

Bài 4

Mã khối hiện đại

Thời lượng: 4 tiết

Lương Thái Lê

Tình huống dẫn nhập

- Cần che giấu nội dung thông tin không cho người không có thẩm quyền xem
- Tạo nên công cụ chuẩn để mọi người dễ dàng sử dụng
- Mã hóa thông điệp nhờ thuật toán chuẩn chung và thông tin bí mật (khóa) chia sẻ giữa người gửi và người nhận
- Người nhận có thể biến đổi ngược lại bản mã nhận được để có thông điệp gốc

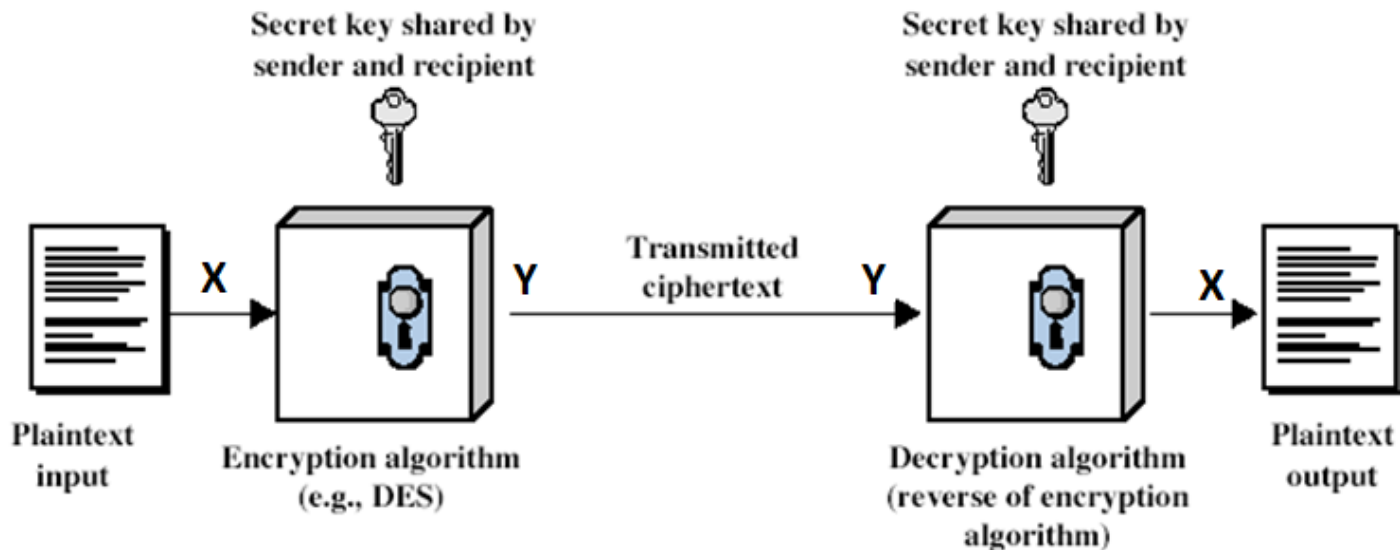
=> Xuất hiện mã khối đối xứng (hiện đại)

Nội dung

- Nguyên lý mã đối xứng và mã khối
- Mã khối Feistel
- Các thuật toán mã khối đối xứng
 - Chuẩn mã dữ liệu DES
 - Chuẩn mã nâng cao AES
- Mã dòng RC4

Nguyên lý mã đối xứng

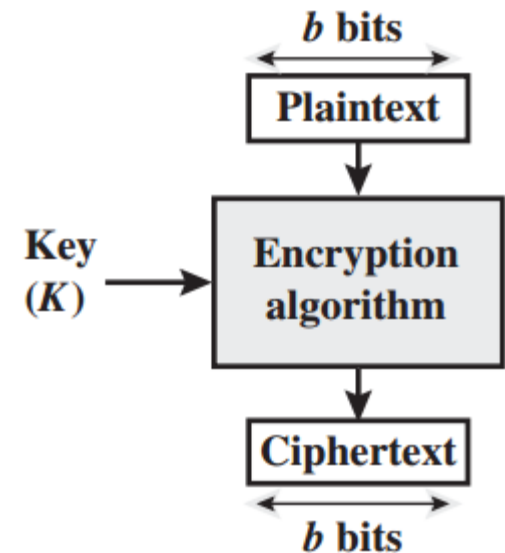
- Sơ đồ mã có 5 khái niệm:



- An ninh phụ thuộc vào bảo mật khóa, chứ không phải bảo mật thuật toán
- Câu hỏi: Tại sao lại cho rằng thuật toán mọi người đều biết?*
- Trả lời: Vì ta phải chuẩn hóa các thuật toán dùng chung trên các giao thức trên mạng, và số thuật toán dùng được cũng không nhiều, nên luôn giả thiết là phổ cập*

Mã dòng vs. Mã khối

- Mã dòng :
 - mã hóa theo dòng bit (hoặc byte)
 - Mỗi bit (byte) một khóa riêng
- Mã khối:
 - Mã hóa từng khối bit
 - Mỗi khối dùng chung 1 khóa
 - Khối có thể gồm 64bit (DES), 128 bit (AES)...



Nội dung

- Nguyên lý mã đối xứng và mã khối
- Mã khối Feistel
- Các thuật toán mã khối đối xứng
 - Chuẩn mã dữ liệu DES
 - Chuẩn mã nâng cao AES
- Mã dòng RC4

Mã khối Feistel

- Hầu hết thuật toán mã khối hiện nay, kể cả DES đều dựa trên cấu trúc của mã khối Feistel (bác học Horst Feistel của IBM, năm 1973)
- Là mã tích có thể giải ngược (reversible)
- Dựa trên 2 nguyên lý của Shannon:
 - *Tính khuếch tán (diffusion)*: làm mất đi cấu trúc thống kê của bản rõ trên bản mã
 - *Tính rối loạn (confusion)*: làm cho quan hệ giữa bản mã và khóa càng phức tạp càng tốt

Đề xuất của Feistel

⇒ Feistel đề xuất khóa độ dài k với số phép biến đổi là 2^k

⇒ Số vòng lặp: n bất kỳ

⇒ kích thước khối: $2w$ bit

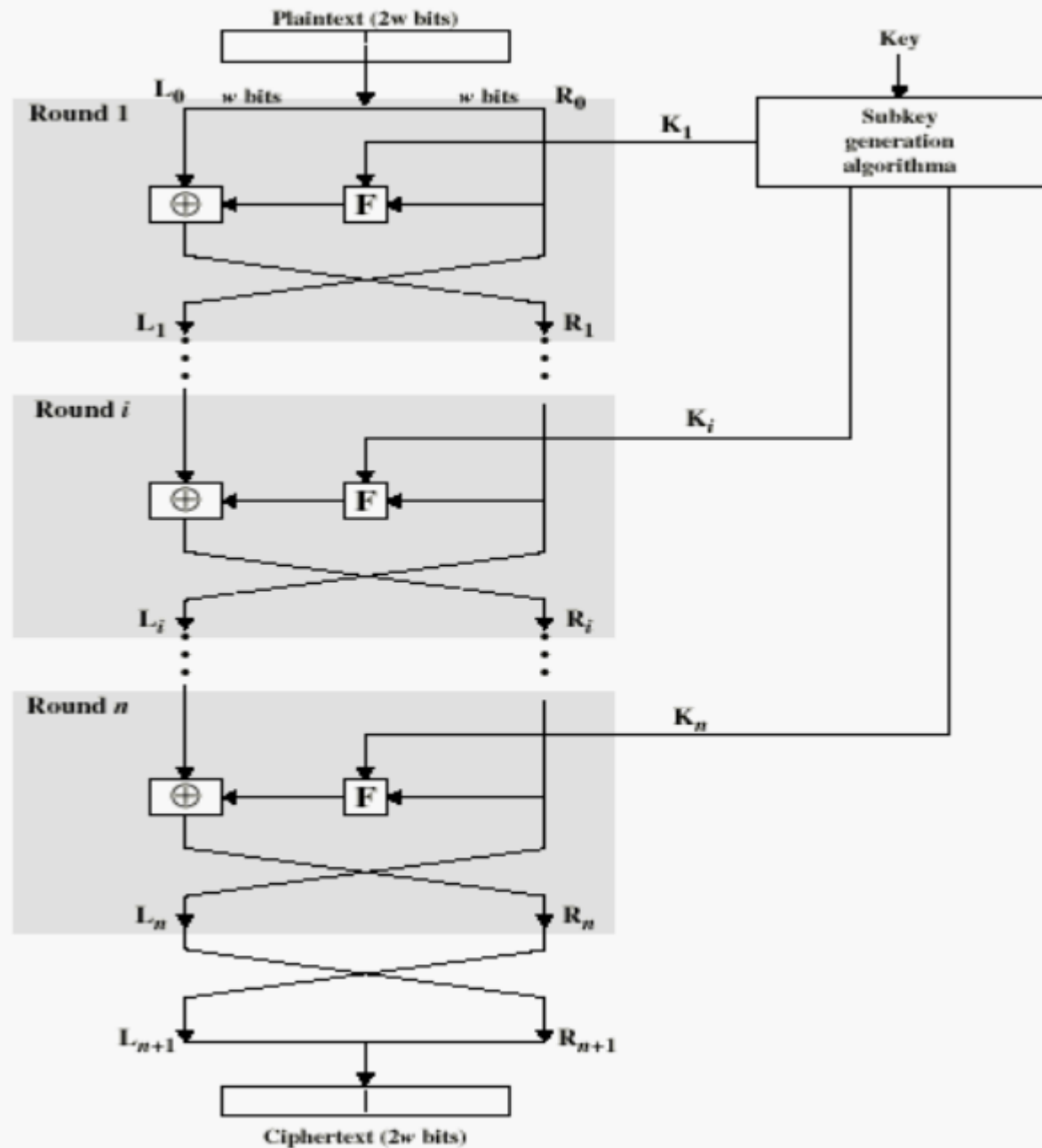


Figure 2.2 Classical Feistel Network

Các đặc trưng của Kiến trúc mã Feistel

- **Kích thước khối:** kích thước khối càng lớn an toàn càng cao
- **Kích thước khóa:** kích thước khóa càng lớn an toàn càng cao
- **Số vòng:** nhiều vòng lặp sẽ tăng cường an ninh
- **Thuật toán sinh khóa con:** độ phức tạp càng lớn thì độ khó thám mã càng cao.
- **Hàm F :** càng phức tạp càng tốt
- **Mã hoá / giải mã nhanh:** tốc độ thực hiện thuật toán mã hóa trở nên rất quan trọng
- **Dễ phân tích:** => dễ bảo vệ

