

BÀI GIẢNG AN TOÀN & BẢO MẬT THÔNG TIN CHƯƠNG 4-MÃ KHỐI HIỆN ĐẠI

TS. NGUYỄN ĐÌNH DƯƠNG
BỘ MÔN KHMT - KHOA CÔNG NGHỆ THÔNG TIN

Email: duongnd@utc.edu.vn

Ngày 03/07/2022

Nội dung

Mã khối hiện đại

- 1.1 Mở đầu
- 1.2 Chuẩn mã dữ liệu DES
- 1.3 Triple DES
- 1.4 Một số bài tập thực hành

Trao đổi

Nội dung

Mã khối hiện đại

- 1.1 Mở đầu
- 1.2 Chuẩn mã dữ liệu DES
- 1.3 Triple DES
- 1.4 Một số bài tập thực hành

Trao đổi

1. Mã khôi hiện đại

1. 1. Mở đầu

- Các hệ mã cổ điển có đặc điểm chung là từng kí tự của bản rõ được mã hoá tách biệt → thám mã dễ dàng hơn.
- Khắc phục: **từng khôi kí tự của bản rõ được mã hoá cùng một lúc** (xem như 1 đơn vị mã hoá) → **mã khôi**
- Trong mã khôi, các tham số quan trọng là **kích thước mỗi khôi** và **kích thước khoá**
- Điều kiện an toàn của mã khôi:
 - Kích thước khôi phải đủ lớn để chống lại phương án tấn công bằng phương pháp thông kê → thời gian mã hoá tăng.
 - Chiều dài khoá phải đủ lớn để chống lại phương pháp tấn công brute-force,khoá phải đủ ngắn để việc tạo khoá, phân phối và lưu trữ được dễ dàng.
- Khi thiết kế mã khôi, phải đảm bảo 2 yêu cầu:
 - **Sự hỗn loạn** (Confusion): sự phụ thuộc giữa bản rõ và bản mã phải phức tạp, tốt nhất là phụ thuộc *kiểu phi tuyến*.
 - **Sự khuếch tán** (Diffusion): mỗi bit của bản rõ và khoá có ảnh hưởng lên càng nhiều bit của bản mã càng tốt.

1. Mã khối hiện đại

1. 1. Mở đầu

- **Giải pháp:**

- Sự hỗn loạn được tạo ra bằng kỹ thuật thay thế (Substitution, S-box)
- Sự khuếch tán được tạo ra bằng kỹ thuật hoán vị (Permutation, P-box)
⇒ Mã khối kết hợp đồng thời 2 kỹ thuật trên → Substitution-Permutation Network (SPN).

- **Đặc điểm chung của các hệ mã khối:**

- Bản rõ được chia thành khối dữ liệu (thường ở dạng xâu bit) có kích thước khác nhau (tối thiểu 64 bit)
- Khoá của hệ cũng là một xâu bit có độ dài cố định (56 bit với DES, các hệ mã khác là 128,256,512 bit)
- Quá trình mã hoá và giải mã được thực hiện qua một số lần lặp, mỗi lần sử dụng một khoá con (được sinh ra từ khoá chính).

1. Mã khối hiện đại

1. 2. Chuẩn mã dữ liệu DES

- Cuối những năm 1960, IBM phát triển mã Lucifer, được lãnh đạo bởi Fiestel. Ban đầu Lucifer sử dụng khối dữ liệu 64 bit và khóa 128 bit.
- NBS (sau này là NIST - Viện chuẩn và công nghệ Quốc gia) đã dàn xếp với IBM để thuật toán này thành miễn phí và phát triển nó thành chuẩn mã hoá dữ liệu (DES), công bố ngày 15/02/1977.
- DES là mã khối với mỗi khối dữ liệu 64 bit và dùng khóa dài 56 bit. Nó được sử dụng rộng rãi và đã được tranh luận kỹ về mặt an toàn.

1. Mã khôi hiện đại

1. 2. Chuẩn mã dữ liệu DES

Mã Feistel

Ví dụ 1.1

Plaintext: $P = 011110100001$

Hàm $F(\cdot)$ thực hiện một thuật toán thay thế, $F(xyz) = zxy$

① Mã hoá:

- *Bước 1:* Chia bản rõ thành 2 nửa $L_0 = 011110$ và $R_0 = 100001$
- *Bước 2:* Mã hoá R_0 bởi $f(\cdot)$ thành $E = F(R_0) = 010100$
- *Bước 3:* Đặt $L_1 = R_0 = 100001$ và $R_1 = L_0 \oplus E = 001010$
- *Bước 4:* Ghép L_1 và R_1 nhận được bản mã $C = 100001001010$.

② Giải mã:

- *Bước 1:* Chia bản mã C thành 2 nửa
- *Bước 2:* $R_0 = L_1$, $L_0 = R_1 \oplus E$, với $E = F(R_0)$
- *Bước 3:* Ghép L_0 và R_0 nhận được bản rõ P .

