INDUSTRIAL UNIVERSITY OF HOCHIMINH CITY

MÃ HÓA
(Cryptography)

Mã hóa đối xứng
SYMMETRIC CIPHERS

NỘI DUNG

1. Giới thiệu
2. Những khái niệm cơ bản về mã hóa
3. Hàm cửa lật một chiều và mã công khai
4. Một số mã công khai thông dụng
5. Nguyên lý thiết kế mã đối xứng
6. Một số mã đối xứng thông dụng

Giới thiệu

MẬT THƯ 1: 45, 24, 34 – 12, 11 - 13- 14, 35, 11, 34 – 31, 15, 45

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

TIN BA C
ĐOÀN KẾT

Giới thiệu

MẬT THƯ 2:
TÍ VỀ - TUẤT THƯƠNG – HỢI NHẤT – SỬU HƯỚNG – DẬU YÊU –DẦN MẬT – MẸO TRỜI – TỊ SẼ - MÙI NGƯỜI – NGỌ THẤY – THÂN BẠN

VỀ HƯỚNG MẶT TRỜI BẠN SẼ THẤY

NGƯỜI BẠN YÊU THƯƠNG NHẤT

☞12 CON GIAP: TI, SỬU, DẦN, MẸO,
THÌN, NGỌ, MÙI, THÂN, DẬU, TUẤT, HỢI

Mã hóa là gì?

- Mã hóa là một phương pháp hỗ trợ rất tốt trong việc chống lại những truy cập bất hợp pháp tới dữ liệu được truyền đi qua các kênh truyền thông
- Mã hoá sẽ khiến cho nội dung thông tin được truyền đi dưới dạng mờ và không thể đọc được đối với bất kỳ ai cố tình muốn lấy thông tin đó

Các khái niệm cơ bản

- **Kỹ thuật mật mã (cryptology)** là ngành khoa học nghiên cứu 2 lĩnh vực: mã hóa (cryptography) và phân tích mật mã (cryptanalysis codebreaking)
- **Mật mã (Cryptography)** là ngành khoa học nghiên cứu các phương pháp và kỹ thuật đảm bảo an toàn và bảo mật dữ liệu trong việc truyền tin.

  W. Stallings (2003), *Cryptography and Network Security: Principles and Practice, Third Edition*, Prentice Hall

Khái niệm mã hóa

- Ứng dụng của mật mã:
  - Trong các cơ quan chính phủ: bảo vệ thông tin mật, các thông tin quân sự, ngoại giao, …
  - Trong lĩnh vực kinh tế: bảo mật thông tin tài khoản ngân hàng, giao dịch thanh toán, thông tin khách hàng, …
  - Trong y tế: bảo vệ thông tin cá nhân,
  - Trong bảo vệ thông tin cá nhân: thông tin riêng tư, tài khoản email, an toàn trên mạng xã hội, …

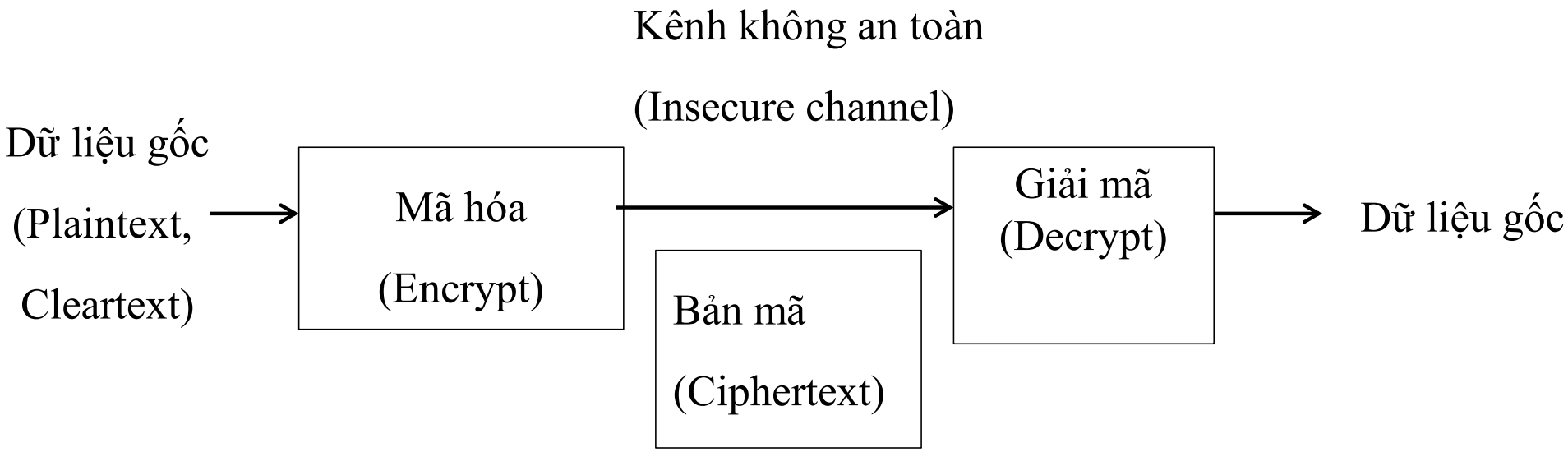
Khái niệm mã hóa

- **Phân tích mật mã (cryptanalysis)**: ngành khoa học nghiên cứu các phương pháp, kỹ thuật nhằm phá vỡ hệ thống mã hóa.
- Trong sự phát triển của mật mã thì lĩnh vực mật mã và phân tích mật mã phát triển song hành với nhau, tuy nhiên trong học tập, nghiên cứu thì lĩnh vực mật mã học được quan tâm rộng rãi hơn do các ứng dụng thực tiễn, hiệu quả mà nó đem lại.

Khái niệm mã hóa

- **Giao thức mật mã (cryptographic protocol)** là tập hợp các quy tắc, trình tự thực hiện sơ đồ mã hóa.
- **Độ an toàn của hệ mã hóa**: là khả năng chống lại việc thám mã, trong nhiều trường hợp được tính bằng số phép toán cần thực hiện để thám mã sử dụng thuật toán tối ưu nhất.
- **Hệ thống mật mã** (cryptosystem) là hệ thống đảm bảo an toàn dữ liệu sử dụng công cụ mã hóa. Hệ thống mật mã bao gồm: sơ đồ, giao thức mật mã, quy tắc tạo và phân phối khóa. Khái niệm hệ thống mật mã có thể hiểu đơn giản hơn là bao gồm: thuật toán (algorithm) và giá trị mật (key).

Sơ đồ mã hóa



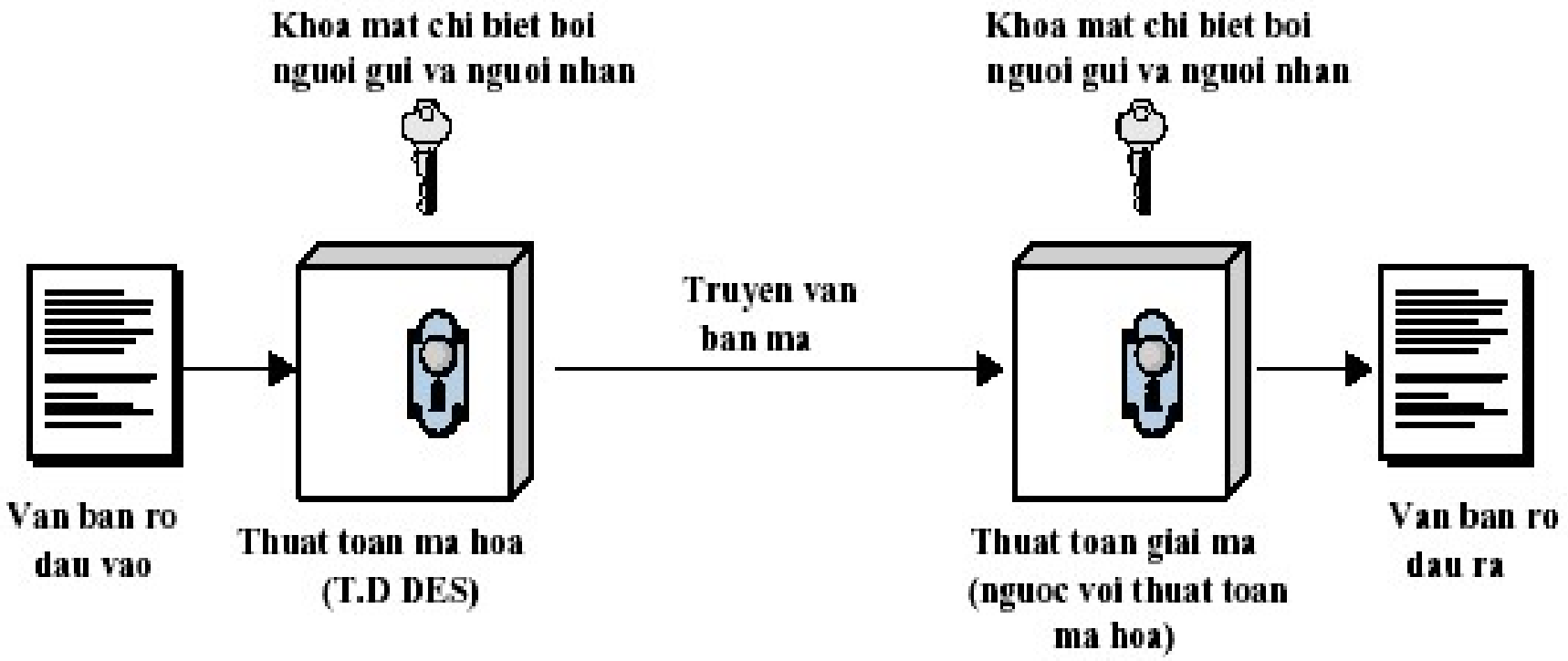
Mô hình toán học tổng quát:

Mã hóa: C=E(P)

Giải mã: P=D(C),

với Plaintext (P), Encrypt (E), Ciphertext (C), Decrypt (D)

Mô hình đơn giản của Mật Mã



Mô hình đơn giản của Mật Mã

- Ví dụ:

Nội dung gốc : "Hello everybody"

Mã hóa : dời nội dung sang phải – Keycode =1

→ "Ifmmp fwfsacpea"

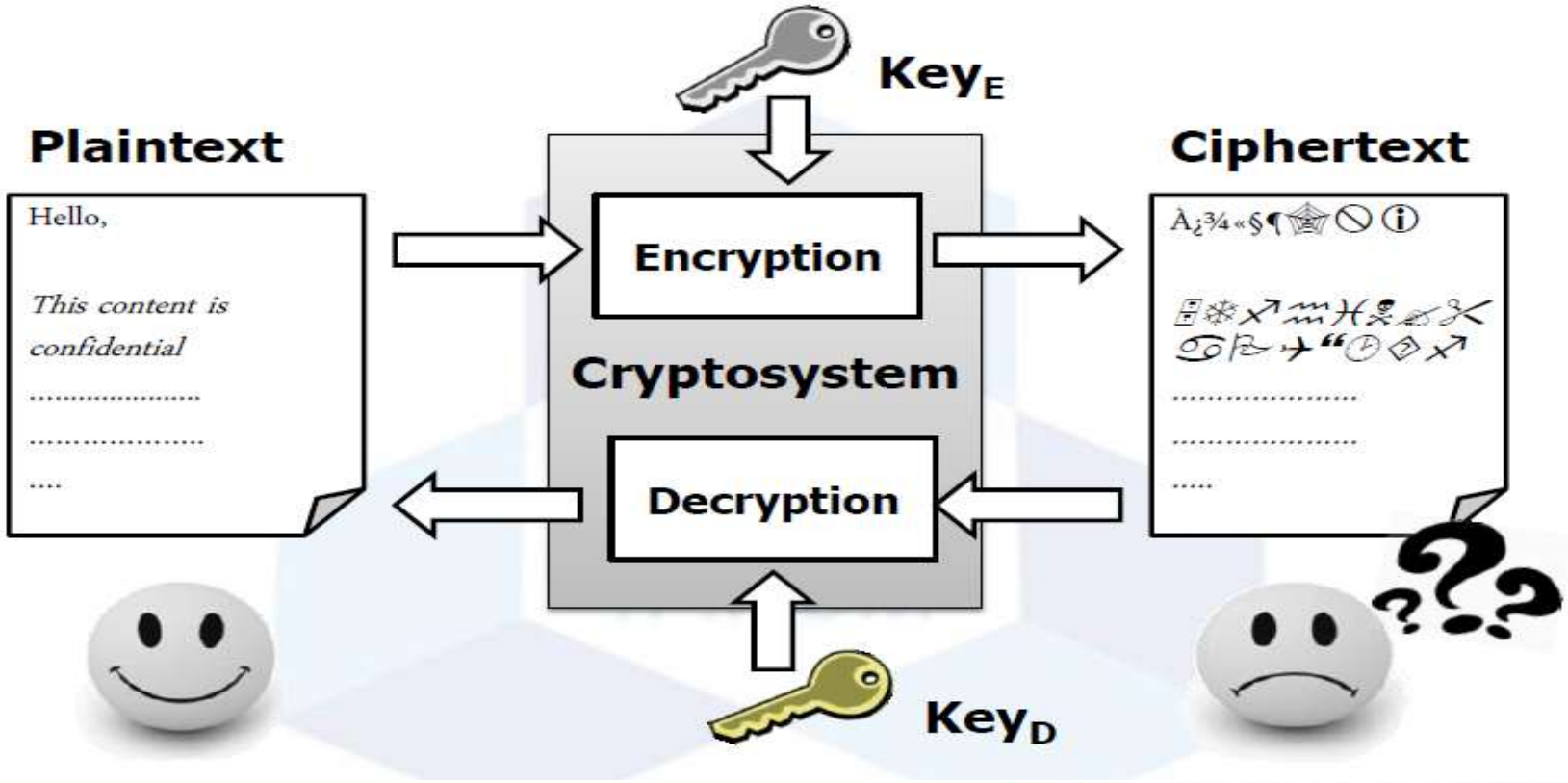
Giải mã : dời nội dung sang trái – Keycode =1

Hệ thống mã hóa

Hệ thống mã hóa (cryptosystem)

Cryptosystem = encryption + decryption algorithms

Khóa (key) được sử dụng trong quá trình mã hóa và giải mã



Phân loại mã hóa

Theo thời gian có thể chia mật mã thành:

- Mã hóa cổ điển (classical cryptographic)
- Mã hóa hiện đại (modern cryptography)

Ngoài ra, dựa theo cách thức xử lý dữ liệu đầu (data input) vào người ta phân chia thành 2 loại:

- **Mã hóa khối (block cipher):** xử lý dữ liệu đầu vào theo khối tại một thời điểm, cho kết quả theo một khối dữ liệu ở đầu ra.
- **Mã hóa luồng (stream cipher):** xử lý tuần tự các phần tử liên tục ở đầu vào và cho kết quả từng phần tử ở đầu ra tại một thời điểm.

Phân loại mã hóa

- **Mã hóa cổ điển (classical cryptographic):** đây là kỹ thuật được hình thành từ xa xưa, ý tưởng bên gởi sử dụng thuật toán mã hóa cổ điển dựa trên hai kỹ thuật cơ bản: thay thế (substitution) và hoán vị (transposition), bên nhận dựa vào thuật toán của bên gởi để giải mã mà không cần dùng khóa.
- Do đó, độ an toàn của kỹ thuật này không cao do chỉ dựa vào sự che giấu thuật toán, hiện nay mã hóa cổ điển ít được sử dụng trong thực tế.

Phân loại mã hóa

- **Mã hóa hiện đại (modern cryptography):** mã hóa đối xứng (symmetric cipher, secret key cryptography – 1 khóa), bất đối xứng (asymmetric cipher, public key cryptography – 2 khóa), hàm băm (hash functions – không có khóa).

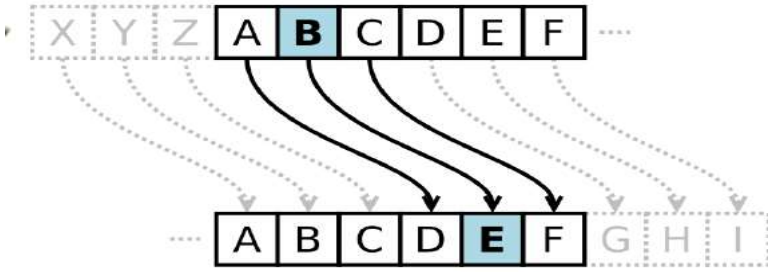
Mã hóa cổ điển (classical cryptographic)

- **Mã hóa cổ điển** dựa trên kỹ thuật thay thế (thay thế kí tự hoặc các kí tự này bằng kí tự hoặc các kí tự khác tương ứng) và hoán vị (thay đổi trật tự, vị trí các ký tự) trong văn bản gốc. Các kỹ thuật này có thể áp dụng đối với một ký tự (monoalphabetic) hoặc nhiều ký tự (polyalphabetic) tùy vào mục đích sử dụng.
- Các loại mã hóa cổ điển:
  - **Mã Caesar (Caesar cipher)**
  - **Mã hóa đơn bảng (Monoalphabetic Substitution Cipher)**
  - **Mã hóa Vigenère Cipher (Vigenère cipher)**
  - **Mã Playfair**
  - **One-Time Pad**
  - Mã hàng rào sắt (rail fence cipher)

Caesar Cipher

- Thế kỷ thứ 3 trước công nguyên, Julius Ceasar đưa ra phương pháp mã hóa này.
- ***Thay thế mỗi ký tự trong bản rõ bằng ký tự đứng sau nó k vị trí trong bảng chữ cái.***
- Giả sử chọn k=3, ta có bảng chuyển đổi:
- ACTIONS→EGXMSRW

Chữ ban đầu:	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
Chữ thay thế:	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



Caesar Cipher

- Ví dụ:

Chữ ban đầu: 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

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

Như vậy bản rõ meet me after the toga party

được mã hóa thành: NJJU NJ ZRUJM UKJ UVSZ DZMUE

Caesar Cipher

- Để tấn công hệ mật Caesar có thể sử dụng một số kỹ thuật sau:
- Vét cạn (brute-force): thử tất cả các khả năng biến đổi có thể xảy ra để tìm được quy tắc thay thế, do hệ mã Caesar chỉ có 26 ký tự (tương ứng 25 quy tắc - khóa) nên việc giải mã không mất nhiều thời gian trong điều kiện hiện nay.
- Tần số xuất hiện kí tự (Character frequencies): dựa vào thống kê xuất hiện của các kí tự trong bản mã, đối chiếu với bảng tần số được khảo sát trước của từng ngôn ngữ.

Caesar Cipher

- Trong 25 trường hợp trên, chỉ có trường hợp k=3 thì bản giải mã tương ứng là có ý nghĩa.
- Do đó đối thủ có thể chắc chắn rằng **"meet me after the toga party"** là bản rõ ban đầu.

