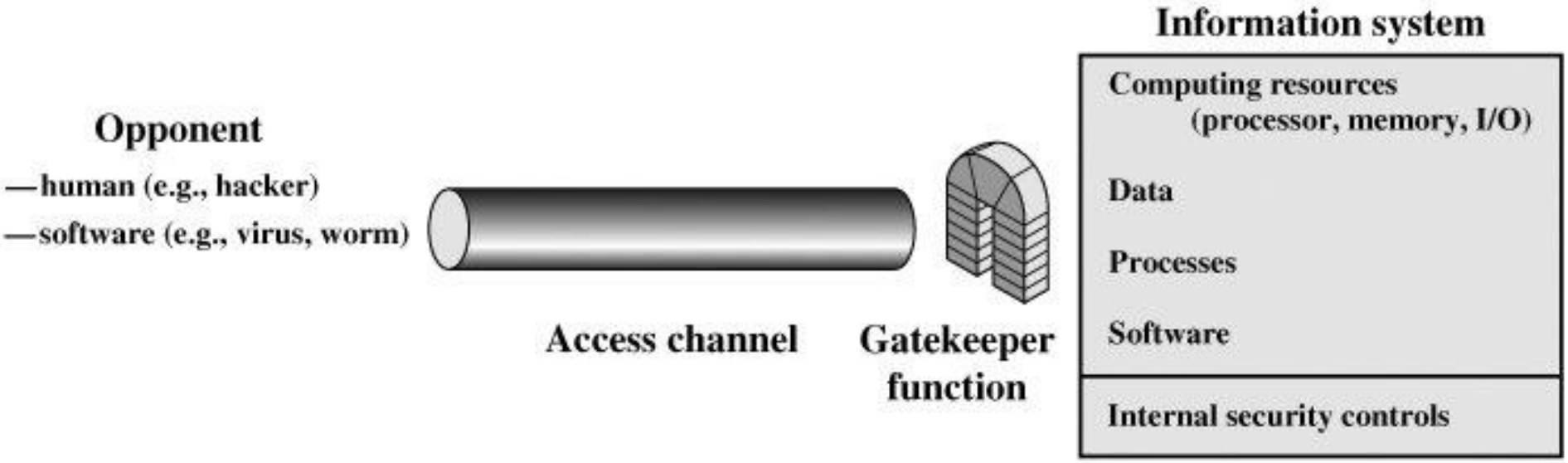
Các cơ chế bảo mật

- Encipherment – mã hóa thông tin
- Digital Signature – chữ ký số
- Access Control – quản lý quyền truy xuất tài nguyên
- Data Integrity – đảm bảo tính toàn vẹn của thông tin
- Authentication Exchange – trao đổi thông tin xác thực
- Traffic Padding – chống phân tích thông tin
- Routing Control – định tuyến truyền tin
- Notarization – xác thực dựa vào tổ chức trung gian (trusted third party)

Mô hình bảo mật mạng máy tính



Mô hình bảo mật mạng máy tính (2)



AN TOÀN VÀ BẢO MẬT THÔNG TIN

Chương 2: Mã hóa khóa bí mật

Nguyễn Duy Phúc
duyphucit@live.com
Vĩnh Long, 02/2014

Giới thiệu

- Các tên gọi:
  - Mã hóa đối xứng (symmetric encryption)
  - Mã hóa truyền thống (conventional)
  - Mã hóa khóa đơn (single-key)
- Là dạng mật mã mà quá trình mã hóa và giải mã sử dụng cùng một khóa

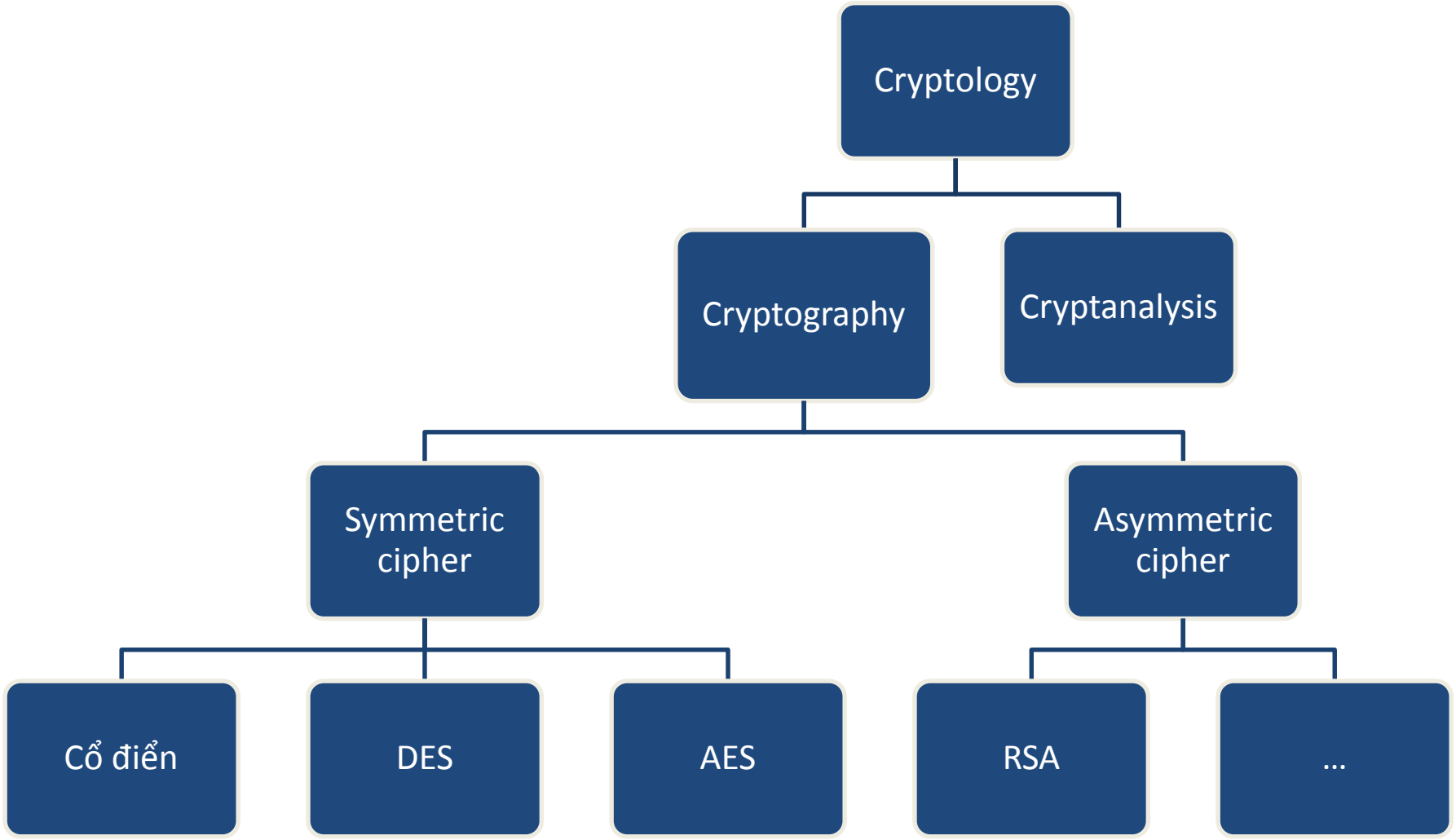
Một số khái niệm

- Plaintext (P): văn bản gốc
- Ciphertext (C): văn bản đã mã hóa
- Enciphering/Encryption (E): quá trình chuyển từ P → C
- Deciphering/Decryption (D): quá trình phục hồi từ C → P

