ĐẠI NAM
UNIVERSITY

CHƯƠNG 2: MÃ KHÓA BÍ MẬT

Giảng viên: Nguyễn Văn Nhân

Điện thoại: 0346542854

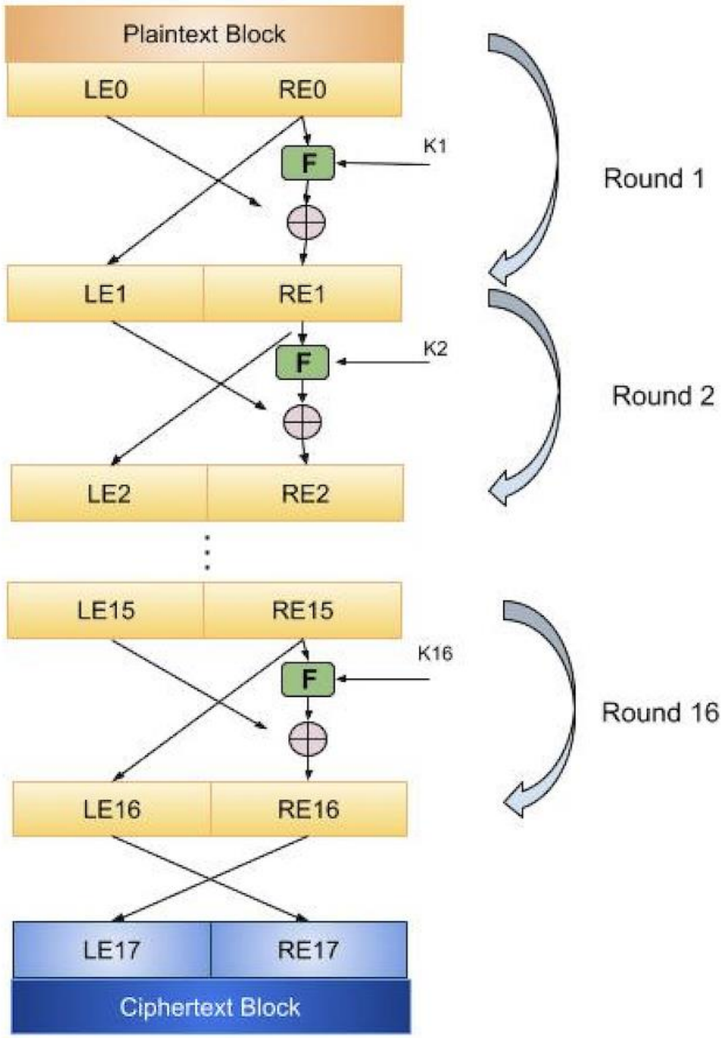
Email: nhannv@dainam.edu.vn

ĐẠI NAM
UNIVERSITY

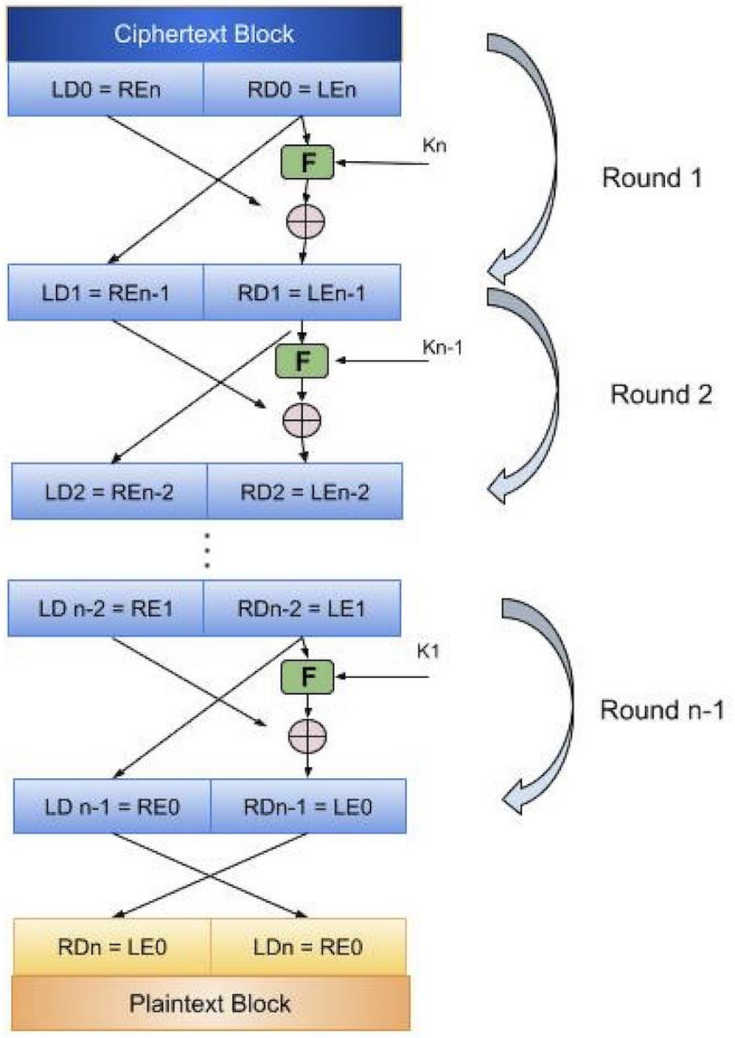
Bài 5

DES và Triple DES

Giới Thiệu Chung về DES (Data Encryption Standard)



Mã khối Feistel



Nội dung DES

1. Giới thiệu

2. Thuật toán DES

3. Thuật toán Triple DES

4. Luyện tập

Nội dung DES

1. Giới thiệu

2. Thuật toán DES

3. Thuật toán Triple DES

4. Luyện tập

Giới thiệu

Thuật toán mã DES (Data Encryption Standard):