KEY	PHHW PH DIWHU WKH WRJD SDUWB
1	oggv og chvgt vjg vqic rctva
2	nffu nf bgufs uif uphb qbsuz
3	**meet me after the toga party**
4	ldds ld zesdq sgd snfz ozqsx
5	kccr kc ydrcp rfc rmey nyprw
6	jbbq jb xcqbo qeb qldx mxoqv
7	iaap ia wbpan pda pkcw lwnpu
8	hzzo hz vaozm ocz ojbv kvmot
9	gyyn gy uznyl nby niau julns
10	fxxm fx tymxk max mhzt itkmr
11	ewwl ew sxlwj lzw lgys hsjlq
12	dvvk dv rwkvi kyv kfxr grikp
13	cuuj cu qvjuh jxu jewq fqhjo
14	btti bt puitg iwt idvp epgin
15	assh as othsf hvs hcuo dofhm
16	zrrg zr nsgre gur gbtn cnegl
17	yqqf yq mrfqd ftq fasm bmdfk
18	xppe xp lqepc esp ezrl alcej
19	wood wo kpdob dro dyqk zkbdi
20	vnnc vn jocna cqn cxpj yjach
21	ummb um inbmz bpm bwoi xizbg
22	tlla tl hmaly aol avnh whyaf
23	skkz sk glzkx znk zumg vgxze
24	rjjy rj fkyjw ymj ytlf ufwyd
25	qiix qi ejxiv xli xske tevxc

Caesar Cipher

Bài tập:

1. Áp dụng mật mã Ceasar mật mã hóa các bản rõ sau với khóa k = 4

   actions speak louder than words

2. Đoán khóa k và giải mã cho bản mật sau:

   ST RFS HFS XJWAJ YBT RFXYJWX

Caesar Cipher

1. Giải mã bản mã sau, giả sử mã hóa Ceasar được sử dụng để mã hóa với k=3: IRXUVFRUHDQGVHYHQBHDUVDJR

2. Khóa

   Plain(a): **abcdefghijklmnopqrstuvwxyz**

   - Cipher(b): **DKVQFIBJWPESCXHTMYAUOLRGZN**

   Mã hóa:

   - Plaintext: **ifwewishtoreplaceletters**
   - Ciphertext: **WIRFRWAJUHYFTSDVFSFUUFYA**

Caesar Cipher

- Eve has intercepted the ciphertext "UVACLYFZLJBYL". Show how she can use a brute-force attack to break the cipher.
- Solution: Eve tries keys from 1 to 7. With a key of 7, the plaintext is "not very secure", which makes sense.

Ciphertext: UVACLYFZLJBYL

K = 1	→	**Plaintext:** tuzbkxeykiaxk
K = 2	→	**Plaintext:** styajwdxjhzwj
K = 3	→	**Plaintext:** rsxzivcwigyvi
K = 4	→	**Plaintext:** qrwyhubvhfxuh
K = 5	→	**Plaintext:** pqvxgtaugewtg
K = 6	→	**Plaintext:** opuwfsztfdvsf
K = 7	→	**Plaintext:** notverysecure

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

Caesar Cipher

- Gán cho mỗi chữ cái một con số nguyên từ 0 đến 25:
- Với mỗi ký tự trong P thay bằng chữ mã hóa *C*, trong đó: TERM→GREZ với k=0

$$C = (P + k) \ mod \ 26$$ (mod: phép chia lấy số dư)

- Và quá trình giải mã đơn giản là:

$$P = (C - k) \ mod \ 26$$

- *k* được gọi là khóa.
- Hiện nay, mã Ceasar không được xem là an toàn.

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

Caesar Cipher

Ví dụ: Bảng chữ cái tiếng Anh có 26 ký tự:
ABCDEFGHIJKLMNOPQRSTUVWXYZ

Mã hóa kí tự x với khoảng cách dịch chuyển 1 đoạn n=13 theo qui tắc sau:y=(x+13) mod 26

Giải mã: x=(y-13) mod 26

Mã hóa chuỗi ký tự sau theo qui tắc trên:

GUIDELINES FOR TERM PAPERS

Kết quả:

THVQRYVARF SBE GREZ CNCREF

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

Caesar Cipher

- Use the additive cipher with key = (?+15) mod 26 to encrypt the message “hello”.
- Solution
- We apply the encryption algorithm to the plaintext, character by character:

Plaintext: h → 07	Encryption: (07 + 15) mod 26	Ciphertext: 22 → W
Plaintext: e → 04	Encryption: (04 + 15) mod 26	Ciphertext: 19 → T
Plaintext: l → 11	Encryption: (11 + 15) mod 26	Ciphertext: 00 → A
Plaintext: l → 11	Encryption: (11 + 15) mod 26	Ciphertext: 00 → A
Plaintext: o → 14	Encryption: (14 + 15) mod 26	Ciphertext: 03 → D

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

Caesar Cipher

- Use the additive cipher with key = (?-15)mod 26 to decrypt the message "WTAAD".
- Solution
- We apply the decryption algorithm to the plaintext character by character:

Ciphertext: W → 22	Decryption: (22 – 15) mod 26	Plaintext: 07 → h
Ciphertext: T → 19	Decryption: (19 – 15) mod 26	Plaintext: 04 → e
Ciphertext: A → 00	Decryption: (00 – 15) mod 26	Plaintext: 11 → l
Ciphertext: A → 00	Decryption: (00 – 15) mod 26	Plaintext: 11 → l
Ciphertext: D → 03	Decryption: (03 – 15) mod 26	Plaintext: 14 → o

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

Caesar Cipher

- Giả sử có bản tin gốc (*bản rõ*): meet me after the toga party
- Mã hóa bản gốc trên?
- Giả sử bạn có được bản mã

PHHW PH DIWHU WKH WRJD SDUWB

và biết được phương pháp mã hóa và giải mã là phép cộng trừ modulo 26→Bạn có suy ra được bản gốc không?

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

Caesar Cipher

- **Với bản chữ cái Tiếng Việt (29 ký tự) với khóa là 3:**
- Gán cho mỗi chữ cái một con số nguyên từ 0 đến 28:
- Phương pháp Ceasar biểu diễn tiếng Việt như sau: với mỗi chữ cái p thay bằng chữ mã hóa C, trong đó:

  C = (p + k) mod 29
- Và quá trình giải mã đơn giản là:

  p = (C – k) mod 29

A	Ă	Â	B	C	D	Đ	E	Ê	G	H	I	K	L	M	N	O	Ô	Ơ	P	Q	R	S	T	U	Ư	V	X	Y
0	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

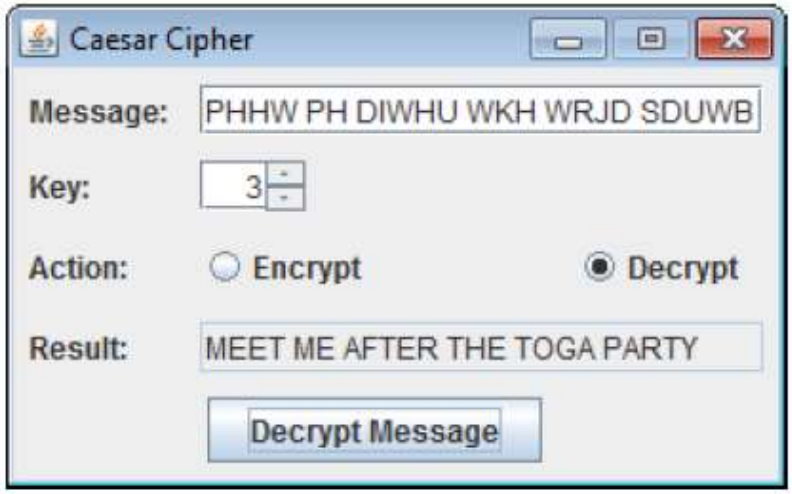
Caesar Cipher

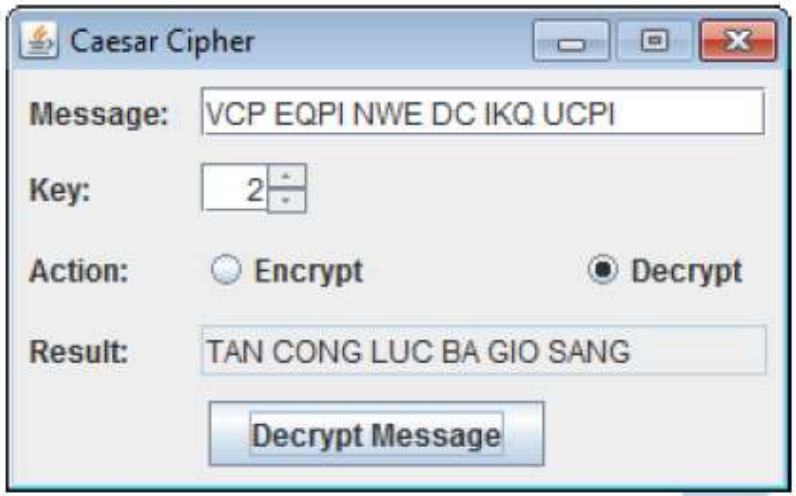
Code Java

```
private String encryptMessage(String msg, int k) {
    String result = "";
    for (int i = 0; i < msg.length(); i++)
    result += encryptChar(msg.charAt(i), k);
    return result;
}
private char encryptChar(char c, int k) {
    if (Character.isLetter(c))
    return (char) ('A' + (c - 'A' + k) % 26); //'A'=65
    else
    return c;
}
```

- **Nếu giải mã: encryptMessage(msg,26-k);**

Caesar Cipher





Mã hóa đơn bảng
(Monoalphabetic Substitution Cipher)

- Phương pháp đơn bảng tổng quát hóa phương pháp Ceasar bằng cách dòng mã hóa không phải là một dịch chuyển *k* vị trí của các chữ cái A, B, C, … nữa mà là một *hoán vị* của 26 chữ cái này. Lúc này mỗi hoán vị được xem như là một khóa.
- Số lượng hoán vị của 26 chữ cái là 26! $=4x10^{26}$(tương đương với số khóa).
- Vì 26! là một con số khá lớn → tấn công phá mã vét cạn khóa là bất khả thi (6400 thiên niên kỷ với tốc độ thử khóa là 109 khóa/giây).

→ phương pháp này được xem là một phương pháp mã hóa an toàn trong suốt 1000 năm sau công nguyên.

Monoalphabetic Ciphers

- Ví dụ:
- *Chữ ban đầu:* 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
- *Khóa :* Z P B Y J R S K F L X Q N W V D H M G U T O I A E C
- Như vậy bản rõ meet me after the toga party
- được mã hóa thành: NJJU NJ ZRUJM UKJ UVSZ DZMUE

Monoalphabetic Ciphers

- Tuy nhiên vào thế kỷ thứ 9, một nhà hiền triết người Ả Rập tên là Al-Kindi đã phát hiện ra một phương pháp phá mã khả thi khác. Phương pháp phá mã này dựa trên nhận xét sau:
  - Trong ngôn ngữ tiếng Anh, tần suất sử dụng của các chữ cái không đều nhau, chữ E được sử dụng nhiều nhất, còn các chữ ít được sử dụng thường là Z, Q, J. Tương tự như vậy, đối với cụm 2 chữ cái (digram), cụm chữ TH được sử dụng nhiều nhất.
  - Nếu chữ E được thay bằng chữ K thì tần suất xuất hiện của chữ K trong bản mã là 13.05%. Đây chính là cơ sở để thực hiện phá mã.

Monoalphabetic Ciphers

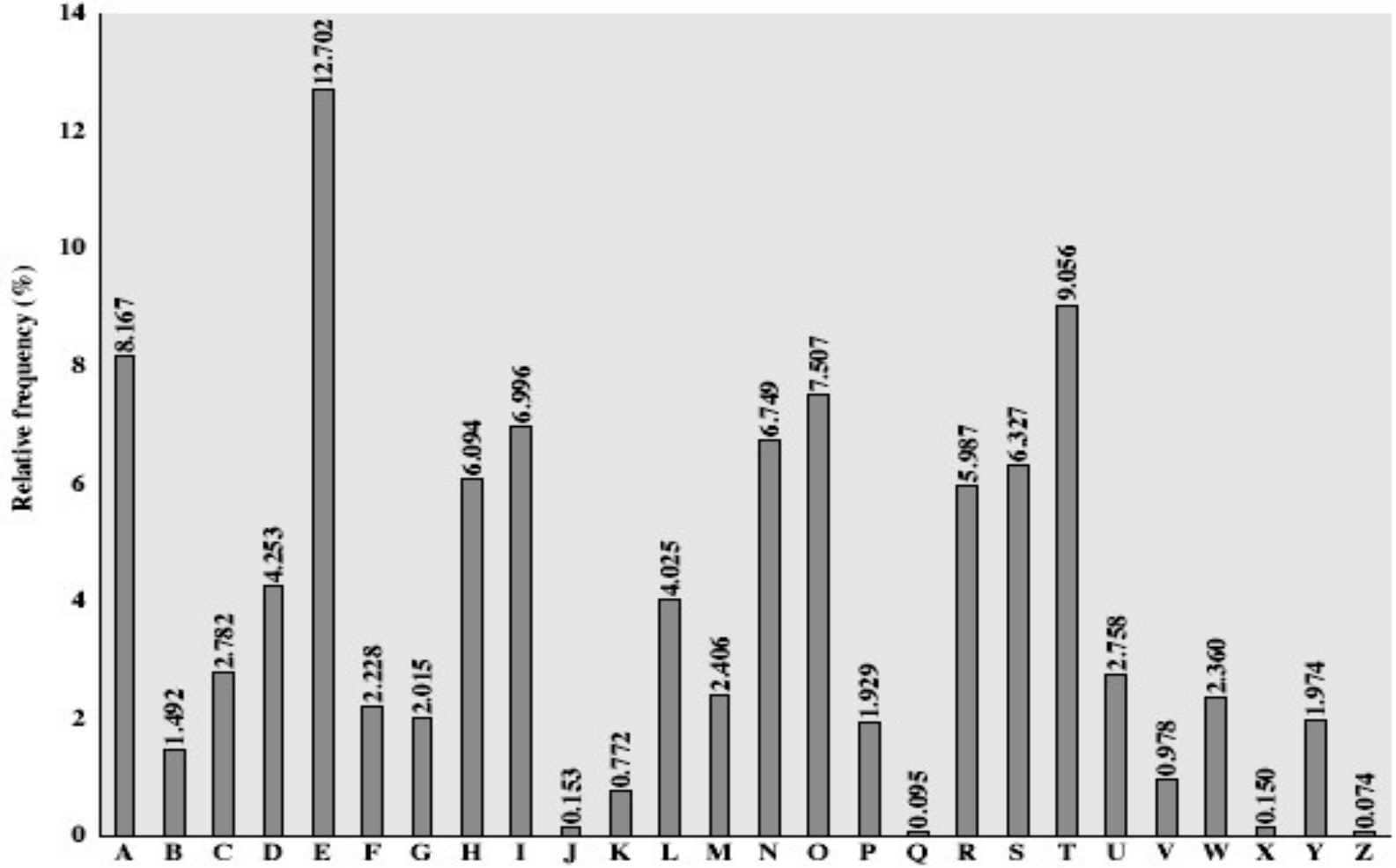
Tần suất của chữ tiếng anh

Chữ cái (%)		Cụm 2 chữ (%)		Cụm 3 chữ (%)		Từ (%)	
E	13.05	TH	3.16	THE	4.72	THE	6.42
T	9.02	IN	1.54	ING	1.42	OF	4.02
O	8.21	ER	1.33	AND	1.13	AND	3.15
A	7.81	RE	1.30	ION	1.00	TO	2.36
N	7.28	AN	1.08	ENT	0.98	A	2.09
I	6.77	HE	1.08	FOR	0.76	IN	1.77
R	6.64	AR	1.02	TIO	0.75	THAT	1.25
S	6.46	EN	1.02	ERE	0.69	IS	1.03
H	5.85	TI	1.02	HER	0.68	I	0.94
D	4.11	TE	0.98	ATE	0.66	IT	0.93
L	3.60	AT	0.88	VER	0.63	FOR	0.77
C	2.93	ON	0.84	TER	0.62	AS	0.76
F	2.88	HA	0.84	THA	0.62	WITH	0.76
U	2.77	OU	0.72	ATI	0.59	WAS	0.72
M	2.62	IT	0.71	HAT	0.55	HIS	0.71
P	2.15	ES	0.69	ERS	0.54	HE	0.71
Y	1.51	ST	0.68	HIS	0.52	BE	0.63
W	1.49	OR	0.68	RES	0.50	NOT	0.61
G	1.39	NT	0.67	ILL	0.47	BY	0.57
B	1.28	HI	0.66	ARE	0.46	BUT	0.56
V	1.00	EA	0.64	CON	0.45	HAVE	0.55
K	0.42	VE	0.64	NCE	0.45	YOU	0.55
X	0.30	CO	0.59	ALL	0.44	WHICH	0.53
J	0.23	DE	0.55	EVE	0.44	ARE	0.50
Q	0.14	RA	0.55	ITH	0.44	ON	0.47
Z	0.09	RO	0.55	TED	0.44	OR	0.45

Bảng 2-2. Bảng liệt kê tần suất chữ cái tiếng Anh

Monoalphabetic Ciphers

Tần suất của chữ tiếng anh



Ví dụ khám mã

Cho bản mã:

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPPDPTGUDTMOHMQ

- Đếm tần suất ký tự

A 2	F 3	K 0	P 17	U 9
B 2	G 3	L 0	Q 3	V 5
C 0	H 6	M 7	R 0	W 4
D 6	I 1	N 0	S 10	X 5
E 6	J 0	O 9	T 4	Y 2
				Z 13

DT 2	HZ 2	PE 2	TS 2	XU 2
DZ 2	MO 2	PO 3	UD 2	ZO 2
EP 3	OH 2	PP 2	UZ 3	ZS 2
FP 3	OP 3	SX 3	VU 2	ZU 2
HM 2	PD 3	SZ 2	WS 2	ZW 3

Số lần xuất hiện của các digram
(xuất hiện từ 2 lần trở lên)

Ví dụ khám mã

- Do đó ta có thể đoán P là mã hóa của e, Z là mã hóa của t. Vì TH có tần suất cao nhất trong các digram nên trong 4 digram ZO, ZS, ZU, ZW có thể đoán ZW là th.
- Chú ý rằng trong dòng thứ nhất có cụm ZWSZ, nếu giả thiết rằng 4 chữ trên thuộc một từ thì từ đó có dạng th_t, từ đó có thể kết luận rằng S là mã hóa của a (vì từ THAT có tần suất xuất hiện cao).
- Như vậy đến bước này, ta đã phá mã được như sau:

Ví dụ khám mã

```
UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
 t a          e   e te   a that e e a         a
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
   e t    ta t ha e ee   a e   th     t   a
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
 e   e e tat   e    the    t
```

Suy luận tiếp tục ta có được bản rõ

```
it was disclosed yesterday that several informal but
direct contacts have been made with political
representatives of the viet cong in moscow
```

Bài tập phá mã sử dụng bảng tần suất

Bài 1

Mã hoá bản rõ: illustrate sử dụng mã thay thế với khoá là 1 hoán vị bất kì sau:

a	b	c	d	e	f	g	h	I	j	k	l	m	n	o	p	q	r	s	t
X	N	Y	A	H	P	O	G	Z	Q	W	B	T	S	F	L	R	C	V	M

u	v	w	x	y	z
U	E	K	J	D	I

CiperText: ZBBUVMCXMH

Bài tập phá mã sử dụng bảng tần suất

- Eve has intercepted the following ciphertext. Using a statistical attack, find the plaintext.