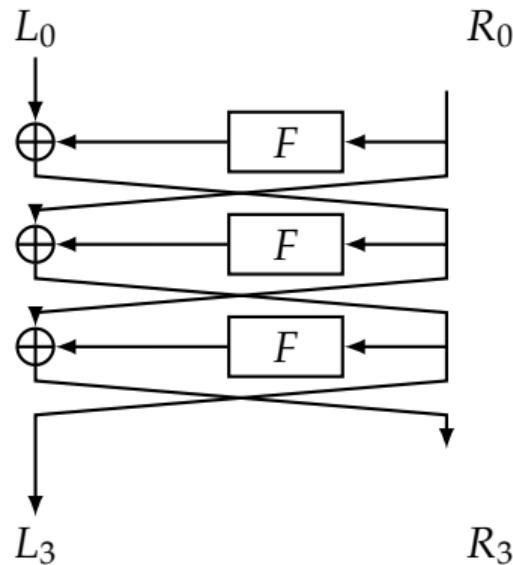


1. Mã khối hiện đại

1. 2. Chuẩn mã dữ liệu DES

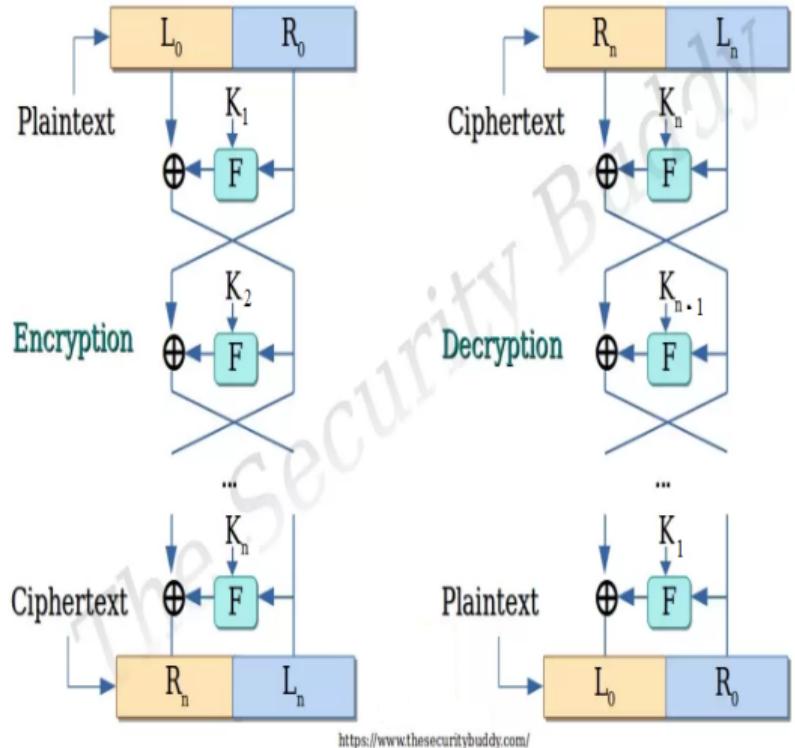
Ưu điểm của mã Feistel

- độ an toàn cao hơn các hệ mã cổ điển
- so với các hệ mã tích **thay thế - hoán vị** khác, thao tác mã hóa/giải mã luôn luôn thực hiện được
- F không cần khả nghịch
- thao tác mã hóa và giải mã giống hệt nhau → kích thước của code hoặc mạch cần thiết để thực hiện hệ mã giảm gần một nửa.





1. Mã khối hiện đại



1. 2. Chuẩn mã dữ liệu DES

Mạng Feistel tổng quát: sử dụng hàm F (hàm vòng) nhận hai đầu vào (một khối dữ liệu và một khóa con) và trả về một đầu ra có cùng kích thước với khối dữ liệu.

- Sử dụng khoá K sinh n khoá con K_1, K_2, \dots, K_n
- Xử lý qua n vòng, ở mỗi vòng lặp
 - khối dữ liệu được chia thành 2 nửa (trái L_i , phải R_i)
 - Áp dụng phép thay thế nên phần trái bằng cách $L_{i-1} \oplus F(R_{i-1}, K_i)$, phần phải giữ nguyên
 - Hoán vị hai nửa cho nhau ($L_i \leftrightarrow R_i$)

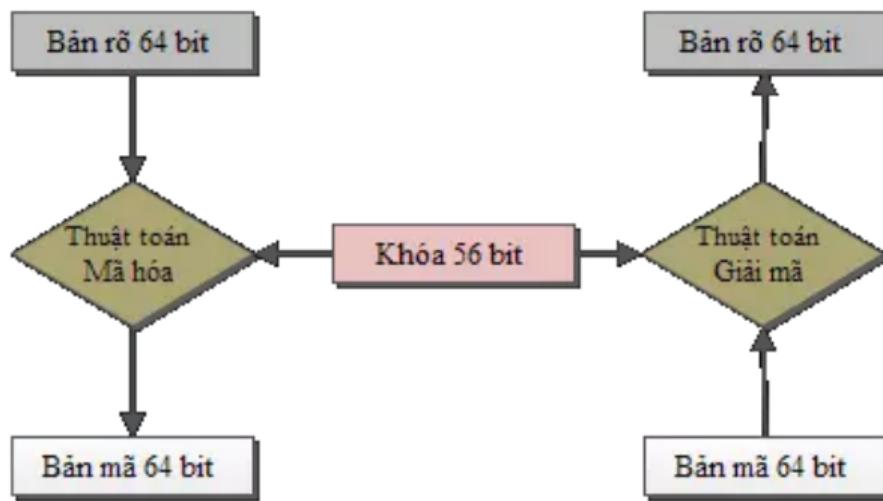
$$\begin{cases} L_i = R_{i-1} \\ R_i = L_{i-1} \oplus F(R_{i-1}, K_i) \end{cases}$$



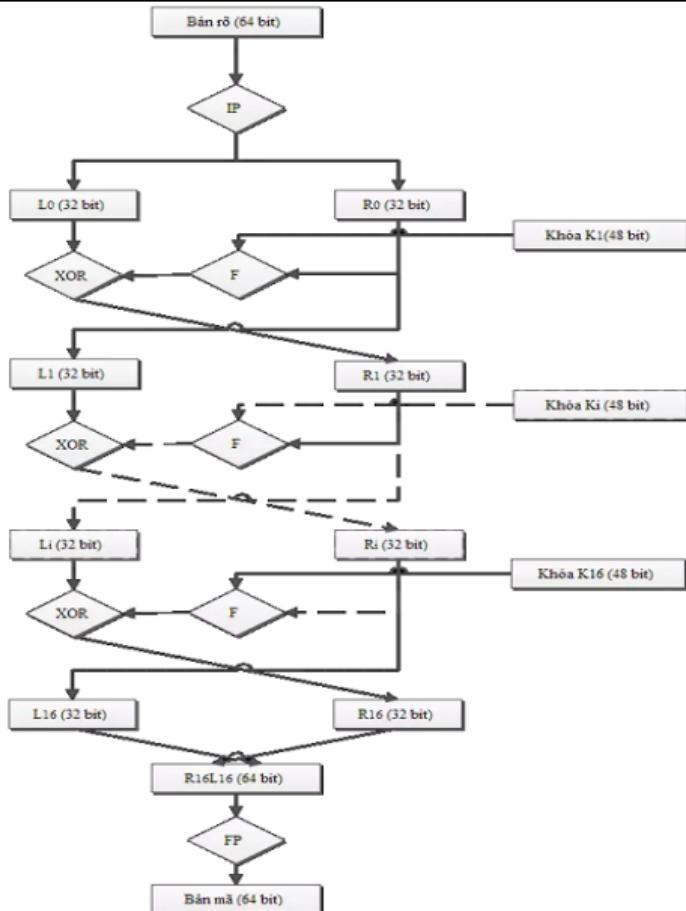
1. Mã khối hiện đại

1. 2. Chuẩn mã dữ liệu DES

Thuật toán mã hóa DES



- **Input:** Bản rõ 64 bit $M = m_1m_2 \dots m_{64}$
- **Output:** Bản mã 64 bit $C = c_1c_2 \dots c_{64}$
- **Khoa:** $K = k_1k_2 \dots k_{64}$ (thực ra 56 bit, các bit ở vị trí chia hết cho 8 sử dụng để kiểm tra tính chẵn lẻ)



- Sinh khoá con: sử dụng thuật toán sinh ra 16 khoá con 48 bit K_1, K_2, \dots, K_{16} từ K
- Sử dụng IP (Initial Permutation) để hoán vị các bit của M , kết quả nhận được chia thành 2 nửa:
 $L_0 = m_{63}m_{62} \dots m_{32}$ và $R_0 = m_{31}m_{30} \dots m_0$.
- Thuật toán thực hiện 16 vòng lặp:

$$\begin{cases} L_i = R_{i-1} \\ R_i = L_{i-1} \oplus F(R_{i-1}, K_i), F(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i)) \end{cases}$$

E : hàm hoán vị mở rộng xâu 32 bit \rightarrow 48 bit;

S : hàm thay thế xâu 48 bit \rightarrow 32 bit;

P : hàm hoán vị khác.

