CHƯƠNG 3
CÁC HỆ MẬT MÃ KHÓA BÍ MẬT (SECRET KEYS)

3.1 .Các hệ mật cổ điển

3.1.1. Hệ mã hoá thay thế (substitution cipher)

Có 4 kỹ thuật thay thế sau đây:

1. Thay thế đơn (A simple substitution cipher): một ký tự của bản rõ được thay bằng một ký tự tương ứng trong bản mã. Một ánh xạ 1-1 từ bản rõ tới bản mã .
2. Thay thế đồng âm (A homophonic substitution cipher): giống như thay thế đơn, song một ký tự của bản rõ có thể ánh xạ tới một trong số nhiều ký tự của bản mã: sơ đồ ánh xạ 1-n (one-to-many).
3. Thay thế đa mẫu tự (A polyalphbetic substitution cipher): dùng nhiều thuật toán mã hoá thay thế đơn. Ánh xạ 1-1 nhưng có thể thay đổi nhiều lần trong phạm vi một thông điệp

4. Thay thế đa ký tự (A polygram substitution cipher): là thuật toán trong đó các khối ký tự được mã hoá theo nhóm. Đây là thuật toán tổng quát nhất, cho phép thay thế các nhóm ký tự của văn bản gốc. Ví dụ, “ABA” có thể tương ứng với “RTQ”, “ABB” có thể tương ứng với “SLL”, v.v

3.1.1.1. Hệ mã Ceasar : Là một hệ mã đơn . Làm việc trên trương modulo 26 của bảng chữ cái Latin (A-Z)

Ta có : $P \in \{a\text{-}z\}$ - Không gian bản rõ (plain text)

$C \in \{a\text{-}z\}$ - Không gian bản mã (cipher text)

$K \in [Z_N]$ - Không gian khóa

- Mã hóa: $E_K(i) = (i + k) \bmod N$.
- Giải mã: $D_K(i) = (i – k) \bmod N$.

- Các phép tính toán số học được thực hiện trên vành Z_{26}, số khóa có thể là 26 nhưng trên thực tế chỉ có 25 khóa có ích.
- Ví dụ: với k=3 (được hoàng đế Caesar sử dụng)

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

"I LOVE YOU" được mã thành "L OSZH CSY"

Trên thực tế hệ mã Caesar có cơ số khóa ít nên hoàn toàn có thể thám mã bằng cách thử tất cả các khóa có thể (kiểu tấn công Brute force).

3.1.1.2.Hệ mã Affine

- Không gian các bản rõ *(P,C) ϵ {A}* - A bảng chữ cái .Giả sử |A| ϵ N. Khi đó không gian khóa của hệ mã được xác định như sau:

$$K = \{ (a, b): a, b \epsilon Z_N, (a, N) = 1\}$$

- Đánh số các chữ cái từ 0 → (N-1)
- Tiến hành mã từng ký tự "x" theo công thức sau :

$$EK(x) = (a*x + b) \bmod N.$$

- Để giải mã ta cần tìm a^{-1} (do (a, N) = 1 ; nên luôn tìm được) và tiến hành tìm "y" (giải mã) theo công thức sau:

$$DK(y) = a*(y - b) \bmod N.$$

3.1.1.3. Hệ mã Vigenere (1523-1596)

- Không gian các bản rõ *(P,C) ϵ {A}* - A bảng chữ cái. Các chữ cái được đánh số từ 0 → (N-1).
- Không gian khóa K được xác định như sau:

  ∀ M ≥0 , khóa có độ dài M là một xâu ký tự :

  $k = k_1, k_2 , \ldots k_M$.
- Để mã hóa ,chia P thành các khối có độ dài M và chuyển thành số thứ tự tương ứng trong {A}, Ví dụ: $x = x_1 x_2 \ldots x_M$.
- Mã hóa : $E_K(x) = (x_1 + k_1, x_2 + k_2, \ldots, x_M + k_M) \bmod N$
- Giải mã : $D_K(y) = (y_1 – k_1, y_2 – k_2, \ldots, y_M – k_M) \bmod N$ với : $(y_1 , y_2, \ldots, y_M)$ là bản rõ.
- Số khóa sử dụng : 26^M

- Ví dụ: xét A là bảng chữ cái tiếng Anh, N = 26 . Giả sử khóa có độ dài 6 và K = “CIPHER”.
  P = “THIS CRYPTOSYSTEM IS NOT SECURE”. Ta có :
  K = 2 8 15 7 4 17, P = 19 7 8 18 2 17 | 24 15 19 14 18 23 | 18 19 4 12 8 18 | 13 14 19 18 4 2 | 20 17 4.
  Quá trình mã hóa :
- P = 19 7 8 18 2 17 | 24 15 19 14 18 23 | 18 19 4 12 8 18 | 13 14 19 18 4 2 | 20 17 4
- K = 2 8 15 7 4 17 | 2 8 15 7 4 17 | 2 8 15 7 4 17 | 2 8 15 7 4 17 | 2 8 15
- C = 21 15 23 25 6 8 | 0 23 8 21 22 14 | 20 1 19 19 12 9 | 15 22 8 25 8 19 | 22 25 19
- Vậy bản mã là C = “VPXZGI AXIVWO UBTTMJ PWIZIT WZT”.

3.1.2. Hệ mã chuyển vị (transposition cipher)

- Hệ mã hoá chuyển vị là hệ mã hoá trong đó các ký tự của bản rõ vẫn được giữ nguyên, nhưng vị trí của chúng được đổi chỗ cho nhau. Ví dụ :
- Bản rõ: COMPUTER GRAPHICS MAY BE SLOW BUT AT LEAST IT‟S EXPENSIVE

  COMPUTERGR

  APHICSMAYB

  ESLOWBUTAT

  LEASTITSEX

  PENSIVE
- Bản mã: CAELPOPSEEMHLANPIOSSUCWTITSBIUEMUTERATSGYA ERBTX

Các kỹ thuật chuyển vị

1. Đảo ngược toàn bộ bản rõ . Đây là phương pháp mã hoá đơn giản nhất vì vậy không đảm bảo an toàn.

Ví dụ : “TRANSPOSITION CIPHER” được mã hoá thành “REHPICNOITISOPSNART”.

2. Mã hoá theo mẫu hình học : bản rõ được sắp xếp lại theo một mẫu hình học nào đó, thường là một mảng hoặc một ma trận hai chiều. Có hai cách:

- Viết theo hàng ngang → Đổi chỗ cột → Lấy ra theo cột
- Viết theo cột → Đổi chỗ cột → Lấy ra theo hàng ngang

3. Chuyển vị các ký tự theo chu kỳ cố định n

- Nếu hàm f(i) là một chuyển vị của một khối gồm n ký tự (i) thì khoá mã hoá được biểu diễn bởi K(n,f).
- Do vậy, bản rõ: $M = m_1\ m_2 ... m_d\ m_{n+1} ... m_{2n}$

  Với m_i là các ký tự , và bản rõ sẽ được mã hoá :
- $Ek(M) = m_{f(1)}\ m_{f(2)} ... m_{f(n)}\ m_{f(n)+1}\ ... m_{f(n)+n}$
- Trong đó : $m_{f(1)}\ m_{f(2)} ... m_{f(n)....}$ là một hoán vị của $m_1\ m_2\ ... m_n$.

Ví dụ: d=6 ,dãy i= 123456 được hoán vị thành f(i)=356214

VỊ TRÍ ĐẦU	CHUYỂN VỊ	KÝ TỰ	BẢN MÃ	CHUYỂN VỊ $^{-1}$	BẢN RÕ
1	3	F	N	5	F
2	5	R	E	4	R
3	6	I	F	1	I
4	2	E	D	6	E
5	1	N	R	2	N
6	4	D	I	3	D

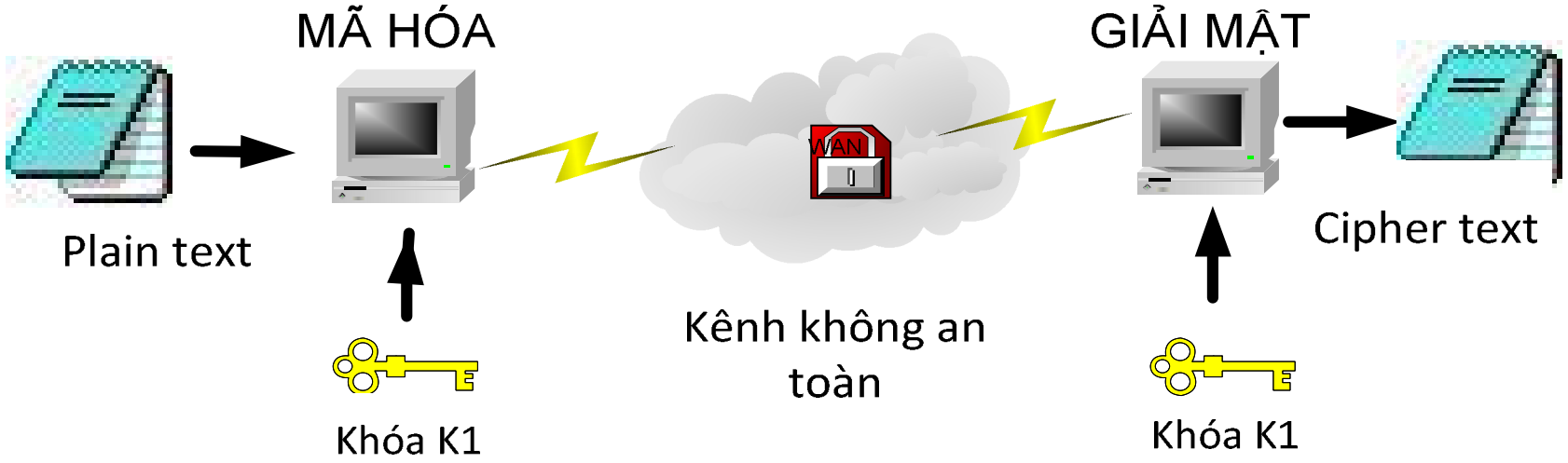
Kết quả mã : NEFDRI

3.2.Các hệ mã khối (Block cipher)