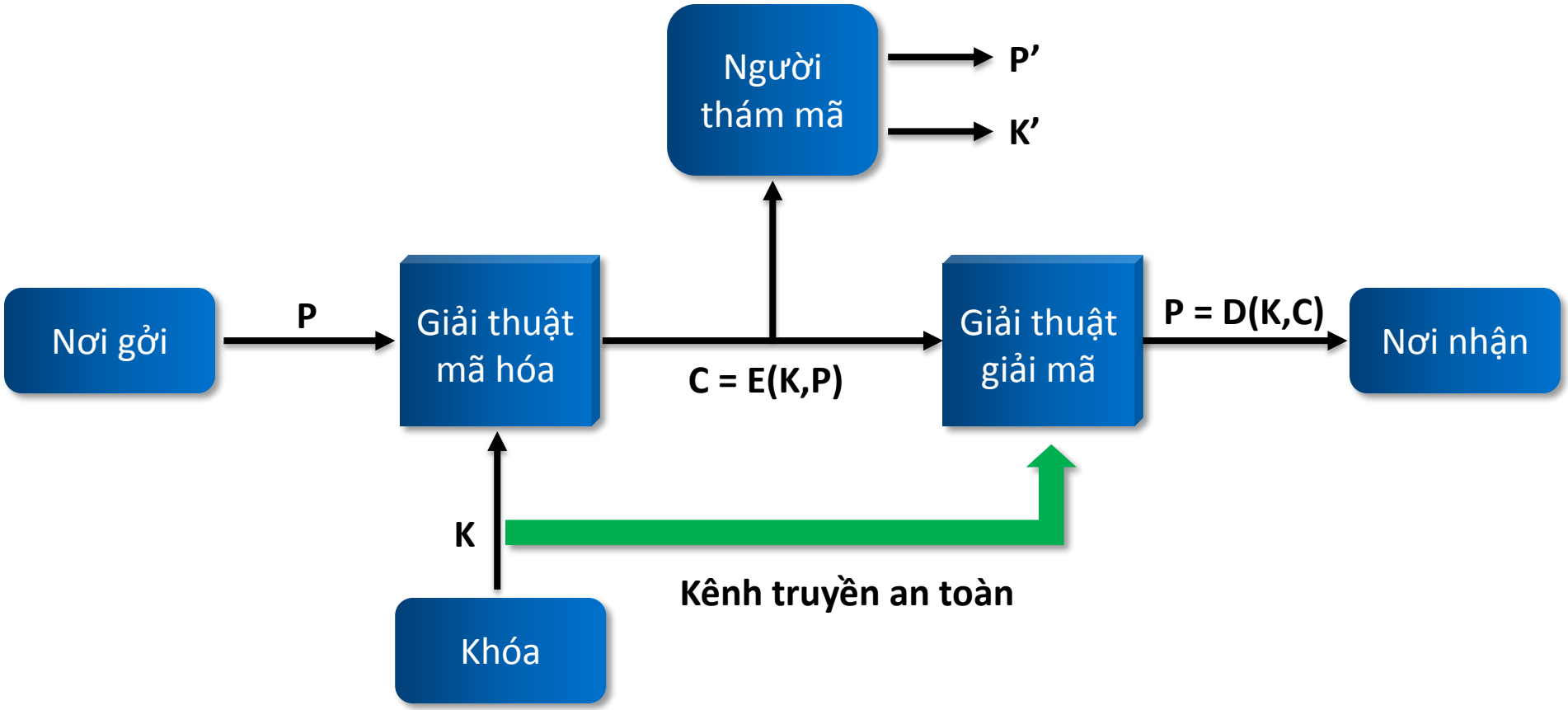
Một số khái niệm (2)

- Cryptography: khoa học mã hóa
- Cryptographic system/cipher: một hệ thống mã hóa cụ thể
- Cryptanalysis: khoa học thám mã
- Cryptology: khoa học mật mã, bao gồm cả cryptography và cryptanalysis

Một số khái niệm (3)



Mô hình mã hóa đối xứng



Mô hình mã hóa đối xứng (2)

Đặc điểm

- E, D mọi người đều có thể biết
- K chỉ có bên gởi và nhận biết
- E phải đủ mạnh để tránh thám mã ra P và K
- Kỹ thuật mã hóa thường dựa trên hai thao tác căn bản là thay thế (substitution) và đổi chỗ (transposition). Hệ thống kết hợp (product system) tăng độ phức tạp bằng cách sử dụng 2 thao tác trên nhiều lần

Thám mã và kỹ thuật vét cạn

Mục tiêu chính khi tấn công vào hệ thống mã hóa là tìm ra K. Có 2 cách:

- Thám mã: khai thác đặc điểm của giải thuật mã hóa để suy luận ra P, K
- Vét cạn (brute-force): giải mã với những khóa có thể có đến khi nào nhận được P "có nghĩa". Trung bình phải thử qua ít nhất ½ tập khóa

→ Về mặt lý thuyết luôn có thể tìm ra khóa bằng vét cạn

Thám mã và kỹ thuật vét cạn (2)

Kiểu thám mã	Người thám mã cần biết
Ciphertext Only	- E, C
Known Plaintext	- E, C - Một hoặc vài cặp P-C tương ứng được mã bằng K
Chosen Plaintext	- E, C - Được chọn P và có C tương ứng được mã bằng K
Chosen Ciphertext	- E, C - Được chọn C và có P tương ứng giải mã bằng K
Chosen Text	- E, C - Được chọn P và có C tương ứng được mã bằng K - Được chọn C và có P tương ứng giải mã bằng K

Độ an toàn của hệ thống mã hóa

Có 2 mức

- An toàn tuyệt đối (unconditionally secure): các thuật toán đều không thỏa mãn được (ngoại lệ: one-time pad)
- An toàn tính toán (computationally secure): thỏa một hoặc cả 2 điều kiện:
  - Chi phí để bẻ khóa cao hơn giá trị của thông tin
  - Thời gian bẻ khóa vượt quá thời gian hiệu lực của thông tin

Độ an toàn của hệ thống mã hóa (2)

Kích thước khóa (bit)	Số lượng khóa	Thời gian (tốc độ giải mã 1 /µs)	Thời gian (tốc độ giải mã 10^6 /µs)
32	$2^{32} = 4.3 \times 10^9$	2^{31}µs = 35.8 phút	2.15 ms
56	$2^{56} = 7.2 \times 10^{16}$	2^{55}µs = 1142 năm	10.01 giờ
128	$2^{128} = 3.4 \times 10^{38}$	2^{127}µs = 5.4×10^{24} năm	5.4×10^{18} năm
168	$2^{168} = 3.7 \times 10^{50}$	2^{167}µs = 5.9×10^{36} năm	5.9×10^{30} năm
Hoán vị 26 chữ cái	$26! = 4 \times 10^{26}$	2×10^{26}µs = 6.4×10^{12} năm	6.4×10^6 năm

Bảng thời gian trung bình thực hiện vét cạn khóa

Mã hóa Caesar

- Phát minh bởi Julius Caesar (100 BC – 44 BC) sử dụng truyền tin trong quân đội
- Thay thế ký tự trong thông điệp bởi ký tự đứng thứ 3 kế tiếp sau nó
- Ví dụ:

```
P:meet me after the toga party
C:PHHW PH DIWHU WKH WRJD SDUWB
```

Mã hóa Caesar mở rộng

- Tổng quát: nếu gán số thứ tự cho bảng chữ cái

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

- Với k = khoảng cách dịch chuyển, ta có:
  - c = E(k,p) = (p + k) mod 26
  - p = D(k,c) = (c - k) mod 26
- k từ 1..25, k = 3 chính là mã hóa Ceasar

Mã hóa Caesar mở rộng (2)

- Chỉ có 25 khóa → nếu biết ngôn ngữ của thông điệp sẽ dễ dàng vét cạn để tìm khóa
- Cách làm: lần lượt thử giải mã C bằng các khóa k = 1, 2, 3, … đến khi P nhận được "có nghĩa" thì k chính là khóa cần tìm
- Ví dụ: Thực hiện vét cạn tìm P, K với

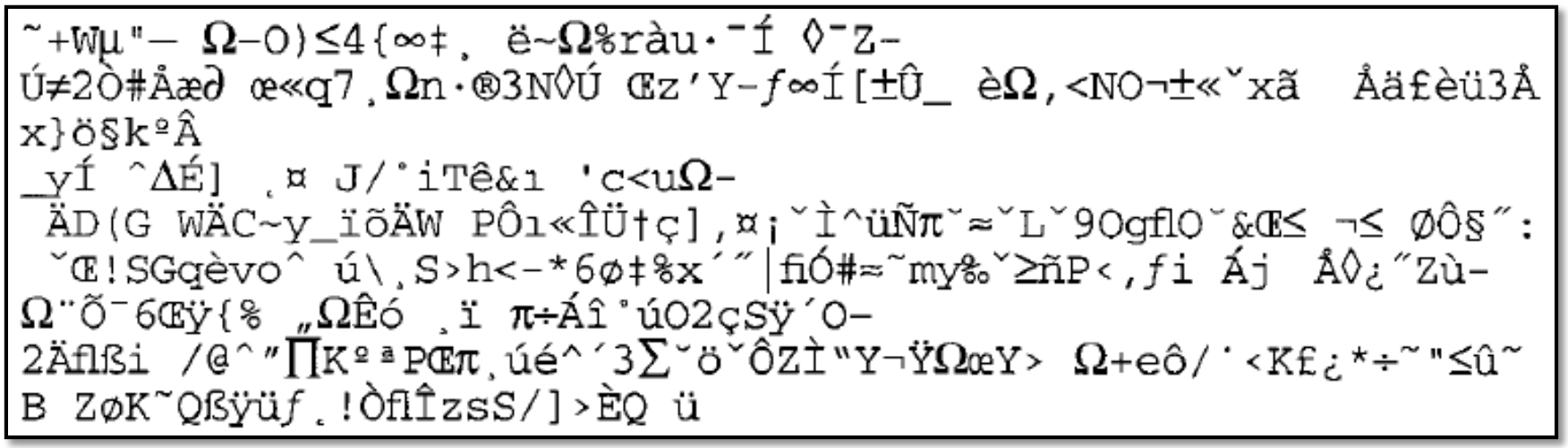
  C: **GCUA VQ DTGCM**

  P:

  K:

Mã hóa Caesar mở rộng (3)

- Điểm mấu chốt là biết được ngôn ngữ gốc, có thể khắc phục bằng cách chuyển P sang dạng khó nhận diện hơn trước khi mã hóa (nén, viết tắt)
- Ví dụ: 1 văn bản đã được nén



Mã hóa thay thế đơn (Monoalphabetic Cipher)