- Hoán vị 2 khối L_{16}, R_{16} ta được $R_{16}L_{16} = b_1b_2 \dots b_{64}$.
- Bản mã $C = FP = IP^{-1}(b_1b_2 \dots b_{64})$
- Chú ý:** Hai hoán vị IP và IP^{-1} không có ý nghĩa gì về mặt mật mã, chỉ tạo điều kiện cho "chip hoá" thuật toán DES.

1. Mã khối hiện đại

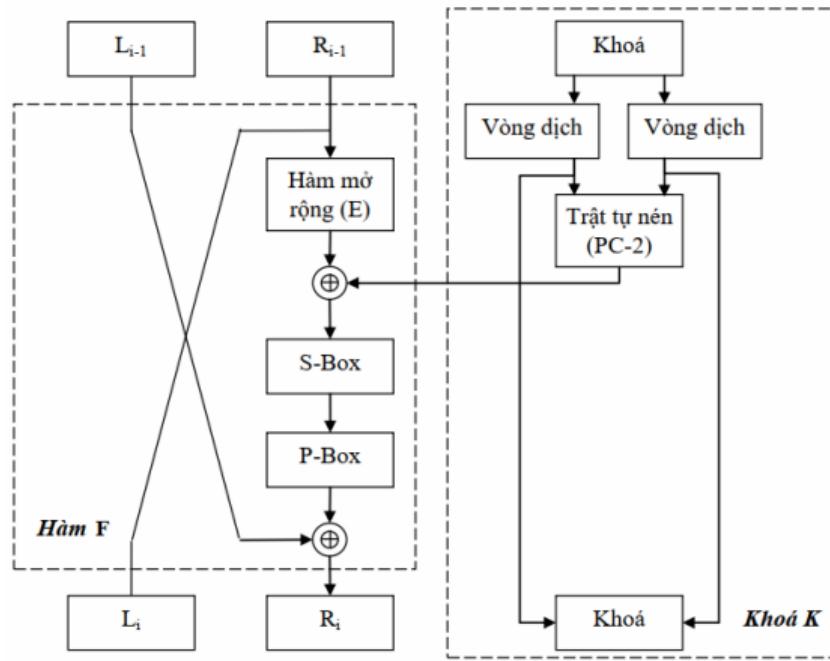
1. 2. Chuẩn mã dữ liệu DES

Sơ đồ 1 vòng lặp

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

trong đó

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

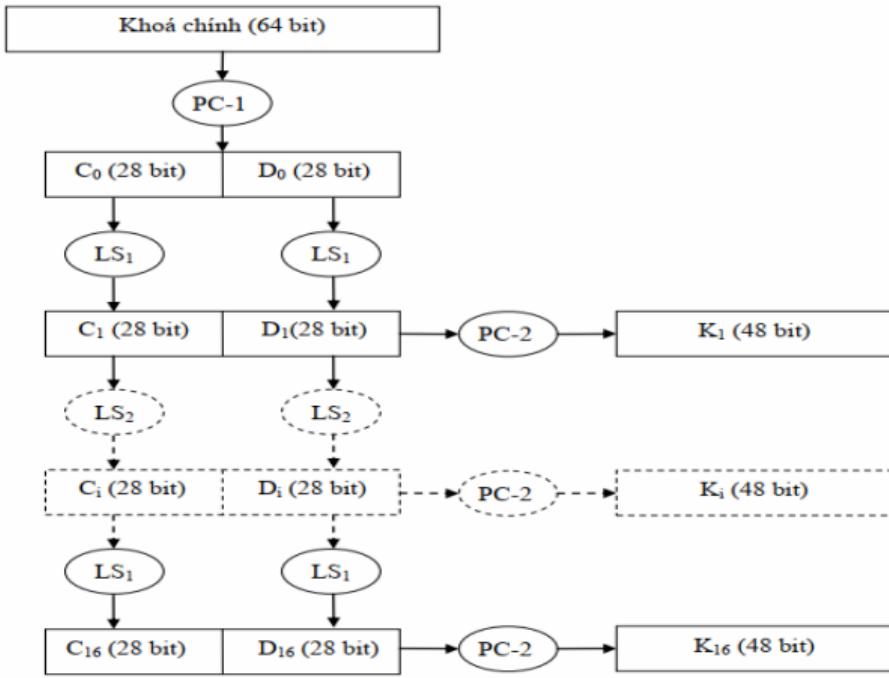




1. Mã khối hiện đại

1. 2. Chuẩn mã dữ liệu DES

Sơ đồ tạo khoá con



- **Input:** $K = k_1k_2 \dots k_{64}$ (bao gồm 8 bit kiểm tra tính chẵn lẻ)
- **Ouput:** 16 khoá con 48 bit K_i , $1 \leq i \leq 16$.

Thuật toán tạo khoá con

- Ban đầu, bỏ 8 bit ở các vị trí chia hết cho 8: $k_8, k_{16}, \dots, k_{64} \rightarrow 56$ bit. Trích lấy 48 bit nhờ 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

- Theo bảng trên: $C_0 = k_{57}k_{49} \dots k_{36}$, $D_0 = k_{63}k_{55} \dots k_4$

Ví dụ 1.2

Cho $K = 133457799BBCDFF1$ (hexadecimal key).

$K = 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001$

$\xrightarrow{\text{PC-1}} 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111$

$\rightarrow C_0 = 1111000 0110011 0010101 0101111, D_0 = 0101010 1011001 1001111 0001111$



1. Mã khôi hiện đại

1. 2. Chuẩn mã dữ liệu DES

Thuật toán tạo khoá con

- Dịch trái từng phần độc lập: $\begin{cases} C_i = LS_i(C_{i-1}) \\ D_i = LS_i(D_{i-1}) \end{cases}, \quad 1 \leq i \leq 16.$
- LS_i là phép dịch bit vòng (cyclic shift) sang trái **1 hoặc 2 vị trí** tùy thuộc vào i .