XLILSYWIMWRSAJSVWEPIJSVJSYVQMPPMSRHSPPEVWMXMWASVX-LQSVILY-VVCFIJSVIXLIWIPPIVVIGIMZIWQSVISJJIVW

- When Eve tabulates the frequency of letters in this ciphertext, she gets: I =14, V =13, S =12, and so on. The most common character is I with 14 occurrences. This means key = 4.

the house is now for sale for four million dollars it is worth more hurry before the seller receives more offers

Bài tập phá mã sử dụng bảng tần suất

“YIFQFMZRWQFYVECFMDZPCVMRZWNMDZV
EJBTXCDDUMJ

NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZRE
XCHZUNMXZ

NZUCDRJXỷYMTMEYIFZWDYVZVYFZUMRZCR
WNZDZJT

XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYU
CFWDINZDIR ”

Mã hóa Vigenère Cipher (Vigenère cipher)

- Thế kỷ thứ 15, một nhà ngoại giao người Pháp tên là **Vigenere** đã tìm ra phương án mã hóa thay thế đa bảng.
- Mã hóa Vigenere được hình thành trên mã hóa Caesar có sử dụng khóa (chuỗi các chữ cái) trên văn bản gốc (gồm các chữ cái).
- Mã hóa Vigenere là sự kết hợp của nhiều phép mã hóa Caesar với các bước dịch chuyển khác nhau.
- Để mã hóa, sử dụng bảng mã Vigenere *(Hình x)* với cột dọc là chuỗi khóa (khóa được lặp đi lặp lại để chiều dài tương ứng với văn bản gốc), cột ngang – văn bản gốc, giao giữa kí tự tương ứng cột chứa khóa và văn bản gốc chính là kí tự mã của thuật toán.

Mã hóa Vigenère Cipher (Vigenère cipher)

KEYWORD LETTER	PLAINTEXT LETTER																									
	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
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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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

Mã hóa Vigenère Cipher (Vigenère cipher)

- Ví dụ: bản tin: "*We are discovered, save yourself*" và khóa là từ *DECEPTIVE*, ta mã hóa như sau:
- Ứng với với chữ w trong bản rõ là chữ D trong khóa, nên dòng mã hóa thứ 4 ứng với khóa D trong bảng Vigenere được chọn. Do đó chữ w được mã hóa thành chữ Z. Tương tự như vậy cho các chữ còn lại.
- Trong ví dụ trên, các chữ e trong bản rõ được mã hóa tương ứng thành I, T, G, T, H, M trong bản mã. Do đó phương pháp phá mã dựa trên thống kê tần suất chữ cái là không thực hiện được. Trong 3 thế kỷ sau đó mã hóa Vigenere được xem là mã hóa không thể bị phá.

Mã hóa Vigenère Cipher (Vigenère cipher)

- Độ an toàn của mã hóa Vigenere phụ thuộc vào độ dài của khóa. Khi đó, kẻ tấn công sẽ các định chiều dài của khóa trước khi thực hiện các bước tiếp theo, như việc phân tích tần số cho các bản mã Caesar khác nhau.

.

Mã Playfair

- Được biết như là **mã thay thế đa ký tự**
- Được đề xuất bởi Charles Wheatstone, được mang tên của người bạn Baron Playfair
- ***Mã hóa Playfair xem hai ký tự đứng sát nhau là một đơn vị mã hóa, hai ký tự này được thay thế cùng lúc bằng hai ký tự khác.***

Mã Playfair

- Mật mã đa ký tự (mỗi lần mã 2 ký tự liên tiếp nhau)
- Giải thuật dựa trên một ma trận các chữ cái 5×5 được xây dựng từ một khóa (chuỗi các ký tự)

1. Xây dựng ma trận khóa
   - Lần lượt thêm từng ký tự của khóa vào ma trận
   - Nếu ma trận chưa đầy, thêm các ký tự còn lại trong
   - bảng chữ cái vào ma trận theo thứ tự A - Z
   - I và J xem như là 1 ký tự
   - Các ký tự trong ma trận không được trùng nhau
2. Mật mã hóa
3. Giải mật mã.

Playfair Cipher

- Dựa trên ma trận 5 ×5 của các ký tự được xây dựng bằng từ khóa (keyword)

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

- Trong ma trận trên, khóa là từ MONARCHY được điền vào các dòng đầu của bảng, các chữ cái còn lại của bảng chữ cái được điền tiếp theo một cách tuần tự. Riêng hai chữ I, J được điền vào cùng một ô

Mã Playfair

- Bản rõ được mã hóa một lần 2 ký tự theo quy tắc sau:

1. Cặp hai ký tự giống nhau xuất hiện trong bản rõ sẽ được tách ra bởi 1 ký tự lọc, chẳng hạn như **x**. Ví dụ trước khi mã hóa **"balloon"** sẽ được biến đổi thành **"balxloxon"**.

2. Hai ký tự trong cặp đều rơi vào cùng một hàng, thì mã mỗi ký tự bằng ký tự bên phải nó trong cùng hàng của ma trận khóa, nếu nó là phần tử cuối của hàng thì vòng sang ký tự đầu cùng của hàng, chẳng hạn "**ar**" mã hóa thành **"rm"**

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

Mã Playfair

3. Hai ký tự mà rơi vào cùng một cột thì nó được mã bằng ký tự ngay dưới, nếu nó ở cuối cột thì vòng sang ký tự ở đầu cột, chẳng hạn "**mu**" được mã hóa thành "**CM**", ov được mã hóa thành HO
4. Trong trường hợp khác, mỗi ký tự của bản rõ trong một cặp được thay bởi ký tự nằm cùng hàng với nó và cột là cùng cột với ký tự cùng cặp (đường chéo của một hình chữ nhật). Chẳng hạn, **"hs"**mã thành **"bp"**, và**"ea"**mã thành **"im"**hoặc **"jm"**

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

Mã Playfair

Ví dụ 2: Hãy tìm hiểu quá trình mã hóa và giải mã bằng phương pháp mã Playfair

Bản rõ: DHCNTPHCM

Khóa: smythework

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

Playfair Cipher

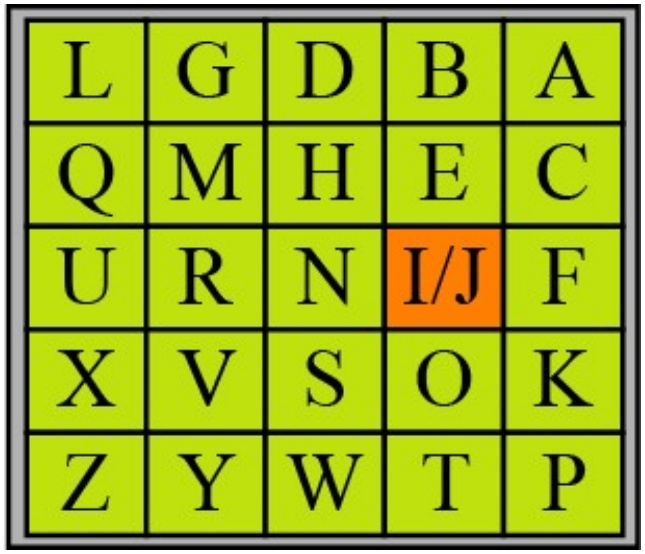
- An toàn được cải tiến hơn với mã hóa đơn bản (simple monoalphabetic ciphers)
- Chỉ xét trên 26 ký tự thì mã khóa Playfair có 26x26=676 cặp ký tự.

  → các cặp ký tự này ít bị chênh lệch về tần suất hơn so với sự chênh lệnh tần suất của từng ký tự. Ngoài ra số lượng các cặp ký tự nhiều hơn cũng làm cho việc phá mã tầnsuất khó khăn hơn. Đây chính là lý do mà người ta tin rằng mã hóa Playfair không thể bị phá và được quân đội Anh sử dụng trong chiến tranh thế giới lần thứ nhất.
- Tuy nhiên, nó có thể bị bẻ khoá nếu cho trước vài trăm chữ, vì bản mã vẫn còn chứa nhiều cấu trúc của bản rõ.

Bài tập Mã Playfair

- An example of a secret key in the Playfair cipher

Secret Key =



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

Let us encrypt the plaintext “hello” using the key in the Figure

he → EC lx → QZ lo → BX

Plaintext: hello Ciphertext: ECQZBX

Bài tập Mã Playfair

1. Mật mã hóa bản rõ sau:

hide the gold in the tree stump

2. Hãy tìm hiểu quá trình mã hóa và giải mã bằng phương pháp mã Playfair

Bản rõ P= “Dai hoc su pham”

Khóa K=tinhoc

One-Time Pad (OTP)

- One-Time Pad – bộ đệm một lần.

  Được đề xuất bởi Joseph Mauborgne
- **Một khóa ngẫu nhiên có chiều dài bằng chiều dài của bản rõ, mỗi khóa dùng một lần**
- Khó bẻ khóa vì không có quan hệ nào giữa bản rõ và bản mã.
- Ví dụ mã hóa bản tin "*wearediscoveredsaveyourself*"
  - Bản tin *P*: wearediscoveredsaveyourself
  - Khóa *K*1: F H W Y K L V M K V K X C V K D J S F S A P X Z C V P
  - Bản mã *C*: LWPOODEMJFBTZNVJNJQOJORGGU
  - Dùng K1 để giải mã thì ta được:

    "*wearediscoveredsaveyourself*"

Joseph Mauborgne

One-Time Pad (OTP)

Xét hai trường hợp giải mã bản mã trên với 2 khóa khác nhau

- *Trường hợp 1*:
  - Bản mã *C*: LWPOODEMJFBTZNVJNJQOJORGGU
  - Khóa *K*2: IESRLKBWJFCIFZUCJLZXAXAAPSY
  - Bản giải mã: theydecidedtoattacktomorrow

    (*they decided to attack tomorrow*)
- *Trường hợp 2*:
  - Bản mã *C*: LWPOODEMJFBTZNVJNJQOJORGGU
  - Khóa *K*3: FHAHDDRAIQFIASJGJWQSVVBJAZB
  - Bản giải mã: wewillmeetatthepartytonight

    (*we will meet at the party tonight*)

One-Time Pad (OTP)

Trong cả hai trường hợp trên thì bản giải mã đều có ý nghĩa.

- Nếu người phá mã thực hiện phá mã vét cạn thì sẽ tìm được nhiều khóa ứng với nhiều bản tin có ý nghĩa => không biết được bản tin nào là bản rõ.
- Điều này chứng minh phương pháp One-Time Pad là phương pháp mã hóa an toàn tuyệt đối.
- Để phương pháp One-Time Pad là an toàn tuyệt đối thì mỗi khóa chỉ được sử dụng một lần.
- Nếu một khóa được sử dụng nhiều lần thì cũng không khác gì việc lặp lại một từ trong khóa (ví dụ khóa có từ DECEPTIVE được lặp lại).

One-Time Pad (OTP)

Thực tế:

- Phương pháp One-Time Pad không có ý nghĩa sử dụng thực tế. Vì chiều dài khóa bằng chiều dài bản tin, mỗi khóa chỉ sử dụng một lần, nên thay vì truyền khóa trên kênh an toàn thì có thể truyền trực tiếp bản rõ mà không cần quan tâm đến vấn đề mã hóa.

Mã hàng rào sắt (rail fence cipher)

- Đây là một mã dùng phép hoán vị hoặc chuyển vị, vì vậy gọi là mã hoán vị hoặc mã chuyển vị (***classical transposition or permutation ciphers***)
- Thực hiện xáo trộn thứ tự các ký tự trong bản rõ. Do thứ tự của các ký tự bị mất đi nên người đọc không thể hiểu được ý nghĩa của bản tin dù các chữ đó không thay đổi.
- Đơn giản nhất của mã hóa kiểu này là mã **rail fence cipher**

Rail Fence Cipher

- Ghi các ký tự trong bản rõ theo từng hàng rào, sau đó kết xuất bản mã dựa trên cột. Sau đó đọc bảng mã theo từng hàng
- Ví dụ: bản rõ "meet me after the toga party" với hành rào sắt độ sâu là 2 (Tách bản rõ thành 2 hàng)
- Ví dụ: bản rõ "meet me at the toga party" được viết thành

```
m e m a t r h t g p r y
 e t e f e t e o a a t
```

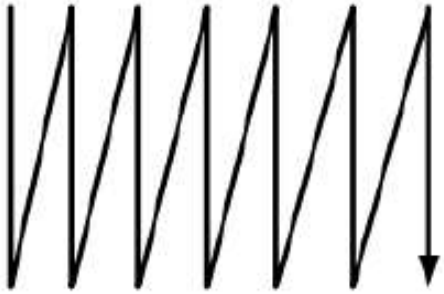
  - Cho bản mã

```
MEMATRHTGPRYETEFETEOAAT
```

Rail Fence cipher

- Ví dụ bản rõ “attackpostponeduntilthisnoon” được viết lại thành bảng 4 x 7 như sau:

```
a t t a c k p
o s t p o n e
d u n t i l t
h i s n o o n
```



- Khi kết xu

“AODHTSUTTNSAPTNCOIOKNLOPETN”

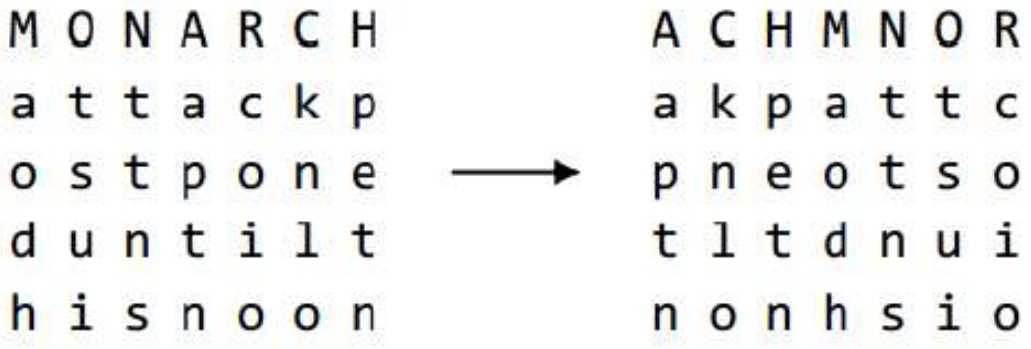
Rail Fence cipher

- Để an toàn hơn nữa, có thể áp dụng phương pháp hoán vị 2 lần (double transposition), tức sau khi hoán vị lần 1, ta lại lấy kết quả đó hoán vị thêm một lần nữa.
- Để phá mã phương pháp hoán vị 2 lần không phải là chuyện dễ dàng vì rất khó đoán ra được quy luật hoán vị.
- Ngoài ra không thể áp dụng được phương pháp phân tích tần suất chữ cái giống như phương pháp thay thế vì tần suất chữ cái của bản rõ và bản mã là giống nhau.

Rail Fence cipher

- Một cơ chế phức tạp hơn là chúng ta có thể hoán vị các cột trước khi kết xuất bản mã.
- Ví dụ chọn một khóa là **MONARCH**, ta có thể hoán vị các cột:

Bản rõ "attackpostponeduntilthisnoon"

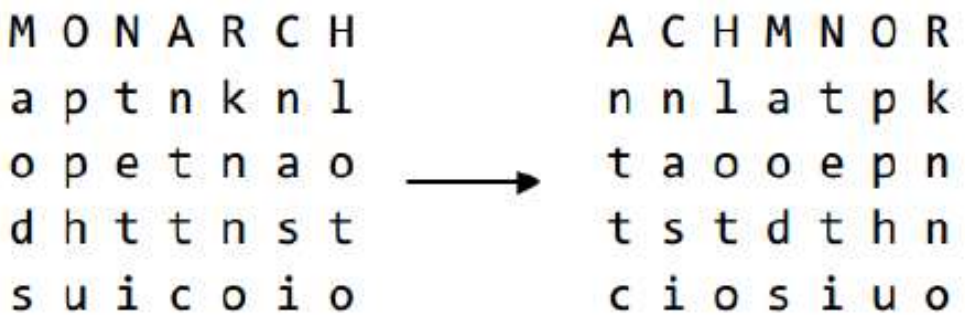


và có được bản mã:

"APTNKNLOPETNAODHTTNSTSUICOIO". Việc giải mã được tiến hành theo thứ tự ngược lại.

Rail Fence cipher

- Để an toàn hơn nữa => **hoán vị 2 lần** (double transposition):
- Sau khi hoán vị lần 1, ta lấy kết quả đó hoán vị lần nữa:



- Và cuối cùng **bản mã** là: "NTTCNASILOTOAODSTETIPPHUKNNO"
- **Phá mã** phương pháp hoán vị 2 lần không phải là chuyện dễ dàng vì rất khó đoán ra được quy luật hoán vị.
- Không thể áp dụng được phương pháp phân tích tần suất chữ cái giống như phương pháp thay thế vì tần suất chữ cái của bản rõ và bản mã là giống nhau.

Các kỹ thuật hoán vị - nâng cao

- Bản rõ được viết trên một hình chữ nhật và đọc theo cột. Thứ tự các cột trở thành khóa của giải thuật.
- Ví dụ: bản rõ "meet me at the toga party"
  - Key 4

1	2	5	3	6	
m	e	e	t	m	e
a	t	t	h	e	t
o	g	a	p	a	r
t	y	x	y	z	w

- bản mật **etgyetaxmeazmaotthpyetrw**
- Để tăng độ mật, có thể áp dụng hoán vị nhiều lần

Các kỹ thuật hoán vị - nâng cao

Bài tập: Ví dụ: mật mã

- Bản rõ:

  hide the gold in the tree stump

- Bản mật:

  BM OD ZB XD NA BE KU DM UI XM MO UV IF

Các kỹ thuật hoán vị - nâng cao

Bài tập: Ví dụ: mật mã Bản rõ:

hide the gold in the tree stump

KEY= midnight

Mã hóa hoán vị 2 lần

- Bản mật:

M	I	D	N	I	G	H	T
H	I	D	E	T	H	E	G
O	L	D	I	N	T	H	E
T	R	E	E	S	T	U	M

P

BẢNG MÃ: DDEHTTEHUILRTNSHOTPEIEGEM

Các kỹ thuật hoán vị - nâng cao

Bài tập: Ví dụ: mật mã

- Bản rõ:

hide the gold in the tree stump

KEY=

midnight→DDEHTTEHUILRTNSHOTPEIEGEM

M	I	D	N	I	G	H	T
D	D	E	H	T	T	E	H
U	I	L	R	T	N	S	H
O	T	P	E	I	E	G	E
M							

BẢNG MÃ: ELPTNEESGDITTTIDUOMHREHHE

Rail Fence Cipher

- Bài tập: bản rõ “attack at midnight” với hành rào sắt độ sâu là 4
- Và hoán vị với từ khóa là NGÀY QUỐC TẾ PHỤ NỮ

Tóm tắt

- Các phương pháp mã hóa cổ điển thường dựa trên hai phương thức.
  1. Dùng phương pháp ***thay thế một chữ cái trong bản rõ thành một chữ cái khác trong bản mã*** (substitution). Các mã hóa dùng phương thức này là mã hóa Ceasar, mã hóa thay thế đơn bảng, đa bảng, one-time pad.
  2. Dùng phương pháp *hoán vị để thay đổi thứ tự ban đầu của các chữ cái trong bản rõ* (Rail Fence)

Câu hỏi

1. Tại sao khi gửi bản mã trên kênh truyền thì không sợ bị lộ thông tin?
2. Khóa là gì? Tại sao cần giữ bí mật khóa chỉ có người gửi và người nhận biết?
3. Tại sao lại gửi khóa qua kênh an toàn mà không gửi trực tiếp bản rõ trên kênh an toàn?
4. Phá mã khác giải mã ở điểm nào?
5. Phá mã theo hình thức vét cạn khóa thực hiện như thế nào? Cần làm gì để chống lại hình thức phá mã theo vét cạn khóa?
6. Các phương pháp Ceasar, mã hóa đơn bảng, đa bảng, one-time pad dùng nguyên tắc gì để mã hóa?
7. Phương pháp hoán vị dùng nguyên tắc gì để mã hóa?
8. Tại sao phương pháp mã hóa đơn bảng có thể bị tấn công phá mã dùng thống kê tần suất?