Nội dung

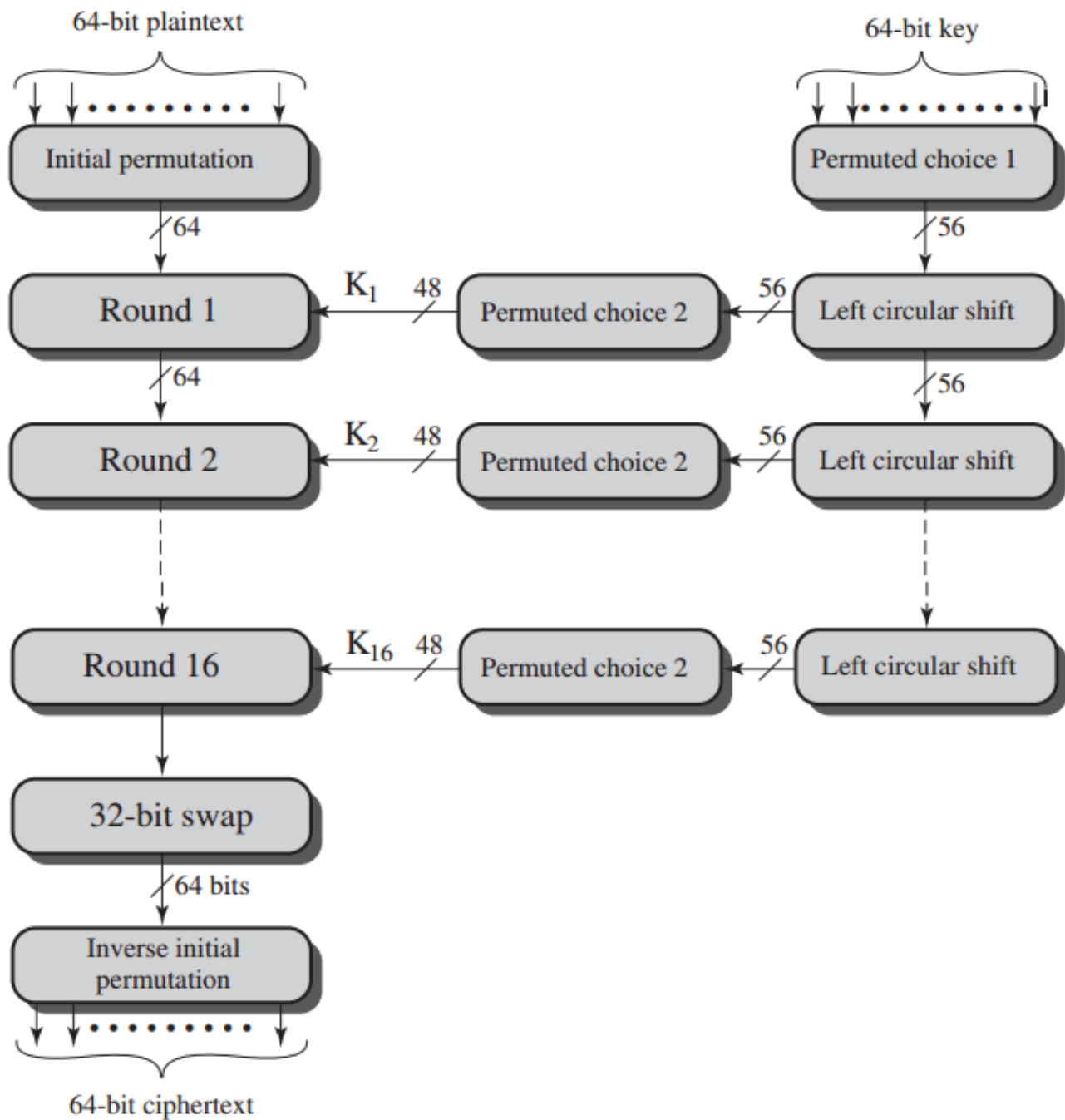
- Nguyên lý mã đối xứng và mã khối
- Mã khối Feistel
- Các thuật toán mã khối đối xứng
 - Chuẩn mã dữ liệu DES
 - Chuẩn mã nâng cao AES
- Mã dòng RC4

Chuẩn mã dữ liệu

Data Encryption Standard (DES)

- Mã khối sử dụng rộng rãi nhất trên thế giới
- Được đưa ra năm 1977 bởi NIST - Viện chuẩn và công nghệ Quốc gia (National Institute of Standard and Technology)
- Dựa trên kiến trúc của mã khối Feistel
- Mã khối dữ liệu 64 bit; dùng khoá dài 56 bit; số vòng 16
- Được tranh luận kỹ về mặt an toàn

Tổng quan của thuật toán mã hóa DES



Hoán vị khởi đầu của DES

(a) Initial Permutation (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

(b) Inverse Initial Permutation (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

Bảng 1

M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}
M_{17}	M_{18}	M_{19}	M_{20}	M_{21}	M_{22}	M_{23}	M_{24}
M_{25}	M_{26}	M_{27}	M_{28}	M_{29}	M_{30}	M_{31}	M_{32}
M_{33}	M_{34}	M_{35}	M_{36}	M_{37}	M_{38}	M_{39}	M_{40}
M_{41}	M_{42}	M_{43}	M_{44}	M_{45}	M_{46}	M_{47}	M_{48}
M_{49}	M_{50}	M_{51}	M_{52}	M_{53}	M_{54}	M_{55}	M_{56}
M_{57}	M_{58}	M_{59}	M_{60}	M_{61}	M_{62}	M_{63}	M_{64}

Bảng 2

M_{58}	M_{50}	M_{42}	M_{34}	M_{26}	M_{18}	M_{10}	M_2
M_{60}	M_{52}	M_{44}	M_{36}	M_{28}	M_{20}	M_{12}	M_4
M_{62}	M_{54}	M_{46}	M_{38}	M_{30}	M_{22}	M_{14}	M_6
M_{64}	M_{56}	M_{48}	M_{40}	M_{32}	M_{24}	M_{16}	M_8
M_{57}	M_{49}	M_{41}	M_{33}	M_{25}	M_{17}	M_9	M_1
M_{59}	M_{51}	M_{43}	M_{35}	M_{27}	M_{19}	M_{11}	M_3
M_{61}	M_{53}	M_{45}	M_{37}	M_{29}	M_{21}	M_{13}	M_5
M_{63}	M_{55}	M_{47}	M_{39}	M_{31}	M_{23}	M_{15}	M_7

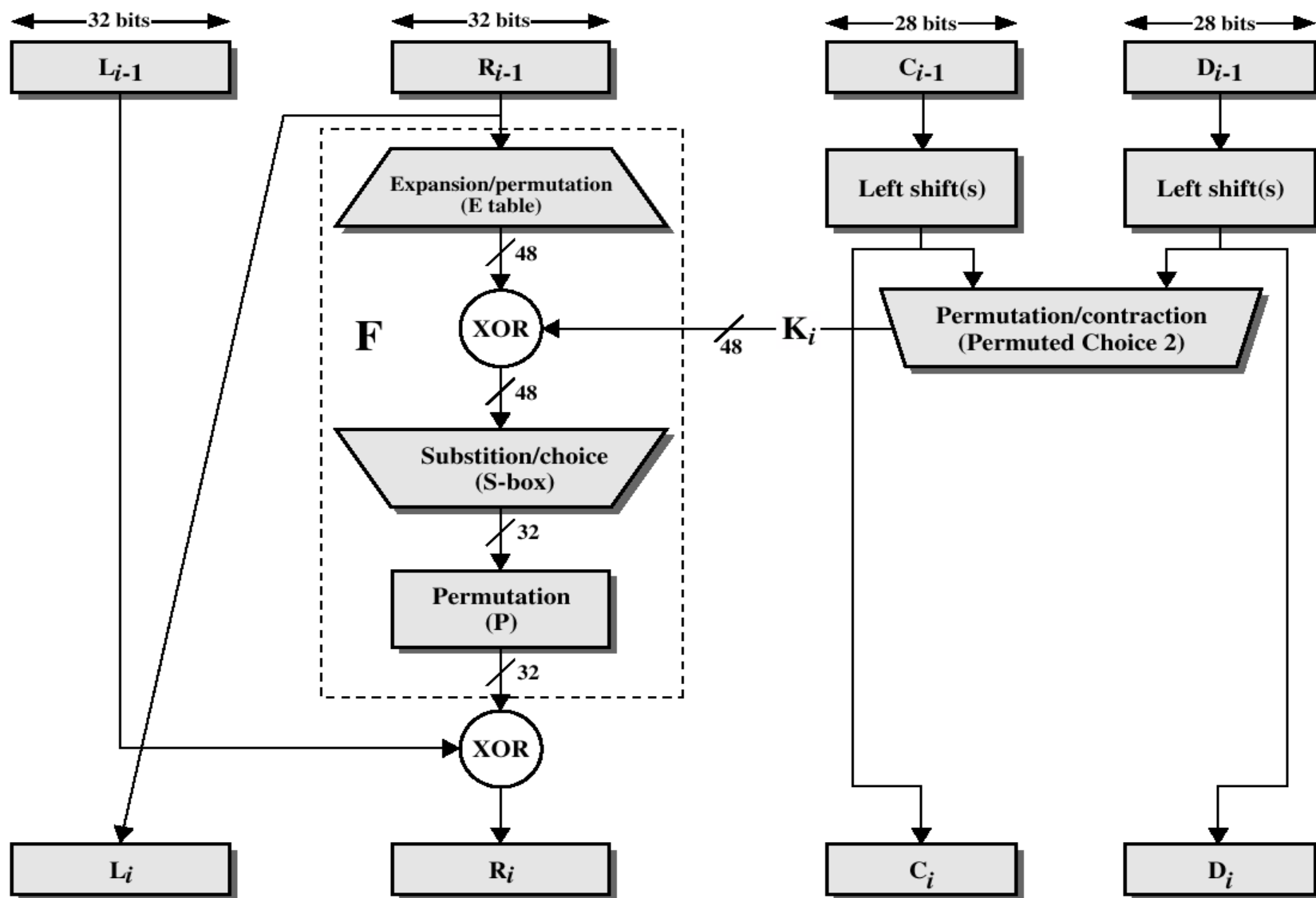


Figure 2.4 Single Round of DES Algorithm

Cấu tạo một vòng của DES

- Sử dụng hai nửa 32 bit trái và 32 bit phải
- Như đối với mọi mã Fiestel có thể biểu diễn

$$L_i = R_{i-1}$$

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

- Hàm F:
 - mở rộng R thành 48 bit nhờ hoán vị E
 - R 48 bit XOR khoá con 48bit
 - Qua 8 S-box để nhận được kết quả 32 bit
 - Đảo lần cuối sử dụng hoán vị 32 bit P

Hoán vị E và hoán vị P

(c) Expansion Permutation (E)

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

(d) Permutation Function (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

Cách thực hiện của S-box (8 box S_i)