Vòng lặp	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Số bit dịch trái	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

C_{n-1} or D_{n-1}

Single left shift



Two left shift

1. Mã khối hiện đại

1. 2. Chuẩn mã dữ liệu DES

Thuật toán tạo khoá con

Vòng lặp	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Số bit dịch trái	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Ví dụ 1.3

C_0 :	1111000011001100101010101111	D_0 :	0101010101100110011110001111
C_1 :	1110000110011001010101011111	D_1 :	1010101011001100111100011110
C_2 :	1100001100110010101010111111	D_2 :	0101010110011001111000111101
C_3 :	0000110011001010101011111111	D_3 :	0101011001100111100011110101
...
C_{14}	1111111000011001100101010101	D_{14}	1110101010101100110011110001
C_{15}	1111100001100110010101010111	D_{15}	1010101010110011001111000111
C_{16}	1111000011001100101010101111	D_{16}	0101010101100110011110001111

Thuật toán tạo khoá con

- Tiếp theo, sử dụng PC-2 để thu được khoá K_i (48 bit)

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

Ví dụ 1.4

- Từ $C_1D_1 = 1110000\ 1100110\ 0101010\ 1011111\ 1010101\ 0110011\ 0011110\ 0011110$
 $\rightarrow K_1 = 000110\ 110000\ 001011\ 101111\ 111111\ 000111\ 000001\ 110010$
- $C_2D_2 = 11000011\ 00110010\ 10101011\ 11110101\ 01011001\ 10011110\ 00111101$
 $\rightarrow K_2 = 011110\ 011010\ 111011\ 011001\ 110110\ 111100\ 100111\ 100101$
- $C_{16}D_{16} = 11110000\ 11001100\ 10101010\ 11110101\ 01010110\ 01100111\ 10001111$
 $\rightarrow K_{16} = 110010\ 110011\ 110110\ 001011\ 000011\ 100001\ 011111\ 110101$



1. Mã khối hiện đại

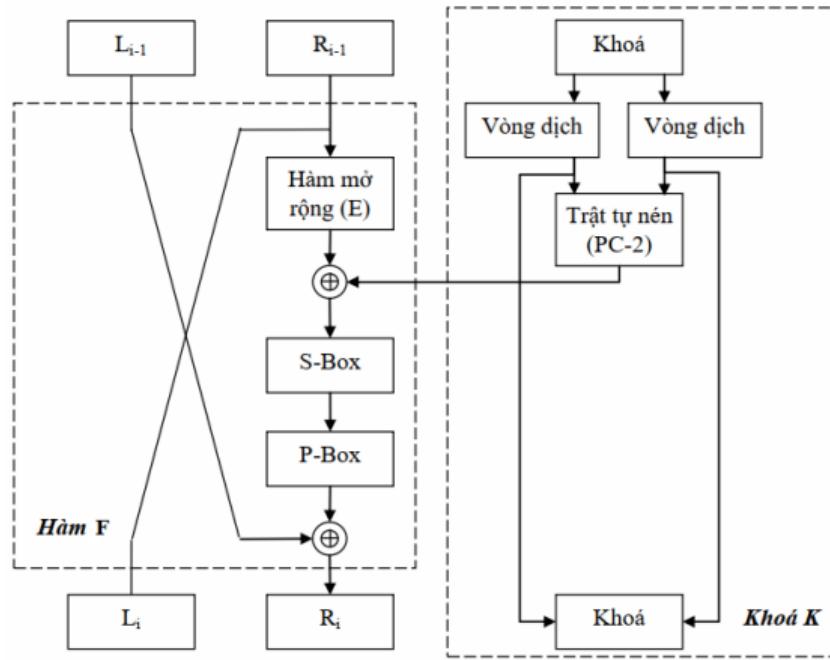
1. 2. Chuẩn mã dữ liệu DES

Mô tả hàm $F(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$

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

trong đó

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



1. Mã khối hiện đại

1. 2. Chuẩn mã dữ liệu DES

Mô tả hàm $F(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$

- **Input** của F : R_{i-1} (32 bit), K_i (48 bit); **Output**: xâu độ dài 32 bit
- "Sức mạnh" của DES nằm ở $F \rightarrow$ cần lựa chọn cẩn thận, tránh bị thám mã dễ dàng
- F thường có tính chất $F^{-1} = F$, tức là $F(F(x)) = x$
- **Thực hiện:** R_{i-1} được mở rộng thành 48 bit theo hàm E . (thực chất $E(R_{i-1})$ là một hoán vị có lặp, trong đó lặp lại 16 bit của R_{i-1})

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

1. Mã khối hiện đại

1. 2. Chuẩn mã dữ liệu DES

Mô tả hàm $F(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$

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

Ví dụ 1.5

$$R_0 = 1111\ 0000\ 1010\ 1010\ 1111\ 0000\ 1010\ 1010$$

$$E(R_0) = 011110\ 100001\ 010101\ 010101\ 011110\ 100001\ 010101\ 010101$$

1. Mã khối hiện đại

1. 2. Chuẩn mã dữ liệu DES

Mô tả hàm $F(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$

- Tính $E(R_{i-1}) \oplus K_i$ và viết kết quả thành 8 xâu 6 bit: $B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8$
- Đưa 8 khối B_i vào 8 bảng S_i cỡ 4×16 cố định (các S-Box):

$$S_1(B_1) S_2(B_2) S_3(B_3) S_4(B_4) S_5(B_5) S_6(B_6) S_7(B_7) S_8(B_8) \quad (32 \text{ bit})$$

- Với mỗi $B_i = b_1 b_2 b_3 b_4 b_5 b_6$

$$S_i(B_i) = S_i(r, c) = C_i \quad (4 \text{ bit}), \text{ trong đó} \quad \begin{cases} r = b_1 b_6 \\ c = b_2 b_3 b_4 b_5 \end{cases}$$

* **Chú ý:** Thuộc tính của S-Box (confusion + diffusion)

- Các bit vào phụ thuộc phi tuyến với các bit ra
- Sửa đổi ở một bit vào làm thay đổi ít nhất hai bit ra
- Khi một bit vào được giữ cố định và 5 bit còn lại thay đổi thì S-Box đầu ra có "phân bố đồng nhất" (số bit 0 và 1 cân bằng)

1. Mã khôi hiện đại

1. 2. Chuẩn mã dữ liệu DES

Mô tả hàm $F(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$

Ví dụ 1.6

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 S_1

$$B = 011011 \Rightarrow \begin{cases} r = 01 \rightarrow 1 \\ c = 1101 \rightarrow 13 \end{cases}$$

$$S_1(B) = S_1(1, 13) = 5 \rightarrow 0101.$$

1. Mã khôi hiện đại

1. 2. Chuẩn mã dữ liệu DES

Mô tả hàm $F(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$

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

Hộp S_2

Ví dụ 1.7

Cho $\begin{cases} R_0 = 1111\ 0000\ 1010\ 1010\ 1111\ 0000\ 1010\ 1010 \\ K_1 = 000110\ 110000\ 001011\ 101111\ 111111\ 000111\ 000001\ 110010 \end{cases}$