- Khóa chính là một cách sắp xếp các ký tự trong bảng chữ cái theo thứ tự tùy ý
- Ký tự trong P sẽ được thế thành ký tự tương ứng với sắp xếp trong khóa
- Ví dụ:

```
   ABCDEFGHIJKLMNOPQRSTUVWXYZ
K: DKVQFIBJWPESCXHTMYAUOLRGZN
P: ifwewishtoreplaceletters
C: WIRFRWAJUHYFTSDVFSFUUFYA
```

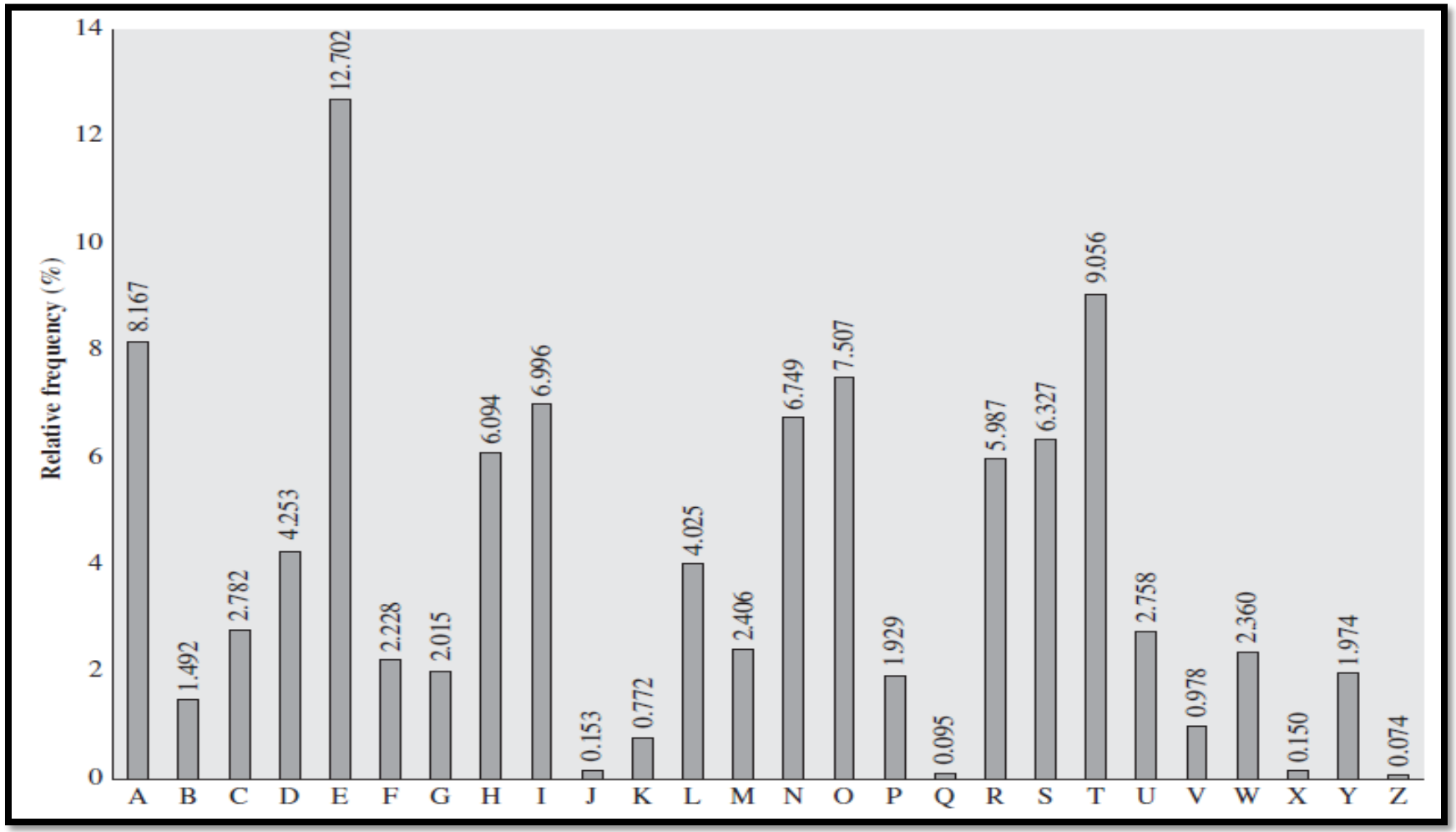
Mã hóa thay thế đơn (Monoalphabetic Cipher) (2)

- Số lượng khóa là 26! → khó vét cạn được khóa
- Nếu biết ngôn ngữ nguồn, có thể thám mã bằng cách phân tích dựa vào tần số xuất hiện của ký tự chữ cái trong ngôn ngữ

→ Cần có C đủ dài để phân tích chính xác

Mã hóa thay thế đơn (Monoalphabetic Cipher) (3)

- Ví dụ: Tần số chữ cái của tiếng Anh



Mã hóa thay thế đơn (Monoalphabetic Cipher) (4)

Cách thám mã dựa vào tần số

- Lập bảng thống kê tần số của các ký tự trong C
- Dựa vào bảng tần số chuẩn để dự đoán các ký tự trong C tương ứng
- Kết hợp với các phân tích: (giả sử là tiếng Anh)
  - Các ký tự liền kề nhau, ví dụ E thường đi theo sau T,R,N,I,O,A,S
  - Các từ biên giới: a, i, in, on, at, that, the, and, for,...
  - Các ký tự đi theo bộ 2, 3,…

Mã hóa Playfair

- Số lượng khóa nhiều chưa hẳn là an toàn
- Hướng tiếp cận là mã hóa cùng lúc nhiều ký tự
- Phát minh bởi Sir Charles Wheatstone năm 1854, đặt theo tên bạn ông là Baron Playfair
- Được sử dụng một thời gian dài bởi quân đội Anh, Mỹ, Đồng minh trong chiến tranh thế giới I, II

Mã hóa Playfair (2)

Ý tưởng

- Từ K xây dựng ma trận ký tự 5 x 5 bằng cách điền các ký tự của K vào ma trận theo thứ tự từ trái qua phải, từ trên xuống, không lặp lại ký tự trùng. Ký tự I và J xem như một
- Phần còn trống điền các ký tự còn lại của bảng chữ cái theo thứ tự
- Ví dụ: với khóa K = “GALOIS” ta có ma trận 5 x 5 như sau

Mã hóa Playfair (3)

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

Mã hóa Playfair (4)

- P được mã hóa theo từng cặp ký tự, nếu cuối cùng không đủ cặp thì thêm vào một ký tự đệm để đủ cặp (vd: x/q)
- Nếu cặp ký tự được chọn giống nhau thì chèn vào giữa một ký tự đệm và bắt cặp lại
- Ví dụ:

P: **little**

→ bắt cặp: **li tq tl eq**

Mã hóa Playfair (5)

- Mã hóa từng cặp ký tự dựa vào ma trận 5x5 theo quy tắc:
  - Nếu cặp ký tự nằm cùng dòng/cột thì thay mỗi ký tự bằng ký tự kế tiếp trong cùng dòng/cột (xoay vòng lại nếu đến cuối dòng/cột)
  - Trường hợp còn lại thì 2 ký tự sẽ là 2 đỉnh đối diện qua 1 đường chéo hình chữ nhật, thay lần lượt từng ký tự bằng ký tự ở đỉnh cùng dòng hoặc cùng cột (tùy theo người sử dụng giải thuật nhưng phải nhất quán)

Mã hóa Playfair (6)