- **Là một phương pháp mã khối, đối xứng, được phát triển vào đầu năm 1970**
- **IBM và được Cơ quan An ninh Quốc gia Mỹ (NSA) sửa đổi và công nhận là tiêu chuẩn mã hóa dữ liệu vào năm 1977.**

Giới thiệu

Năm 2000:

- **DES không còn đủ mạnh để bảo vệ thông tin nhạy cảm**

=> AES (Advanced Encryption Standard).

DES là gì?

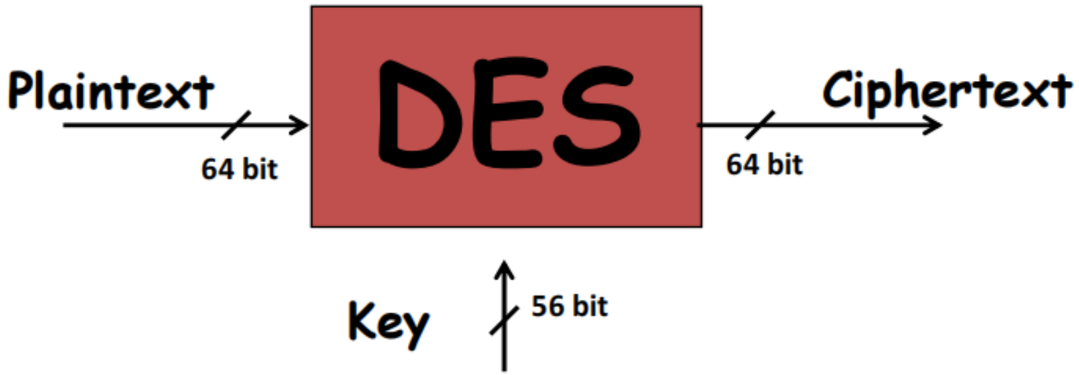
- DES là một thuật toán mã hóa khối, nó mã hóa dữ liệu theo từng khối có kích thước cố định (64-bit).

- DES sử dụng cấu trúc Feistel(16 round) mỗi vòng sử dụng một phiên bản của khóa mã hóa để thực hiện các phép toán trên dữ liệu.