Bài tập

1. Giải mã bản mã sau, giả sử mã hóa Ceasar được sử dụng để mã hóa với k=3:

   IRXUVFRUHDQGVHYHQBHDUVDJR

2. Mã hóa bản rõ sau: “enemy coming“, dùng phương pháp mã hóa thay thế đơn bảng với khóa hoán vị K là:

   IAUTMOCSNREBDLHVWYFPZJXKGQ

3. Mã hóa thông điệp sau bằng phương pháp hoán vị:

   we are all together

   biết khóa 24153

4. Phá mã bản mã sau, giả sử mã hóa Ceasar được sử dụng:

   CSYEVIXIVQMREXIH

Bài tập

5. Phá mã bản mã sau (tiếng Anh), biết phương pháp mã hóa sử dụng là phương pháp thay thế đơn bảng:

GBSXUCGSZQGKGSQPKQKGLSKASPCGBGBKGUKGCEUKUZKGGBSQEICA
CGKGCEUERWKLKUPKQQGCIICUAEUVSHQKGCEUPCGBCGQOEVSHUNSU
GKUZCGQSNLSHEHIEEDCUOGEPKHZGBSNKCUGSUKUASERLSKASCUGB
SLKACRCACUZSSZEUSBEXHKRGSHWKLKUSQSKCHQTXKZHEUQBKZAEN
NSUASZFENFCUOCUEKBXGBSWKLKUSQSKNFKQQKZEHGEGBSXUCGSZQ
GKGSQKUZBCQAEIISKOXSZSICVSHSZGEGBSQSAHSGKHMERQGKGSKR
EHNKIHSLIMGEKHSASUGKNSHCAKUNSQQKOSPBCISGBCQHSLIMQGKG
SZGBKGCGQSSNSZXQSISQQGEAEUGCUXSGBSSJCQGCUOZCLIENKGCA
USOEGCKGCEUQCGAEUGKCUSZUEGBHSKGEHBCUGERPKHEHKHNSZKG
GKAD

Bài tập

6. Xét bản mã được mã hóa bằng phương pháp One-Time Pad như sau: KITLKE. Nếu bản rõ là "thrill" thì khóa là gì? Nếu bản rõ là "tiller" thì khóa là gì?

8. Một trường hợp tổng quát của mã hóa Ceasar là mã Affine, trong đó ký tự p được mã hóa thành ký tự C theo công thức:

$$C = E(p, [a, b]) = (ap + b) \ mod \ 26$$

- Một yêu cầu của thuật toán mã hóa là tính đơn ánh, tức nếu $p \neq q$ thì $E(p) \neq E(q)$. Mã Affine không phải là đơn ánh với mọi a. Ví dụ, với a=2, b=3 thì $E(0) = E(13) = 3$.

  a) Có điều kiện gì đặt ra cho b hay không? Tại sao?

  b) Xác định những giá trị của a làm cho mã Affine không đơn ánh.

Mã hóa hiện đại (modern cryptography)

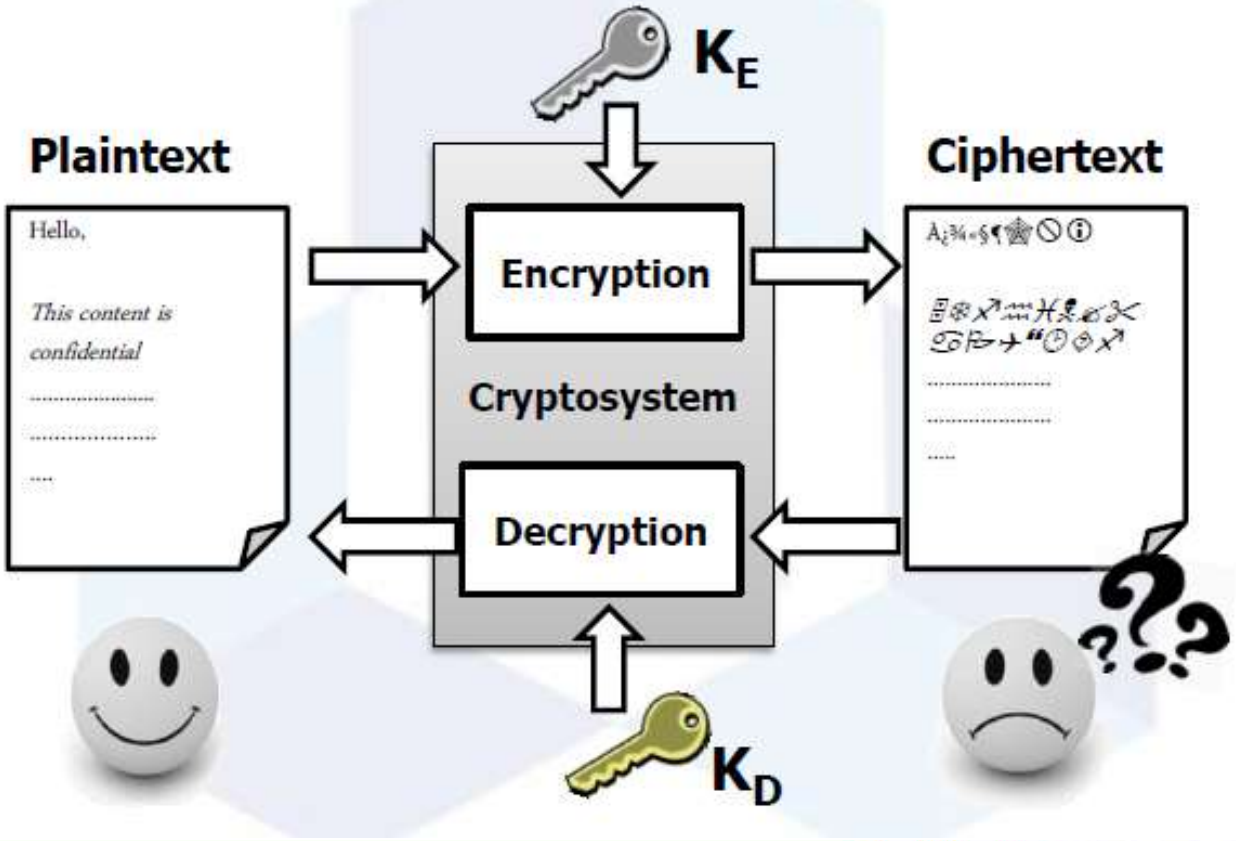
- **Mã hóa hiện đại (modern cryptography):** mã hóa đối xứng (symmetric cipher, secret key cryptography – 1 khóa), bất đối xứng (asymmetric cipher, public key cryptography – 2 khóa), hàm băm (hash functions – không có khóa).

Mã hóa hiện đại (modern cryptography)

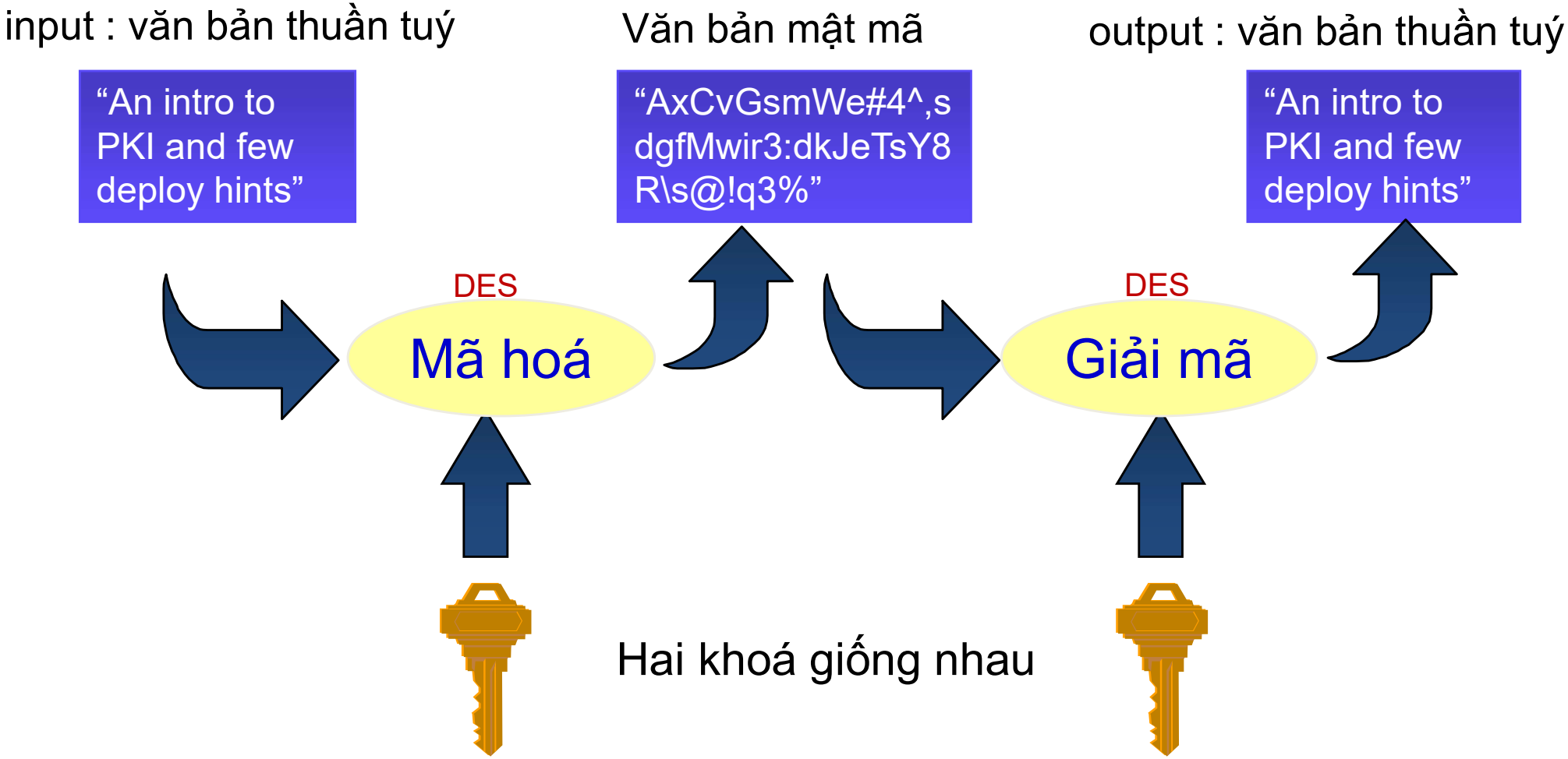
- ***Hệ thống mã hóa đối xứng (Symmetric cryptosystem)*** là hệ thống mã hóa sử dụng một khóa bí mật chia sẻ (shared-secret-key) cho cả hai quá trình mã hóa và giải mã.
- ***Hệ thống mã hóa bất đối xứng (Asymmetric cryptosystem)*** là hệ thống mã hóa sử dụng một khóa công khai (public key) và một khóa bí mật (private key) cho quá trình mã hóa và giải mã.
- Hệ thống mã hóa bất đối xứng còn được gọi là hệ thống mã hóa khóa công khai (public-key cryptosystem)

Mã hóa hiện đại (modern cryptography)

- Mã hóa đối xứng: $K_E = K_D$
- Mã hóa bất đối xứng: $K_E \neq K_D$



Mã hóa đối xứng (symmetric cipher)

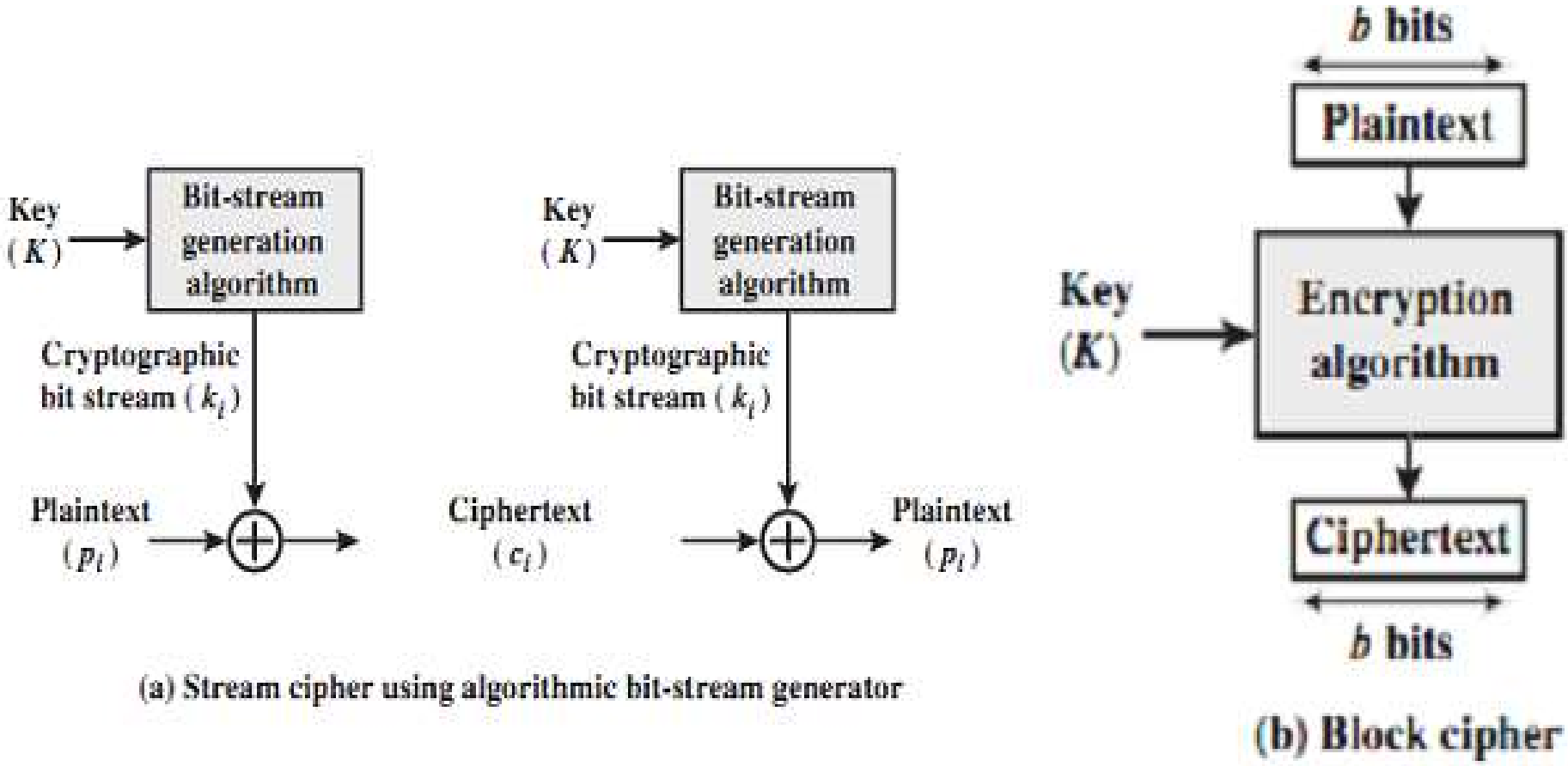


Phân loại

- **Mã dòng (stream cipher)** là một kỹ thuật mã hóa mà mã hóa một dòng dữ liệu số một bit hoặc một byte tại một thời điểm.

- **Mã khối (block cipher)** là một cơ chế mã hóa/giải mã mà trong đó một khối của bản rõ được xử lý như một tổng thể và dùng để tạo ra khối bản mã có độ dài bằng nhau. Thông thường kích cỡ khối là 64 hoặc 128 bit được sử dụng.

Stream Ciphers và Block Ciphers



(a) Stream cipher using algorithmic bit-stream generator

(b) Block cipher

Stream Ciphers

Mã dòng có các đặc tính sau:

- Kích thước một đơn vị mã hóa: gồm *k* bít. Bản rõ được chia thành các đơn vị mã hóa:

- Một bộ sinh dãy số ngẫu nhiên: dùng một khóa *K* ban đầu để sinh ra các số $P \rightarrow p_0 p_1 p_2 \dots p_{n-1} \quad (p_i : k\ bít)$ ch thước đơn vị mã hóa:

- Mỗi số ngẫu nhiên được XOR với đơn vị mã hóa của bản rõ để có đu $StreamCipher(K) \rightarrow S = s_0 s_1 s_2 \dots s_{n-1} \quad (s_i : k\ bít)$

$$c_0 = p_0 \oplus s_0,\ c_1 = p_1 \oplus s_1 \dots \ ; C = c_0 c_1 c_2 \dots c_{n-1}$$

Stream Ciphers

- Quá trình giải mã được thực hiện ngược lại, bản mã *C* được XOR với dãy số ngẫu nhiên *S* để cho ra lại bản rõ ban đầu:
- Trong ví dụ p=111100000011 đơn vị mã hóa có chiều dài k = 4 bít, n = 3:

$$p_0 = c_0 \oplus s_0,\ p_1 = c_1 \oplus s_1 \ldots$$

$$p_0 = 1111,\ p_1 = 0000,\ p_2 = 0011$$

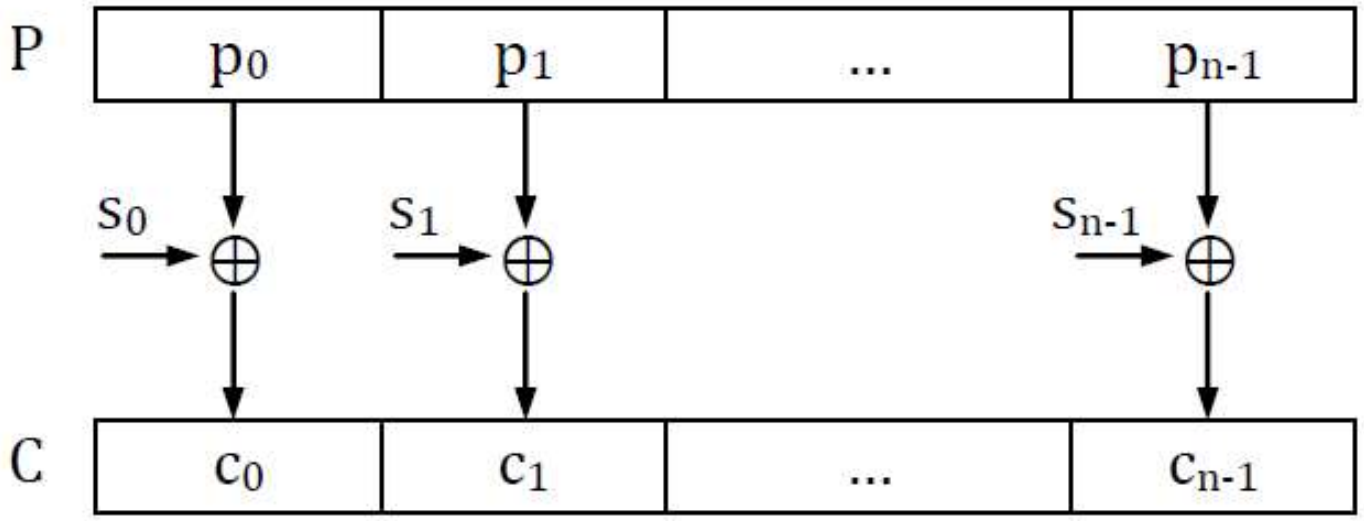
$$s_0 = s_1 = s_2 = K = 0101$$

$$c_0 = 1010,\ c_1 = 0101,\ c_2 = 0110$$

Ví dụ này không phải là mã dòng vì s_0, s_1, s_2 lặp lại khóa K

Stream Ciphers

- Đối với mã dòng, các số *si* được sinh ra phải đảm bảo một độ ngẫu nhiên nào đó (chu kỳ tuần hoàn dài):



Hình 3-1. Mô hình mã dòng

Điểm quan trọng nhất của các mã dòng là bộ sinh số ngẫu nhiên

Stream Ciphers

- Quá trình giải mã được thực hiện ngược lại, bản mã *C* được XOR với dãy số ngẫu nhiên *S* để cho ra lại bản rõ ban đầu:
- Trong ví dụ p=1111 0000 0011 đơn vị mã hóa có chiều dài k = 4 bít, n = 3, với các khóa phát sinh ngẫu nhiên sau: s0=0101, s1=1010, s2=1100, s3 $p_0 = c_0 \oplus s_0,\ p_1 = c_1 \oplus s_1$ ...