- Ví dụ: theo ma trận 5x5

  AO → LI hoặc LJ

  LR → CX

  MY → TO

  BT → DQ hoặc QD

  AR → LQ hoặc QL

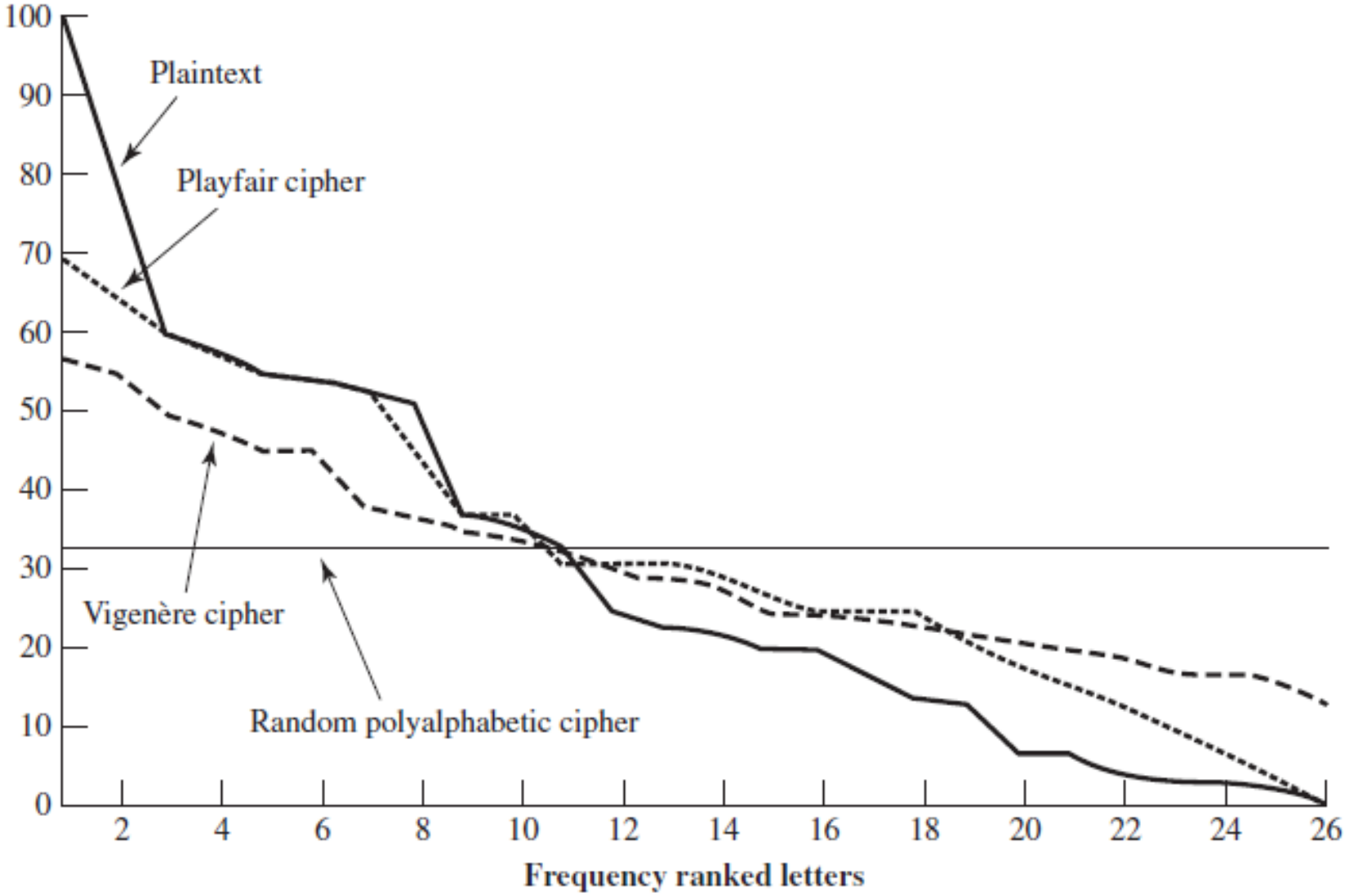
G	A	L	O	I/J
S	B	C	D	E
F	H	K	M	N
P	Q	R	T	U
V	W	X	Y	Z

Mã hóa Playfair (7)

Độ an toàn

- Có 26 x 26 = 676 cặp ký tự khác nhau → Khó bẻ khóa bằng phương pháp phân tích tần số
- Tuy nhiên khi lượng ciphertext đủ lớn thì vẫn có thể phân tích được
- Một dạng cải tiến là Double Playfair được quân Đức sử dụng ở WW II, thám mã bởi quân Anh
- Với tốc độ của máy tính hiện nay thì việc thám mã chỉ trong vài giây

Tần số xuất hiện ký tự của một số cipher



Các kỹ thuật mã hóa thay thế đa từ (Polyalphabetic Ciphers)

Đặc điểm chung

- Một tập các luật thay thế đơn được định nghĩa
- Khóa được sử dụng để chọn luật thay thế cho một lần chuyển đổi
- Mục đích là làm giảm sự chênh lệch tần số của ký tự → khó bẻ khóa bằng phân tích tần số
- Đại diện: Vigenère, Autokey system, One-time pad

Mã hóa Vigenère

- Đơn giản, dựa trên 26 khóa của Caesar
- Viết lại K nhiều lần để có chiều dài bằng P
- Mã lần lượt từng ký tự của P, lấy ký tự tương ứng của K làm khóa
- Như vậy với khóa K có độ dài m thì:

$$C_i = (P_i + K_{i \bmod m}) \bmod 26$$

- Để đơn giản ta sử dụng bảng Vigenère, trong đó C_i sẽ là giao của cột P_i và dòng $K_{i \bmod m}$

		Plaintext																								
		a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**	**n**	**o**	**p**	**q**	**r**	**s**	**t**	**u**	**v**	**w**	**x**	**y**	**z**
Key	*a*	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	b	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
	c	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
	d	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
	e	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
	f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
	g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
	h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
	i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
	j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
	k	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
	l	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
	m	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
	n	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
	o	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	p	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
	r	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
	s	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
	t	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
	u	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
	v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
	w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
	x	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
	y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
	z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Mã hóa Vigenère (3)

- Ví dụ:

  P: **wearediscoveredsaveyourself**

  K: **deceptivedeceptivedeceptive**

  C: **ZICVTWQNGRZGVTWAVZHCQYGLMGJ**

Mã hóa Vigenère (4)

Độ an toàn

- Làm mờ độ lệch tần số của ký tự, tuy nhiên vẫn có thể thám mã được (xem biểu đồ trước)
- Có thể đoán được độ dài khóa dựa vào khoảng cách các ký tự lặp lại. Ví dụ: khoảng cách giữa các cụm ký tự **VTW** trong ví dụ trước cho ta dự đoán khóa có độ dài 3 hoặc 9
- Sau đó có thể thám mã như thuật toán thay thế đơn. Ví dụ: các ký tự tại vị trí 1, 10, 19

Mã hóa khóa tự động (Autokey System)

- Để tránh việc lặp lại khóa nhiều lần, Vigenère đề nghị dùng khóa tự động
- Thay vì lặp lại khóa thì ghép phần đầu của thông điệp vào sau khóa để mã hóa
- Khi giải mã dựa vào khóa để giải mã ra phần đầu của thông điệp sau đó ghép vào khóa để giải mã tiếp

Mã hóa khóa tự động (Autokey System) (2)

- Ví dụ:

  P: **wearediscoveredsaveyourself**

  K: **deceptivewearediscoveredsav**

  C: **ZICVTWQNGKZEIIGASXSTSLVVWLA**

Mã hóa One-time Pad

- Do Joseph Mauborgne của Army Signal Corp đề nghị
- Mỗi thông điệp sẽ được mã hóa với một khóa riêng có chiều dài đúng bằng với thông điệp
- Do khóa hoàn toàn không có liên hệ gì với thông điệp nên thuật toán này không thể bẻ khóa
- Tuy nhiên ứng dụng thực tế rất khó do vấn đề tạo và trao đổi khóa

Mã hóa One-time Pad (2)

- Ví dụ: thám mã với C như sau

  ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS

- Ta tìm được 2 khóa đều tạo ra P có nghĩa

pxlmvmsydofuyrvzwc tnlebnecvgdupahfzzlmnyih

mr mustard with the candlestick in the hall

Và

pxlmvmsydofuyrvzwc tnlebnecvgdupahfzzlmnyih

miss scarlet with the knife in the library

- Khó xác định được P/K nào là đúng, hơn nữa mỗi P lại có một K khác

Kỹ thuật che giấu thông tin (Steganography)

- Không phải mã hóa thông tin
- Ý tưởng là giấu thông điệp cần gởi vào một thông tin khác theo cách chỉ có người gởi và người nhận biết
- Không an toàn nếu bị phát hiện cách giấu

→ cải thiện bằng cách mã hóa rồi giấu

- **Mục đích là không muốn người khác phát hiện có sự trao đổi thông tin, biết được người gởi và người nhận**

Kỹ thuật che giấu thông tin (Steganography) (2)

Một số cách giấu thông tin

- Đánh dấu các ký tự: ví dụ viết đè lên ký tự cần gởi bằng viết chì, chỉ thấy khi nghiêng giấy
- Mực không màu: chỉ thấy khi đốt nóng hoặc dùng hóa chất đặc biệt
- Sắp xếp các ký tự theo một vị trí đặc biệt
- Trên máy tính: giấu thông tin trong file ảnh, âm thanh, ..., sử dụng blog, diễn đàn, …

AN TOÀN VÀ BẢO MẬT THÔNG TIN

Chương 3: DES
(Data Encryption Standard)