THUẬT TOÁN MÃ HÓA DES

THUẬT TOÁN MÃ HÓA DES

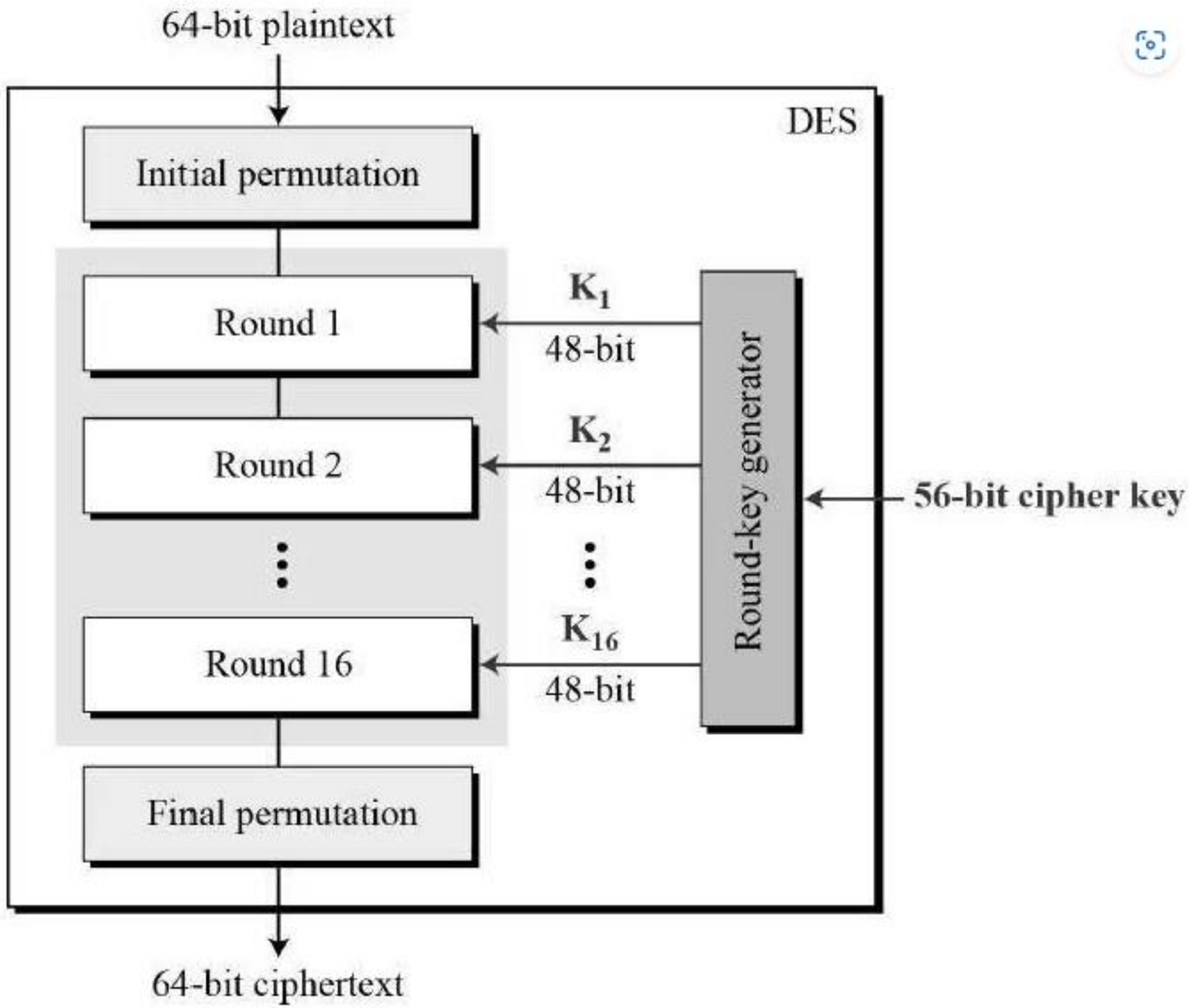
- Block size = 64 bits
- Key size = 56 bits (64 bits, 8 are used as parity-check bits for error control)
- Number of rounds = 16



Các thành phần của DES

- Hoán Vị Ban Đầu (Initial Permutation): Vai trò và cách thực hiện.
- Các Thành Phần của DES: Khóa, bảng S-box, hàm mở rộng (Expansion Function), v.v.
- Số Vòng Lặp (Rounds): DES có 16 vòng – ý nghĩa và cách hoạt động.

Các thành phần và hoạt động của DES



- Kích thước mỗi khối là 64 bits, kích thước khóa là 56 bits
- Tương tự Feistel, DES sử dụng một hàm F chung trong khi mỗi round lại sử dụng một sub-key riêng (được sinh từ master key).

Hoạt động thuật toán của DES

B1: Đầu tiên, khối plaintext (64 bits) được đưa qua bước Initial permutation để thực hiện hoán vị các bit data (Quy tắc hoán vị được định nghĩa trong một bảng gọi là Inital permutaion table).