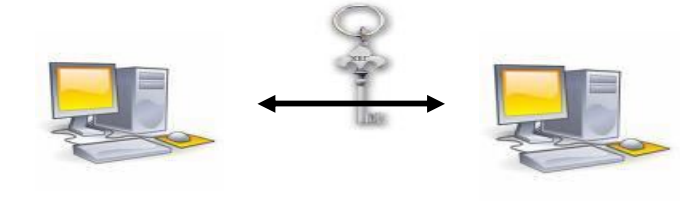
1. Khái niệm :
   - Còn được gọi là mật mã đối xứng
   - Dữ liệu đầu vào (văn bản rõ) được chia thành các khối (M_i) có độ dài cố định (≥ 64 bit).
   - Xử lý (mã hóa) tuần tự từng khối.
   - Độ dài không gian khóa (K) bằng độ dài khối "rõ".
   - Khóa cần được phân phối trước. (preshare keys)

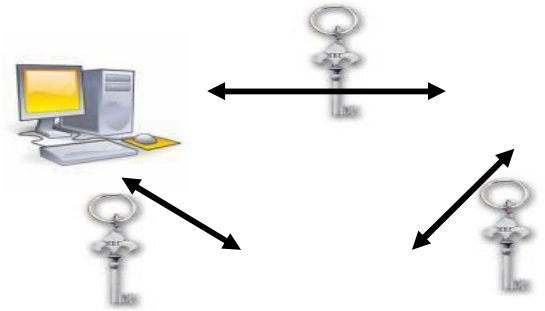
Khái niệm mật mã đối xứng

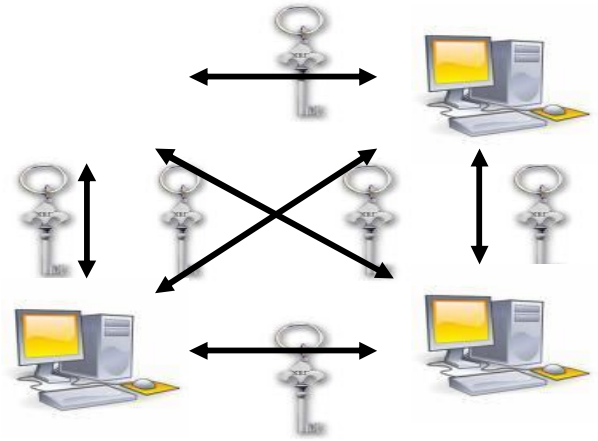
- Bên nhận (giải mã) và bên gửi (mã hóa) sử dụng một khoá mật mã duy nhất (Tính đối xứng).
- Số lượng khoá¸tăng lên tỷ lệ với số người dùng

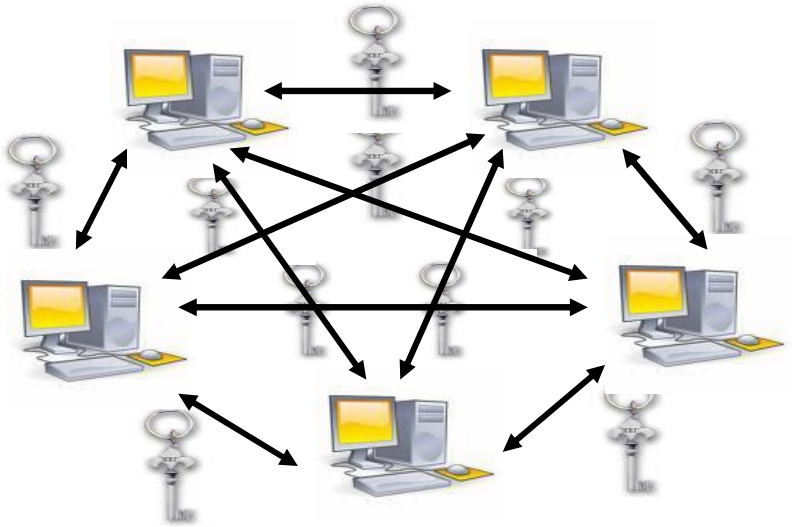


Vấn đề sử dụng khóa









2. Các hệ mã khối hiện đại

- **DES** Data Encryption Standard (DES) xuất hiện vào giữa 1970s. Là thuật toán mạnh vào lúc bấy giờ.DES dùng khoá 64/128-bit .
- **AES** *Advanced Encryption Standard (AES)* thay thế DES sử dụng thuật giải *Rijndael* . AES hỗ trợ khoá có kích thước 128, 192, và 256 bit.
- **3DES** *Triple-DES (3DES)* bản nâng cấp của DES.3DES an toàn hơn DES.
- **CAST** do Carlisle Adams và Stafford Tavares phát triển.CAST sử dụng khoá có chiều dài từ 40-bit đến 128-bit , chạy nhanh và hiệu quả.

- **RC** do phòng thí nghiệm RSA phát triển.Có các loải CR4, RC5 và RC6. RC5 sử dụng khoá 2,048 bit .Là một hệ mật mã mạnh.
- **Blowfish** do *"Counterpane systems"* phát triển (Bruce Schneier). AES hỗ trợ thêm khoá mã 448 bits.
- **IDEA** *International Data Encryption Algorithm (IDEA)* thuật giải dùng 128-bit key. An toàn hơn DES, IDEA được sử dụng trong giaot hức PGP. Pretty Good Privacy (PGP) là hệ mật mã sử dụng trong vùng bảo mật e-mail công cộng.

  "Block cipher" còn được biết với tên gọi :" Hệ mật đối xứng"

- Chuẩn mã hóa dữ liệu DES (Data Encryption Standard), một trong số các hệ mã khối được sử dụng rộng rãi nhất và là nền tảng cho rất nhiều các hệ mã khối khác.
- Được Uỷ ban Tiêu chuẩn quốc gia Hoa Kỳ công bố vào 15/02/1977. DES được xây dựng trên một hệ mã khối phổ biến có tên là LUCIFER do IBM phát triển .
- DES có nhiều ưu điểm (nhanh, thuật toán công khai, dễ cài đặt) và đã được sử dụng trong một thời gian rất dài (trước những năm 90).
- Chuẩn mã hóa mới ra đời AES thay thế cho DES (1997).AES được xây dựng dựa trên thuât toán Rijndael (2011).

3. Điều kiện an toàn của hệ mật mã khối:

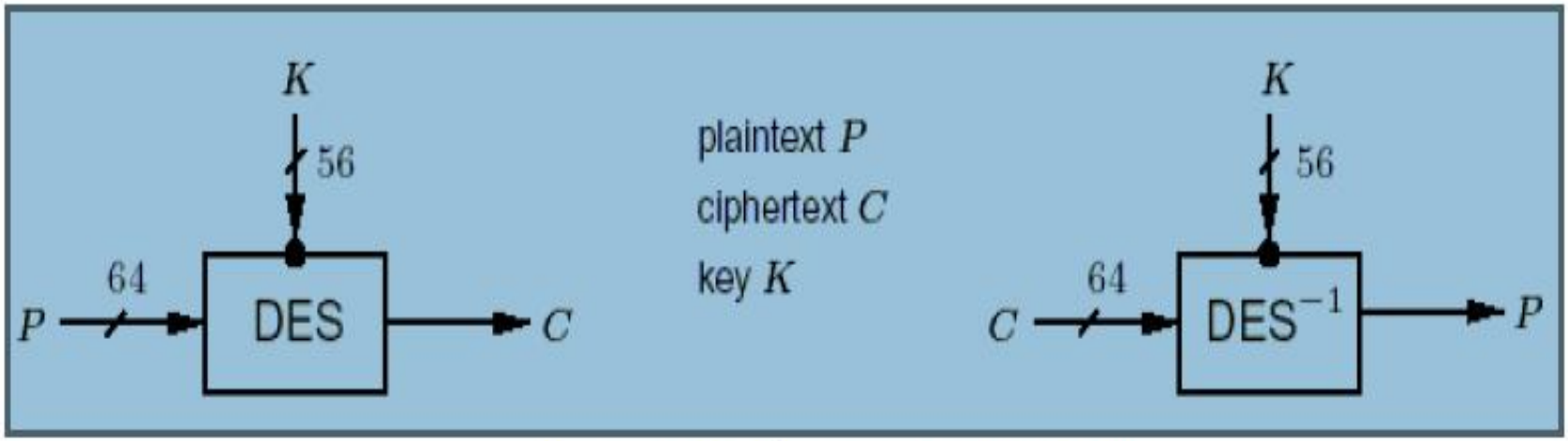
- Kích thước khối phải đủ lớn .Tuy nhiên điều này sẽ dẫn đến thời gian mã hoá sẽ tăng lên.
- Không gian khoá, tức chiều dài khoá phải đủ lớn (chống brute force attack).Tuy nhiên không gian khóa quá lớn sẽ gây khó khăn cho việc tạo khoá, phân phối, quản lý và lưu trữ khoá .
- Khi thiết kế hệ mã khối, phải đảm bảo hai yêu cầu sau:

✓Sự hỗn loạn (confusion): sự phụ thuộc giữa bản rõ và bản mã phải thực sự phức tạp . Mối quan hệ này tốt nhất là phi tuyến.