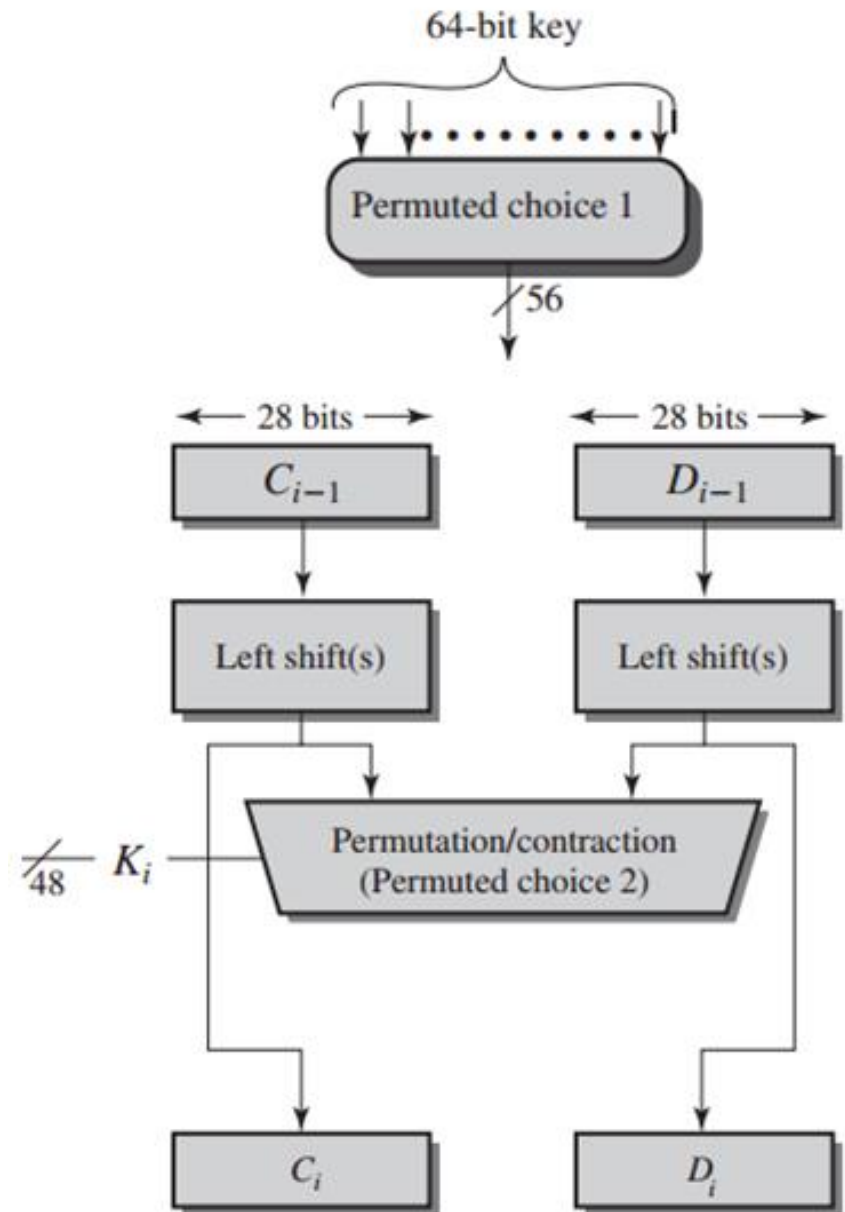
S_1	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

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

- 6-bit input qua mỗi S_i -> cho ra 4-bit output:
- Ví dụ:
 - với input 011001
 - => lấy hàng 01 = 1; lấy cột 1100 = 12
 - => được 9 = 1001
 - output 1001

Sinh khóa

- Đầu vào của khóa là khối 64 bit
=> bỏ 8 bit ở vị trí bội của 8
=> được key 56 bit
- Thực hiện phép hoán vị
Permuted Choice 1
- Chia thành 2 nửa 28 bit
- Thực hiện Left shift với mỗi nửa
dịch chuyển trái 1 hoặc 2 bit
dựa vào bảng shift cho mỗi
vòng
- Hoán vị choice 2 => 48 bit



(a) Input Key

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

(b) Permuted Choice One (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

(c) Permuted Choice Two (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

(d) Schedule of Left Shifts

Round Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits Rotated	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Một ví dụ của DES

Plaintext:	02468aceeca86420
Key:	0f1571c947d9e859
Ciphertext:	da02ce3a89ecac3b

Độ an toàn của DES

- **Sử dụng khóa 56-Bit** (có tất cả 2^{56} xấp xỉ $7.2 \cdot 10^{16}$ khóa)
 - 7/1998, Electronic Frontier Foundation thông báo đã phá vỡ một mã hóa DES bằng một chiếc máy tính "DES cracker" đặc biệt, được chế tạo với giá dưới 250.000 USD. Cuộc tấn công diễn ra **ít hơn ba ngày**.
- **Những cải tiến của DES**
 - DES được công nhận 7/1977 và được tái khẳng định trong 1983, 1988, 1993, và 1999.
 - Chuẩn FIPS 46-2 nâng cấp thành Triple DES.

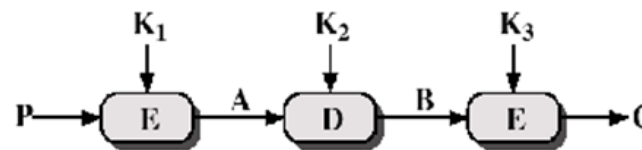
Triple DES với 2 khoá và 3 khóa

- Cần sử dụng 3 mã theo trình tự E-D-E, vậy có thể dùng 3 khoá khác nhau
- Nhưng có thể sử dụng 2 khoá theo trình tự:

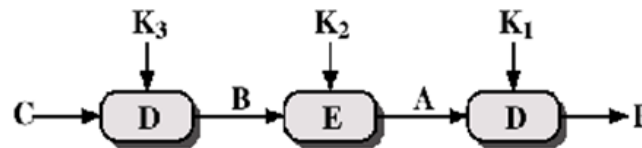
Triple DES với 2 khóa

$$C = E_{K1} (D_{K2} (E_{K1} (P)))$$

- Nếu $K1 = K2$ thì tương đương làm việc với 1 DES
- Triple DES với 3 khóa



(a) Encryption



(b) Decryption

Nội dung

- Nguyên lý mã đối xứng và mã khối
- Mã khối Feistel
- Các thuật toán mã khối đối xứng
 - Chuẩn mã dữ liệu DES
 - Chuẩn mã nâng cao AES
- Mã dòng RC4

Chuẩn mã hóa nâng cao (Advanced Encryption Standard - AES)

- AES được NIST công bố năm 2001.
 - chống lại các tấn công đã biết
 - Đơn giản trong thiết kế
- AES sử dụng kích thước khối 128-bit và kích thước khóa 128, 192 hoặc 256 bit.
- Trong AES:
 - Các phép toán số học gồm phép cộng, nhân và chia được thực hiện trong trường hữu hạn $GF(2^8)$.

Trường hữu hạn dạng $GF(2^8)$

- Trường hữu hạn $GF(2^8)$ là tập Z_2^8 có trang bị các phép toán cộng và nhân được định nghĩa:

- Một số nguyên $a \in [0, 2^8 - 1]$, $a = a_7a_6a_5a_4a_3a_2a_1a_0$, sẽ tương đương với một đa thức có dạng:

$$f(x) = a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

- **ĐN Phép cộng:** $(f + g)(x) = \sum_{i=0}^7 ((a_i + b_i) \bmod 2) x^i$

- **ĐN Phép nhân:**

$$(f \times g)(x) = \left[\sum_{i=0}^7 \sum_{j=0}^7 ((a_i \times b_j) \bmod 2) x^{i+j} \right] [\bmod m(x)]$$

- đa thức tối giản $m(x) = x^8 + x^4 + x^3 + x + 1$

Ví dụ : Thực hiện phép cộng hai số {57} và {83} trong $GF(2^8)$.

{57} = 0101.0111 tương đương $f(x) = x^6 + x^4 + x^2 + x + 1$

{83} = 1000.0011 tương đương $g(x) = x^7 + x + 1$.

$$\begin{aligned} f(x) + g(x) &= x^6 + x^4 + x^2 + x + 1 + x^7 + x + 1 \\ &= x^7 + x^6 + x^4 + x^2 \text{ tương đương } 1101.0100 = \{D4\} \end{aligned}$$

$$\rightarrow \{57\} + \{83\} = \{D4\}$$

Ví dụ : Thực hiện phép nhân hai số {57} và {83} trong $GF(2^8)$.

$$\begin{aligned} f(x) \times g(x) &= x^{13} + x^{11} + x^9 + x^8 + x^7 + x^7 + x^5 + x^3 + x^2 + x + x^6 + x^4 + x^2 + x + 1 \\ &= x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1 \end{aligned}$$

Thực hiện $f(x) \times g(x) \bmod m(x)$

$$(x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1) \bmod (x^8 + x^4 + x^3 + x + 1)$$

$$= x^7 + x^6 + 1 \text{ tương đương } 1100.0001 = \{C1\}$$

$$\rightarrow \{57\}^* \{83\} = \{C1\}$$

$x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$	$x^8 + x^4 + x^3 + x + 1$
$x^{13} + x^9 + x^8 + x^6 + x^5$	$x^5 + x^3$
$x^{11} + x^4 + x^3 + 1$	
$x^{11} + x^7 + x^6 + x^4 + x^3$	
$x^7 + x^6 + 1$	

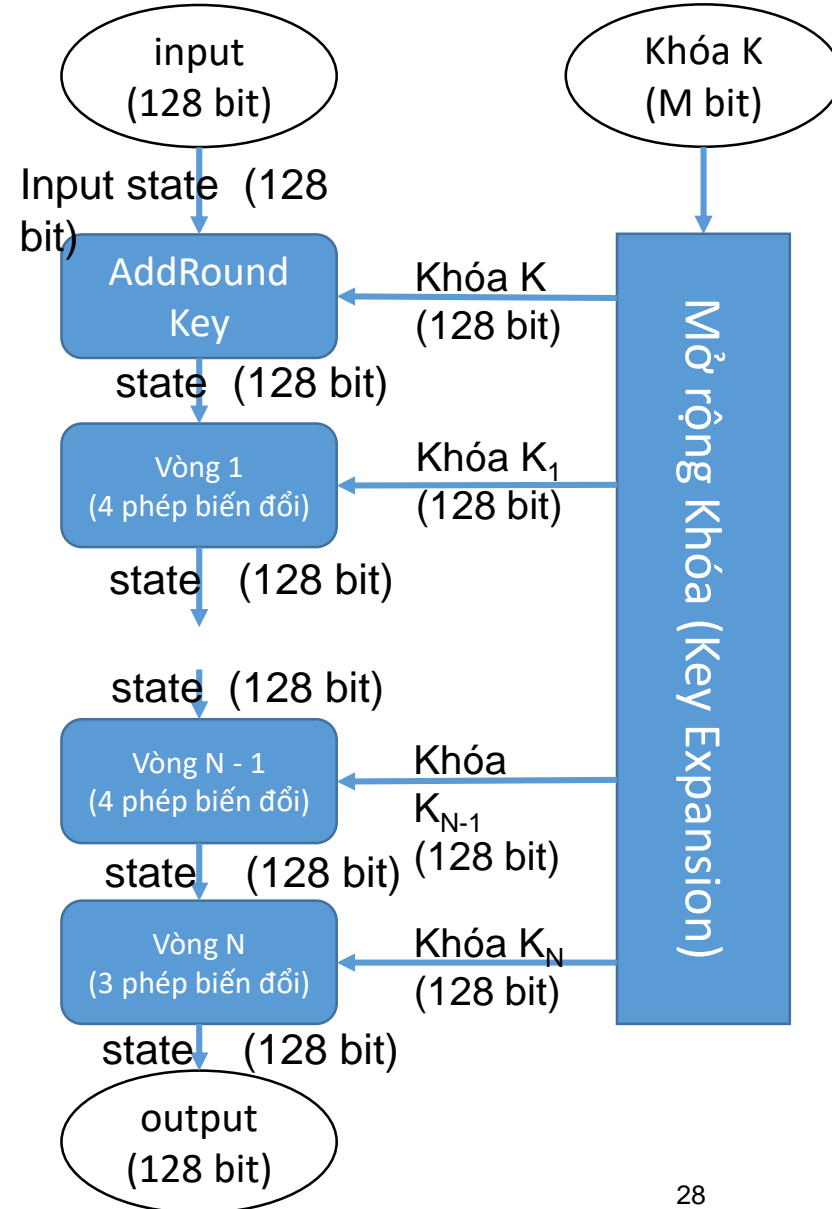
Cấu tạo chuẩn mã nâng cao AES

Chi tiết một vòng lặp
(từ vòng 1 đến $N - 1$)