B2: Sau đó, cho khối chạy 16 rounds để mã hóa, các bước chạy giống như Feistel đã trình bày ở trên.

B3: Cuối cùng, thực hiện bước Final permutation để thu ciphertext (bước này thực chất là đảo ngược của quá trình Initial permutation).

Hoạt động 1

mã hóa chuỗi "ILOVEYOU" bằng DES

B1: khối plaintext (64 bits) được đưa qua bước Initial permutation để thực hiện hoán vị các bit data

"ILOVEYOU" sang nhị phân:

Chuyển từng ký tự sang mã ASCII

- I: 73 (01001001)
- L: 76 (01001100)
- O: 79 (01001111)
- V: 86 (01010110)
- E: 69 (01000101)
- Y: 89 (01011001)
- O: 79 (01001111)
- U: 85 (01010101)

Ghép thành chuỗi nhị phân

64 bit (plaintext): 01001001 01001100 01001111 01010110 01000101 01011001 01001111 01010101

Hoạt động 1

B1: khối plaintext (64 bits) được đưa qua bước **Initial permutation để thực hiện hoán vị các bit data**

01001001 01001100 01001111 01010110 01000101 01011001 01001111 01010101

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

Idea of IP

Bit position in the plain-text block	To be overwritten with the contents of the bit position
1	58
2	50
3	42
....	....
64	7

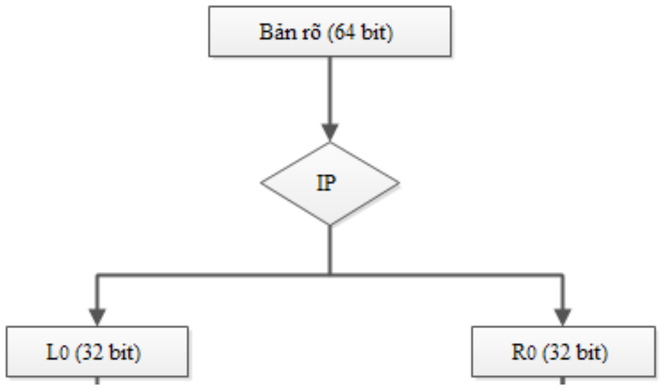
Sau IP khối này được chia thành hai nửa 32 bit: **L0** (nửa trái) | **R0** (nửa phải)