$E(R_0) = 011110\ 100001\ 010101\ 010101\ 011110\ 100001\ 010101\ 010101$

$K_1 \oplus E(R_0) = 011000\ 010001\ 011110\ 111010\ 100001\ 100110\ 010100\ 100111$

$S_1(B_1) S_2(B_2) S_3(B_3) S_4(B_4) S_5(B_5) S_6(B_6) S_7(B_7) S_8(B_8) = 0101\ 1100\ 1000\ 0010\ 1011\ 0101\ 1001\ 0111$

1. Mã khôi hiện đại

1. 2. Chuẩn mã dữ liệu DES

Mô tả hàm $F(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$

- Cuối cùng $F = P(S_1(B_1) S_2(B_2) \dots S_8(B_8))$.
- Hàm P chỉ đơn thuần thực hiện hoán vị kết quả thu được sau S nhờ P -Box

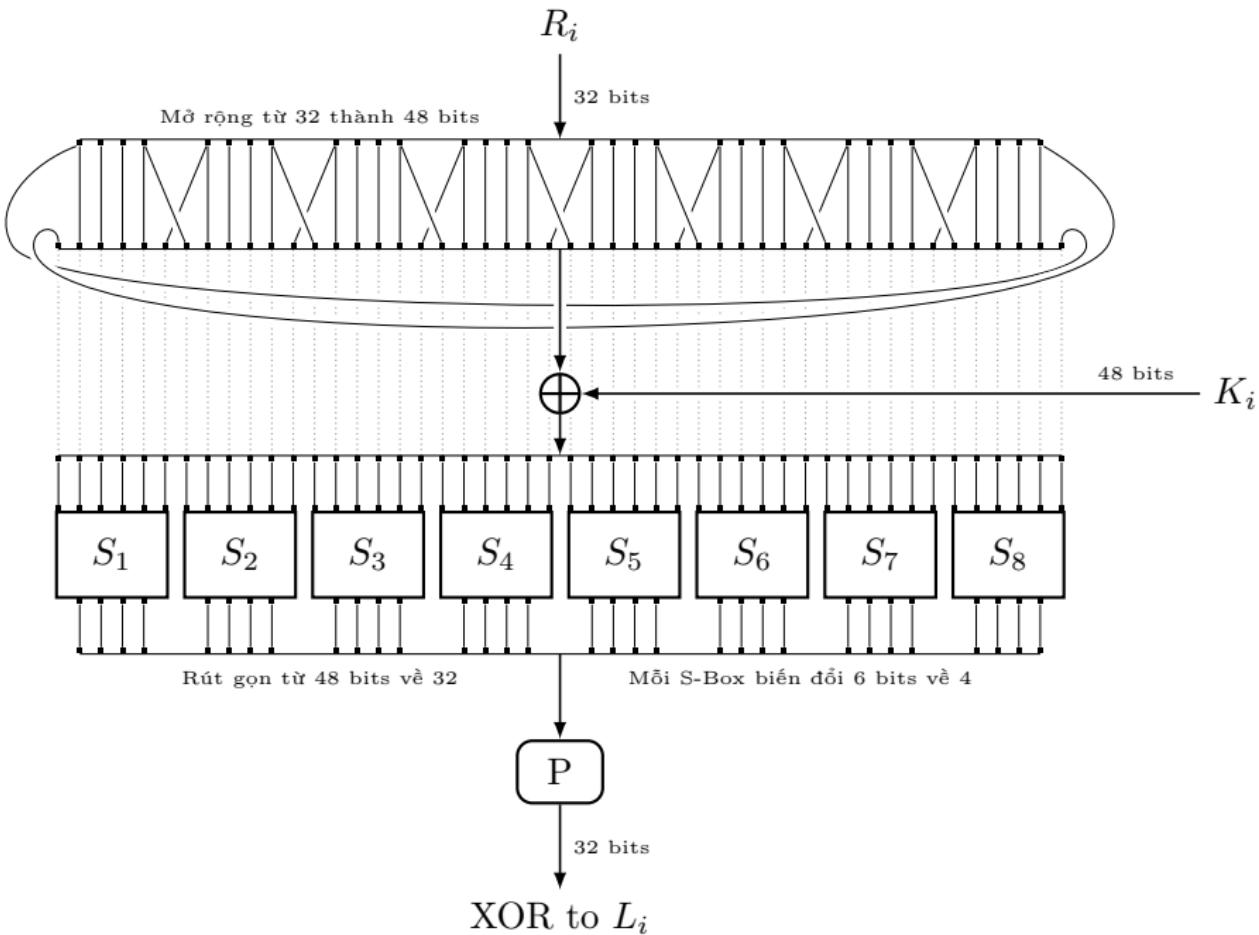
16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