- 1010 1010 1111

Cho biết kết quả sau khi mã hóa

Các hệ mã dòng

- ***Chú ý:*** Nếu ta coi "*0*" biểu thị giá trị "sai" và "*1*" biểu thị giá trị "đúng" trong đại số Boolean thì phép cộng theo *moulo 2* sẽ ứng với phép hoặc loại trừ (XOR).
- Bảng chân lý phép cộng theo modulo 2 giống như bảng chân lý của phép toán XOR

a	b	$c = a + b \bmod 2$
0	0	$0 + 0 = 0 \bmod 2$
0	1	$0 + 1 = 1 \bmod 2$
1	0	$1 + 0 = 1 \bmod 2$
1	1	$1 + 1 = 0 \bmod 2$

Các hệ mã dòng

- Ví dụ: mã hóa ký tự ‘A’ bởi Alice
- Ký tự ‘A’ trong bảng mã ASCII được tướng ứng với mã $65_{10}=1000001_2$ được mã hóa bởi hệ khóa $z_1,\ldots,z_7=0101101$
- Hàm mã hóa:

plaintext x_i:	1000001	= ‘A’	(ASCII symbol)
key stream z_i:	0101101		
ciphertext y_i:	1101100	= ‘l’	(ASCII symbol)

- Hàm giải mã:

ciphertext y_i:	1101100	= ‘l’	(ASCII symbol)
key stream z_i:	0101101		
plaintext x_i:	1000001	= ‘A’	(ASCII symbol)

Mã khối (Block Cipher)

- Trong máy tính các chữ cái được biểu diễn bằng mã ASCII.

Bản tin: attack

Mã ASCII: 97 116 116 97 99 107

Biểu diễn nhị phân: 01100001 01110100 01110100 01100001 01100011 01101011

- Trong bản tin nhị phân cũng tồn tại một số đặc tính thống kê nào đó mà người phá mã có thể tận dụng để phá bản mã, dù rằng bản mã bây giờ tồn tại dưới dạng nhị phân.
- Mã hóa hiện đại quan tâm đến vấn đề *chống phá mã trong các trường hợp biết trước bản rõ (known-plaintext), hay bản rõ được lựa chọn (chosen-plaintext)*.

Mã khối (Block Cipher)

- Ví dụ: chúng ta sử dụng bản rõ là các chữ cái

của một *ngôn ngữ* gồm có 8 chữ cái

A, B, C, D, E, F, G, H trong đó mỗi chữ cái được biểu diễn bằng 3 bít.

- Như vậy nếu có bản rõ là 'head' thì biểu diễn nhị phân tương ứng là: 111100000011
- Giả sử dùng một khóa K gồm 4 bít 0101 để mã hóa bản rõ trên bằng phép XOR ⊕:

Chữ cái	Nhị phân
A	000
B	001
C	010
D	011
E	100
F	101
G	110
H	111

bản rõ: 1111 0000 0011 (head)

khóa: 0101 0101 0101

bản mã: 1010 0101 0110 (FBCG)

Trong phép mã hóa trên, đơn vị mã hóa không phải là một chữ cái mà là một khối 4 bít. Để giải mã, lấy bản mã XOR một lần nữa với khóa thì có lại bản rõ ban đầu.

Mã khối (Block Cipher)

Mã khối an toàn lý tưởng

- Phép toán XOR có một hạn chế là chỉ cần biết *một cặp khối* bản rõ và bản mã, người ta có thể dễ dàng suy ra được khóa và dùng khóa đó để giải các khối bản mã khác (knownplaintext attack). :

bản rõ:	1111 0000 0011	(head)
khóa:	0101 0101 0101	
bản mã:	1010 0101 0110	(FBCG)

Nếu biết bản mã $c_0 = 1010$ có bản rõ tương ứng là $p_0 = 1111$, thì có thể dễ dàng suy ra khóa là 0101.

Mã khối (Block Cipher)

- Do đó để chống phá mã trong trường hợp known-plaintext hay choosen-plaintext, chỉ có thể là làm cho *P* và *C* không có mối liên hệ toán học. Điều này chỉ có thể thực hiện được nếu ta lập một bản tra cứu ngẫu nhiên giữa bản rõ và bản mã.

→ Đây là *mã khối an toàn lý tưởng vì* Người gởi cũng như người nhận phải biết toàn bộ bảng trên để mã hóa và giải mã

→ Không khả thi vì kích thước khối lớn thì số dòng của bảng khóa cũng lớn và gây trở ngại cho việc lưu trữ cũng như trao đổi khóa giữa người gởi và người nhận

Bản rõ	Bản mã
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

Các mô hình ứng dụng mã khối

- Mã khối (như mã DES) được áp dụng để mã hóa một khối dữ liệu có kích thước xác định. Để mã hóa một bản tin dài, bản tin được chia ra thành nhiều khối ($P=P_0P_1P_2...P_{n-1}$) và áp dụng mã khối cho từng khối một. Có nhiều mô hình áp dụng mã khối là
  - Electronic Code Book (ECB)
  - Cipher Block Chaining Mode (CBC)
  - Cipher Feedback Mode (CTR)
  - Output Feedback Mode (OFB)
  - Counter Mode

Giới thiệu một số thuật toán

- Mã Feistel (Feistel Cipher)
- Mã DES (Data Encryption Standard)
- Mã Triple DES
- Mã AES (Advanced Encryption Standard)
- RC 4, RC 5, RC 6 (Rivest Cipher 4,5,6)
- Skipjack
- Blowfish
- CAST-128

Mã Feistel (Feistel Cipher)

- Được đề xuất bởi Horst Feistel
- Bản rõ sẽ được biến đổi qua một số vòng để cho ra bản mã cuối cùng

$$P \xrightarrow{K_1} C_1 \xrightarrow{K_2} C_2 \xrightarrow{K_3} \ldots \xrightarrow{K_{n-1}} C_n$$

- Trong đó bản rõ P và các bản mã Ci được chia thành nửa trái và nửa phải:

$$P = (L_0, R_0) \ ; \qquad C_i = (L_i, R_i) \ i = 1, 2, \ldots n$$

Mã Feistel (Feistel Cipher)

- Quy tắc biến đổi các nửa trái phải này qua các vòng được thực hiện như sau:

$$L_i = R_{i-1} \; ; \quad R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

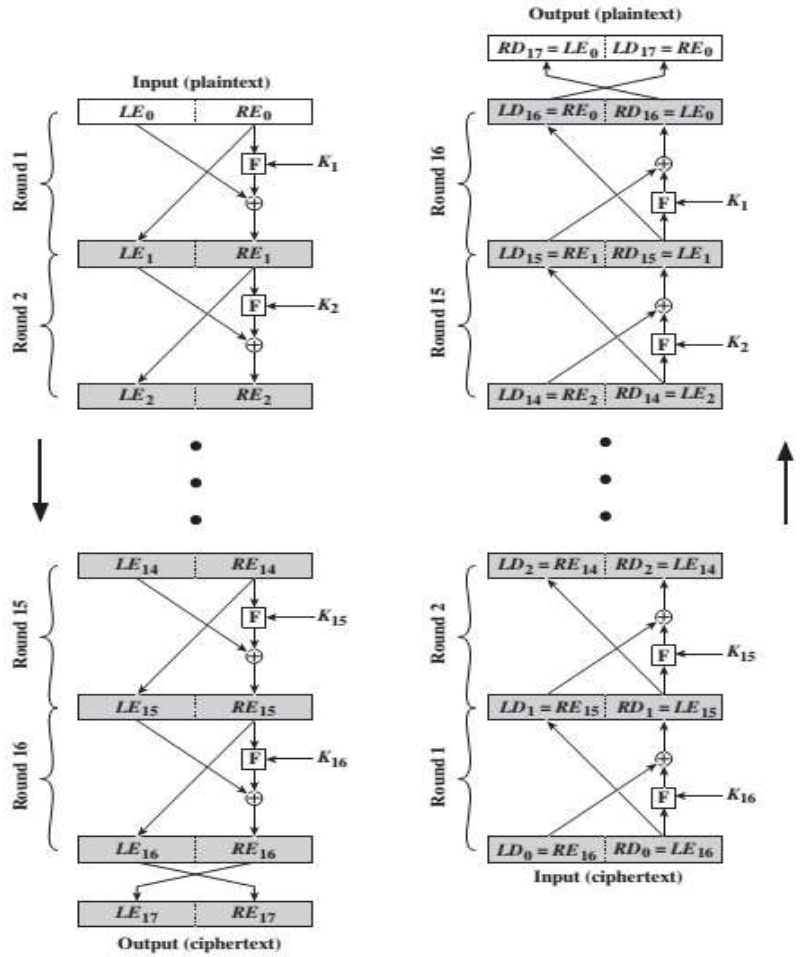
  - K_i là một khóa con cho vòng thứ i. Khóa con này được sinh ra từ khóa K ban đầu theo một thuật toán sinh khóa con (key schedule): $K \rightarrow K_1 \rightarrow K_2 \rightarrow \ldots \rightarrow K_n$
  - F là một hàm mã hóa dùng chung cho tất cả các vòng. Hàm F đóng vai trò như là phép thay thế còn việc hoán đổi các nửa trái phải có vai trò hoán vị.

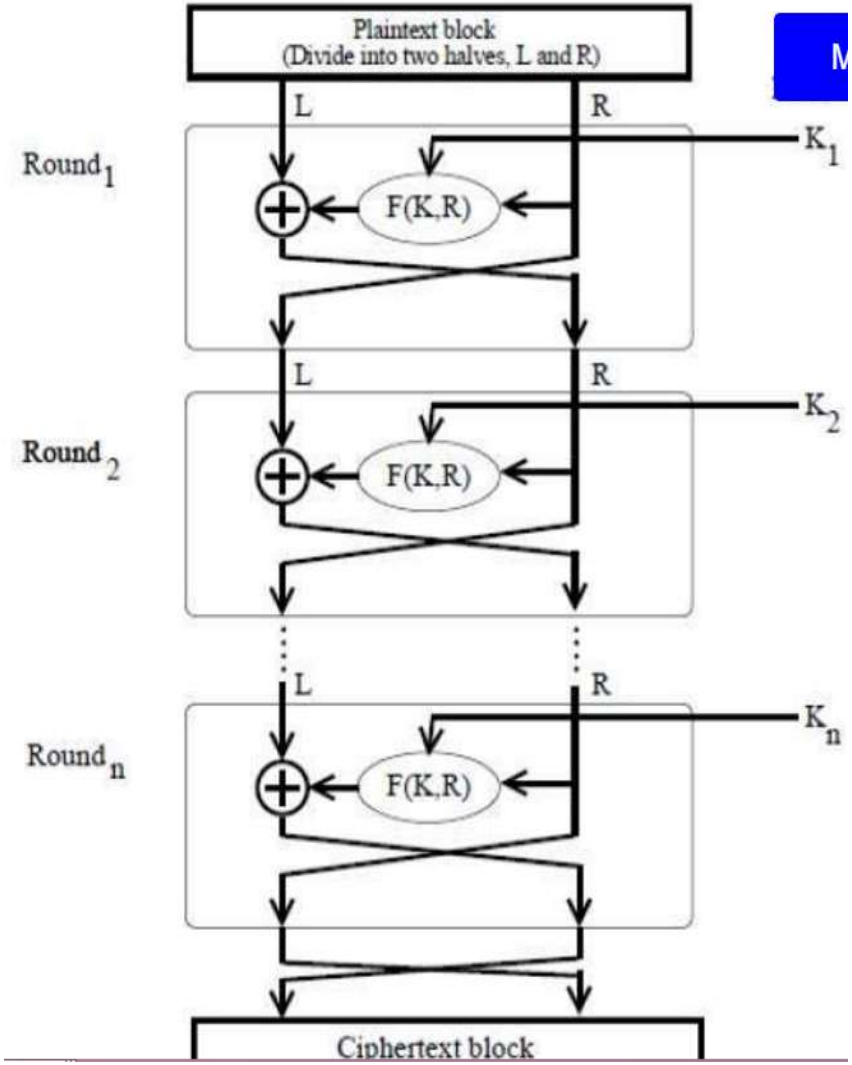
- Bản mã C được tính từ kết xuất của vòng cuối cùng:

$$C = C_n = (L_n, R_n)$$

Mã Feistel (Feistel Cipher)

- Sơ đồ tính toán của hệ mã Feistel





Feistel Cipher

- Để giải mã quá trình được thực hiện qua các vòng theo thứ tự ngược lại:

$C \rightarrow L_n, R_n$

$R_{i-1} = L_i$ (*theo mã hóa Li = Ri-1*)

$L_{i-1} = R_i \oplus F(R_{i-1}, K_i)$

(*theo mã hóa* $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$)

Và cuối cùng bản rõ là $P = (L_0, R_0)$.

Sơ đồ tính toán của hệ mã Feistel

Feistel Cipher

Việc thực hiện chính xác một mã Feistel tùy thuộc vào:

- **Kích cỡ khối (Block size)**: Kích cỡ khối càng lớn có nghĩa là bảo mật càng cao, nhưng tốc độ mã hóa/giải mã (encryption/decryption) giảm (block size: 64 bits).
- **Kích cỡ của khóa (Key size)**: Khóa càng dài thì bảo mật càng cao nhưng cũng giảm tốc độ mã hóa/giải mã. Bảo mật cao có nghĩa là kháng cự lại được tấn công vét cạn (brute-force attacks) và sự hỗn độn (Key size: 128 bits).

Feistel Cipher

- **Số dòng (Number of rounds)**: Bản chất thuật toán mã Feistel là một dòng duy nhất là đã cung cấp đầy đủ tính an toàn nhưng nếu số vòng càng tăng thì tính an toàn càng cao. (thông thường 16 vòng).
- **Thuật toán phát sinh khóa con (Subkey generation algorithm)**: Thuật toán càng phức tạp thì sẽ khó khăn hơn trong việc thám mã.
- **Hàm vòng F (Round function F)**: Càng phức tạp thì đề kháng càng cao đối với thám mã.

Feistel Cipher

Ưu điểm:

- Quá trình mã hóa và giải mã trùng nhau, chỉ khác nhau ở thứ tự khóa con, điều này sẽ tiết kiệm được nữa tài nguyên khi thực hiện thuật toán trên phần cứng.
- Hàm F có thể chọn với độ khó bất kỳ, vì không phải tìm hàm nghịch.

Feistel Cipher

Nhược điểm:

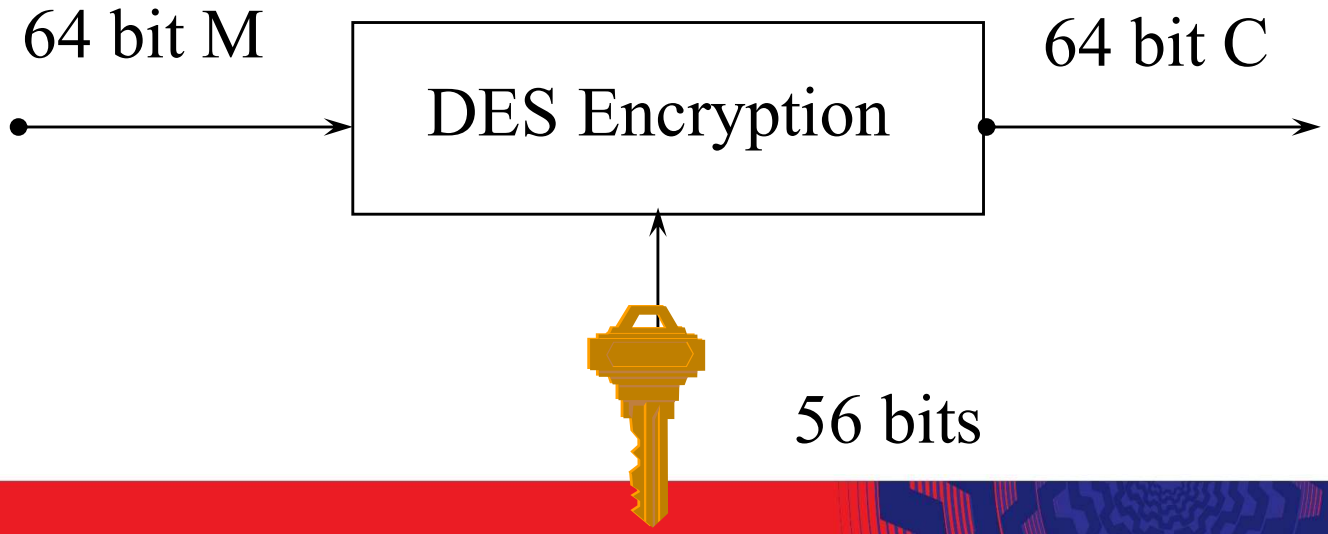
- Vì mỗi vòng mã chỉ thực hiện biến đổi nữa khối dữ liệu, nên cần số vòng mã hóa lớn để đảm bảo độ an toàn của hệ mật, điều này làm giảm đáng kể tốc độ mã.
- Ngoài ra xây dựng trên cơ sở mạng Feistel tồn tại lớp khóa tương đương, nên làm không gian khóa giảm đi một nữa.