10000101 01010100 00110011 11001010 01010100 11001010 00110011 01010100

Hoạt động 1

Bảng hoán vị ban đầu (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

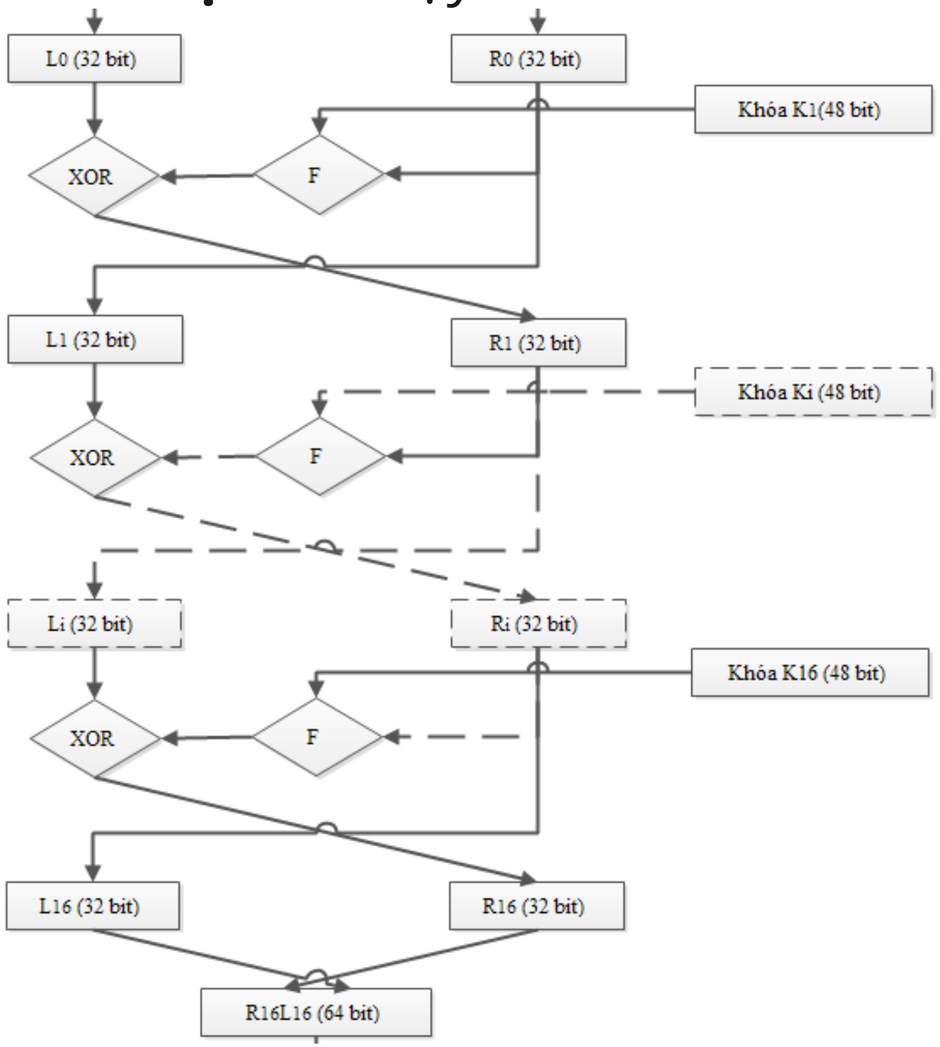


- Bảng này có 8 hàng và 8 cột, tổng cộng 64 ô, tương ứng với 64 bit.
- Các số trong bảng là vị trí của bit trong khối đầu vào (tính từ 1 đến 64, từ trái sang phải).
- Kết quả đầu ra sẽ là một khối 64 bit mới, với các bit được sắp xếp lại theo thứ tự trong bảng.

Sau IP khối này được chia thành hai nửa 32 bit: **L0** (nửa trái) | **R0** (nửa phải)

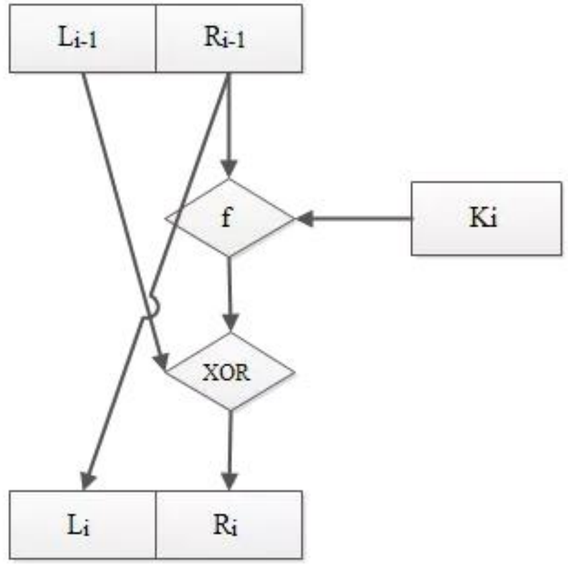
Hoạt động 2

Giai đoạn 2: Chạy 16 rounds để mã hóa



Bước 2: Sinh khóa con - Đặc điểm DES

- **Feistel cổ điển**: Khóa có thể cố định hoặc đơn giản, không quy định cơ chế sinh khóa.
- **DES**: Sử dụng thuật toán sinh khóa phức tạp (PC-1, dịch vòng, PC-2) để tạo 16 khóa con Ki từ khóa chính.



Hoạt động 2

Sinh khóa con K1 (ví dụ minh họa):

1.PC-1: Lọc 64 bit thành 56 bit (bỏ 8 bit kiểm tra chẵn lẻ):

```
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
```

Kết quả:
1111000 0110011 0101010 1100110 1001100 1111100 0011111 (56 bit).

2. Chia đôi và dịch vòng: Chia thành C0 (28 bit) và D0 (28 bit), dịch trái 1 bit:

- C0 = 1111000 0110011 0101010 1100 → C1 = 1110000 1100110 1010101 1001
- D0 = 110 1001100 1111100 0011111 → D1 = 101 0011001 1111000 0111111

3. PC-2: Chọn 48 bit từ 56 bit:

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

Kết quả K1 (giả định):
110011 001100 111100 000011 101010 010101.