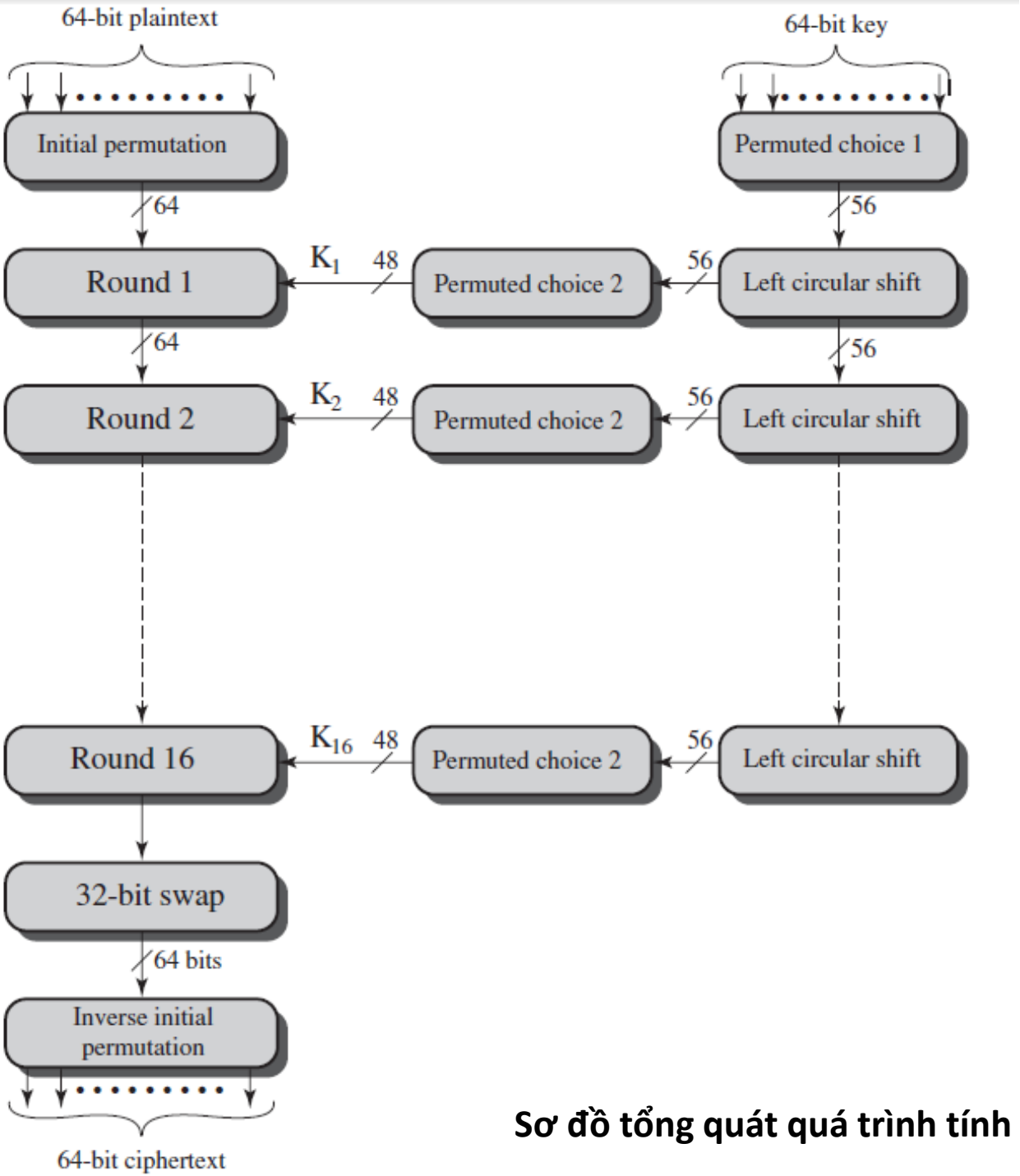
Nguyễn Duy Phúc
duyphucit@live.com
Vĩnh Long, 03/2014

Giới thiệu DES

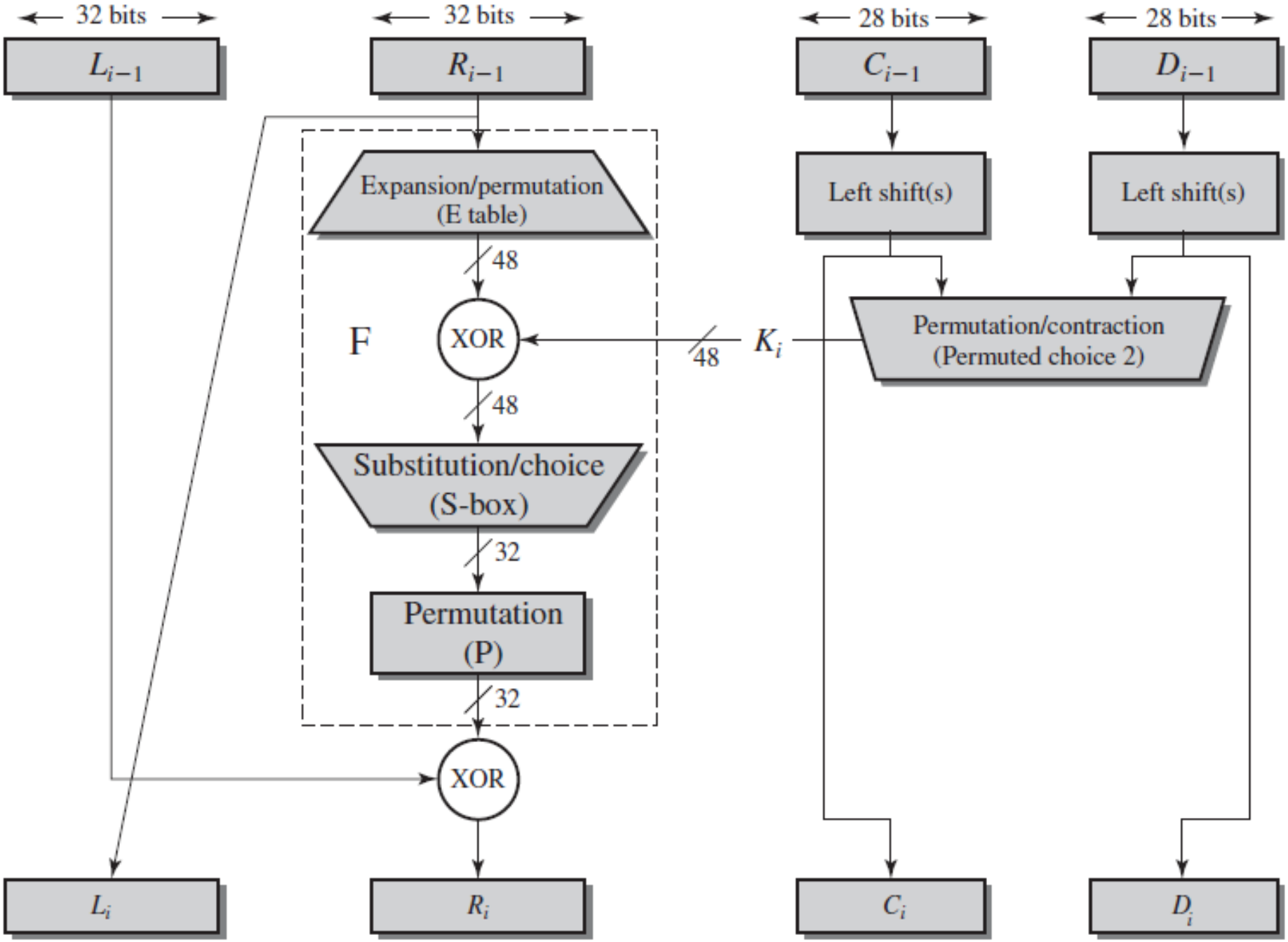
- Là thuật toán được sử dụng phổ biến nhất
- Được IBM phát triển dựa trên thuật toán Lucifer
- Được NIST (National Institute of Standards) công nhận năm 1977 (FIPS PUB 46)
- Sử dụng rộng rãi trong các ứng dụng thông thường, đặc biệt là trong tài chính
- Định kỳ 5 năm được xét duyệt lại, hiện đang dần được thay thế bởi AES

Khái quát quá trình mã hóa

- DES mã hóa dữ liệu theo từng khối 64 (block) bit
- Khóa yêu cầu có kích thước 64 bit, thực chất sử dụng chỉ 56 bit
- Quy trình mã hóa dựa theo cấu trúc Feistel, số vòng lặp là 16



Sơ đồ tổng quát quá trình tính toán của DES



Sơ đồ 1 vòng tính của DES

Tạo khóa con

- Khóa ban đầu 64 bit
- Loại bỏ 8 bit tại các vị trí 8, 16, 24, 32, 40, 48, 56, 64 và thực hiện hoán vị thông qua bảng PC1
- 56 bit kết quả được chia thành 2 khối C_0 , D_0 mỗi khối 28 bit
- Mỗi vòng lặp C_i, D_i sẽ có được từ phép quay trái các bit của C_{i-1} , D_{i-1}
- Vòng 1, 2, 9, 16 quay 1 bit, còn lại là 2 bit
- Áp dụng bảng PC2 cho C_i , D_i ta được K_i (48 bit)

Tạo khóa con (2)

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Khóa ban đầu (64 bit)

Tạo khóa con (3)

1	2	3	4	5	6	7
9	10	11	12	13	14	15
17	18	19	20	21	22	23
25	26	27	28	29	30	31
33	34	35	36	37	38	39
41	42	43	44	45	46	47
49	50	51	52	53	54	55
57	58	59	60	61	62	63

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

PC1 (56 bit)

Tạo khóa con (3)

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

PC2 (48 bit)

Mã hóa dữ liệu

- Dữ liệu vào 64 bit
- Đầu tiên là hoán vị khởi tạo (IP – Initial Permutation), chia thành 2 khối L_0, R_0 (32 bit)
- Thực hiện 16 vòng lặp, tính:
  - $L_i = R_{i-1}$
  - $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$
- Hoán vị L_{16} và R_{16}
- Cuối cùng thực hiện hoán vị nghịch đảo (IP^{-1}) cho ra kết quả mã hóa

Hoán vị khởi tạo (IP)