✓Sự khuếch tán (diffusion): Tăng độ dư thừa của bản mã.

3.2.1.Chuẩn mã hoá dữ liệu DES (Data Encryption Standard)

3.2.1.1. Sơ đồ tổng quát của DES



Hình 3.1: Chuẩn mã hóa dữ liệu DES

- Input block : 64 bit
- Key length : 56 (RD)+ 8 (parity)=64 bit

3.2.1.2. Thuật giải DES

- DES thực hiện 16 vòng lặp (i=1 đến i=16)
- Hàm mã hóa :

$$L_i = R_{i-1} ; \quad (1)$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i) ; \text{ trong đó :}$$

$$f(R_{i-1},K_i) = P(S(E(R_{i-1}) \oplus K_i)); (2)$$

- Trong đó:
  - ✓ $\oplus$ phép tuyển loại trừ của hai xâu bit theo modulo 2.
  - ✓ Hàm f là một hàm phi tuyến.
  - ✓ E là hoán vị mở rộng ánh xạ R_{i-1} tới 32 bit thành 48 bit .
  - ✓ P là hoán vị cố định khác của 32 bit.
  - ✓ IP - hoán vị bit khởi đầu .Ở đầu ra dùng hoán vị nghich đảo IP^{-1}

a. Thuật toán chi tiết:

- Input: Bản rõ M = $m_1\ m_2 \ldots m_{64}$
  Khóa 64 bit K = $k_1\ k_2 \ldots k_{64}$ (bao gồm cả 8 bit parity)
- Output: Bản mã 64 bit c = $c_1\ c_2 \ldots c_{64}$

  1. Sinh khóa con.

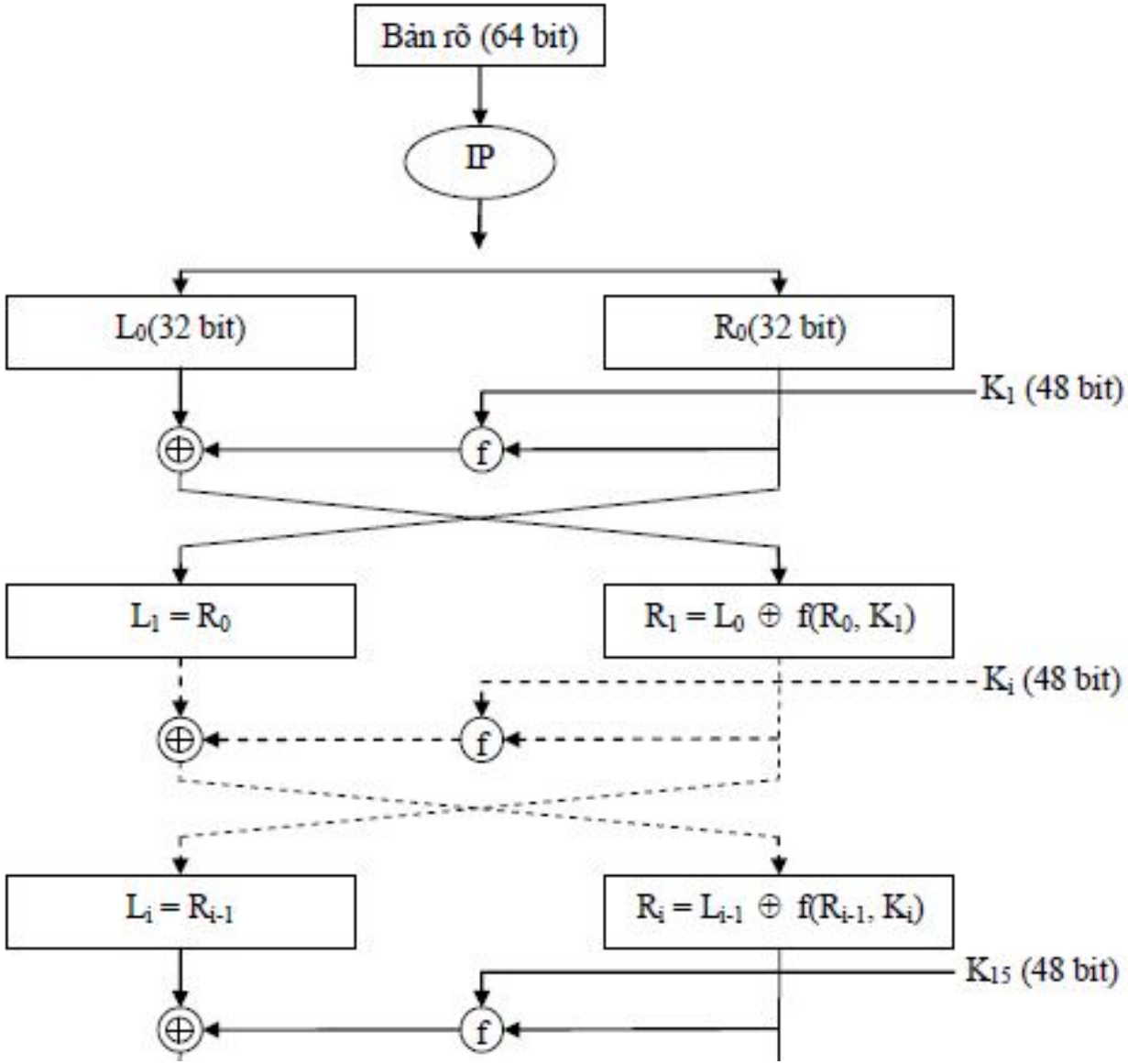
  2. Khởi tạo (L_0 ,R_0) từ IP($m_1\ m_2 \ldots m_{64}$) .

  Kết quả nhận được $L_0 = m_{58}\ m_{50} \ldots m_8$.

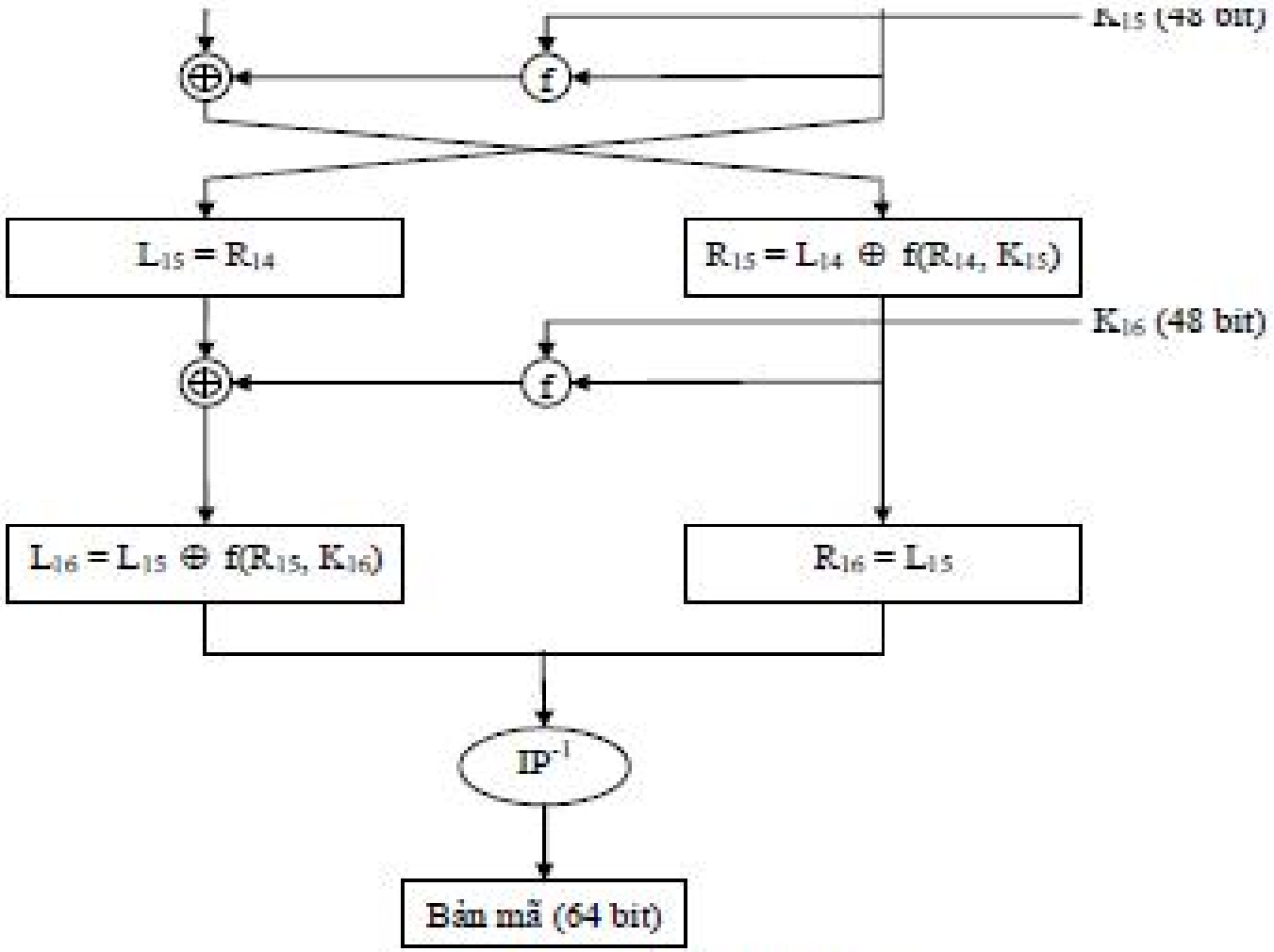
  $R_0 = m_{57}\ m_{49} \ldots m_7$.