Mã DES (Data Encryption Standard)

- DES được công nhận vào năm 1977 bởi Viện nghiên cứu quốc gia về chuẩn của Mỹ (NIST –National Institut of Standards and Technology), chuẩn hóa 1979.
- Là mã thuộc hệ mã Feistel gồm 16 vòng, ngoài ra DES có thêm một hoán vị khởi tạo trước khi vào vòng 1 và một hoán vị khởi tạo sau vòng 16
- Kích thước của khối là 64 bít.
- Ví dụ bản tin "*meetmeafterthetogaparty*" biểu diễn theo mã ASCII thì mã DES sẽ mã hóa làm 3 lần, mỗi lần 8 chữ cái (64 bít): *meetmeaf - tertheto - gaparty.*

Đặc điểm của thuật toán DES

- Khóa dùng trong DES có độ dài toàn bộ là 64 bit. Tuy nhiên chỉ có 56 bit thực sự được sử dụng; 8 bit còn lại chỉ dùng cho việc kiểm tra.
- Mỗi vòng của DES dùng khóa con có kích thước 48 bít được trích ra từ khóa chính.
- Des xuất ra bãn mã 64 bit.
- Mã hoá và giải mã được sử dụng cùng một khoá.
- DES được thiết kế để chạy trên phần cứng.

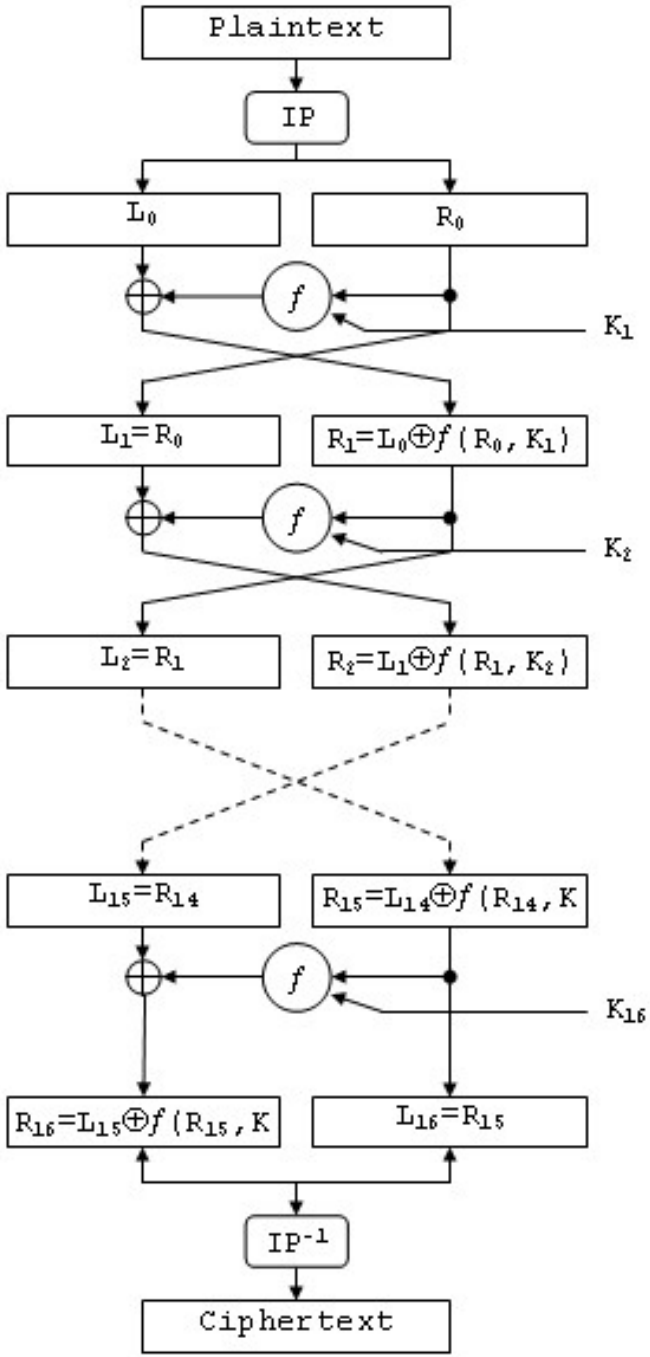


Mô tả thuật toán

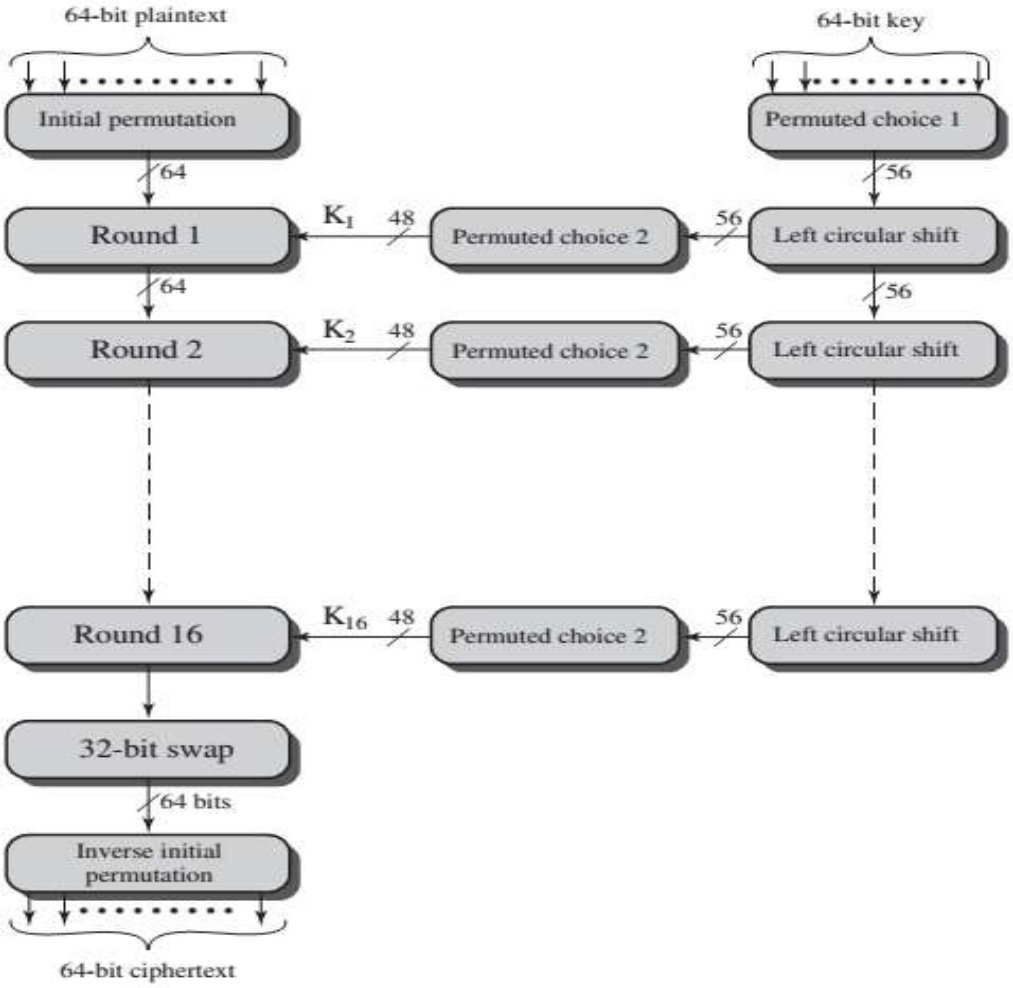
- DES nhận vào một thông điệp M 64 bit, một khóa K 56 bit và cho ra một bảng mã C 64 bit.
- Bước 1: áp dụng một phép hoán vị(bit khởi tạo IP vào M cho ra M': M'=IP(M).
- Bước hai, chia M' thành hai phần: nửa trái L0 =32 bit và nửa phải R_0=32 bit.
- Bước ba, thi hành các phép toán sau với i = 1, 2,…16 (có 16 vòng).

  Li = Ri-1

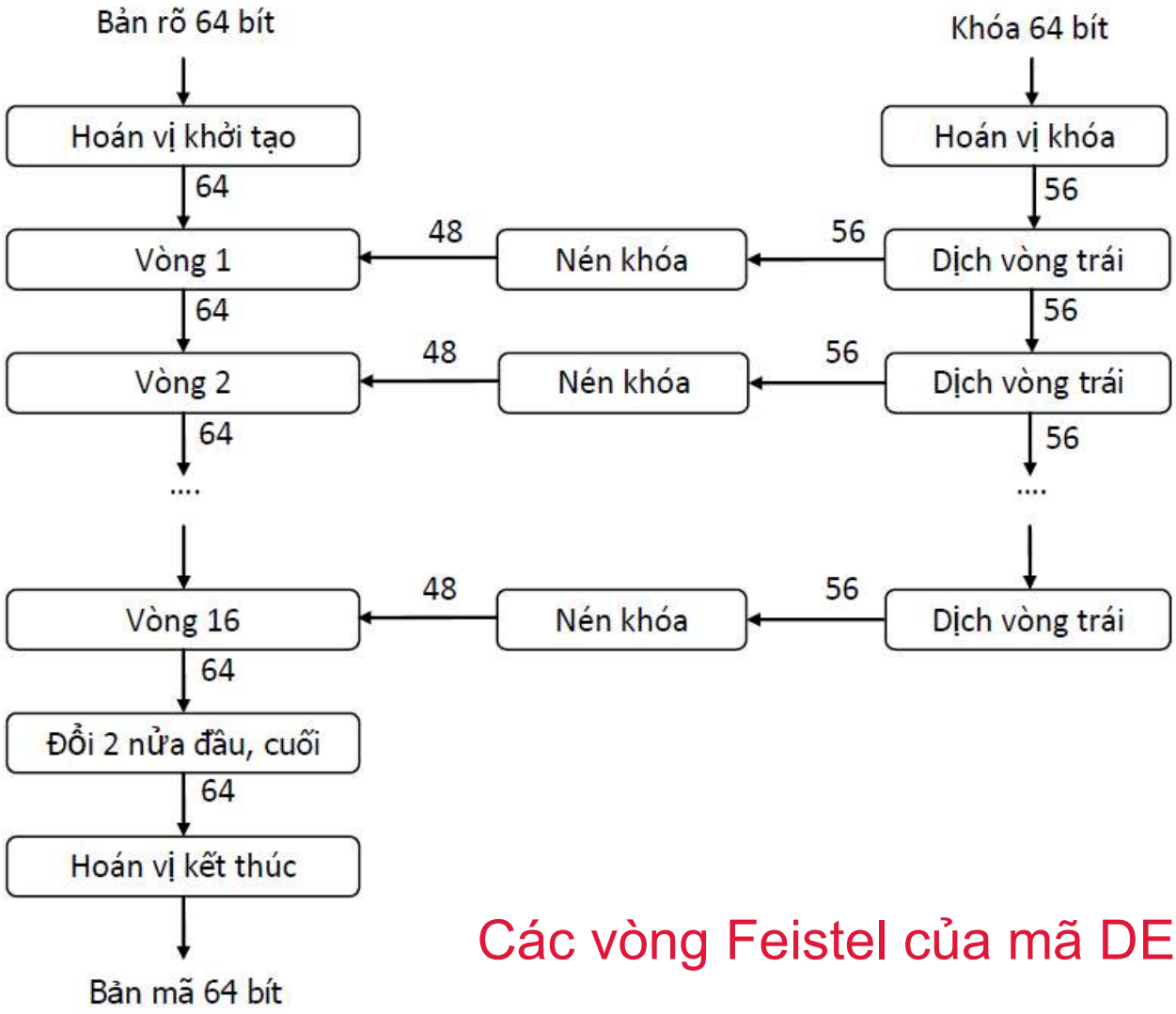
  Ri = Li-1 ⊕ f(Ri-1, Ki)
- Cuối cùng hoán vị với phép hoán vị(IP^{-1}) để được bản mã cuối cùng C.



Các vòng Feistel của mã DES

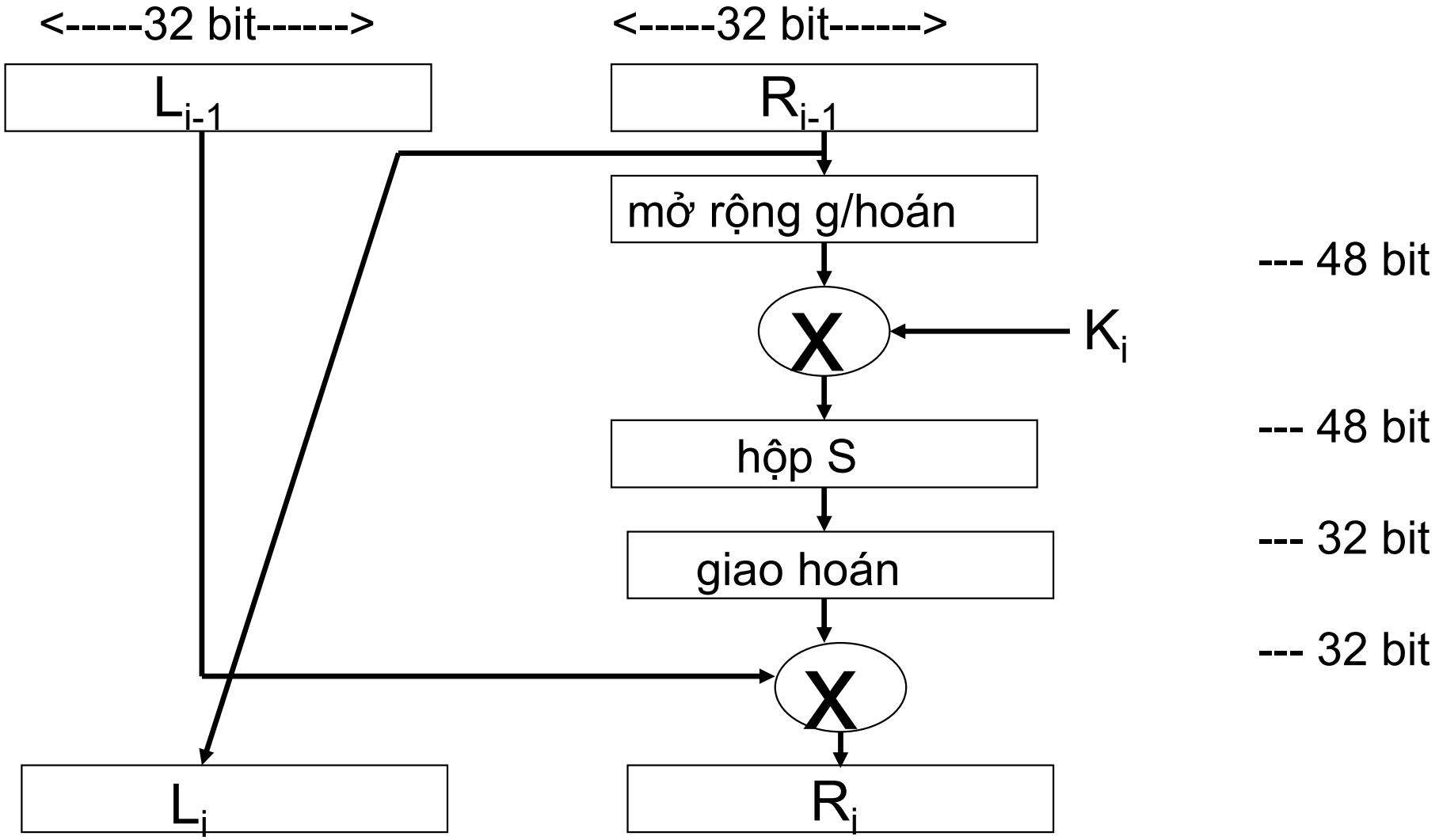


Các vòng Feistel của mã DES



Các vòng Feistel của mã DES

Cấu trúc một vòng của mã DES



Mô tả thuật toán

Thuật toán được thực hiện trong 3 giai đoạn:

1. Cho bản rõ x (64bit) được hoán vị khởi tạo IP (Initial Permutation) tạo nên xâu bit x_0.

   $x_0=IP(x)=L_0R_0$

   L_0 là 32 bit đầu tiên của x_0.
   R_0 là 32 bit cuối của x_0.

Mô tả thuật toán

- Hoán vị khởi tạo và hoán vị kết thúc:
- Ta đánh số các bít của khối 64 bít theo thứ tự từ trái sang phải là 0, 1, …, 62, 63: $b_0 b_1 b_2 \ldots b_{62} b_{63}$
- Hoán vị khởi tạo sẽ hoán đổi các bít theo quy tắc sau :

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
56	48	40	32	24	16	8	0
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6

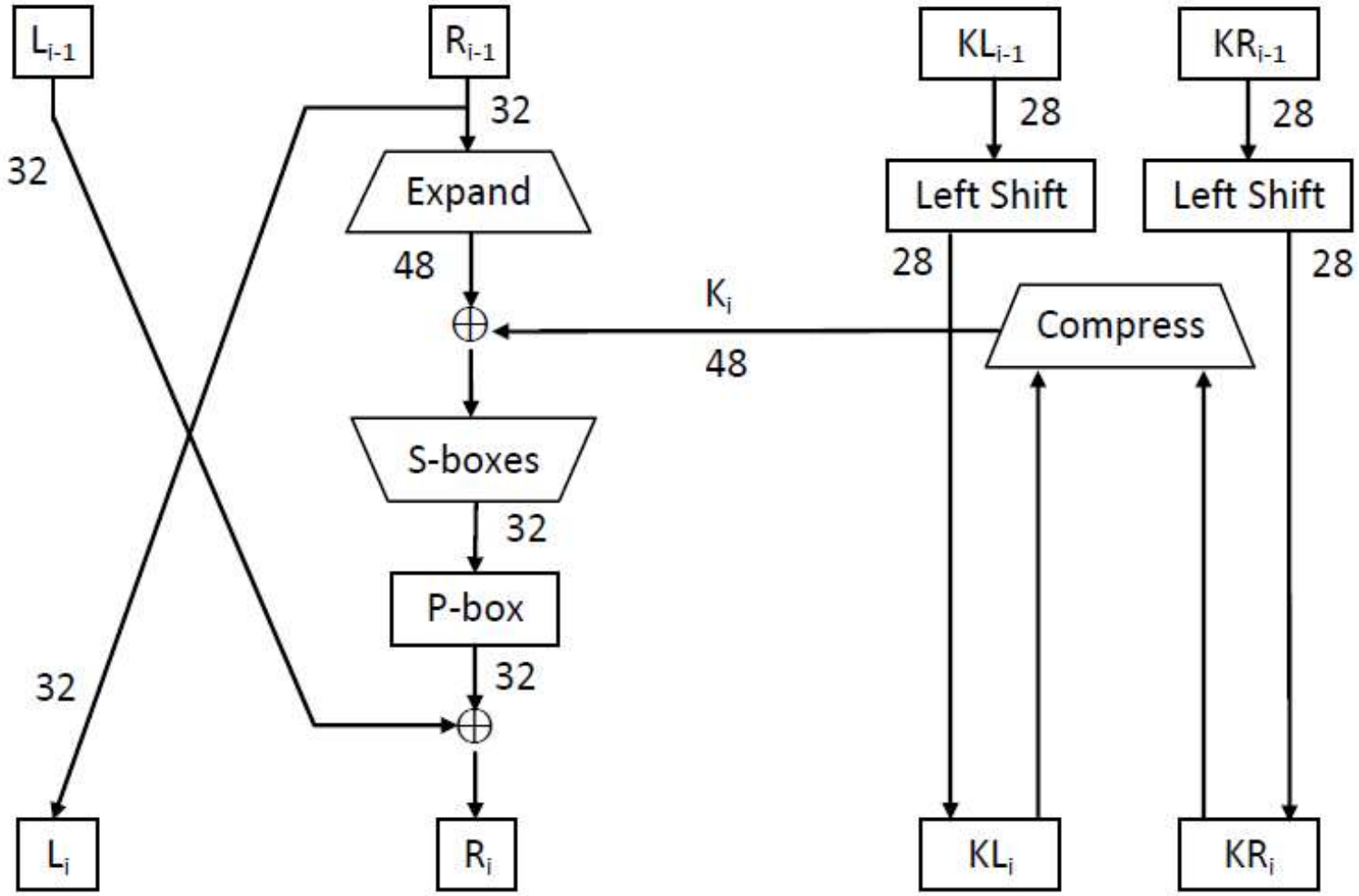
$$(b_0 b_1 b_2 \ldots b_{62} b_{63} \rightarrow b_{57} b_{49} b_{41} \ldots b_{14} b_6)$$

Mô tả thuật toán

- Hoán vị kết thúc hoán đổi các bít theo quy tắc sau:
- Hoán vị kết thúc chính là hoán vị nghịch đảo của hoán vị khởi tạo. Đối với knownplaintext hay chosen-plaintext attack, hoán vị khởi tạo và hoán vị kết thúc không có ý nghĩa bảo mật, sự tồn tại của hai hoán vị trên được nhận định là do yếu tố lịch sử.