Hoạt động 2

Sinh khóa con K1 (ví dụ minh họa):

1.PC-1: Lọc 64 bit thành 56 bit (bỏ 8 bit kiểm tra chẵn lẻ):

3. PC-2: Chọn 48 bit từ 56 bit:

2. Chia đôi và dịch vòng: Chia thành C0 (28 bit) và D0 (28 bit), dịch trái 1 bit:

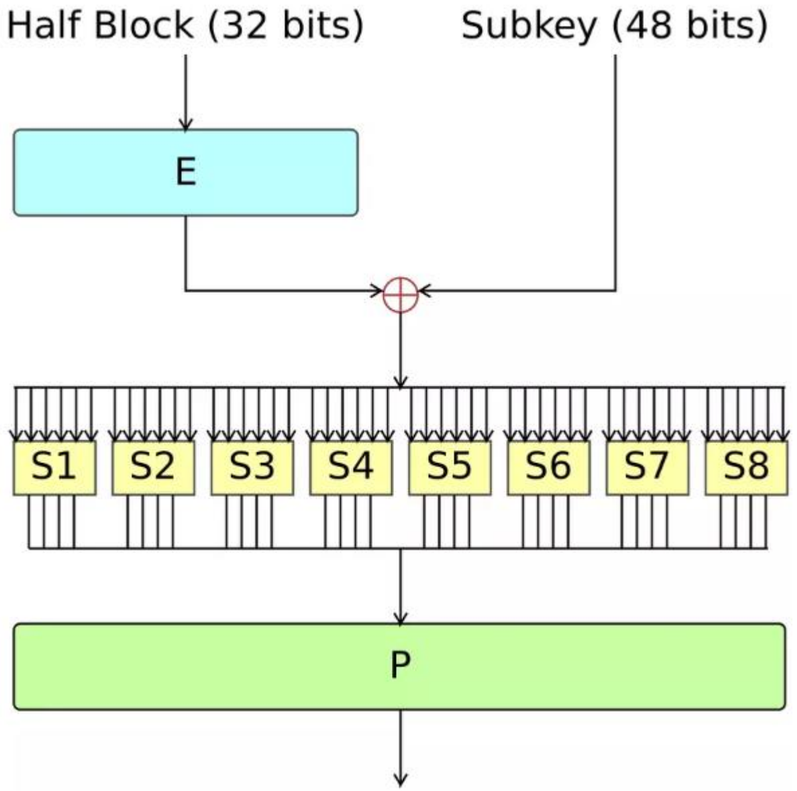
- C0 = 1111000 0110011 0101010 1100 → C1 = 1110000 1100110 1010101 1001
- D0 = 110 1001100 1111100 0011111 → D1 = 101 0011001 1111000 0111111