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

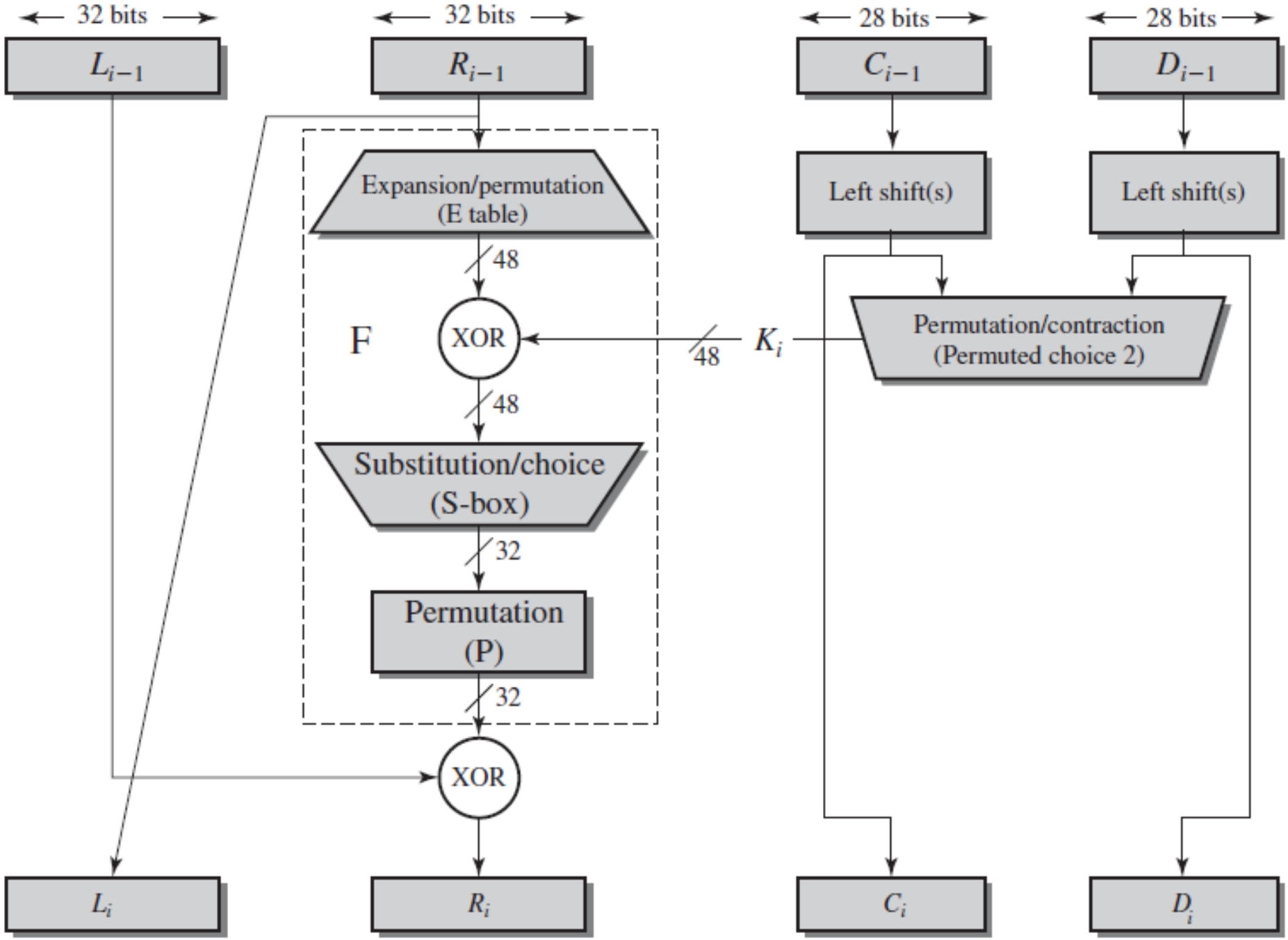
L_0 (32 bit)

R_0 (32 bit)

Vòng lặp biến đổi dữ liệu

- Tổng quát
  - $L_i = R_{i-1}$
  - $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$
- Trong đó hàm F
  - Hoán vị mở rộng $E(R_{i-1})$ từ 32 → 48 bit
  - XOR kết quả với K_i
  - Thay thế qua 8 S-box (6 → 4 bit) : $S_1..S_8$
  - Hoán vị P

  Tóm lại: $F(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$


32 bits
32 bits
28 bits
28 bits
L_{i-1}
R_{i-1}
C_{i-1}
D_{i-1}
Expansion/permutation
(E table)
Left shift(s)
Left shift(s)
48
F
XOR
48
K_i
Permutation/contraction
(Permuted choice 2)
48
Substitution/choice
(S-box)
32
Permutation
(P)
32
XOR
L_i
R_i
C_i
D_i

Hoán vị mở rộng (E)

$E(R_{i-1})$

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

S-box

- Dữ liệu vào 6 bit cho ra kết quả 4 bit
- S-box là bảng 4 x 16, 6 bit đầu vào thì 4 bit giữa để chọn cột, 2 bit ngoài để chọn dòng, dữ liệu ra là giao của hàng và cột
- Ví dụ: Dữ liệu vào là $(111010)_2$ cho qua S_1
  - Hàng là $(10)_2 = 2$
  - Cột là $(1101)_2 = 13$
  - $S_1(111010_2) = 10 = (1010)_2$

  Do giao của hàng 2 cột 13 trong S_1 là 10

S_1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

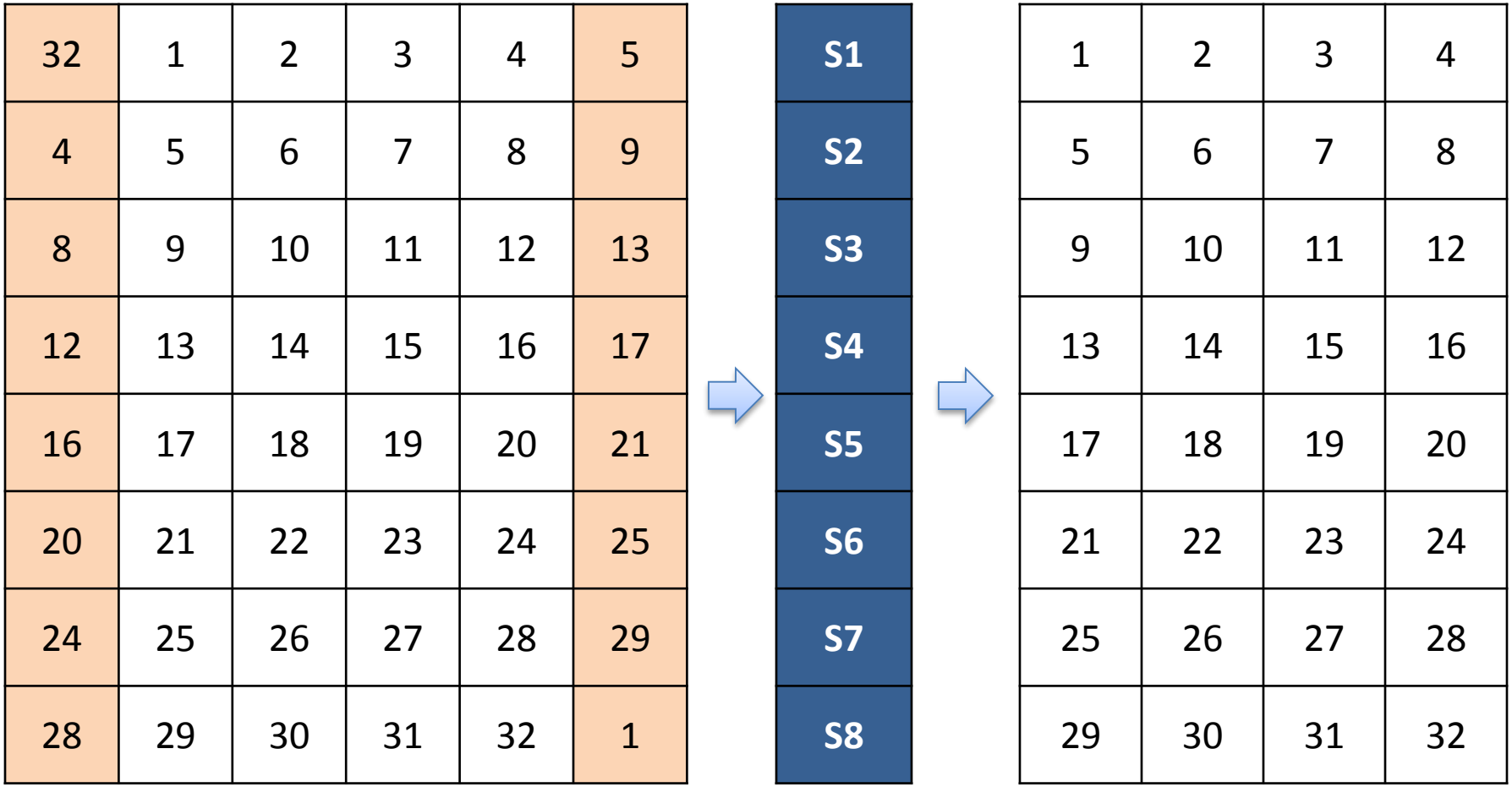
S_7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

S-box (2)



$E(R_{i-1})$ $S(E(R_{i-1}))$

Hoán vị P

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

$P(S(E(R_{i-1})))$

Hoán vị nghịch đảo (IP^{-1})