  3. (16 vòng) for i = 1 to 16
- Tính các L_i và R_i theo các công thức (1) và (2)

b. Sơ đồ các vòng lặp của DES



Sơ đồ các vòng lặp của DES (tiếp)



Hình 3.2: Sơ đồ mã hoá DES

c. Hoán vị IP và hoán vị ngược IP^{-1}

Bảng hoán vị IP được đưa ra trong bảng dưới đây:

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

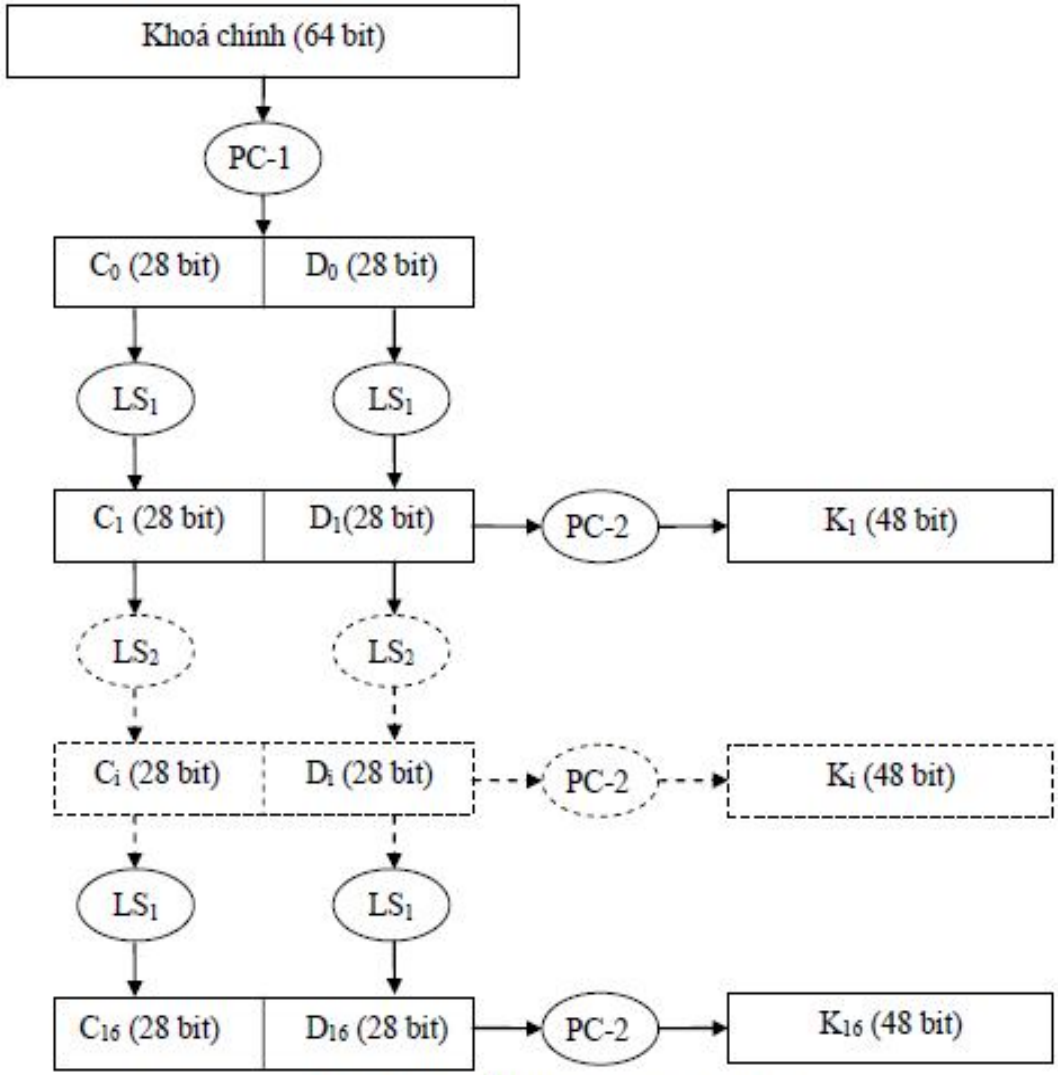
Bảng 3.6: Bảng hoán vị IP

Bảng hoán vị ngược 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 3.7: Bảng hoán vị ngược IP^{-1}

d.Tạo khóa con (dùng cho mỗi vòng lặp)



Hình 3.4: Sơ đồ tạo khoá con của DES

Tạo khóa con (tiếp)

- Từ khóa chính 64 bit bỏ đi các bit parity (các bit thứ 8) nhờ bảng PC-1 .Nhận khối khóa 56 bit.
- Khối 56 bit này chia thành hai khối C_0 và D_0

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 3.8: Bảng PC-1

- Bảng hoán vị nén PC-2

  Sau mỗi một vòng lặp , bảng hoán vị nén PC-2 được sử dụng

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

Bảng 3.10: Bảng PC-2

- Output là một khối khóa K_i = 48 bit

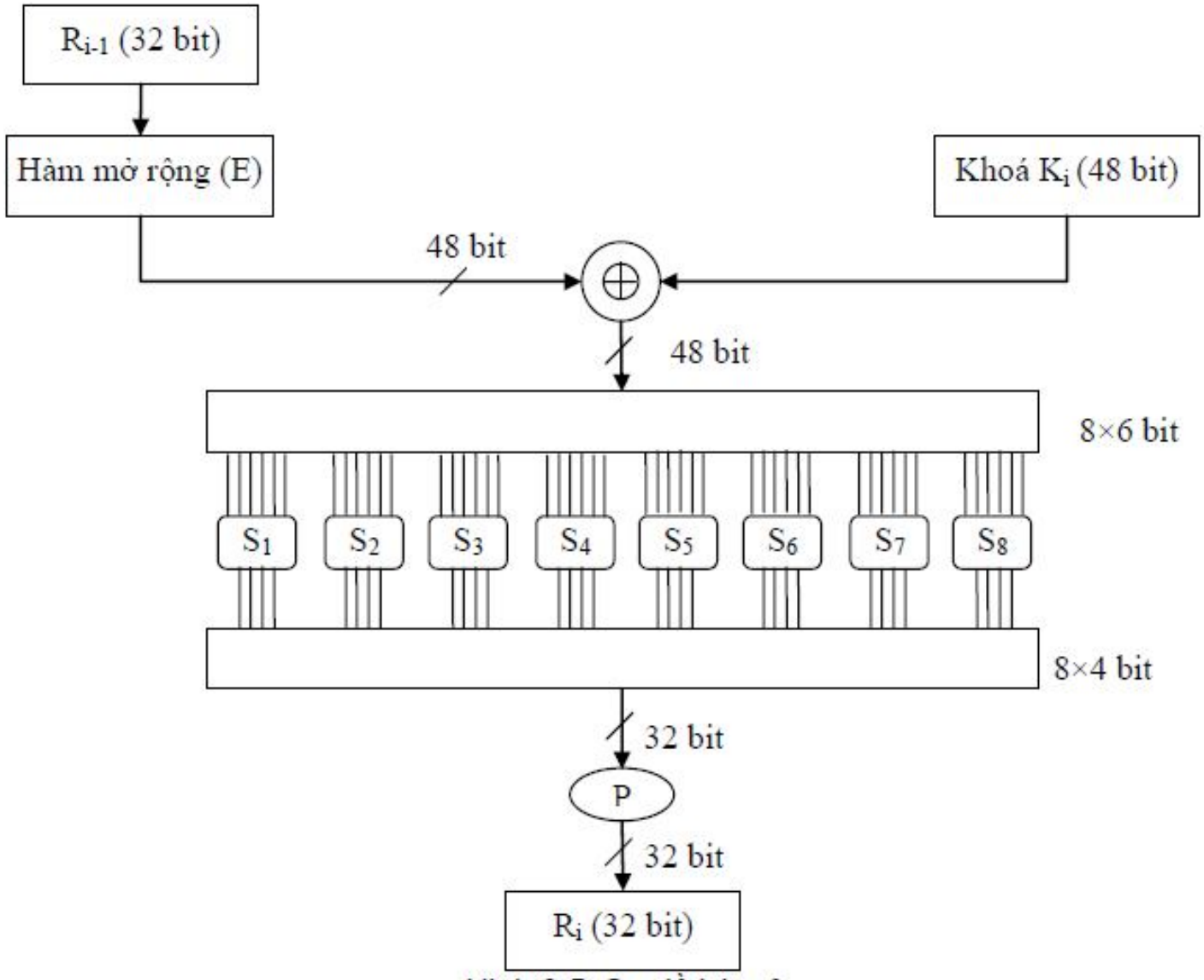
- Sau đó, hai khối 28 bit này được dịch trái 1 hoặc 2 bit phụ thuộc vào bảng sau :