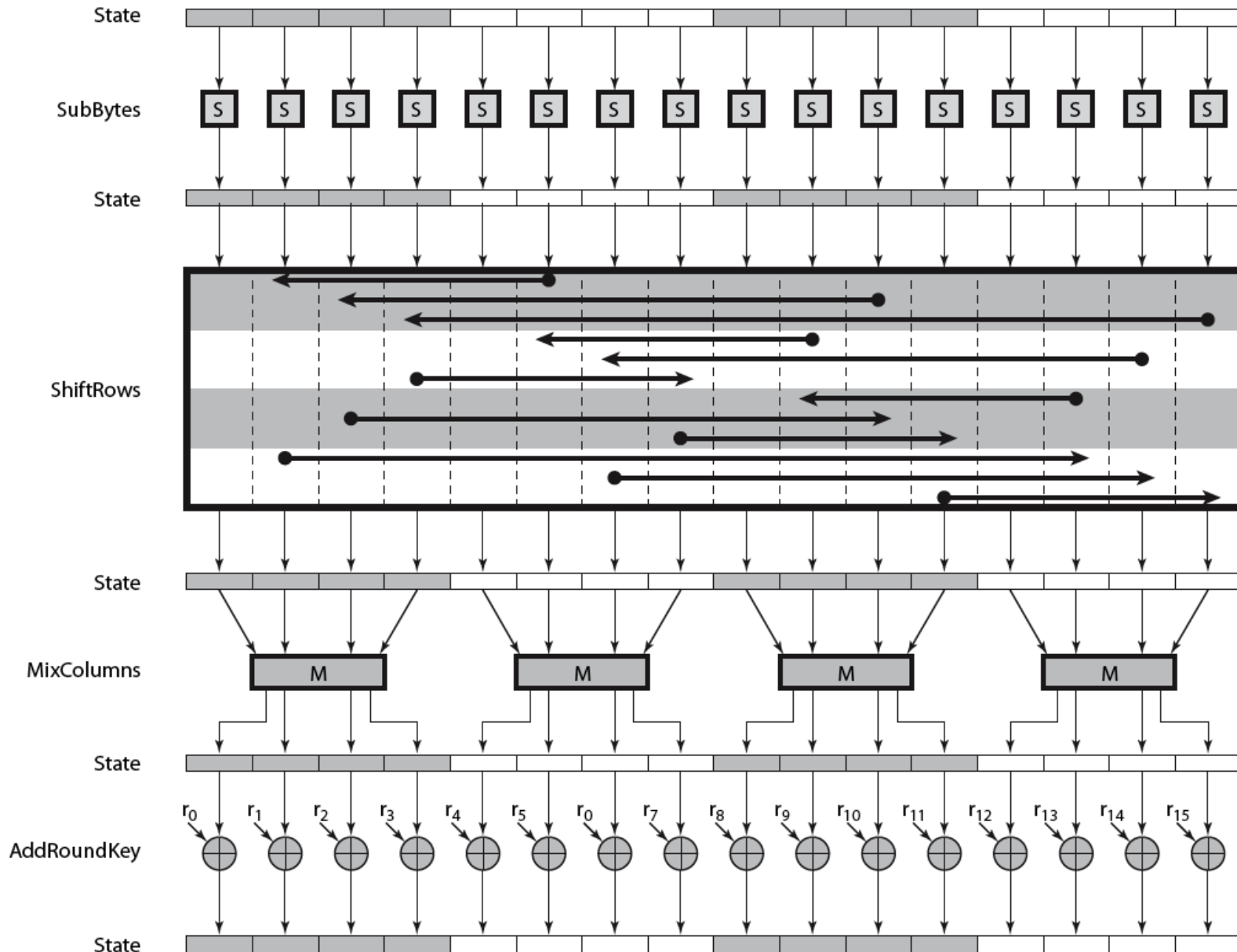
1. **Substitute bytes**
2. **ShiftRows**
3. **MixColumns**
4. **AddRoundKey**

Riêng vòng thứ N không có phép **MixColumns**.

Số vòng lặp (N)	10	12	14
Khóa (bit)	128	192	256
Input (bit)	128	128	128
Khóa vòng lặp (bit)	128	128	128
Khóa mở rộng (bytes)	176	208	240



Mỗi vòng AES xử lý từng nhóm 4 byte



Ví dụ một bản mã AES -128

Plain text	0123456789abcdeffedcba9876543210
Key	0f1571c947d9e8590cb7add6af7f6798
cipher text	ff0b844a0853bf7c6934ab4364148fb9

01 23 45 67 . 89 ab cd ef . fe dc ba 98 . 76 54 32 10


Phép *SubBytes*

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

- Phép ***SubBytes*** thay thế mỗi byte trong state bằng 1 byte trong bảng **S-box**.
- Ví dụ:
 - Byte {95} được thay thế thành {2A} (giá trị tại hàng 9, cột 5 của bảng S-box)
 - $\text{SubBytes}(\{95\}) = \{2A\}$
 - $\text{SubByte}(\{59\}) = \{CB\}$

Phép ShiftRows

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6



87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

1. Hàng đầu không thay đổi.
2. Hàng hai, dịch vòng trái 1-byte.
3. Hàng ba, dịch vòng trái 2 byte.
4. Hàng tư, dịch vòng trái 3-byte.

Phép MixColumns

- **MixColumns** được định nghĩa bằng phép nhân ma trận sau

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

$$\begin{aligned} s'_{0,j} &= (2 \cdot s_{0,j}) \oplus (3 \cdot s_{1,j}) \oplus s_{2,j} \oplus s_{3,j} \\ s'_{1,j} &= s_{0,j} \oplus (2 \cdot s_{1,j}) \oplus (3 \cdot s_{2,j}) \oplus s_{3,j} \\ s'_{2,j} &= s_{0,j} \oplus s_{1,j} \oplus (2 \cdot s_{2,j}) \oplus (3 \cdot s_{3,j}) \\ s'_{3,j} &= (3 \cdot s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \cdot s_{3,j}) \end{aligned}$$

- Các phép toán thực hiện trong GF(2⁸)

- Ví dụ:

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

➔

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

Minh họa cho phép Mixcolumn

$$\begin{aligned}
 (\{02\} \cdot \{87\}) \oplus (\{03\} \cdot \{6E\}) \oplus \{46\} &\oplus \{A6\} &= \{47\} \\
 \{87\} \oplus (\{02\} \cdot \{6E\}) \oplus (\{03\} \cdot \{46\}) \oplus \{A6\} &= \{37\} \\
 \{87\} \oplus \{6E\} \oplus (\{02\} \cdot \{46\}) \oplus (\{03\} \cdot \{A6\}) &= \{94\} \\
 (\{03\} \cdot \{87\}) \oplus \{6E\} \oplus \{46\} \oplus (\{02\} \cdot \{A6\}) &= \{ED\}
 \end{aligned}$$

For the first equation, we have $\{02\} \cdot \{87\} = (0000\ 1110) \oplus (0001\ 1011) = (0001\ 0101)$ and $\{03\} \cdot \{6E\} = \{6E\} \oplus (\{02\} \cdot \{6E\}) = (0110\ 1110) \oplus (1101\ 1100) = (1011\ 0010)$. Then,

$$\begin{aligned}
 \{02\} \cdot \{87\} &= 0001\ 0101 \\
 \{03\} \cdot \{6E\} &= 1011\ 0010 \\
 \{46\} &= 0100\ 0110 \\
 \{A6\} &= \underline{1010\ 0110} \\
 &0100\ 0111 = \{47\}
 \end{aligned}$$

Chú ý : Nhân $\{a\}$ với $\{02\}$ thực hiện bằng cách dịch trái 1 bit. Nếu bit bên trái nhất của số $\{a\} = 1$ thì sau khi dịch trái XOR với $(0001\ 1011 \leftrightarrow x^4 + x^3 + x + 1)$

Phép AddRoundKey

- Trong **phép biến đổi cộng khóa vòng thuận**, được gọi là **AddRoundKey**, 128 bit của **State** được thực hiện phép XOR bitwise với 128 bit của khóa vòng.
- **Phép biến đổi cộng khóa vòng ngược** cũng chính là phép cộng khóa vòng thuận, bởi phép XOR tự nó đảo ngược.
- Dưới đây là một ví dụ của AddRoundKey:

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

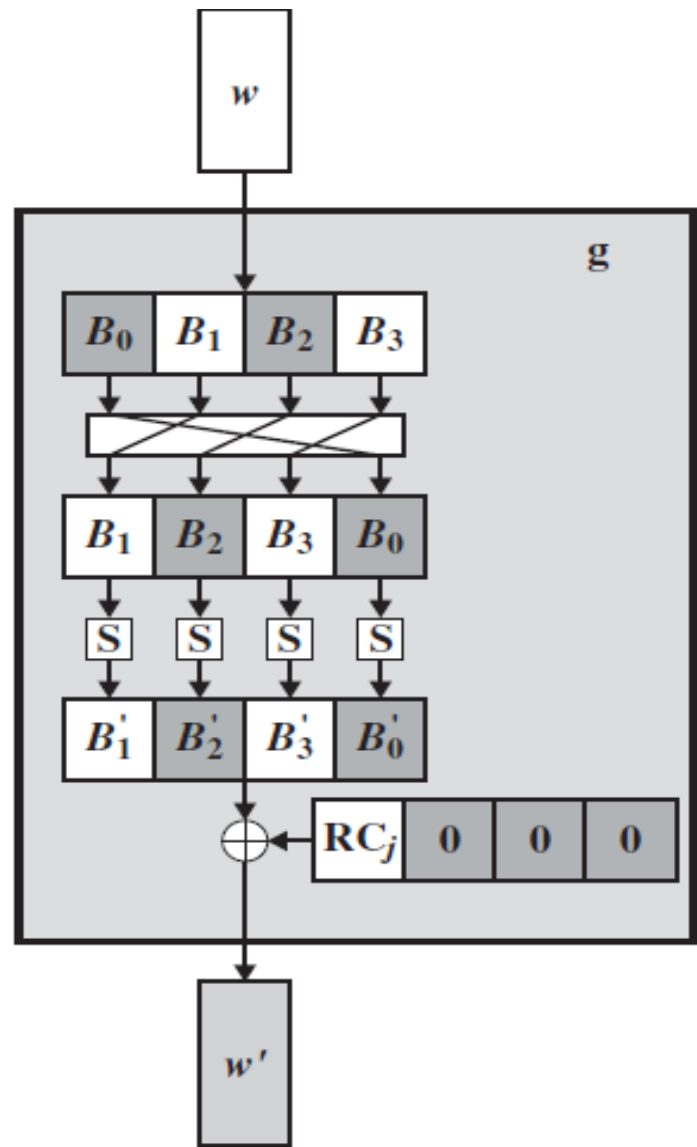
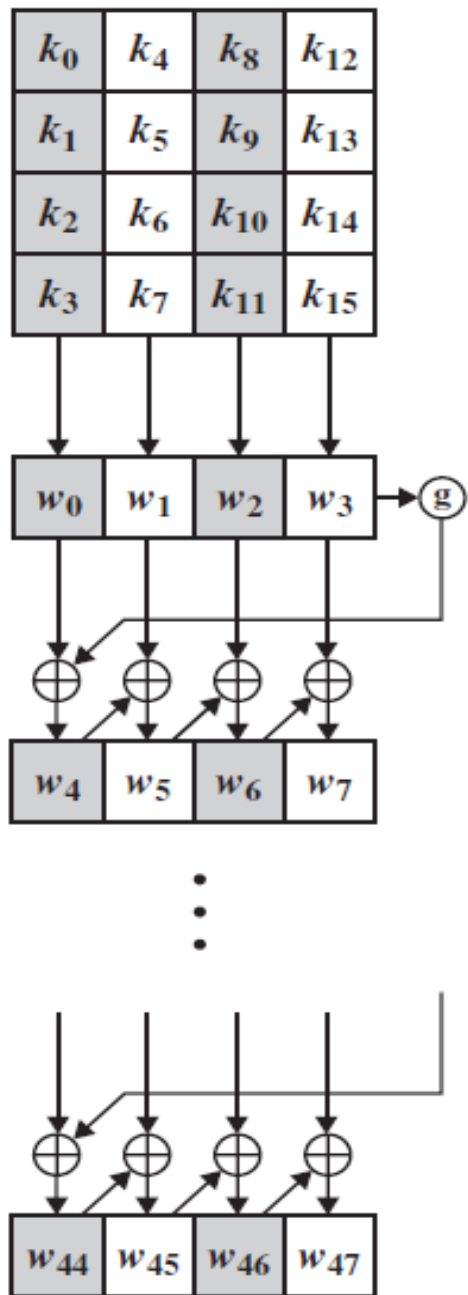
 \oplus

AC	19	28	57
77	FA	D1	5C
66	DC	29	00
F3	21	41	6A

 $=$

EB	59	8B	1B
40	2E	A1	C3
F2	38	13	42
1E	84	E7	D6

Mô hình mở rộng khóa



(b) Function g

Giải mã cho Key Expansion

```
KeyExpansion (byte key[16], word w[44])
{
    word temp
    for (i = 0; i < 4; i++)    w[i] = (key[4*i], key[4*i+1],
                                       key[4*i+2],
                                       key[4*i+3]);

    for (i = 4; i < 44; i++)
    {
        temp = w[i - 1];
        if (i mod 4 = 0)    temp = SubWord (RotWord (temp))
                                $\oplus$  Rcon[i/4];

        w[i] = w[i-4]  $\oplus$  temp
    }
}
```

Mở rộng khóa AES

- **Rcon** là một word (4 bytes):
- $Rcon[j] = (RC[j], 0, 0, 0)$, với $RC[1] = 1$, $RC[j] = 2 * RC[j-1]$ với phép nhân được định nghĩa trên trường $GF(2^8)$.
- Các giá trị của $RC[j]$ trong cơ số 16 là:

j	1	2	3	4	5	6	7	8	9	10
RC[j]	01	02	04	08	10	20	40	80	1B	36

- Ví dụ:
- giả sử rằng khóa vòng cho vòng lặp 8 ($i=32 \rightarrow 35$) là
EA D2 73 21 B5 8D BA D2 31 2B F5 60 7F 8D 29 2F
- 4 byte đầu tiên (w đầu tiên) của khóa vòng cho vòng lặp 9 được tính như sau:

i (decimal)	temp	After RotWord	After SubWord	Rcon (9)	After XOR with Rcon	w[i- 4]	w[i] = temp \oplus w[i- 4]
36	7F8D292F	8D292F7F	5DA515D2	1B000000	46A515D2	EAD27321	AC7766F3

Nhận xét về AES

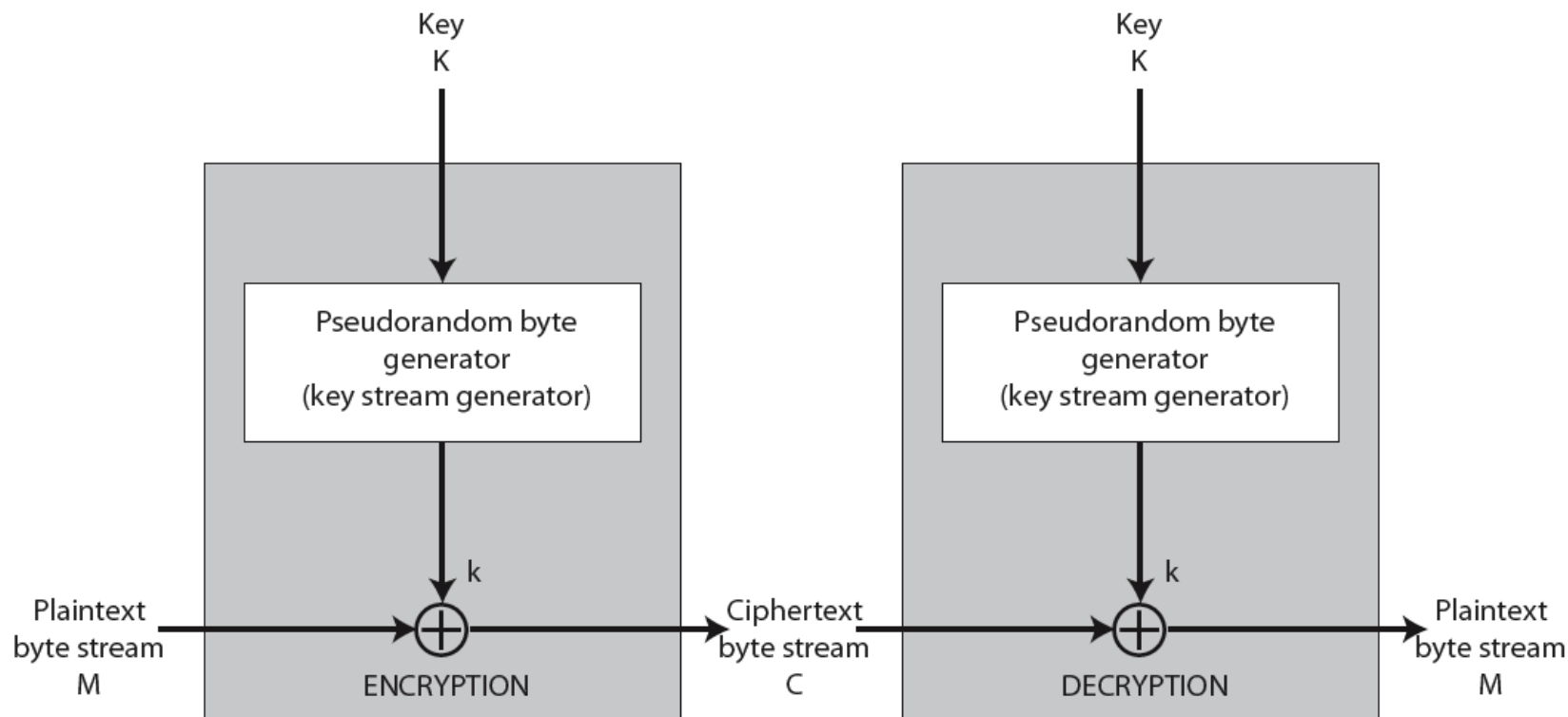
- Thay thế DES trong nhiều ứng dụng
- Không sử dụng kiến trúc của Feistel
- Đảm bảo tính dễ phân tích nhưng an toàn

Key Size (bits)	Number of Alternative Keys	Time required at 1 decryption/ μ s	Time required at 10^6 decryptions/ μ s
32	$2^{32} = 4.3 \times 10^9$	$2^{31} \mu\text{s} = 35.8 \text{ minutes}$	2.15 milliseconds
56 (DES)	$2^{56} = 7.2 \times 10^{16}$	$2^{55} \mu\text{s} = 1142 \text{ years}$	10.01 hours
128 (AES)	$2^{128} = 3.4 \times 10^{38}$	$2^{127} \mu\text{s} = 5.4 \times 10^{24} \text{ years}$	$5.4 \times 10^{18} \text{ years}$
168 (Triple DES)	$2^{168} = 3.7 \times 10^{50}$	$2^{167} \mu\text{s} = 5.9 \times 10^{36} \text{ years}$	$5.9 \times 10^{30} \text{ years}$
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26} \mu\text{s} = 6.4 \times 10^{12} \text{ years}$	$6.4 \times 10^6 \text{ years}$

Nội dung

- Nguyên lý mã đối xứng và mã khối
- Mã khối Feistel
- Các thuật toán mã khối đối xứng
 - Chuẩn mã dữ liệu DES
 - Chuẩn mã nâng cao AES
- Mã dòng RC4

Mã dòng



- Xử lý plain text lần lượt theo từng byte (hoặc bit)
- Mỗi byte được mã hóa bởi khóa riêng

Mã dòng

- Khóa K được đưa vào **bộ sinh số giả ngẫu nhiên** để tạo ra dòng khóa (keystream)

$$X_{n+1} = (aX_n + c) \bmod m$$

{7, 17, 23, 1, 7, etc.}

{5, 25, 29, 17, 21, 9, 13, 1, 5, etc.}

- Thực hiện XOR khóa với bản rõ theo từng bit -> bản mã

Mã hóa		
input	\oplus	11001100
Khóa		01101100
<hr/>		
Bản mã		10100000

Giải mã		
Bản mã	\oplus	10100000
Khóa		01101100
<hr/>		
Bản giải mã		11001100

- Mã hóa và giải mã dùng chung 1 dòng khóa

Yêu cầu và đặc điểm của mã dòng

- Yêu cầu:

1. Chuỗi số giả ngẫu nhiên được tạo ra để làm khóa phải có chu kỳ lặp lớn
2. Chuỗi giả ngẫu nhiên nên gần với ngẫu nhiên, tần suất các số nên tương đối như nhau
3. Khóa K đầu vào nên có độ dài đủ lớn (ít nhất là 128 bit)

- Ưu điểm: Đơn giản, an toàn và tốc độ nhanh

- Nhược điểm: Không dùng lại khóa được

Cipher	Key Length	Speed (Mbps)
DES	56	9
3DES	168	3
RC2	Variable	0.9
RC4	Variable	45

Mã dòng RC4

- RC4 là một hệ mật mã dòng được thiết kế năm 1987 bởi Ron Rivest.
- Thuật toán được dựa trên việc sử dụng một hoán vị ngẫu nhiên.
- RC4 đơn giản, nhưng hiệu quả, có nhiều cỡ khóa khác nhau.
- RC4 được sử dụng trong SSL/TLS (chuẩn giao tiếp giữa trình duyệt Web và Server),
- thuật toán bảo mật cho mạng không dây WEP). Khóa thực hiện hoán vị ngẫu nhiên cả 8 giá trị bit. Sử dụng hoán vị đó để khuấy thông tin đầu vào được xử lý từng byte.