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Ví dụ

- Plain: 02468aceeca86420
- Key: 0f1571c947d9e859
- Cipher: da02ce3a89ecac3b

Vòng lặp	K_i	L_i	R_i
IP		5a005a00	3cf03c0f
1	1e030f03080d2930	3cf03c0f	bad22845
2	0a31293432242318	bad22845	99e9b723
3	23072318201d0c1d	99e9b723	0bae3b9e
4	05261d3824311a20	0bae3b9e	42415649
5	3325340136002c25	42415649	18b3fa41
6	123a2d0d04262a1c	18b3fa41	9616fe23
7	021f120b1c130611	9616fe23	67117cf2
8	1c10372a2832002b	67117cf2	c11bfc09
9	04292a380c341f03	c11bfc09	887fbc6c
10	2703212607280403	887fbc6c	600f7e8b
11	2826390c31261504	600f7e8b	f596506e
12	12071c241a0a0f08	f596506e	738538b8
13	300935393c0d100b	738538b8	c6a62c4e
14	311e09231321182a	c6a62c4e	56b0bd75
15	283d3e0227072528	56b0bd75	75e8fd8f
16	2921080b13143025	75e8fd8f	25896490
IP^{-1}		da02ce3a	89ecac3b

Độ an toàn của DES

- Số lượng khóa là 2^{56} (~ 7.2×10^{16})
- 7/1998 EFF (Electronic Frontier Foundation) dò khóa (brute-force) chưa tới 3 ngày bằng máy tính có giá trị khoảng 250,000$
- Ngoài ra DES có thể bị bẻ khóa bằng các kỹ thuật: Timing Attack, Differential Cryptanalysis, Linear Cryptanalysis
- Trong thực tế, để tăng độ an toàn người ta thường mã hóa DES 3 lần với khóa khác nhau (triple DES)

AN TOÀN VÀ BẢO MẬT THÔNG TIN

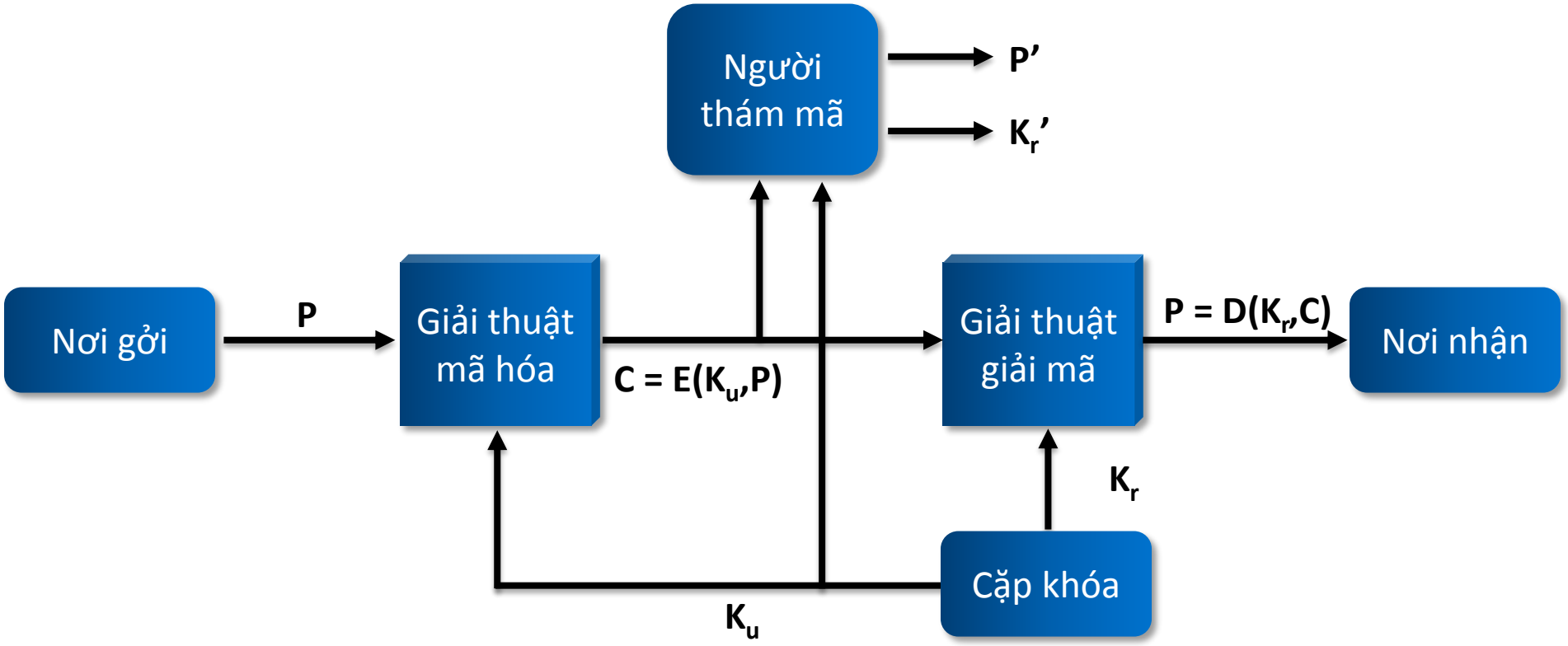
Chương 4: Mã hóa khóa công khai

Nguyễn Duy Phúc
duyphucit@live.com
Vĩnh Long, 03/2014

Giới thiệu

- Các tên gọi:
  - Mã hóa khóa công khai (Public-Key)
  - Mã hóa bất đối xứng (Asymmetric)
- Là dạng mật mã mà quá trình mã hóa và giải mã sử dụng khóa khác nhau – khóa công khai (public key) và khóa bí mật (private key)
- Được sử dụng để bảo mật (confidentiality), chứng thực (authentication)
- Thuật toán thường được sử dụng là RSA

Mô hình mã hóa công khai



Khóa bí mật và khóa công khai

Khóa bí mật

- Thuật toán mã hóa, giải mã tương tự nhau, sử dụng khóa giống nhau
- Hai bên phải chia sẻ khóa với nhau
- Khóa phải được giấu kín

Khóa công khai

- Thuật toán mã hóa, giải mã khác nhau, khóa này dùng mã hóa thì khóa kia dùng giải mã và ngược lại
- Mỗi bên phải có được 1 khóa tương ứng
- 1 trong 2 khóa phải được giữ bí mật

Thuật toán RSA

- Phát triển bởi Ron **R**ivest, Adi **S**hamir và Len **A**dleman tại MIT năm 1977
- Cơ sở thuật toán dựa vào phép lũy thừa trên trường Galoa của các số nguyên theo modulo của số nguyên tố
- Sự an toàn của RSA dựa trên độ khó của bài toán phân tích thừa số nguyên tố và bài toán logarit rời rạc

Cài đặt RSA

- Tạo cặp khóa công khai và cá nhân
  - Chọn 2 số nguyên tố lớn **$p \neq q$** (>120 chữ số)
  - Tính **$N = p.q$** và **$\phi(N) = (p-1).(q-1)$**
  - Chọn e sao cho **$0 < e < \phi(N)$** và **$\gcd(e, \phi(N)) = 1$**
  - Tính **$d = e^{-1} \bmod \phi(N)$**
  - Khóa công khai **$K_u = \{e, N\}$**
  - Khóa cá nhân **$K_r = \{d, n\}$**

Sử dụng RSA

- Mã hóa thông điệp **0 < M < N**
  - Người mã hóa sử dụng khóa công khai của người nhận $\mathbf{K_u = \{e, N\}}$
  - Tính $\mathbf{C = M^e \bmod N}$
- Giải mã
  - Người nhận sử dụng khóa cá nhân của mình $\mathbf{K_r = \{d, N\}}$
  - Tính $\mathbf{P = C^d \bmod N}$

Ví dụ RSA

- Giả sử B muốn gởi cho A thông điệp **M = 26**
- B1: A tính toán để có được $\mathbf{K_r}$ và $\mathbf{K_u}$ của mình
  - Chọn **p = 11, q = 47**
  - Tính **N = p.q = 517** $\quad$ **ϕ(N) = (p-1).(q-1) = 460**
  - Chọn ngẫu nhiên **e = 3**, kiểm tra e thỏa điều kiện **0 < 3 < 460** và **gcd(3, 460) =1**
  - Tính **d = e^{-1} mod ϕ(N) = 3^{-1} mod 460 = 307**
  - Khóa chung: $\mathbf{K_u}$ **= {3, 517}**
  - Khóa riêng: $\mathbf{K_r}$ **= {307, 517}**

Ví dụ RSA (2)

- B2: A công bố khóa chung $\mathbf{K_u = \{3, 517\}}$ cho B biết
- B3: B dùng $\mathbf{K_u}$ để mã hóa **M → C** rồi gởi cho A
  - Tính $\mathbf{C = M^e \bmod N = 26^3 \bmod 517 = 515}$
- B4: A dùng khóa riêng $\mathbf{K_r = \{307, 517\}}$ để giải mã **C → P**
  - Tính $\mathbf{P = C^d \bmod N = 515^{307} \bmod 517 = 26}$