VÒNG LẶP	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
SỐ BIT DỊCH	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

e. Hàm mã hóa $f(R_{i-1},K_i)$

- Biến thứ nhất R_{i-1} được mở rộng có độ dài 48 bit theo một hàm E. $E(R_{i-1})$ là một bảng hoán vị có lặp trong đó lặp lại 16 bit của R_{i-1}.(Hình vẽ 3.6)
  - Tính $E(R_{i-1})$ XOR K_i và viết kết quả là 8 chuỗi 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 $S_1, S_2, ..., S_8$ (S-Box). Mỗi hộp S-Box là một bảng 4*16 cố định có các cột từ 0 đến 15 và các hàng từ 0 đến 3.

Sơ đồ hàm f(R_{i-1} ,K_i)



Hình 3.5: Sơ đồ hàm f

Tính $C = C_1\ C_2\ C_3 \ldots\ C_8$

Hai bít B_1 và B_6 tạo ra một dãy số từ 0 đến 3

Bốn bít giữa B_2- B_5 tạo thành dãy số từ 0 đến 15.

Dùng bít B_1 và B_2 tạo số dòng và bốn bít giữa B_2-B_5 tạo số cột . Khối 4 bít được nhận bởi đối chiếu số dòng và số cột tương ứng .

Ví dụ : Khối 6 bít có dạng : 1 1 0 0 1 0

$B_1 = 1$

$B_6 = 0$ số cột là $(10)_2$ hay $(2)_{10}$

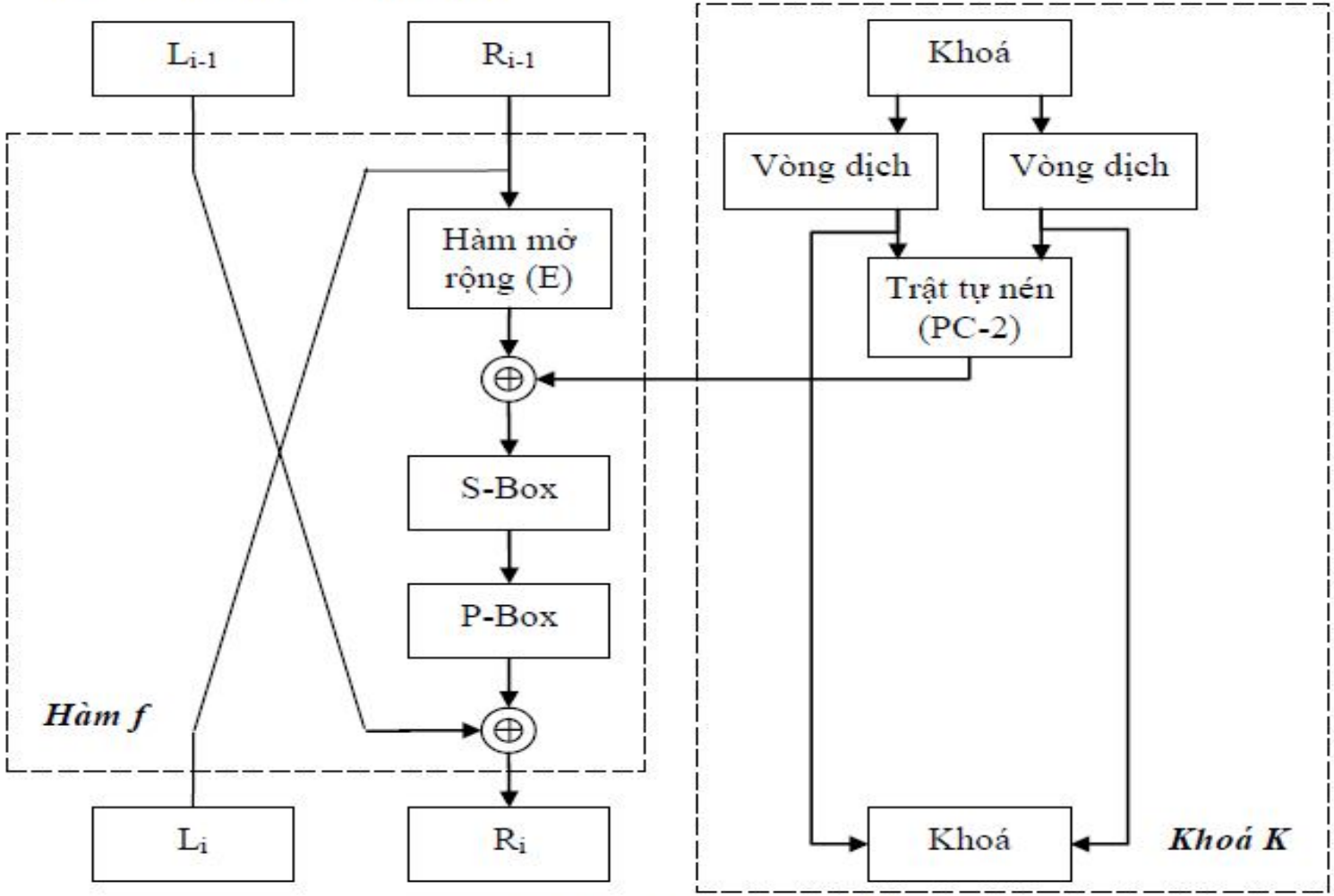
$B_2 - B_5 = (1001)_2$ hay $(9)_{10}$

Đối chiếu với bảng của hàm S_j ta có số cột là 9 , số dòng là 2,ta nhận được mã của khối 4 bit là 12 hay “1 1 0 0 ”

Tại từng nút lặp ta sử dụng các bảng S_j khác nhau.Có tất cả 8 bảng S_i

Sơ đồ tổng quát 1 vòng của DES

Sơ đồ cấu trúc một vòng DES:



Hình 3.3: Sơ đồ một vòng DES

f. Hộp S-box

- Ba thuộc tính của hộp S (đảm bảo tính confusion và diffusion) của thuật toán (NSA).
  - ✓Các bit vào luôn phụ thuộc phi tuyến với các bit ra.
  - ✓ Sửa đổi ở một bit vào làm thay đổi ít nhất là hai bit ra.
  - ✓ Phân bố các bit "1" và "0" trong hộp S-box phải tuân theo luật "phân bố đều" , "đồng xác xuất".
- Sau khi cộng modulo với khoá K, thu được chuỗi 48 bit chia làm 8 khối đưa vào 8 hộp S-Box. Mỗi hộp S-Box có 6 bit đầu vào và 4 bit đầu ra (tổng bộ nhớ yêu cầu cho 8 hộp S-Box chuẩn DES là 256 bytes). Kết quả thu được là một chuỗi 32 bit tiếp tục vào hộp P-Box.
- Có thể tự thiết kế S-box