Sinh khoá RC4

- Bắt đầu từ mảng S với biên độ: 0..255
- Sử dụng khoá để xáo trộn đều thực sự
- S tạo trạng thái trong của mã

```
for i = 0 to 255 do
    S[i] = i
    T[i] = K[i mod keylen]
j = 0
for i = 0 to 255 do
    j = (j + S[i] + T[i]) (mod 256)
    swap (S[i], S[j])
```

Mã RC4

- Dùng khóa K có chiều dài từ 1 -> 256 byte để tạo vectơ S = (S[0], S[2], ..., S[255])
- Mã tiếp tục hoán vị các giá trị của mảng
- Tổng của các cặp trộn cho giá trị khoá dòng k từ hoán vị
- XOR S[t] (khóa k) với byte tiếp theo của bản tin để mã/giải mã

$i = j = 0$

for each message byte M_i

$i = (i + 1) \pmod{256}$

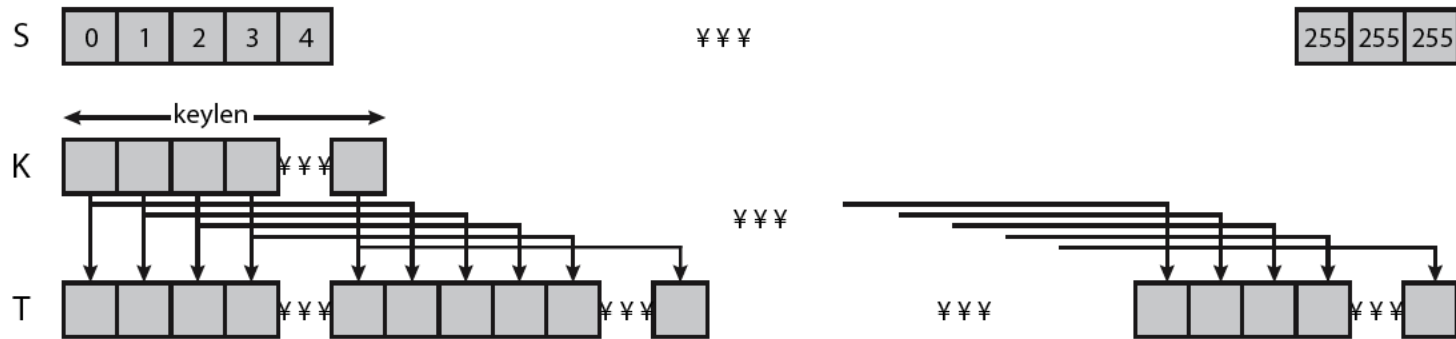
$j = (j + S[i]) \pmod{256}$

swap(S[i], S[j])

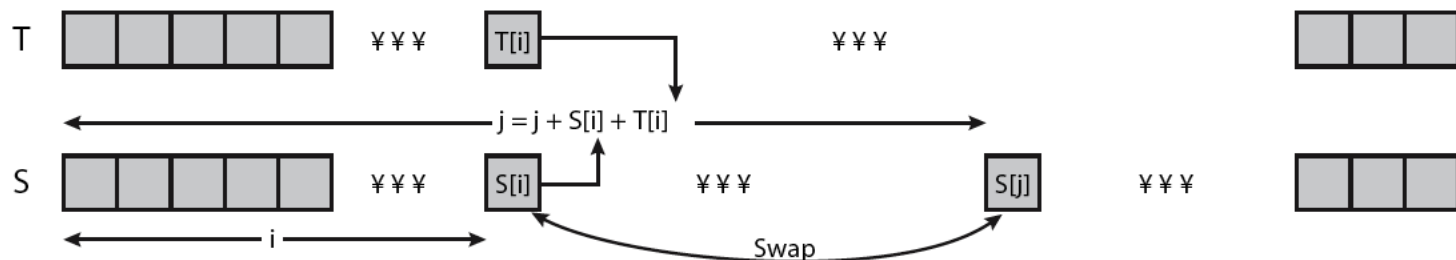
$t = (S[i] + S[j]) \pmod{256}$

$C_i = M_i \text{ XOR } S[t]$

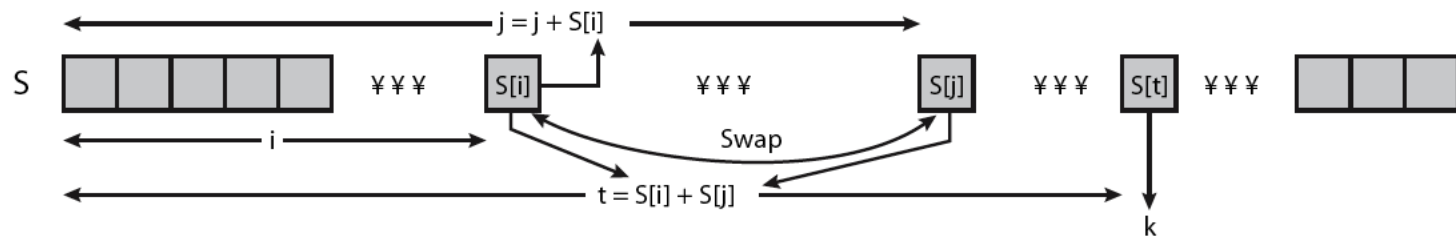
Tổng quan RC4



(a) Initial state of S and T



(b) Initial permutation of S



(c) Stream Generation

2.5 Các kiểu thao tác mã khối

- Mã khối mã các block có kích thước cố định
- Chẳng hạn DES mã các block 64 bit với khoá 56 bit
- Cần phải có cách áp dụng vào thực tế vì các thông tin cần mã có kích thước tùy ý.
- Có 4 cách được định nghĩa cho DES theo chuẩn ANSI

ANSI X3.106-1983 Modes of Use

- Bây giờ có 5 cách cho DES và AES
- Có kiểu khối và dòng

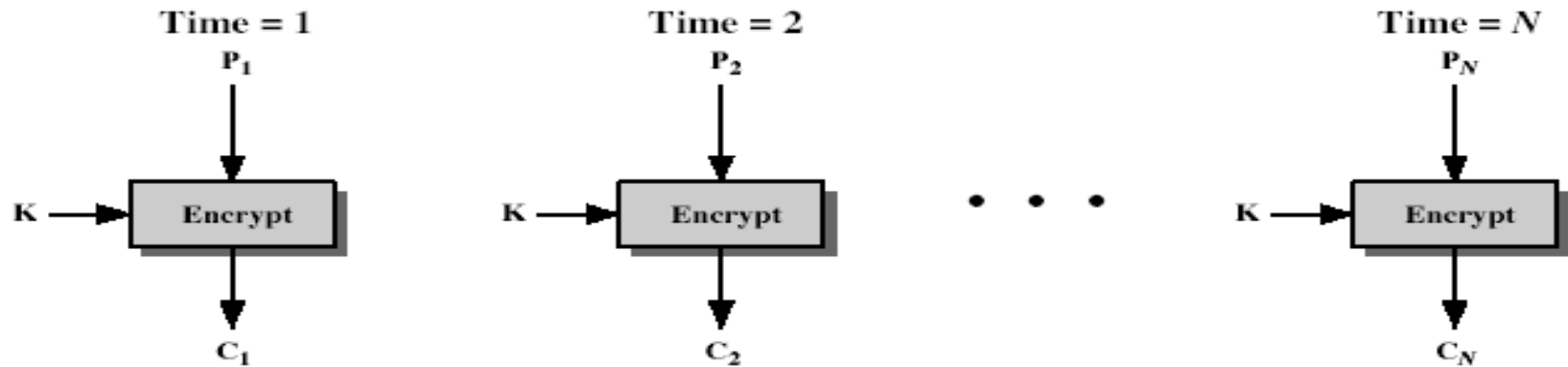
Sách mật mã điện tử (ECB)

- Mẫu tin được chia thành các khối độc lập, sau đó mã từng khối
- Mỗi khối là giá trị cần thay thế như dùng sách mã, do đó có tên như vậy
- Mỗi khối được mã độc lập với các mã khác

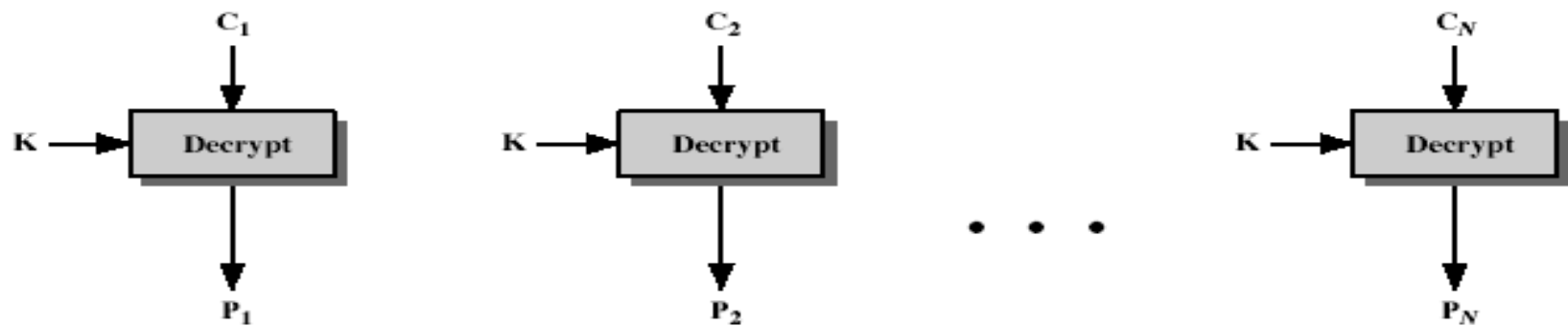
$$C_i = \text{DES}_{K1}(P_i)$$

- Khi dùng: truyền an toàn từng khối riêng lẻ
- *Câu hỏi: tại sao gọi là sách điện tử? Lặp trên bản rõ có tạo nên lặp trên bản mã không?*
- *Trả lời: vì giống như khi đã biết khóa, ta mã và giải mã từng khối bằng cách tra sách mã*