Tính nghịch đảo a^{-1} theo modulo N

- Sử dụng thuật toán Euclid mở rộng
- Ví dụ: tính 299^{-1} mod 323

Vòng lặp	y	g	V
	_	323	0
	_	299	1
0	323 div 299 = 1	323 mod 299 = 24	0 - 1.1 = -1
1	299 div 24 = 12	299 mod 24 = 11	1 - (-1.12) = 13
2	24 div 11 = 2	24 mod 11 = 2	-1 - (13.2) = -27
3	11 div 2 = 5	11 mod 2 = 1 (dừng)	13 - (-27.5) = 148

Kết quả: 299^{-1} mod 323 = 148

Tính a^b mod n

- Sử dụng thuật toán bình phương và nhân
- Thuật toán:
  - Biểu diễn b dưới dạng nhị phân $b_k b_{k-1}..b_0$
  - Tính

```
c ← 0; f ← 1
for i ← k downto 0
    do  c ← 2 × c
        f ← (f × f) mod n
    if  b_i = 1
        then c ← c + 1
             f ← (f × a) mod n
return f
```

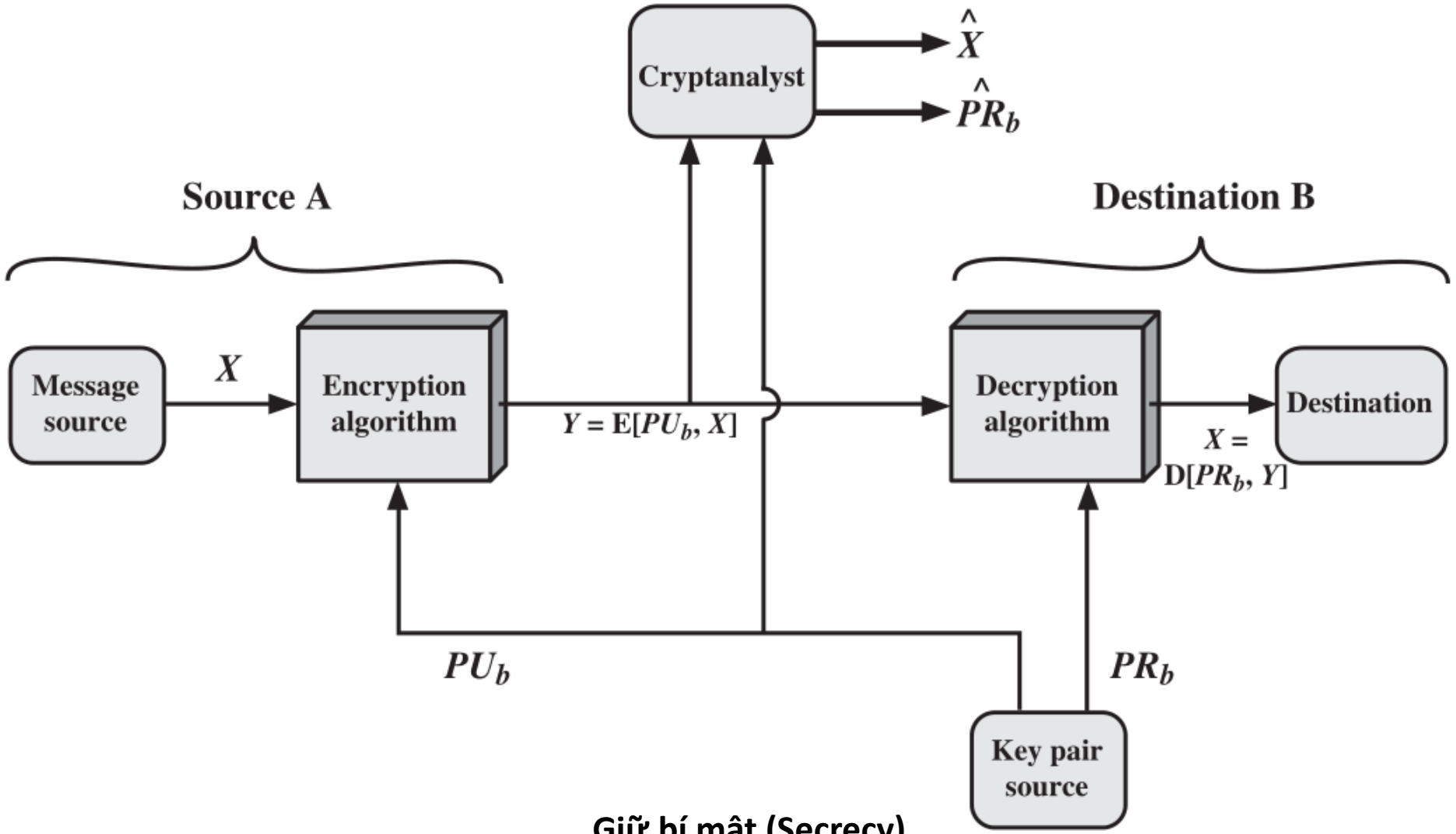
Tính a^b mod n (2)

- Ví dụ: tính 7^6 mod 19
  - $6 = (110)_2$

Vòng lặp	b_i	c	f
Khởi động		0	1
2	1	2.0 +1 = 1	1.1.7 mod 19 = 7
1	1	2.1 + 1 = 3	7.7.7 = 1
0	0	2.3 = 6	1.1 = 1

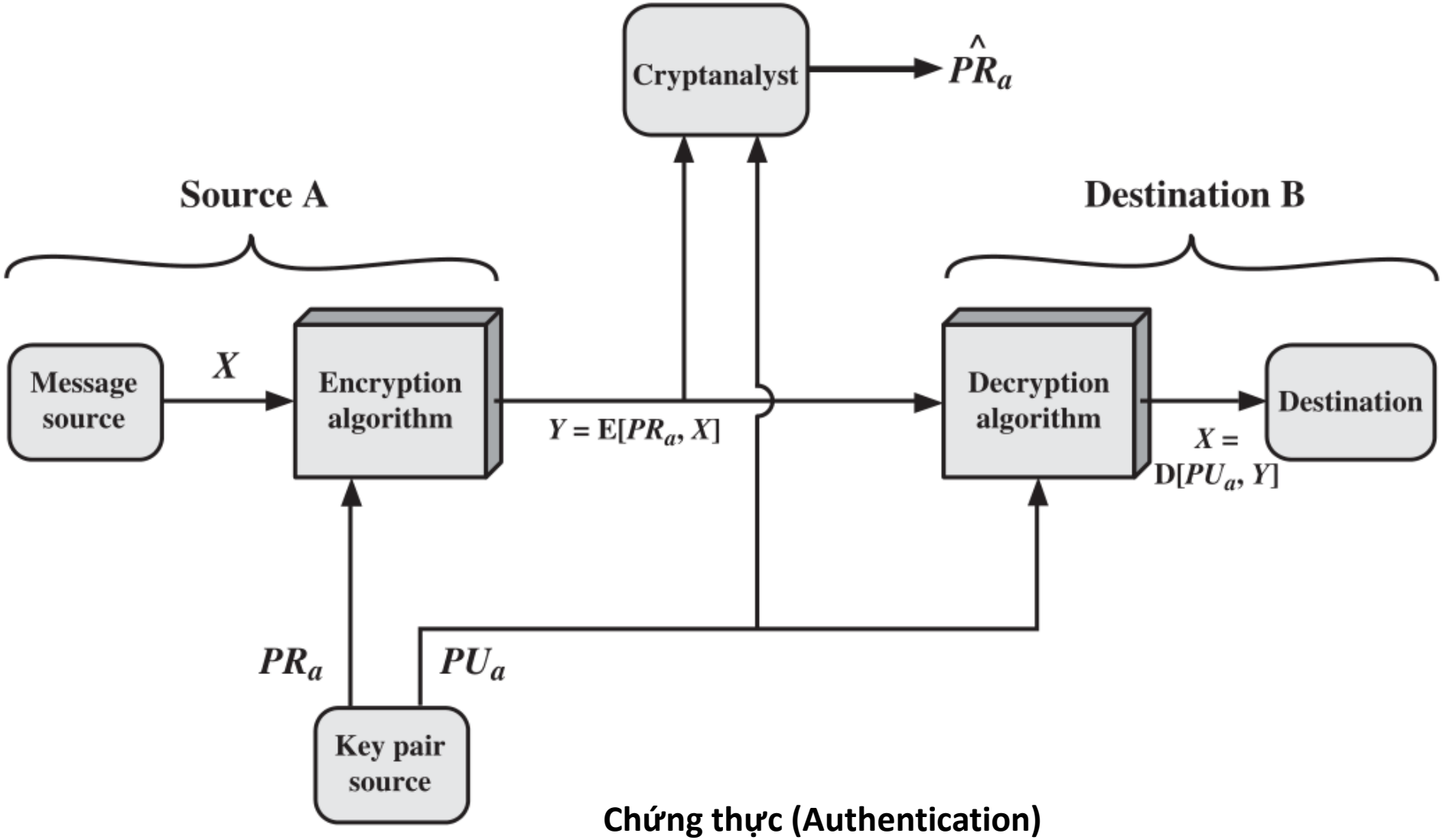
Kết quả 7^6 mod 19 = 1

Mô hình ứng dụng khóa công khai



Giữ bí mật (Secrecy)

Mô hình ứng dụng khóa công khai (2)



Chứng thực (Authentication)

Mô hình ứng dụng khóa công khai (3)