Một số S-box chuẩn

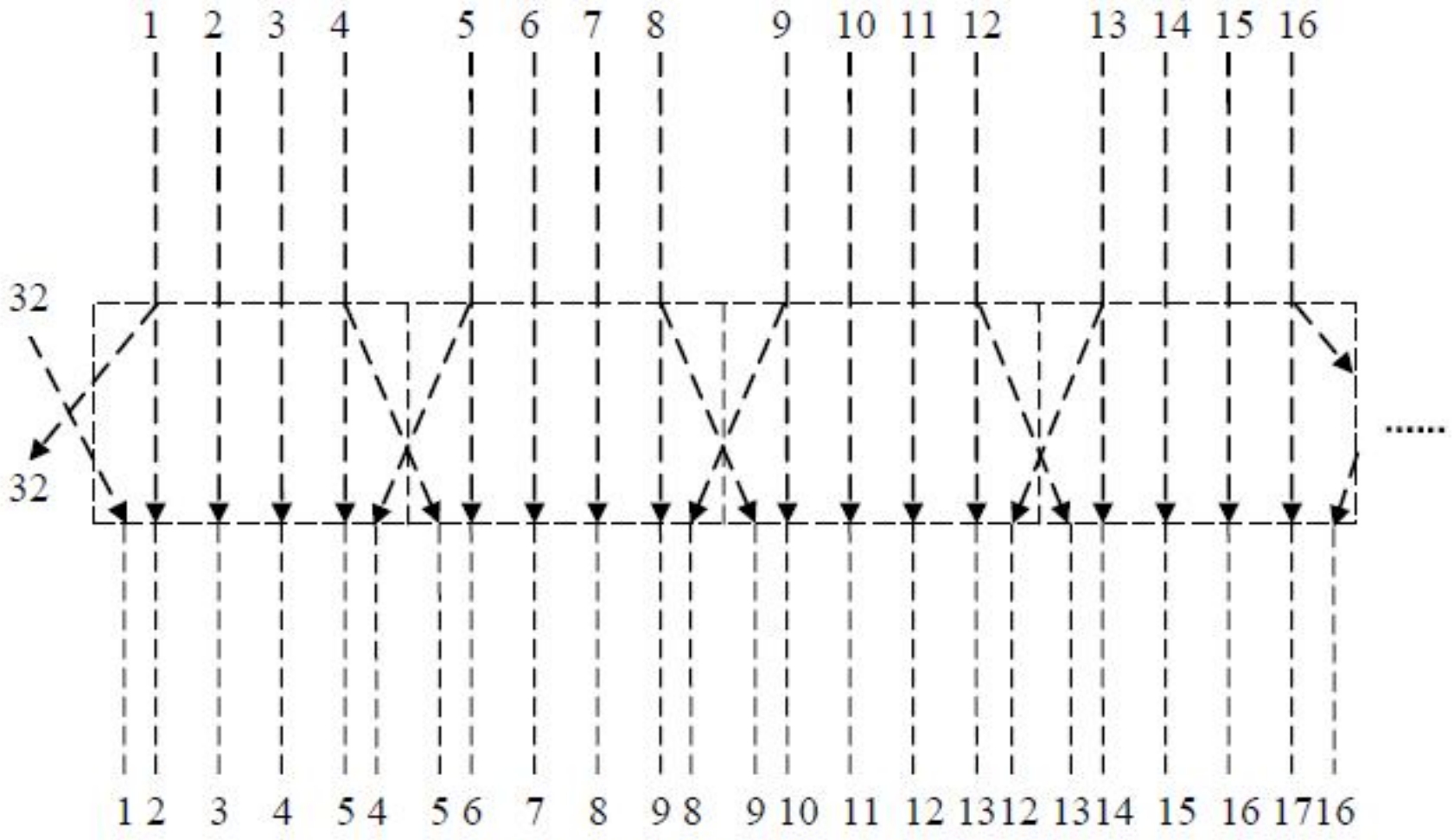
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Bảng 3.12: Hộp S_1

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

Bảng 3.13: Hộp S_2

g.Hàm hoán vị mở rộng E-box



Hình 3.6: Sơ đồ hàm mở rộng (E)

- Hàm E-Box, mỗi 4 bit của khối vào, bit thứ nhất và bit thứ tư tương ứng với 2 bit của đầu ra, trong khi bit thứ 2 và 3 tương ứng với 1 bit ở đầu ra. Bảng sau đây miêu tả vị trí của bit ra so với bit vào.

Bảng mô tả hàm mở rộng (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

Bảng 3.11: Bảng mô tả hàm mở rộng E

h. Hộp P-Box

- Mang tính đơn ánh, nghĩa là một bit đầu vào sẽ cho một bit ở đầu ra, không bit nào được sử dụng hai lần hay bị bỏ qua.

Bảng mô tả hộp P-Box (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

Bảng 3.20: Bảng hoán vị P

3.2.1.3.Giải mã DES

- Có thể sử dụng cùng chức năng để mã hoá giải mã hoặc một khối.
- Lúc này các khoá phải được sử dụng theo thứ tự ngược lại. Nghĩa là, nếu các khoá mã hoá cho mỗi vòng là k_1, k_2, k_3 ,... , k_{15},k_{16} thì các khoá giải mã là k_{16}, k_{15},... , k_3, k_2, k_1.
- Giải thuật để tạo khoá cho mỗi vòng cũng tương tự. Các khoá được dịch phải và số vị trí bit để dịch được lấy theo chiều ngược lại.

3.2.1.4. Các yếu điểm của DES

1. Tính bù

- Nếu ta ký hiệu $\overline{u}$ *là phần bù của u (ví dụ : 0100101 là phần bù của 1011010) thì* DES có tính chất sau:

$$y = DES(x,k) \rightarrow \overline{y} = DES(\overline{x}, \overline{k})$$

- Nếu biết y được mã hoá từ x với khoá K thì ta suy ra được bản mã $\overline{y}$ *được mã hoá từ bản rõ* $\overline{x}$ *với khoá* $\overline{k}$ *.*
- *Tính chất này chính là một yếu điểm* của DES , hacker có thể loại bỏ đi một số khoá phải thử khi tiến hành thử giải mã theo kiểu vét cạn.

2. Khoá yếu

- Khoá yếu là các khoá sinh ra cả 16 khoá con như nhau:

$$K_1 = K_2 = \ldots = K_{15} = K_{16}$$

- Điều đó khiến cho việc mã hóa và giải mã đối với khoá yếu là giống hệt nhau.
- Có tất cả 4 khoá yếu sau:

Khoá yếu (Hex)				C_0	D_0
0101	0101	0101	0101	$\{0\}^{28}$	$\{0\}^{28}$
FEFE	FEFE	FEFE	FEFE	$\{1\}^{28}$	$\{1\}^{28}$
1F1F	1F1F	0E0E	0E0E	$\{0\}^{28}$	$\{1\}^{28}$
E0E0	E0E0	F1F1	F1F1	$\{1\}^{28}$	$\{0\}^{28}$

Bảng 3.22: Các khóa yếu của DES

3. Không gian khóa *K*

- DES có $2^{56} = 10^{17}$ khoá. Nếu chúng ta biết được một cặp "tin/mã" thì chúng ta có thể thử tất cả 10^{17} khả năng này để tìm ra khoá cho kết quả khớp nhất. Giả sử một phép thử mất 10^{-6}s, thì chúng sẽ mất 10^{11}s, tức 7300 năm.
- Với các máy tính được chế tạo theo xử lý song song. Với 10^7 con chipset mã DES chạy song song thì mỗi một con chipset chỉ phải tính toán với 10^{10} phép thử.
- Chipset mã DES ngày nay có thể xử lý tốc độ 4.5×10^7 bit/s tức có thể làm được hơn 10^5 phép mã DES trong một giây

- Vào năm 1976 và 1977, Diffie và Hellman đã ước lượng rằng có thể chế tạo một máy tính chuyên dụng để vét cạn không gian khoá DES trong ½ ngày với cái giá 20 triệu đô la.
- Năm 1984, chipset mã hoá DES với tốc độ mã hoá 256000 lần/giây.
- Năm 1987, đã tăng lên 512000 lần/giây.
- Năm 1993, Michael Wiener đã thiết kế một máy tính chuyên dụng với giá 1 triệu đô la sử dụng phương pháp vét cạn để giải mã DES trung bình trong vòng 3,5 giờ (và chậm nhất là 7 giờ).