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
32	0	40	8	48	16	56	24

Cấu trúc một vòng của mã DES



Hình 3-7. Cấu trúc một vòng của mã DES

Mô tả thuật toán

2. Từ L_0 và R_0 sẽ lặp 16 vòng, tại mỗi vòng tính:

$L_i=R_{i-1}$

$R_i=L_{i-1}\oplus f(R_{i-1},K_i)$ với i= 1, 2,…,16

với:

⊕ là phép XOR của hai xâu bit:

$0 \oplus 0=0$, $1 \oplus 1=0$

$1 \oplus 0=1$, $0 \oplus 1=1$

f là hàm mà ta sẽ mô tả sau.

K_i là các xâu có độ dài 48 bit được tính như là các hàm của khóa K.

K_1 đến K_{16} lập nên một lịch khóa.

Mô tả thuật toán

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

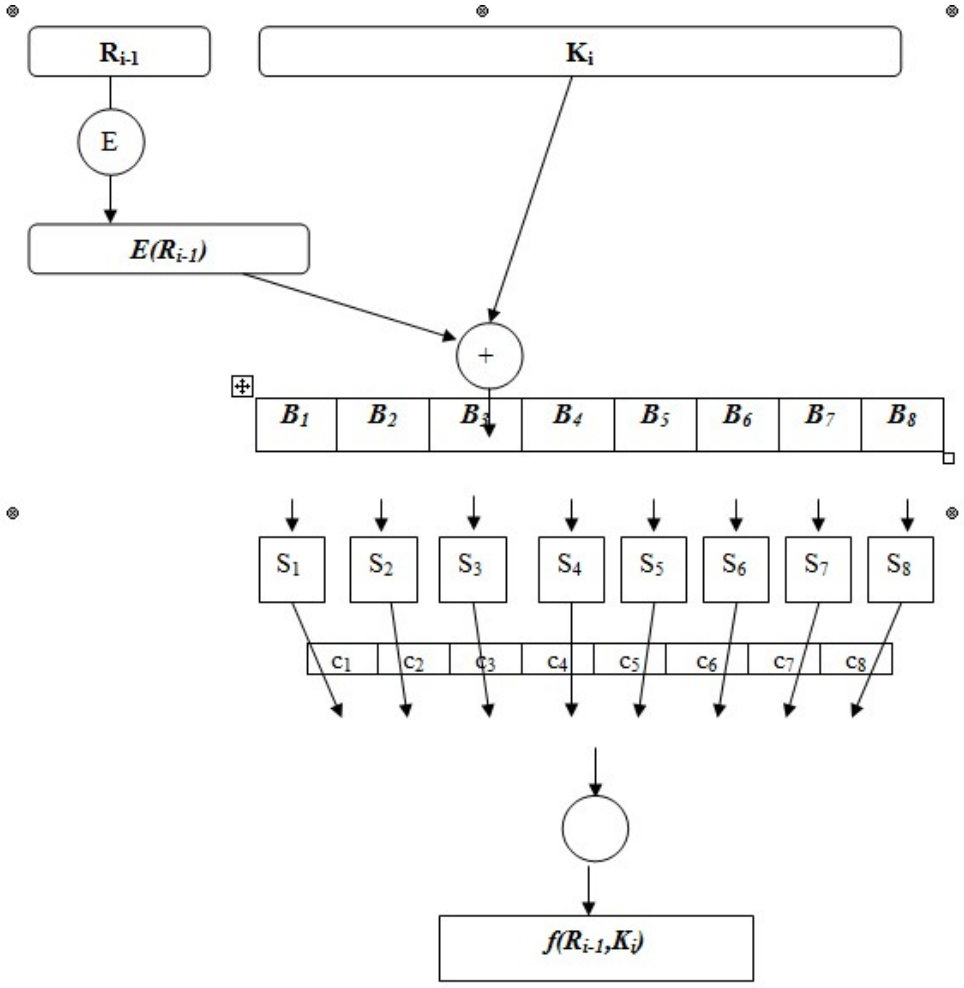
3. Tại vòng thứ 16, R16 đổi chỗ cho L16. Sau đó ghép 2 nửa R16, L16 cho đi qua hoán vị nghịch đảo của hoàn vị IP sẽ tính được bản mã. Bản mã cũng có độ dài 64 bít.

Hoán vị IP^{-1}

Mô tả thuật toán

Hàm f

Sơ đồ tính hàm f(Ri-1,Ki)



Hàm f

- Đối số đầu Ri-1 sẽ được "mở rộng" thành xâu có độ dài 48 bit tương ứng với hàm mở rộng E cố định. E(Ri) bao gồm 32 bit từ Ri, được hoán vị theo một cách thức xác định, với 16 bit được tạo ra 2 lần.

- Hàm f lấy đối số đầu là xâu nhập Ri-1 (32 bit) đối số thứ hai là Ki (48 bit) và tạo ra xâu xuất có độ dài 32 bit. Các bước sau được thực hiện.

Hàm f

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

Hàm mở rộng E

Hàm f

2. Tính $E(R_{i-1}) \oplus K_i$ kết quả được một khối có độ dài 48 bit. Khối này sẽ được chia làm 8 khối $B=B_1B_2B_3B_4B_5B_6B_7B_8$. Mỗi khối này có độ dài là 6 bít.

3. Bước kế tiếp là cho các khối Bi đi qua hộp S_i sẽ biến một khối có độ dài 6 bit thành một khối C_i có độ dài 4 bít.

S-box

- Mỗi hộp S-box là một bảng gồm 4 hàng và 16 cột được đánh số từ 0. Như vậy mỗi hộp S có hàng 0,1,2,3. Cột 0,1,2,…,15. Mỗi phần tử của hộp là một số 4 bít. Sáu bít vào hộp S sẽ xác định số hàng và số cột để tìm kết quả ra.
- Mỗi khối Bi có 6 bít kí hiệu là b1, b2, b3, b4, b5 và b6. Bít b1 và b6 được kết hợp thành một số 2 bít, nhận giá trị từ 0 đến 3, tương ứng với một hàng trong bảng S. Bốn bít ở giữa, từ b2 tới b5, được kết hợp thành một số 4 bít, nhận giá trị từ 0 đến 15, tương ứng với một cột trong bảng S.

S-box

Ví dụ: Ta có B1=011000 thì b_1b_6=00 (xác định r=0), $b_2b_3b_4b_5$=1100 (xác định c=12), từ đó ta tìm được phần tử ở vị trí (0,12) --> S1(B1)=0101 (tương ứng với số 5).

$b_2b_3b_4b_5$=1100

b_1b_6=00

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

Hộp S1

- Mỗi xâu xuất 4 bit của các hộp S được đưa vào các Cj tương ứng: Cj = Sj(Bj) (1<=j<=8).

S-box

Hộp S1

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

Hộp S2

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-box

Hộp S3

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

Hộp S4

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-box

Hộp S5

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

Hộp S6

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-box

Hộp S7

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

Hộp S8

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

Hàm f

4. Xâu bit C = $C_1C_2C_3C_4C_5C_6C_7C_8$ có độ dài 32 bit được hoán vị tương ứng với hoán vị cố định P. Kết quả có P(C)= f(R_i,K_i).

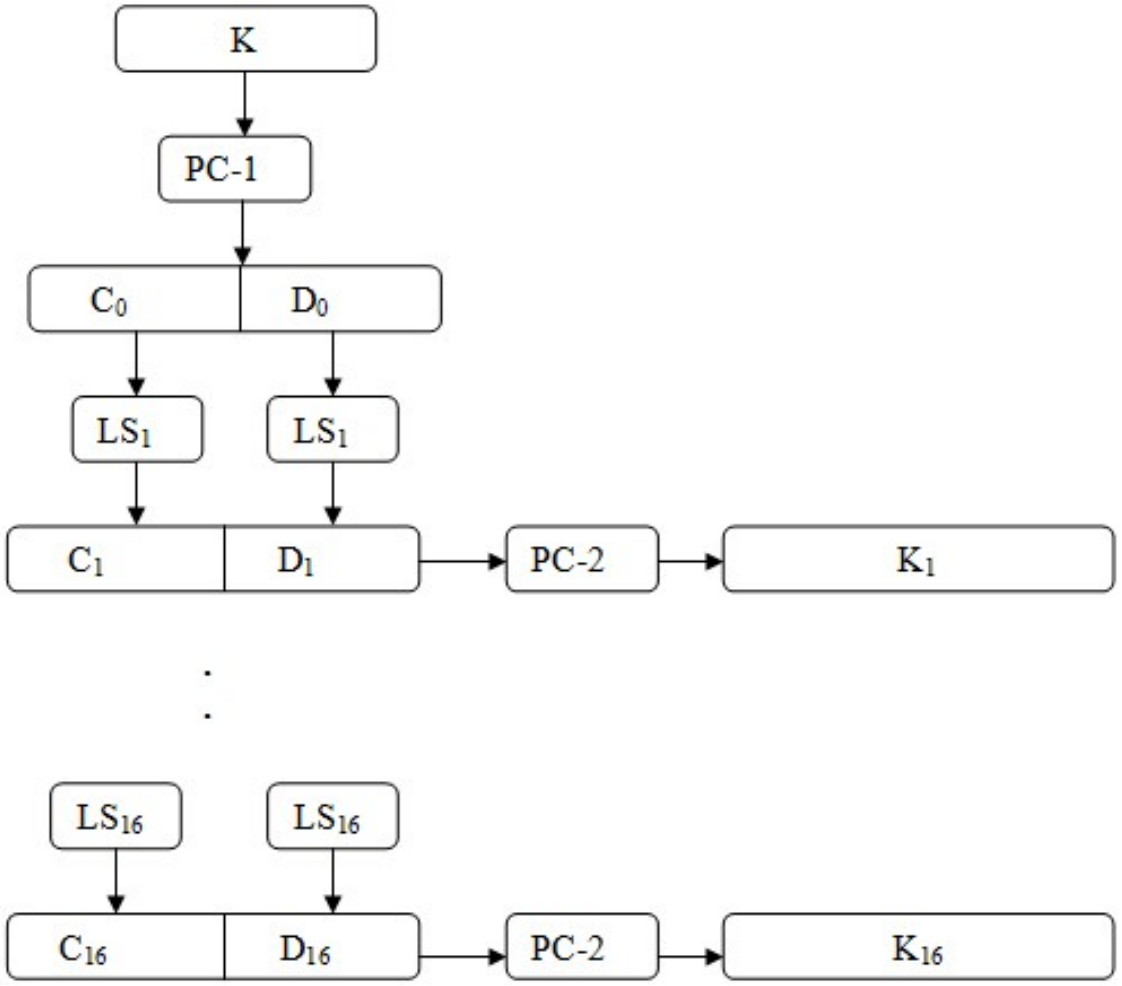
Hoán vị P

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

Khóa K

- K là một xâu có độ dài 64 bit trong đó 56 bit dùng làm khóa và 8 bit dùng để kiểm tra sự bằng nhau (phát hiện lỗi).
- Các bit ở các vị trí 8, 16,…, 64 được xác định, sao cho mỗi byte chứa số lẻ các số 1, vì vậy từng lỗi có thể được phát hiện trong mỗi 8 bit.
- Các bit kiểm tra sự bằng nhau là được bỏ qua khi tính lịch khóa.

Sơ đồ tính khóa K1, K2, …, K16



Khóa K

- Quá trình tạo các khóa con (subkeys) từ khóa K được mô tả như sau:

- Cho khóa K 64 bit, loại bỏ các bit kiểm tra và hoán vị các bit còn lại của K tương ứng với hoán vị cố định PC-1. Ta viết PC1(K) = C0D0, với C0 bao gồm 28 bít đầu tiên của PC-1(k) và D0 là 28 bit còn lại.

Trong đó bảng số bít dịch trái tại mỗi vòng là:

Vòng i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Số bít dịch	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Khóa K

Các hoán vị cố định PC-1 và PC-2:

Bảng trật tự khoá (PC-1):

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

Bảng trật tự nén(PC-2):

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

Giải mã

- Việc giải mã dùng cùng một thuật toán như việc mã hoá.
- Để giải mã dữ liệu đã được mã hoá, quá trình giống như mã hoá được lặp lại nhưng các chìa khoá phụ được dùng theo thứ tự ngược lại từ K_{16} đến K_1, nghĩa là trong bước 2 của quá trình *mã hoá dữ liệu đầu vào* ở trên R_{i-1} sẽ được XOR với K_{17-i} chứ không phải với K_i.

Đặc điểm của mã DES

Tính chất bù của mã DES:

DES có tính chất bù:

$$E_K(P) = C \Leftrightarrow E_{\overline{K}}(\overline{P}) = \overline{C}$$

trong đó :

Ā là phần bù của A theo từng bít (1 thay bằng 0 và ngược lại).

E_K là bản mã hóa của E với khóa K. P và C là văn bản rõ (trước khi mã hóa) và văn bản mã (sau khi mã hóa).

Do tính bù, ta có thể giảm độ phức tạp của tấn công duyệt toàn bộ xuống 2 lần (tương ứng với 1 bít) với điều kiện là ta có thể lựa chọn bản rõ.

Đặc điểm của mã DES

Các khóa yếu trong mã Des:

Ngoài ra DES còn có 4 khóa yếu (weak keys). Khi sử dụng khóa yếu thì mã hóa (E) và giải mã (D) sẽ cho ra cùng kết quả:

$$E_K(E_K(P)) = P \text{ or equivalently, } E_K = D_K$$

Bên cạnh đó, còn có 6 cặp *khóa nửa yếu* (semi-weak keys). Mã hóa với một khóa trong cặp, $K1$, tương đương với giải mã với khóa còn lại, $K2$:

$$E_{K1}(E_{K2}(P))=P \text{ or equivalently } E_{K1}=D_{K2}$$

Tuy nhiên có thể dễ dàng tránh được những khóa này khi thực hiện thuật toán, có thể bằng cách thử hoặc chọn khóa một cách ngẫu nhiên. Khi đó khả năng chọn phải khóa yếu là rất nhỏ.

Đặc điểm của mã DES

Triple DES:

Triple-DES chính là DES với hai chìa khoá 56 bit. Cho một bản tin cần mã hoá, chìa khoá đầu tiên được dùng để mã hoá DES bản tin đó.

Kết quả thu được lại được cho qua quá trình giải mã DES nhưng với chìa khoá là chìa khoá thứ hai.

Bản tin sau qua đã được biến đổi bằng thuật toán DES hai lần như vậy lại được mã hoá DES một lần nữa với chìa khoá đầu tiên để ra được bản tin mã hoá cuối cùng.

Quá trình mã hoá DES ba bước này được gọi là Triple-DES.

Độ an toàn của DES

- ***Tấn công vét cạn khóa (Brute Force Attack)***
  - Khóa của DES: **56 bít** nên để tiến hành vét cạn khóa cần kiểm tra 2^{56} khóa khác nhau.→ rất lớn
  - 1998, tổ chức Electronic Frontier Foundation (EFF) thông báo đã xây dựng được một thiết bị phá mã DES gồm nhiều máy tính chạy song song, trị giá khoảng 250.000$. Thời gian thử khóa là 3 ngày.
  - Hiện nay mã DES vẫn còn được sử dụng trong thương mại, tuy nhiên người ta đã bắt đầu áp dụng những phương pháp mã hóa khác có chiều dài khóa lớn hơn (128 bít hay 256 bít) như TripleDES hoặc AES.

Độ an toàn của DES

- ***Phá mã DES theo phương pháp vi sai*** (differential cryptanalysis):
  - Năm 1990 Biham và Shamir đã giới thiệu phương pháp phá mã vi sai. Phương pháp vi sai tìm khóa ít tốn thời gian hơn brute-force. Tuy nhiên phương pháp phá mã này lại đòi hỏi phải có 247 cặp bản rõ - bản mã được lựa chọn (chosen-plaintext). Vì vậy phương pháp này là bất khả thi dù rằng số lần thử có thể ít hơn phương pháp brute-force.

Độ an toàn của DES

- ***Phá mã DES theo phương pháp thử tuyến tính*** (linear cryptanalysis)
  - Năm 1997 Matsui đưa ra phương pháp phá mã tuyến tính. Trong phương pháp này, cần phải biết trước 243 cặp bản rõ-bản mã (known-plaintext). Tuy nhiên 243 cũng là một con số lớn nên phá mã tuyến tính cũng không phải là một phương pháp khả thi.

Mã Triple DES

- Một trong những cách để khắc phục yếu điểm kích thước khóa ngắn của mã hóa DES là sử dụng mã hóa DES nhiều lần với các khóa khác nhau cho cùng một bản tin. Đơn giản nhất là dùng DES hai lần với hai khóa khác nhau, cách thức này được gọi là Double DES

$$C=E(E(P, K_1),K_2)$$

- Điều này giống như là Double DES dùng một khóa có kích thước là 112 byte, chỉ có một hạn chế là tốc độ chậm hơn DES vì phải dùng DES hai lần. Tuy nhiên người ta đã tìm một phương pháp tấn công Double DES có tên gọi là gặp-nhau-ở-giữa (meet-in-the middle).Đây là một phương pháp tấn công chosen-plaintext.