Sách mật mã điện tử (ECB)



(a) Encryption



(b) Decryption

Dãy chuyền mã khối (CBC)

- Các mẫu tin được chia thành các khối, sẽ khắc phục việc lặp trên dữ liệu tạo ra việc lặp trên mã
- Nhưng chúng được liên kết với nhau trong quá trình mã hoá
- Các block được sắp thành dãy, vì vậy có tên như vậy
- Sử dụng véctơ ban đầu IV để bắt đầu quá trình

$$C_i = \text{DES}_{K1}(P_i \text{ XOR } C_{i-1})$$

$$C_{-1} = \text{IV}$$

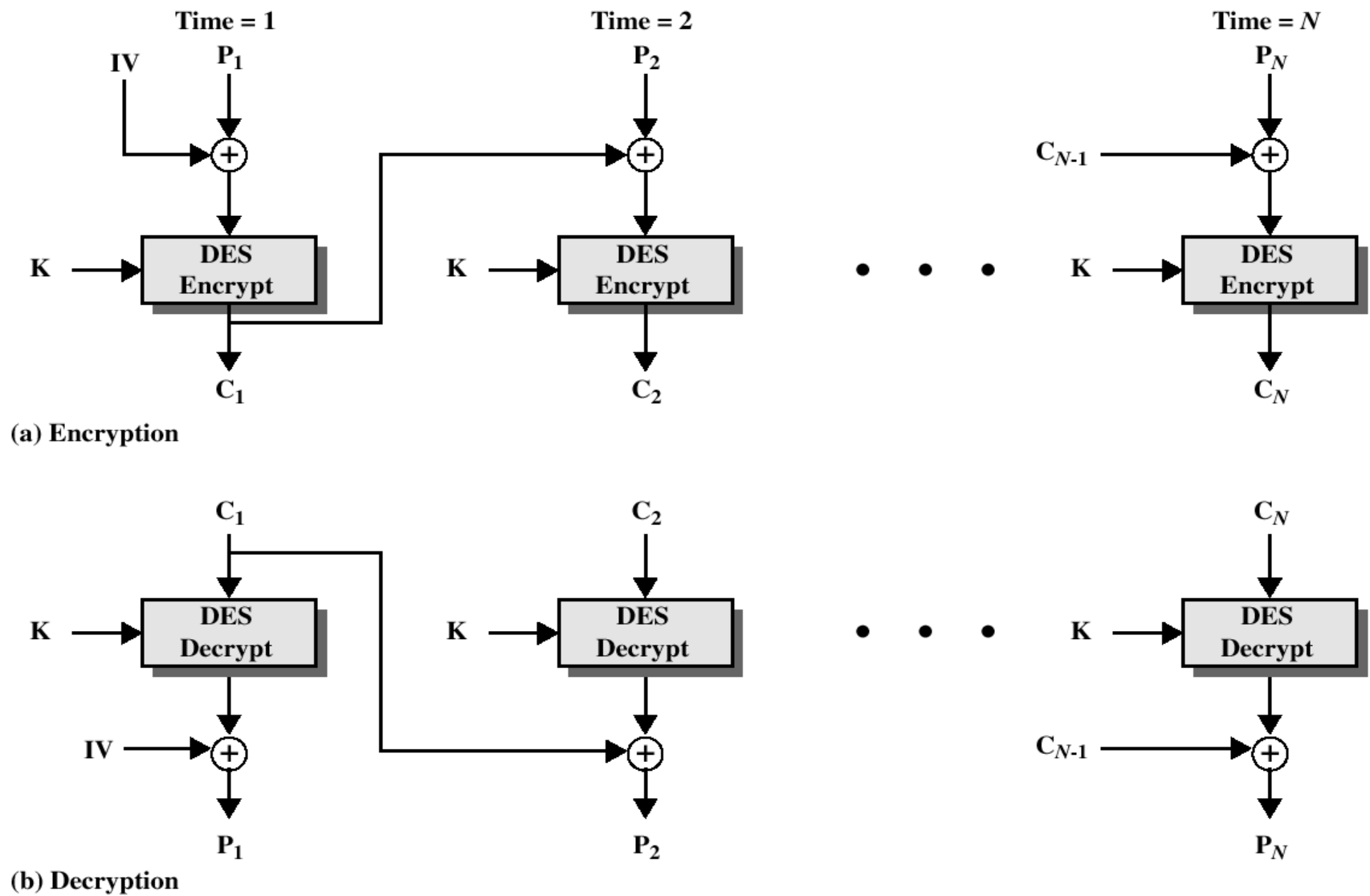


Figure 2.7 Cipher Block Chaining (CBC) Mode

Phản hồi ngược đầu ra (OFB)

- Mẫu tin xem như dòng bit
- Đầu ra của mã được bổ sung cho mẫu tin
- Đầu ra do đó là phản hồi, do đó có tên như vậy
- Phản hồi ngược là độc lập đối với bản tin
- Có thể được tính trước

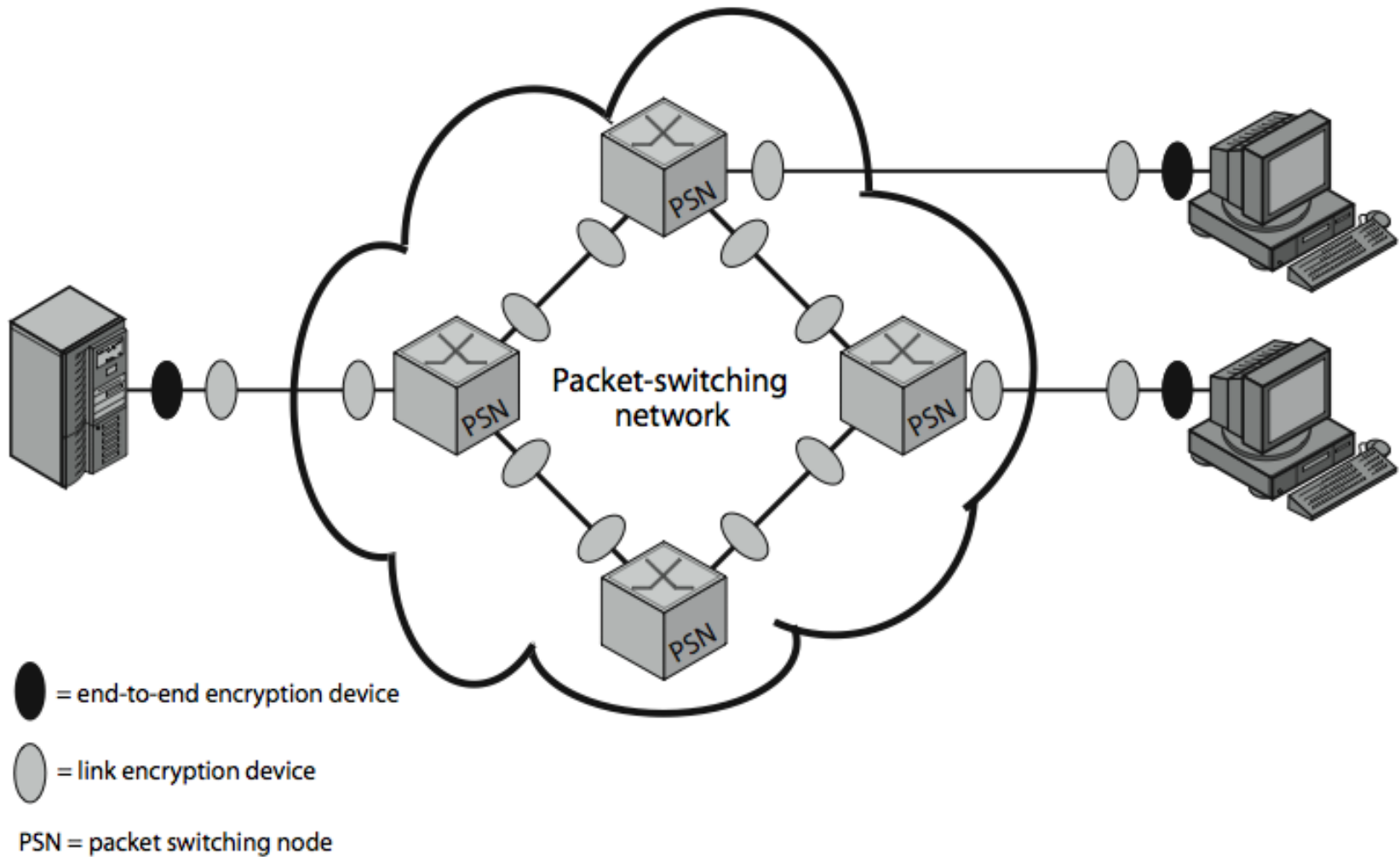
$$\begin{aligned}C_i &= P_i \text{ XOR } O_i \\O_i &= \text{DES}_{K1}(O_{i-1}) \\O_{-1} &= \text{IV}\end{aligned}$$

- Được dùng cho mã dòng trên các kênh âm thanh
- *Câu hỏi: tại sao dùng như mã dòng được?*
- *Trả lời: vì ta có thể thỏa thuận chọn trước số s bit tùy ý cho một lần mã, giống như trong mã dòng.*

2.6 Chỗ đặt mã

- Có hai phương pháp chính xác định chỗ đặt mã trên mô hình mạng
- Mã kết nối (Link Encryption)
 - Mã xảy ra độc lập trên mỗi kết nối.
 - Suy ra cần phải giải mã truyền tin giữa các kết nối
 - Đòi hỏi nhiều thiết bị và các cặp khoá
- Mã đầu cuối (End to end Encryption) AES là mã mới thay thế
 - Mã xảy ra giữa điểm gốc và điểm đích
 - Cần thiết bị tại mỗi đầu cuối và khoá chia sẻ

Chỗ đặt mã



Chỗ đặt mã

- Khi dùng mã đầu cuối cần phải để thông tin đầu của nó rõ ràng, vì như vậy mạng mới định hướng đúng dẫn thông tin
- Vì vậy tuy nội dung tin được bảo vệ, nhưng khuôn dòng tin truyền thì không
- Lý tưởng là muốn bí mật cả hai
- Mã đầu cuối bảo vệ nội dung thông tin trên cả đường truyền và cung cấp danh tính
- Mã kết nối bảo vệ luồng truyền khỏi việc theo dõi.