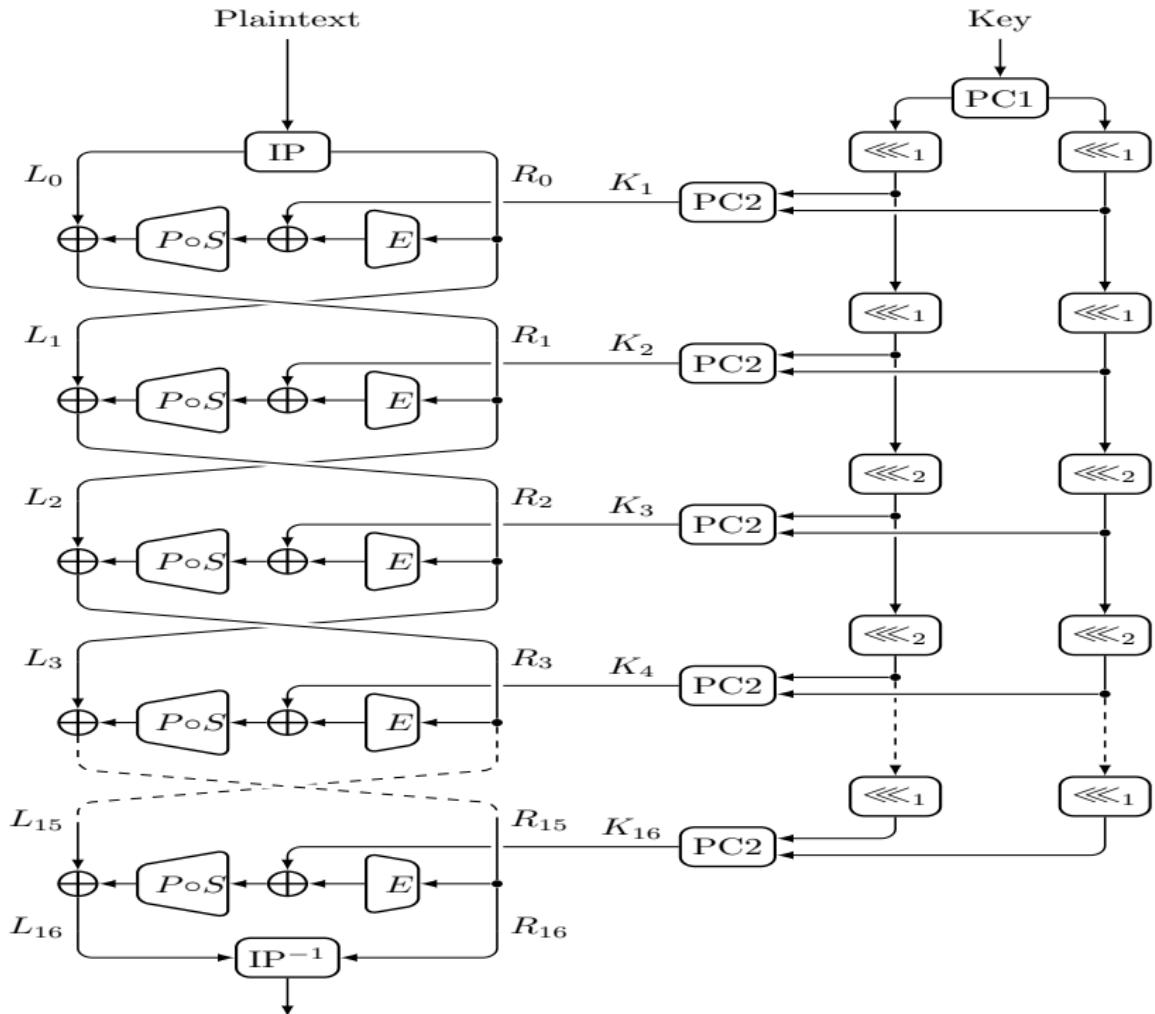
Bảng hoán vị P (P -Box)

Ví dụ 1.8

Từ kết quả của các S -Box:

$$\begin{aligned} S_1(B_1) S_2(B_2) S_3(B_3) S_4(B_4) S_5(B_5) S_6(B_6) S_7(B_7) S_8(B_8) &= 0101\ 1100\ 1000\ 0010\ 1011\ 0101\ 1001\ 0111 \\ \rightarrow F = P(\dots) &= 0010\ 0011\ 0100\ 1010\ 1010\ 1001\ 1011\ 1011 \end{aligned}$$





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

Bảng IP

Ví dụ 1.9

- Cho $M = 0123456789ABCDEF$. Viết dưới dạng nhị phân:

$M = 0000\ 0001\ 0010\ 0011\ 0100\ 0101\ 0110\ 0111\ 1000\ 1001\ 1010\ 1011\ 1100\ 1101\ 1110\ 1111$

- Áp dụng bảng IP ta thu được

$IP = 1100\ 1100\ 0000\ 0000\ 1100\ 1100\ 1111\ 1111\ 1111\ 0000\ 1010\ 1010\ 1111\ 0000\ 1010\ 1010$

$L_0 = 1100\ 1100\ 0000\ 0000\ 1100\ 1100\ 1111\ 1111;\ R_0 = 1111\ 0000\ 1010\ 1010\ 1111\ 0000\ 1010\ 1010$

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 IP⁻¹

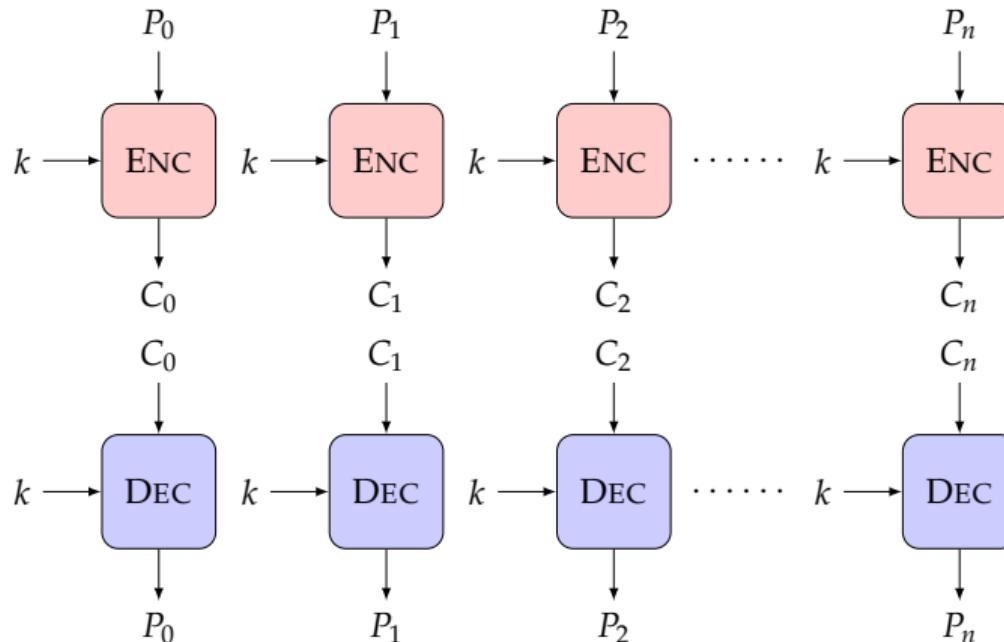
- $L_{16} = 0100\ 0011\ 0100\ 0010\ 0011\ 0010\ 0011\ 0100$; $R_{16} = 0000\ 1010\ 0100\ 1100\ 1101\ 1001\ 1001\ 0101$
- Hoán vị 2 khối rồi áp dụng IP⁻¹
 $R_{16}L_{16} = 00001010\ 01001100\ 11011001\ 10010101\ 01000011\ 01000010\ 00110010\ 00110100$
 $IP^{-1} = 10000101\ 11101000\ 00010011\ 01010100\ 00001111\ 00001010\ 10110100\ 00000101 \rightarrow$
 $85E813540F0AB405$
- Vậy bản rõ $\mathbf{M} = 0123456789ABCDEF \xrightarrow{DES} \mathbf{C} = 85E813540F0AB405$.
- Việc giải mã thực hiện tương tự mã hoá nhưng khoá được áp dụng theo **thứ tự ngược lại**