Mã Triple DES

- Nếu dùng DES ba lần với ba khóa khác nhau, cách thức này được gọi là Triple DES:

$$C=E(E(E(P, K_1),K_2),K_3)$$

- Chiều dài khóa là 168 bít sẽ gây phức tạp hơn nhiều cho việc phá mã bằng phương pháp tấn công gặp-nhau-ở-giữa. Trong thực tế người ta chỉ dùng Triple DES với hai khóa $K1$, $K2$ mà vẫn đảm bảo độ an toàn cần thiết.

$$C=E(E(E(P, K_1),K_2),K_1)$$

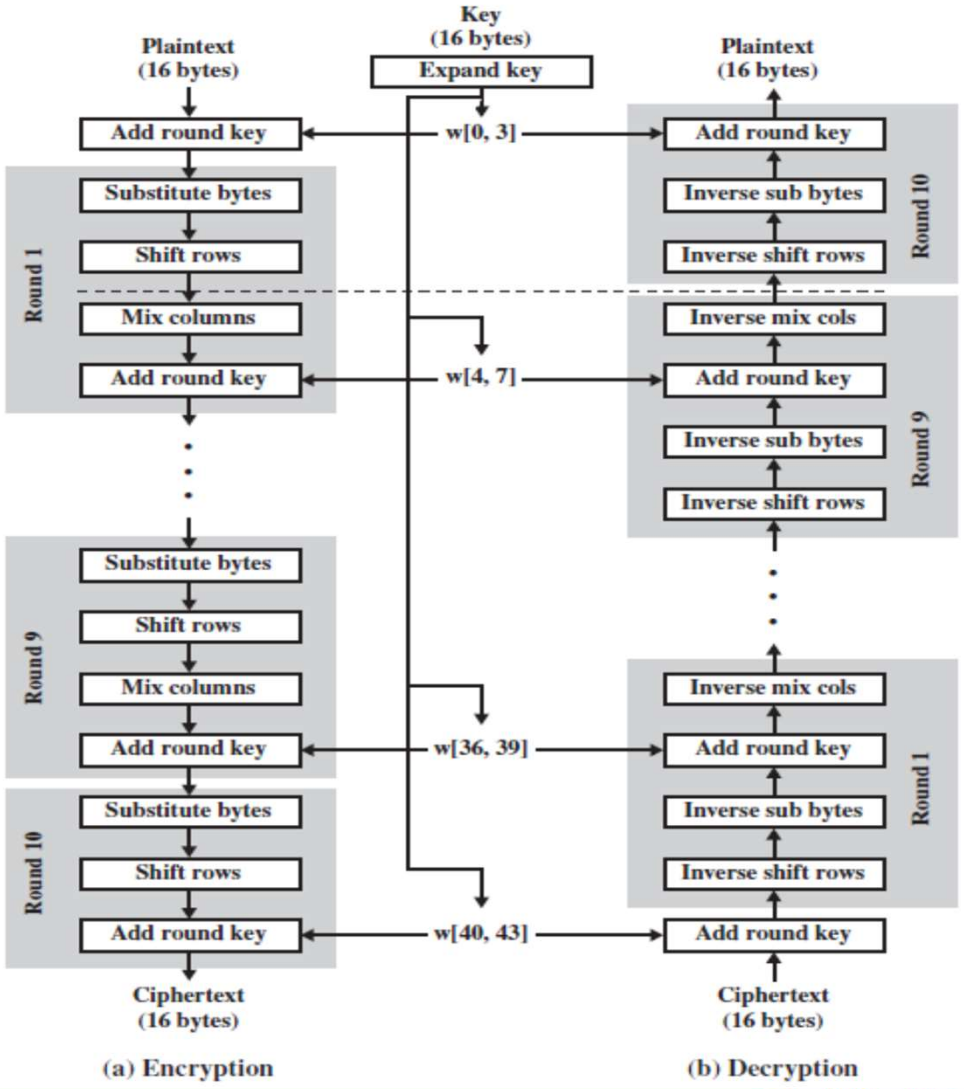
Mã AES (Advanced Encryption Standard)

- Tiêu chuẩn mã hóa tiên tiến (Advanced Encryption Standard - AES) được giới thiệu vào năm 2001 bởi NIST với mục đích thay thế DES có nhiều hạn chế, và được sử dụng rộng rãi trong các ứng dụng hiện nay, thuật toán được thiết kế bởi Joan Daemen và Vincent Rijmen với tên gọi ban đầu là Rijndael.
- Khóa có kích thước 256 bít là ***"an toàn mãi mãi"*** bất kể những tiến bộ trong ngành kỹ thuật máy tính.
- AES là thuật toán mã hóa đối xứng dạng khối 128-bit

Mã AES (Advanced Encryption Standard)

- Như DES, mã AES là mã khối, nhiều vòng, nhưng mã AES không phải là một mã Feistel. Thuật toán AES khá phức tạp.
- Đặc điểm chính của AES:
  - Cho phép lựa chọn kích thước khối mã hóa là 128, 192 hay 256 bít.
  - Cho phép lựa chọn kích thước của khóa một cách độc lập với kích thước khối: là 128, 192 hay 256 bít.
  - Số lượng vòng có thể thay đổi từ 10 đến 14 vòng tùy thuộc vào kích thước khóa.

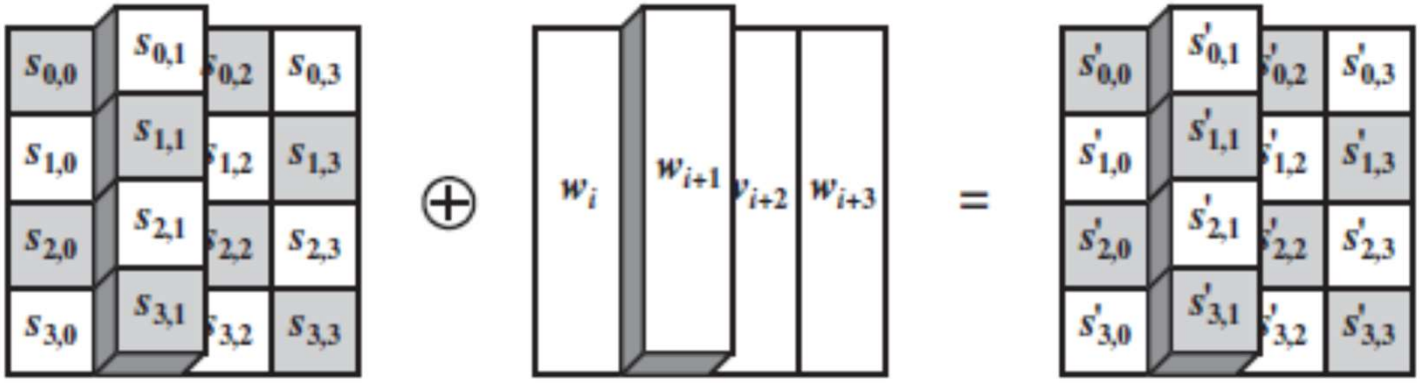
Mã AES (Advanced Encryption Standard)



Mã AES (Advanced Encryption Standard)

- Quá trình mã hóa bao gồm 4 bước:

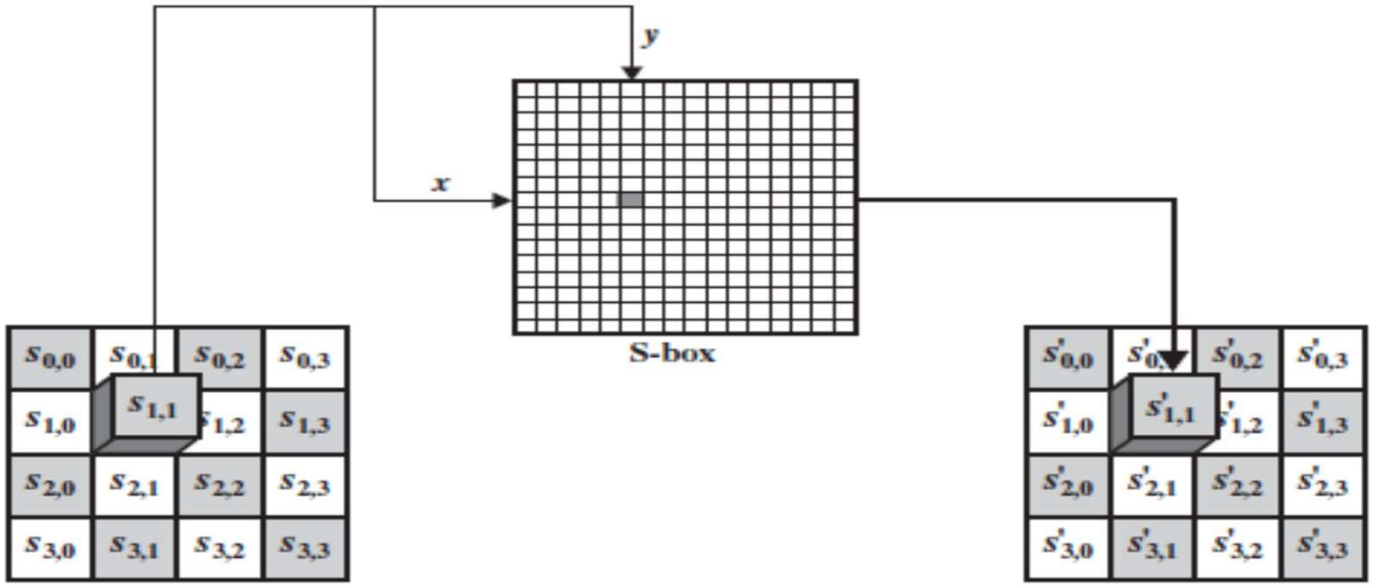
B1: AddRoundKey - mỗi byte của khối được kết hợp với khóa con, các khóa con này được tạo ra từ quá trình tạo khóa con Rijndael.



Add round key transformation

Mã AES (Advanced Encryption Standard)

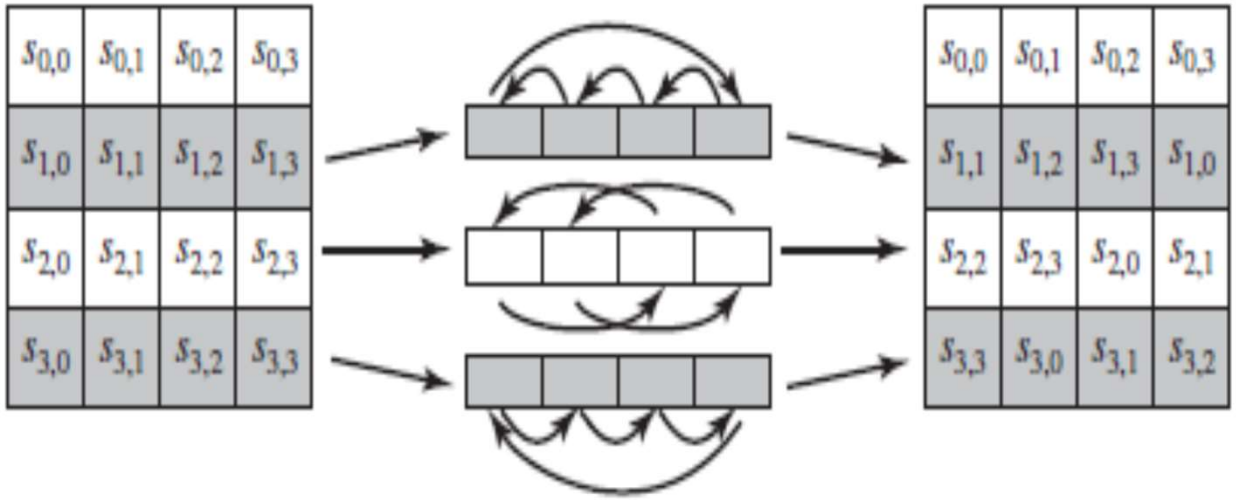
- B2: SubBytes - đây là quá trình thay thế trong đó mỗi byte sẽ được thay thế bằng một byte khác theo bảng tra.



Substitute byte transformation

Mã AES (Advanced Encryption Standard)

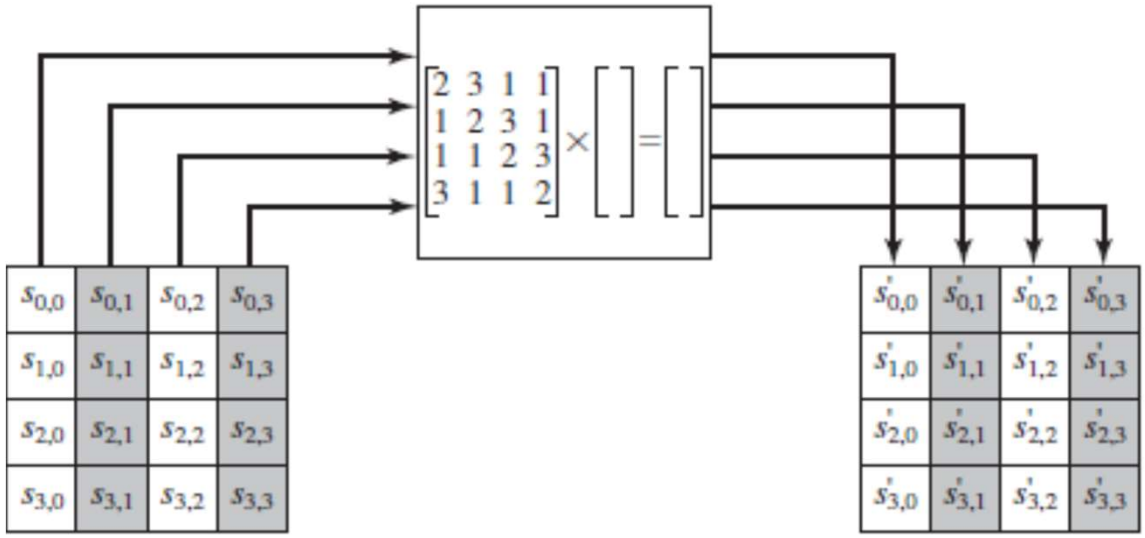
- B3: ShiftRows - đổi chỗ, các hàng trong khối được dịch vòng.



Shift row transformation

Mã AES (Advanced Encryption Standard)

- B4: MixColumns - quá trình trộn làm việc theo các cột trong khối theo một chuyển đổi tuyến tính.
- Tại chu trình cuối thì MixColumns được thay thế bằng AddRoundKey



Mã AES (Advanced Encryption Standard)

- Độ an toàn của AES làm cho AES được sử dụng ngày càng nhiều và trong tương lai sẽ chiếm vai trò của DES và Triple DES.

Ưu và nhược điểm của mã hóa đối xứng

Ưu điểm:

- Độ an toàn cao (phụ thuộc vào thuật toán và khóa), quá trình mã hóa và giải mã nhanh do đó mã hóa đối xứng được sử dụng phổ biến trong việc truyền dữ liệu.

Ưu và nhược điểm của mã hóa đối xứng

Nhược điểm: Một số vấn đề cần quan tâm của hệ thống mã hóa đối xứng liên quan đến khóa, bao gồm:

- Do quá trình mã hóa và giải mã sử dụng chung một khóa nên khóa (secret key) sử dụng cần phải được bảo quản an toàn tuyệt đối.
- Vấn đề phân phối khóa
- Vấn đề quản lý khóa (với hệ thống có n nút khác nhau thì số lượng khóa cần thiết cho hệ thống là n(n+1)/2
- Không cung cấp tính chống thoái thác thông tin

Câu hỏi và bài tập

1. Mã hóa đối xứng hiện đại và mã hóa đối xứng cổ điển khác nhau ở điểm nào.
2. Mã dòng hoạt động dựa trên nguyên tắc thay thế hay hoán vị?
3. Hệ mã Fiestel có thuận lợi gì trong việc thực hiện mã khối?
4. Tại sao mã hóa DES lại dùng các phép biến đổi phức tạp chỉ để mã hóa một khối 64 bít?

Câu hỏi và bài tập

1. Khái niệm mã hóa, tại sao phải mã hóa thông tin khi truyền tin trên mạng?
2. Khái niệm mã hóa đối xứng, cơ chế, các thành phần của hệ mã hóa đối xứng.
3. Tại sao gửi bản mã (cipher) trên kê truyền thì không sợ bị lộ thông tin?
4. Khóa là gì? Tại sao cần giữ bí mật khóa chỉ có người gửi và người nhận biết?
5. Khám mã khác giải mã ở điểm nào?
6. Khám mã theo hình thức vét cạn khóa thực hiện như thể nào? Cần làm gì để chống lại hình thức khám mã theo kiểu vét cạn khóa?

Câu hỏi và bài tập

7. Các phương pháp Ceasar, mã hóa đơn bảng, đa bảng, one-time pad dùng nguyên tắc gì để mã hóa?
8. Mã hóa bản rõ "DAI HOC CONG NGHIEP", dùng phương pháp mã hóa Ceasar với k=3
9. Giải mã bản mã sau, giải sự mã hóa Ceasar được sử dụng để mã hóa với k=3

   IRXUVFRUHDQGVHYHQBHDUVDJR

7. Mã hóa bản rõ "DAI HOC CONG NGHIEP", dùng phương pháp mã hóa thay thể đơn bản (Monoalphabetic Ciphers) với khóa hoán vị K là: IAUTMOCSNREBDLHVWYFPZJXKGQ
8. Mã hóa bản rõ "DAI HOC CONG NGHIEP", dùng phương pháp mã hóa Playfair với khóa k là "monarchy".
9. Bài tập trang 59 (File BaiGiangATTT.pdf)

Câu hỏi và bài tập

Find the output of the initial permutation box when the input is given in hexadecimal as:

0x0000 0080 0000 0002

Solution

Only bit 25 and bit 64 are 1s; the other bits are 0s. In the final permutation, bit 25 becomes bit 64 and bit 63 becomes bit 15. The result is

0x0002 0000 0000 0001

Câu hỏi và bài tập

Prove that the initial and final permutations are the inverse of each other by finding the output of the final permutation if the input is

0x0002 0000 0000 0001

Solution

The input has only two 1s; the output must also have only two 1s. Using Table 6.1, we can find the output related to these two bits. Bit 15 in the input becomes bit 63 in the output. Bit 64 in the input becomes bit 25 in the output. So the output has only two 1s, bit 25 and bit 63. The result in hexadecimal is

Câu hỏi và bài tập

The input to S-box 1 is 100011. What is the output?

Solution

If we write the first and the sixth bits together, we get 11 in binary, which is 3 in decimal. The remaining bits are 0001 in binary, which is 1 in decimal. We look for the value in row 3, column 1, in Table 6.3 (S-box 1). The result is 12 in decimal, which in binary is 1100. So the input 100011 yields the output 1100.

Câu hỏi và bài tập

The input to S-box 8 is 000000. What is the output?

Solution

If we write the first and the sixth bits together, we get 00 in binary, which is 0 in decimal. The remaining bits are 0000 in binary, which is 0 in decimal. We look for the value in row 0, column 0, in Table 6.10 (S-box 8). The result is 13 in decimal, which is 1101 in binary. So the input 000000 yields the output 1101.

Câu hỏi và bài tập

What is the probability of randomly selecting a weak, a semi-weak, or a possible weak key?

Solution

DES has a key domain of 2^{56}. The total number of the above keys are 64 (4 + 12 + 48). The probability of choosing one of these keys is 8.8×10^{-16}, almost impossible.

THANKS YOU

INNOVATION - UNITY - HUMANITY

www.iuh.edu.vn