Chỗ đặt mã

- Có thể đặt mã ở nhiều tầng khác nhau trong mô hình OSI
- Mã kết nối thực hiện ở tầng 1 hoặc 2
- Mã đầu cuối có thể thực hiện ở tầng 3, 4, 6, 7 (tùy sự tương thích của kiến trúc)
- Dịch chuyển đến tầng càng cao, càng ít thông tin được mã hoá, nhưng càng an toàn hơn tuy nhiên phức tạp hơn với nhiều đối tượng và khoá hơn.
- *Câu hỏi: tại sao lại có câu khẳng định trên?*
- *Trả lời: Vì tại giao thức ở mỗi tầng, cần bổ sung thông tin tham số cho đầu mỗi gói tin ở dạng tường minh để trao đổi giữa bên gửi và bên nhận cùng tầng*

Tóm tắt

- Khái niệm mã đối xứng
- Cấu trúc mã khối Feistel
- Chuẩn mã dữ liệu DES và các chế độ mã
- Triple DES và chuẩn mã nâng cao
- Mã dòng
- Chỗ đặt mã: mã link và mã đầu cuối

Câu hỏi trắc nghiệm 1

- Câu 1: Mục nào không phải là thành phần của Khóa đối xứng
 - A. Một khóa chia sẻ người gửi và người nhận
 - B. Hai thuật toán mã hóa và giải mã
 - C. Bản rõ và bản mã
 - D. Thuật toán nén văn bản
- Câu 2: Nói chung coi độ khó thám mã không phụ thuộc vào
 - A. Độ phức tạp của thuật toán mã hóa
 - B. Độ lớn của không gian khóa
 - C. Che giấu thuật toán mã hóa
 - D. Che giấu khóa mật

Câu hỏi trắc nghiệm 2

- Câu 3: Thao tác xử lý dữ liệu sau nào không dùng trong mã đối xứng:
 - A. Dùng phép thế xâu ký tự bản rõ bằng xâu ký tự bản mã
 - B. Dùng phép hoán vị đảo chỗ các ký tự bản rõ tạo ra bản mã
 - C. Kết hợp cả hai phép toán trên và có thể xử lý nhiều vòng
 - D. Che giấu dữ liệu trong môi trường khác
- Câu 4: Trong mã hóa DES điều khẳng định nào là không đúng
 - A. Mã khối với mỗi khối 64 bit, khóa có độ dài 56 bit
 - B. Hoán vị đầu, nghịch đảo cuối và lặp 16 vòng
 - C. Khóa dùng chung cho 16 vòng
 - D. Mỗi vòng: hoán vị 2 nửa, xử lý một nửa dùng phép thế qua hộp và cộng khóa con

Câu hỏi trắc nghiệm 3

- **Câu 5. Trong thuật toán mã AES điều khẳng định nào là không đúng:**
 - A. Có 128/192/256 bit khoá và 128 bit dữ liệu
 - B. Có 9/11/13 vòng, mỗi vòng: thế byte, dịch hàng, trộn cột, cộng khóa
 - C. Mỗi vòng: chia 2 nửa, đảo chỗ và xử lý một nửa
 - D. Xử lý dữ liệu như 4 nhóm của 4 byte
- **Câu 6. Các chế độ làm việc của DES. Khẳng định nào sau đây là sai**
 - A. ECB: khối mã trước quay vòng tác động vào khối mã sau
 - B. CBC: khối mã trước cộng nhị phân với khối bản tin sau rồi mã
 - C. CFB: bản tin như dòng bit cộng nhị phân đầu ra của mã, rồi phản hồi
 - D. OFB: đầu ra mã phản hồi và cộng nhị phân với dòng bit của bản tin

Câu hỏi trắc nghiệm 4

- **Câu 7. Thành phần của mã dòng không bao gồm:**
 - A. Khóa chia sẻ người gửi và người nhận
 - B. Bộ sinh số giả ngẫu nhiên dưới tác động của khóa
 - C. Bộ sinh các khóa con từ khóa chính
 - D. Bộ cộng loại trừ bit giữa dữ liệu với dãy giả ngẫu nhiên
- **Câu 8. Mã đối xứng không đặt ở vị trí đặt nào:**
 - A. Mã đầu cuối ở máy tính đầu - cuối bảo vệ thông tin nội dung trên cả đường truyền và cung cấp danh tính, tầng 3, 4, 6, 7
 - B. Mã kết nối đặt ở máy tính đầu cuối và mạng chuyển gói tin bảo vệ luồng truyền khỏi việc theo dõi, tầng 1, 2
 - C. Kết hợp cả hai dạng trên
 - D. Mã đầu cuối ở tầng 5 - tầng phiên

Đáp án câu hỏi trắc nghiệm

- Câu 1

- D, Khóa, thuật toán mã hóa, giải mã, bản rõ bản mã đều là các thành phần của mã đối xứng, nên không dùng đến.

- Câu 2

- C, luôn coi thuật toán mã hoá là mọi người đều biết

- Câu 3

- D, Che giấu dữ liệu không phải là thao tác mã hóa, mà là giấu sự tồn tại của dữ liệu mật trong môi trường nào đó

- Câu 4

- C, trong 16 vòng, mỗi vòng dùng một khóa con riêng

Đáp án câu hỏi trắc nghiệm - tiếp

- Câu 5
 - C, mỗi vòng xử lý cả khối 128 bit, tức là cả 4 nhóm - mỗi nhóm 4 byte
- Câu 6
 - A, ECB là chế độ sách mã, tức là mã các khối độc lập
- Câu 7
 - C, Không cần sinh khóa con vì không có vòng lặp
- Câu 8
 - D, Tầng phiên kiểm soát hội thoại giữa các máy tính, nên thông thường không dùng mã ở tầng này.

Glossary - Từ điển thuật ngữ

- Bản rõ là bản tin gốc.
- Bản mã là bản tin gốc đã được mã hoá.
- Mã là thuật toán chuyển bản rõ thành bản mã
- Khoá là thông tin dùng để mã hoá, chỉ có người gửi và người nhận biết.
- Mã hoá chuyển bản rõ thành bản mã
- Giải mã chuyển bản mã thành bản rõ.
- Mật mã nghiên cứu các nguyên lý và phương pháp mã hoá.
- Thám mã nghiên cứu các nguyên lý và phương pháp giải mã mà không biết khoá.
- Lý thuyết mã bao gồm cả mật mã và thám mã

Glossary - Từ điển thuật ngữ - tiếp

- Mã đối xứng là mã ở đó hai người nhận và gửi chia sẻ chung một khóa.
- DES - chuẩn mã dữ liệu là mã khối 64 bit, khóa 56 bit
- 3DES là cải tiến của DES, dùng lặp 3 lần DES với 2 hoặc 3 khóa
- AES - chuẩn mã nâng cao là mã khối 128 bit, khóa 128 bit thay thế DES.
- EBC: sách mã điện tử - mã riêng biệt từng khối mã với cùng một khóa
- CBC: dây chuyền mã khối - khối mã trước cộng nhị phân với khối bản tin sau rồi mã
- CFB: mã phản hồi ngược - bản tin như dòng bit cộng nhị phân đầu ra của mã, rồi phản hồi
- OFB: phản hồi đầu ra - đầu ra mã phản hồi và cộng nhị phân với dòng bit của bản tin

FAQ – Câu hỏi thường gặp

1. **Nêu sự khác biệt giữa mã thế và mã hoán vị**
2. **Thế nào là mã đối xứng mạnh. Nó cần có các tính chất gì?**
3. **Mô tả kiến trúc mã đối xứng Fiestel**
4. **Nêu các đặc trưng của DES. Trong mỗi vòng nó thực hiện các thao tác gì trên dữ liệu?**
5. **Để tăng cường an ninh cho DES người ta đưa ra các giải pháp gì?**
6. **Nêu các đặc trưng của AES. Trong mỗi vòng nó thực hiện các thao tác gì trên dữ liệu?**

FAQ – Câu hỏi thường gặp (tiếp)

7. Phân biệt mã khối và mã dòng

8. Nêu các chế độ thao tác trên mã khối
9. Mô tả thao tác mã dòng RC4
10. Có những vị trí nào đặt mã đối xứng trên mô hình mạng?
11. Chức năng nhiệm vụ của mã đầu cuối? Ưu, nhược điểm
12. Chức năng nhiệm vụ của mã kết nối? Vị trí đặt, ưu, nhược điểm

Trả lời câu hỏi:

1. Mã thế là thay mỗi ký tự bản rõ bằng 1 xâu bản mã; mã hoán vị đảo thứ tự các ký tự trong bản rõ để tạo nên bản mã.
2. Mã đối xứng mạnh cần có các tính chất sau:
 - Kích thước khối dữ liệu mã và khóa tương đối lớn: cân bằng với tốc độ
 - Thuật toán mã hóa mạnh: lặp nhiều vòng, mỗi vòng kết hợp hoán vị với thế; thuật toán sinh khóa còn phức tạp cho mỗi vòng. Bản mã có tính chất khuếch tán và tác dụng đồng loạt để khó thám mã
3. Kiến trúc mã khối Feistel
 - Lặp nhiều vòng, sinh khóa con riêng cho từng vòng
 - Quá trình giải mã ngược lại với quá trình mã hóa
 - Cân đối việc tăng kích thước khối, khóa và số vòng để đảm bảo an ninh với tốc độ thực hiện
4. Các đặc trưng của DES:
 - Mã khối 64 bit, khóa 56 bit, 16 vòng
 - Mỗi vòng chia 2 nửa, đảo hai nửa, xử lý nửa phải bằng cách hoán vị, mở rộng, cộng khóa vòng, thế nhờ các hộp box và lại hoán vị
 - Khó thám mã, cài đặt phần mềm, phần cứng

Trả lời câu hỏi – (tiếp)

5. Ban đầu đưa ra giải pháp 3DES – mã 3 lần 2 khóa hoặc 3 khóa. Sau này xây dựng Chuẩn mã nâng cao để tăng tốc độ xử lý.
6. Chuẩn mã nâng cao AES: dùng sơ đồ Fiestel, khối dữ liệu 128 bit, khóa 128/192/256, vòng 9/11/13. Mỗi vòng thực hiện các thao tác: thế byte, dịch hàng, trộn cột, cộng khóa vòng. An toàn tương đương 3DES, tốc độ nhanh hơn.
7. Mã khối thao tác trên từng khối dữ liệu; Mã dòng thao tác trên từng bit hoặc từng byte. Mã dòng thường dùng trên các tầng mạng thấp.
8. Có 5 chế độ thao tác mã khối như DES và AES: sách mã điện tử ECB, dây mã khối dây chuyền mã khối CBC, mã phản hồi ngược CFB, phản hồi ngược đầu ra OFB và Bộ đếm CTR
9. Dùng mảng S gồm 256 số tự nhiên đầu tiên và mảng T lập từ khóa K. Sau đó trộn S nhờ T và đảo S dựa vào chính nó.
10. Mã đầu cuối đặt ở các máy chủ và mã kết nối đặt ở mạng chuyển gói tin
11. Mã đầu cuối bảo vệ nội dung thông tin trên cả đường truyền và cung cấp danh tính
12. Mã kết nối bảo vệ luồng truyền khỏi việc theo dõi, có thể đặt ở nhiều tầng mạng, tầng càng cao, càng ít thông tin được mã hoá,