1. Mã khối hiện đại

1. 2. Chuẩn mã dữ liệu DES

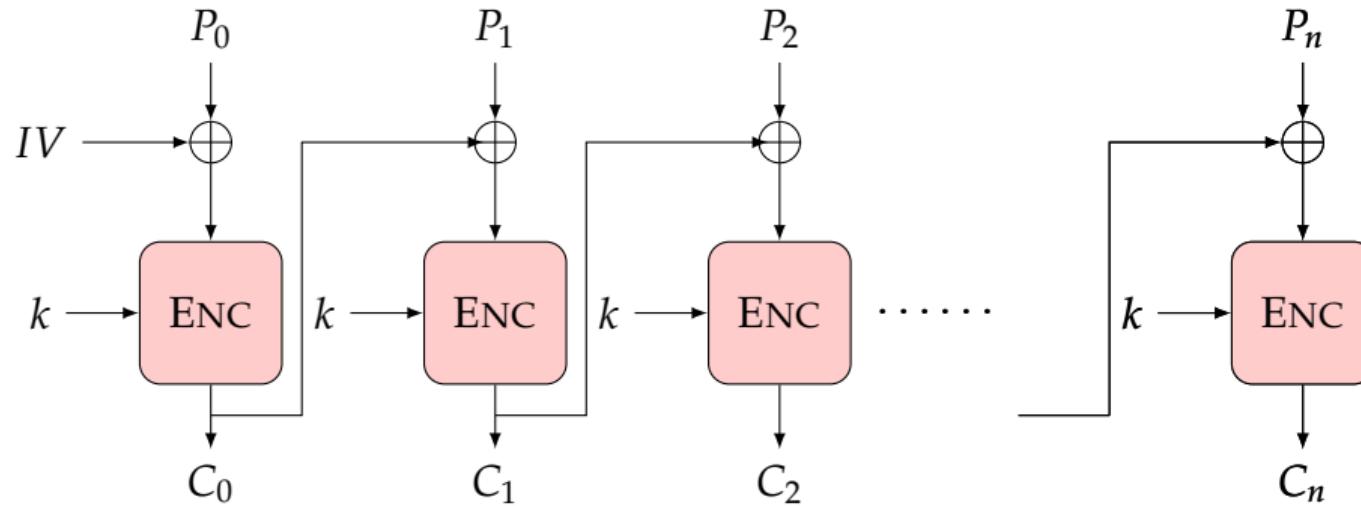
Một số cơ chế hoạt động của DES

- **Electronic Code Book (ECB):** mỗi khối 64 bit của bản rõ được mã hoá độc lập



Một số cơ chế hoạt động của DES

- **Chain Block Coding (CBC):** việc mã hoá khối sau phụ thuộc vào bản mã của khối trước

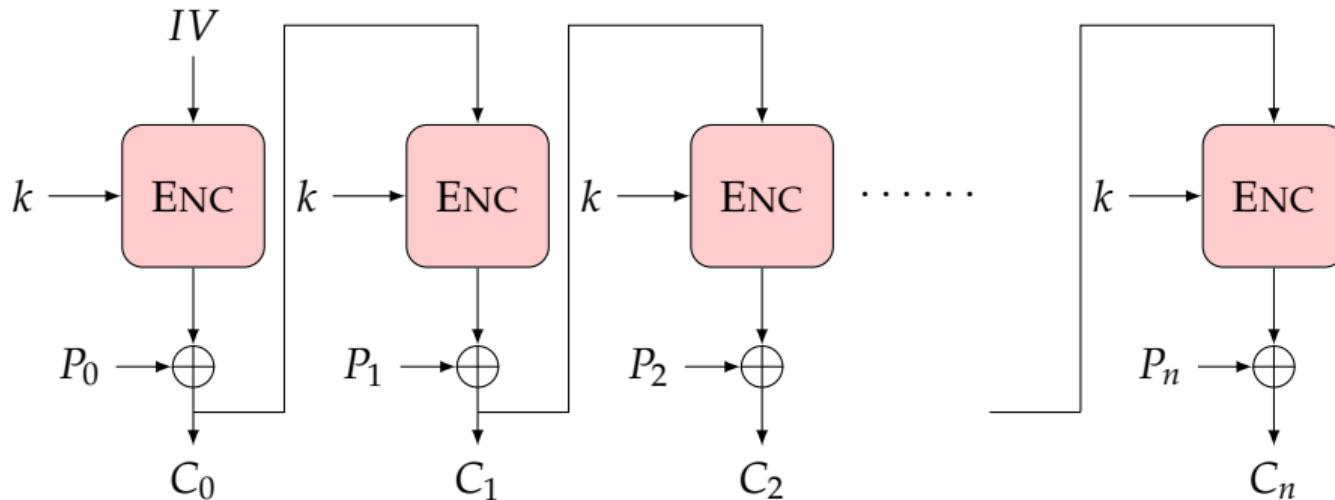


1. Mã khối hiện đại

1. 2. Chuẩn mã dữ liệu DES

Một số cơ chế hoạt động của DES

- Cipher Feedback (CFB)



1. Mã khối hiện đại

1. 2. Chuẩn mã dữ liệu DES

Thám mã DES

- DES có $2^{56} = 10^{17}$ khoá. Giả sử một phép thử mất 10^{-6} s thì sẽ mất 10^{11} s ≈ 7300 năm
- Năm 1976, Diffie và Hellman đề xuất chế tạo một "máy tính song song sử dụng 1 triệu chip để thử 1 triệu khoá mỗi giây". Máy tính này có thể vét cạn không gian khoá DES trong $\frac{1}{2}$ ngày với giá 20 triệu \$.
- Năm 1993, Michael Wiener đã thiết kế máy tính chuyên dụng với giá 1 triệu \$ sử dụng phương pháp vét cạn giải mã DES trung bình trong 3,5 giờ (chậm nhất là 7 giờ).
- Năm 1998, John Gilmore và cộng sự đã chi 220.000\$ để chế tạo máy tính có thể vét cạn không gian khoá DES trung bình 5,6 giờ. Máy tính có tên là Deep Crack, sử dụng 27 bảng mạch, mỗi bảng chứa 64 chip và có khả năng thử 90 tỷ khoá mỗi giây.
- Năm 1990, hai nhà toán học Do Thái Biham và Shamir đã phát minh ra phương pháp phá mã vi sai (differential cryptanalysis) và chứng minh nó hiệu quả hơn phương pháp brute-force.