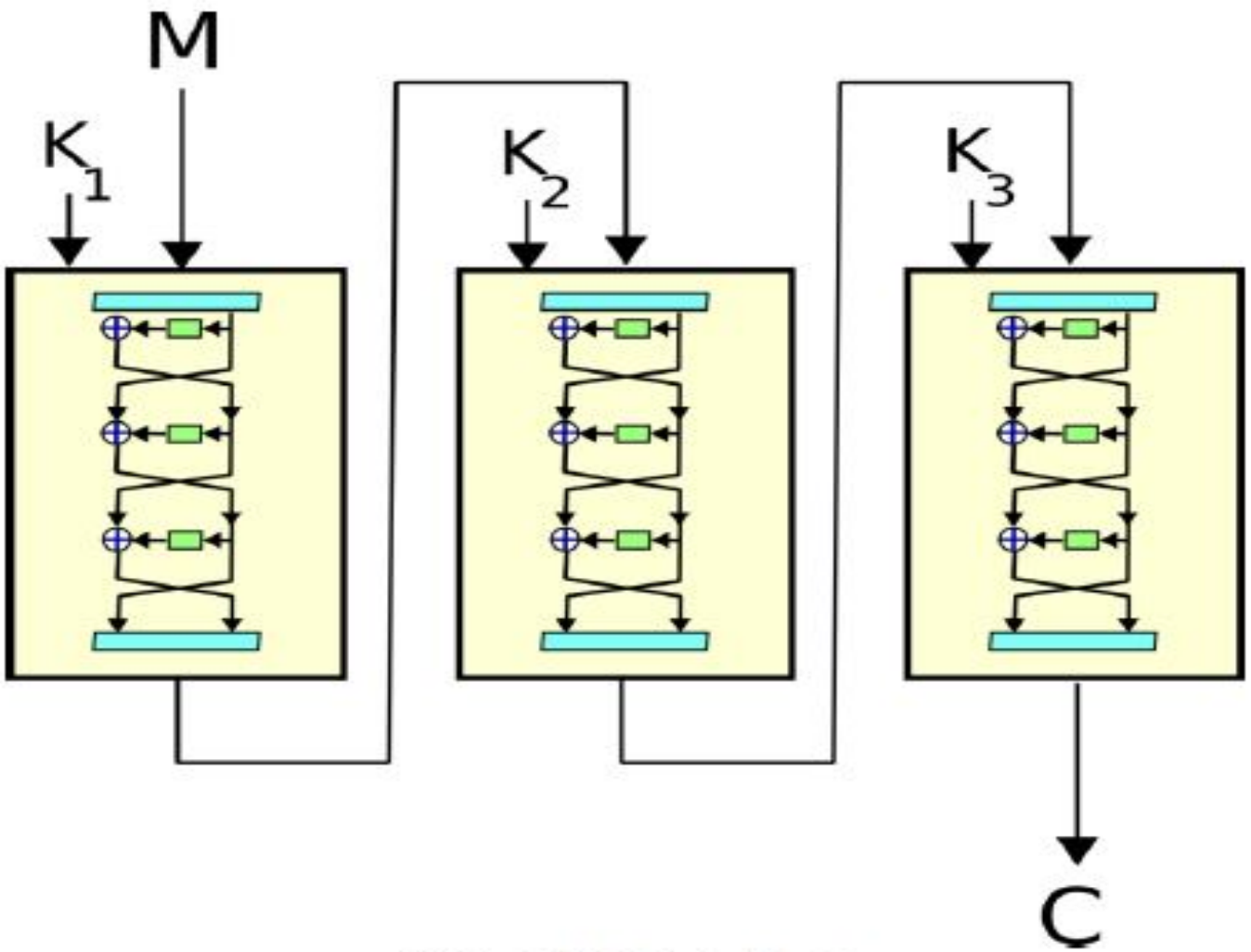
3.2.2.Triple DES

- Hệ mã DES với không gian khóa vẻn vẹn có 2^{54} khóa nên thực tế hiện nay có thể bị thám mã trong khoảng thời gian vài giờ đồng hồ .
- Triple DES (TDES) hay 3DES thực hiện mã hóa DES ba lần (Triple Data Encryption Algorithm).
- Bản mã : $C = DESK_3(DESK_2(DESK_1(M))$,

  Mô hình EEE ,cả ba bước sử dụng ba khóa ở đây đều sử dụng DES ,
- Một biến thể khác của mô hình này gọi là EDE với bước ở giữa sử dụng thuật toán giải mã của DES:

$$C = DES_{K3}(DES^{-1}_{K2}(DES_{k1}(M)).$$

Sơ đồ Triple DES



Hình 3.7: Triple DES

- Khóa của Triple DES là 168 bit, một số biến thể của Triple DES sử dụng khóa có độ dài 112 bit (K1=K3) (Two key Triple DES).
- Các chứng minh về mặt lý thuyết và các tấn công đối với Triple DES cho thấy hệ mã này vẫn sẽ còn được sử dụng trong một tương lai dài mặc dù trên trên thực tế nó chậm hơn so với AES 6 lần.

3.2.3. Thuật toán cao cấp AES

- AES là một hệ mã khóa bí mật có tên là Rijndael (Joan Daemen và Vincent Rijmen) trở thành chuẩn từ năm 2002)
- AES xử lý các khối dữ liệu input có kích thước 128 bit .
- Khóa có độ dài 128, 192 hoặc 256 bit.
- Còn được biết với các tên AES-128, AES-192, AES-256 tương ứng với độ dài khóa sử dụng).
- Chi tiết tham khảo tài liệu.

3.2.4. Các MODE làm việc của hệ mật đối xứng

Dựa vào việc xử lý input data của hệ mã ,cơ chế sử dụng các hệ mã khối sau có 2 loại:

1. Các chế độ khối (Block Mode): xử lý các thông điệp theo các khối (ECB, CBC)

2. Các chế độ luồng, dòng (Stream Modes): xử lý các thông điệp như là một chuỗi bit/byte (CFB, OFB).

- Các chế độ khối sử dụng để mã hóa các dữ liệu mà chúng ta biết trước về vị trí, độ lớn khi mã hóa (các file, các email)
- Chế độ chuỗi được sử dụng cho việc mã hóa các dữ liệu không được biết trước về độ lớn cũng như vị trí như các tín hiệu gửi về từ vệ tinh hoặc các tín hiệu do một bộ cảm biến đọc từ bên ngoài vào.

3.2.4.1. ECB (Electronic CodeBook Book) mode

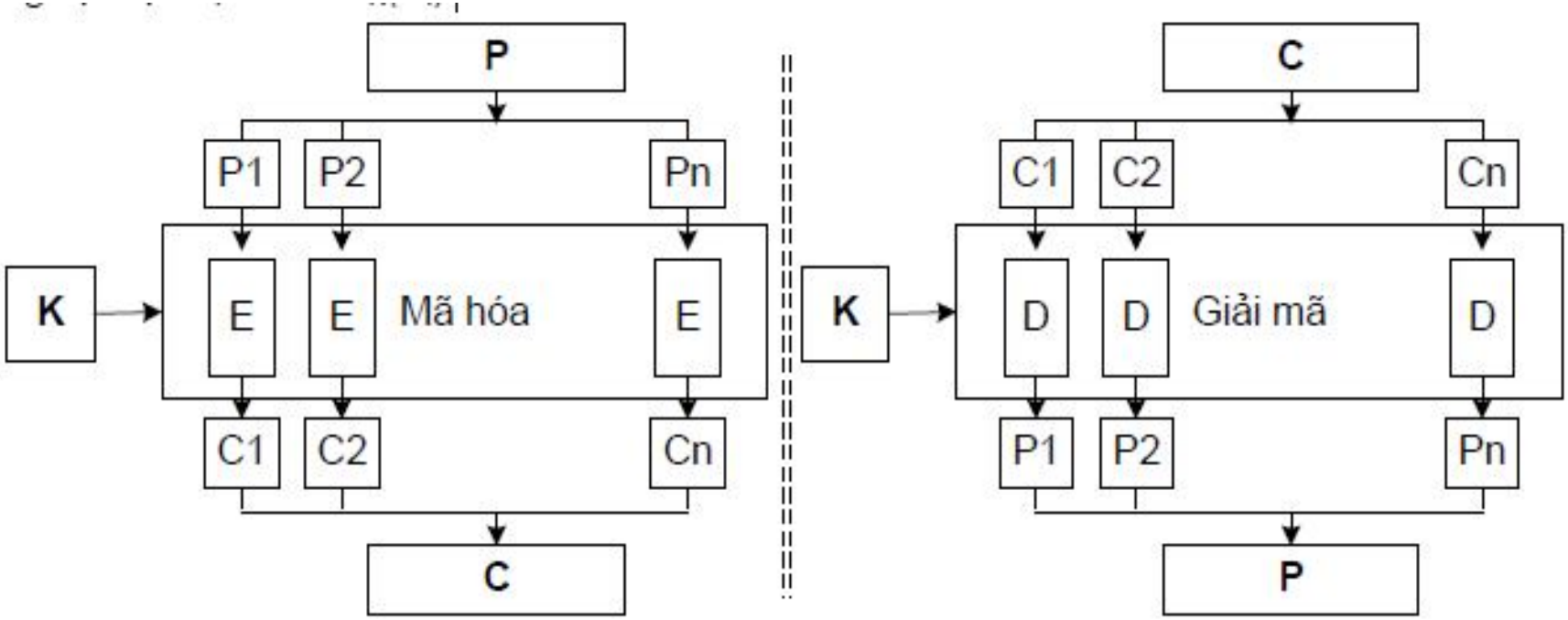
- Thông điệp cần mã hóa được chia thành các khối độc lập để mã hóa, mỗi khối bản mã là kết quả của việc mã hóa riêng biệt khối bản rõ tương ứng với nó và độc lập với khối khác. (giống như thay thế các khối bản mã bằng các khối bản rõ tương ứng nên có tên gọi là bảng tra mã điện tử.

  Plain text : $P = P_1P_2...P_N$
- Mã hóa: $C_i = DES_K(P_i)$,

  Cipher text: $C = C_1C_2..C_N$.
- Giải mã tiến hành ngược lại:

  $P_i = DES^{-1}_K(C_i)$.

ECB MODE



Hình 3.14: Cơ chế ECB

• ECB - đơn giản và dễ cài đặt , sử dụng khi chỉ một khối đơn để mã thông tin cần được gửi đi (ví dụ khóa session key được mã hóa bằng cách dùng một khóa chính).

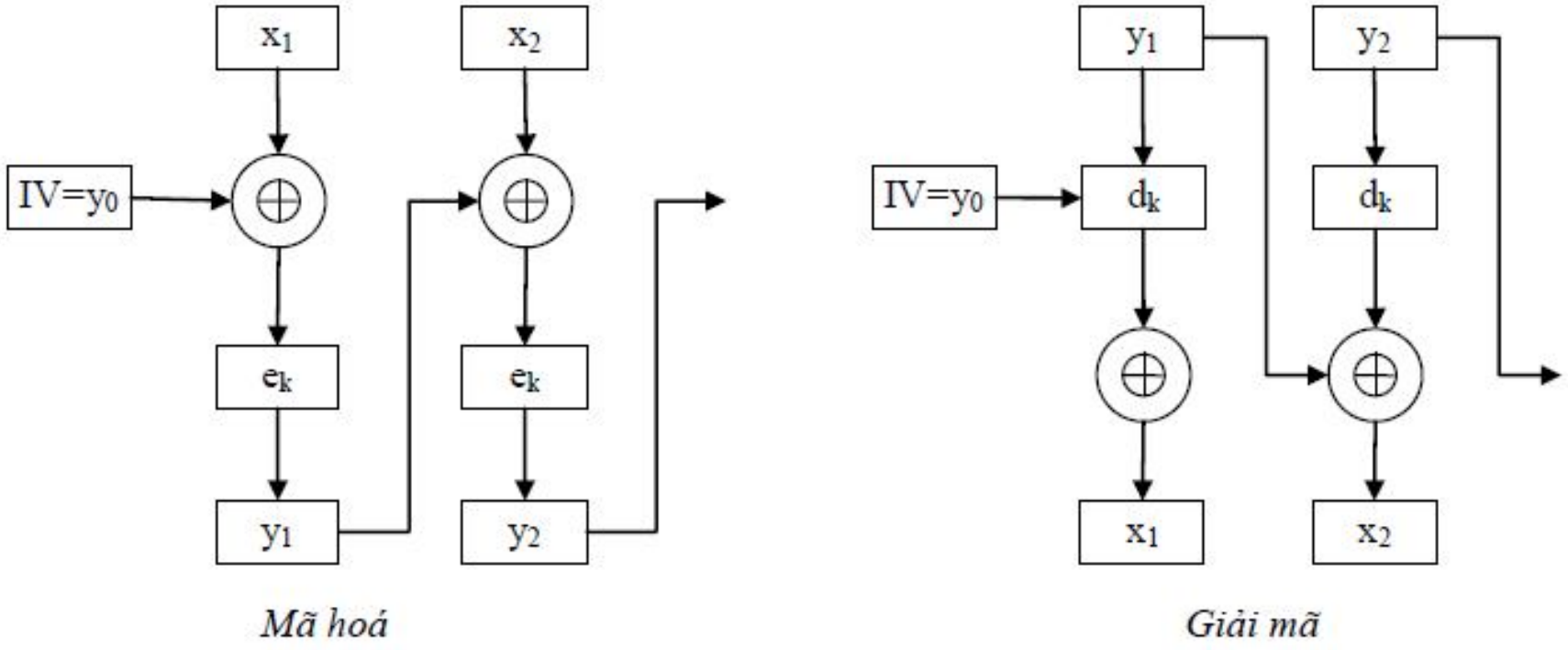
3.2.4.2. CBC (Cipher Block Chaining) MODE

- Giống như EBC mode , trong CBC mode bản rõ sẽ được chia thành các khối nhưng chúng sẽ được liên kết với nhau trong quá trình mã hóa để tạo thành bản rõ. Chính vì các khối bản mã được móc nối với bản rõ nên có tên là CBC mode
- CBC sử dụng một vector khởi tạo IV (Initial Vector) để bắt đầu:

$$C_0 = IV,\ P = P_1P_2..P_N$$

- Mã hóa: $C_i = DES_K\ (P_i \oplus C_{i-1}),\ C = C_1C_2..C_N$
- Giải mã: $P_i = DES^{-1}_K(C_i) \oplus C_{i-1}$, ; $P = P_1P_2..P_N$.

CBC MODE



Hình 3.15: Chế độ CBC

Phù hợp với các dữ liệu có khối lượng lớn như các file , Email , WEB….

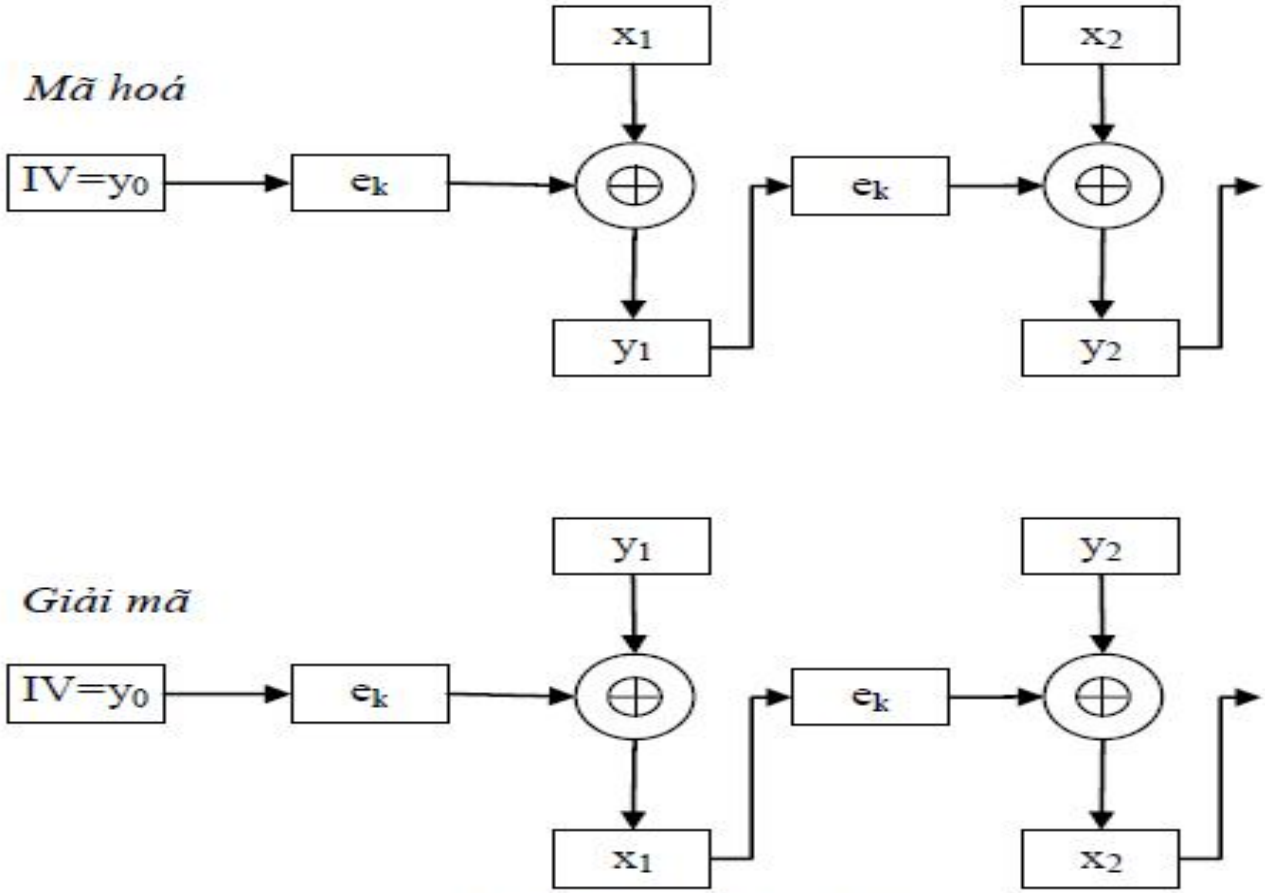
3.2.4.3. CFB (Cipher Feedback) và OFB (Output Feedback) mode

- Các mode CFB và OFB được sử dụng để mã hóa các dữ liệu được cung cấp rời rạc (tín hiệu nhận được từ vệ tinh hoặc do một bộ cảm biến nào đó truyền về).
- Trong chế độ OFB và CFB dòng khoá được tạo ra sẽ được cộng modulo 2 với bản rõ.
- OFB là một hệ mã đồng bô : Lập các vector khởi tạo 64 bit (vector IV). $z_0 = IV$; $z_i = e_k(z_{i-1})$ với $i \geq 1$.

  Mã hóa bản rõ $x_1x_2 \ldots x_n$: $y_i = x_i \oplus z_i$ với $i \geq 1$.
- Mode CFB, tạo $y_0 = IV$ (vector khởi tạo 64 bit) và tạo phần tử z_i của dòng khoá : $z_i = e_k(y_{i-1})$ với $i \geq 1$ và

$$y_i = x_i \oplus z_i \text{ với } i \geq 1.$$

CFB-OFB MODE



Hình 3.16: Chế độ CFB

Kết thúc chương 3