1. Mã khối hiện đại

1. 3. Triple DES

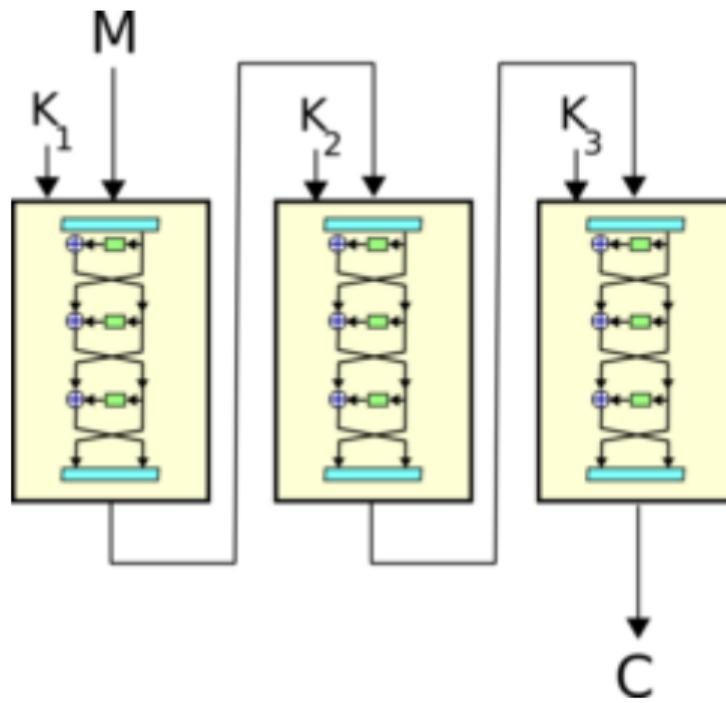
- DES có thể bị thám mã trong vòng vài giờ → cần tìm kiếm hệ mã khác thay thế DES
- **Giải pháp:** có thể tận dụng DES nhưng **mã hoá nhiều lần**
- **Cách 1:** $C = \text{DES}_{K_2}(\text{DES}_{K_1}(P))$
 - gọi tên là double DES hay 2DES
 - các chứng minh chỉ ra hệ mã này không an toàn hơn DES (thuật toán brute-force chỉ yêu cầu số phép tính gấp đôi so với DES)
- **Cách 2:** mã hoá DES 3 lần → triple DES hay 3DES hay TDEA (Triple Data Encryption Algorithm)
 - độ dài khoá 168 bit
 - có một số biến thể khác nhau



1. Mã khối hiện đại

1. 3. Triple DES

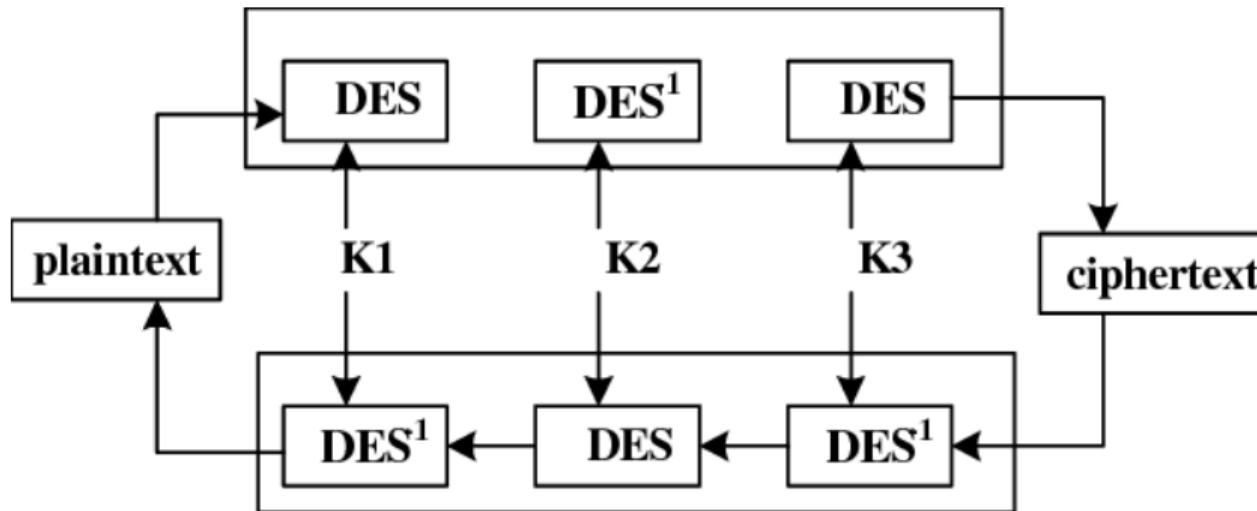
- $C = \text{DES}_{K_3}(\text{DES}_{K_2}(\text{DES}_{K_1}(M))) \rightarrow$ mô hình EEE



1. Mã khối hiện đại

1. 3. Triple DES

- $C = \text{DES}_{K_3} \left(\text{DES}_{K_2}^{-1} (\text{DES}_{K_1}(M)) \right) \rightarrow \text{mô hình EDE}$



1. Mã khối hiện đại

1. 4. Một số bài tập thực hành

Cho plaintext = "Hello World!", Khoá K = "00110100 00101101 10110101 10101000 00011101 11011011 10010000 00000100 "

- ① Derive K_1 , the first-round subkey
- ② Derive L_0, R_0
- ③ Expand R_0 to get $E[R_0]$, where $E[\cdot]$ is the expansion function of Table S.1
- ④ Calculate $A = E[R_0] \oplus K_1$
- ⑤ Group the 48-bit result above into sets of 6 bits and evaluate the corresponding S-Box substitution
- ⑥ Concatenate the results above to get a 32-bit result, B.
- ⑦ Apply the permutation to get $P(B)$
- ⑧ Calculate $R_1 = P(B) \oplus L_0$
- ⑨ Write down the ciphertext.

1. Mã khôi hiện đại

1. 4. Một số bài tập thực hành

Sử dụng một trong các ngôn ngữ lập trình C, C++, Java để làm các bài tập sau:

- ① Viết chương trình đếm tần số xuất hiện của cái chữ cái trong một văn bản tiếng Anh ở dạng file text
- ② Viết chương trình cài đặt thuật toán mã hoá và giải mã của hệ mã Ceasar
- ③ Viết chương trình cài đặt thuật toán mã hoá và giải mã của hệ mã Affine
- ④ Viết chương trình cài đặt thuật toán mã hoá và giải mã của hệ mã Vigenere
- ⑤ Viết chương trình cài đặt thuật toán mã hoá và giải mã file theo hệ DES với các cơ chế ECB, CBC
- ⑥ Viết chương trình cài đặt thuật toán mã hoá và giải mã file theo hệ AES với các cơ chế ECB, CBC

Nội dung

Mã khối hiện đại

- 1.1 Mở đầu
- 1.2 Chuẩn mã dữ liệu DES
- 1.3 Triple DES
- 1.4 Một số bài tập thực hành

Trao đổi

TRAO ĐỔI