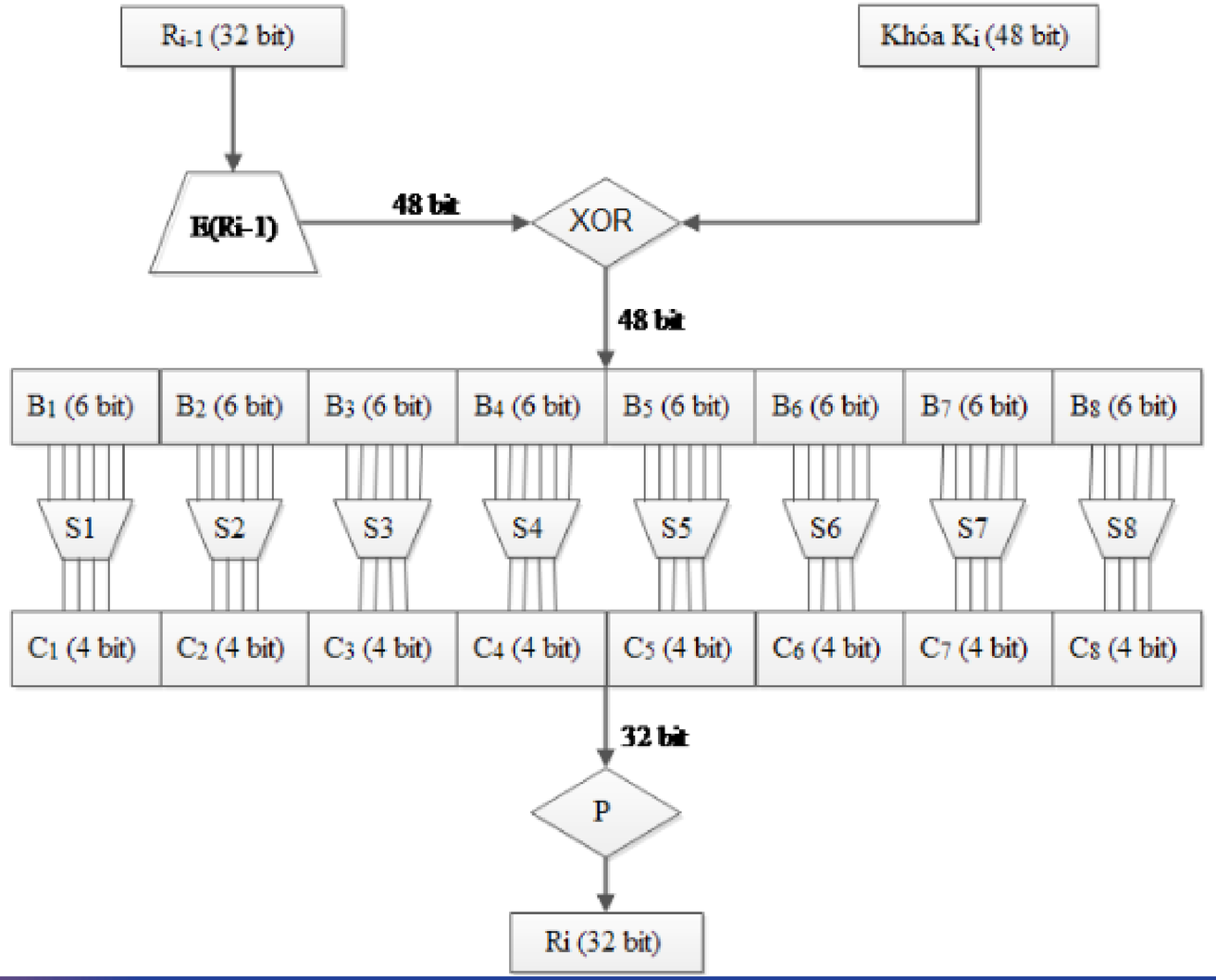
Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
#key bits shifted	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Figure - number of key bits shifted per round

Hoạt động 2

Hàm Feistel F của DES





Hàm Feistel F:

- **Expansion (E-box)**:
- R0 = 01010100 11001010 00110011 01010100
- Bảng E: 32 1 2 3 4 5 4 5 6 7 8 9 ...
- Kết quả 48 bit (giả định): 001010 101001 110010 100011 001101 010100.

- **XOR với K1**:
- K1 = 110011 001100 111100 000011 101010 010101
- XOR: 001010 101001 110010 100011 001101 010100 XOR K1 = 111001 100101 001110 100000 100111 000001.

Hàm Feistel F:

- **S-box**:
- Chia thành 8 nhóm 6 bit, qua 8 S-box (mỗi S-box cho 4 bit đầu ra).
- Ví dụ S1 với 111001: Hàng 11 (3), cột 1100 (12) → Giá trị 5 (0101).
- Kết quả 32 bit (giả định): 0101 1100 0011 1111 1010 0101 1100 0011.

- **P-box**:
- Bảng P: 16 7 20 21 ...
- Kết quả: 1100 0011 0101 1100 1010 1111 0011 0101.

Hàm Feistel F:

1.Cập nhật:

1.L1 = R0 = 01010100 11001010 00110011 01010100

2.R1 = L0 XOR f = 10000101 01010100 00110011 11001010 XOR 1100 0011 0101 1100 1010 1111 0011 0101 = 01000110 00010011 01100100 10000011.

Lặp lại 16 vòng với các K2, K3,..., K16.

Ý nghĩa: Hàm f phức tạp (E → XOR → S-box → P) là điểm nổi bật của DES => làm tăng độ khuếch tán (diffusion) và nhầm lẫn (confusion) so với Feistel cổ điển chỉ có XOR

Sau 16 vòng, hai nửa L16 và R16 được ghép lại (theo thứ tự R16 || L16).

Final Permutation (FP)

- **Feistel cổ điển**: Không có hoán vị cuối.
- **DES**: Dùng bảng FP cố định để hoàn thiện ciphertext.

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

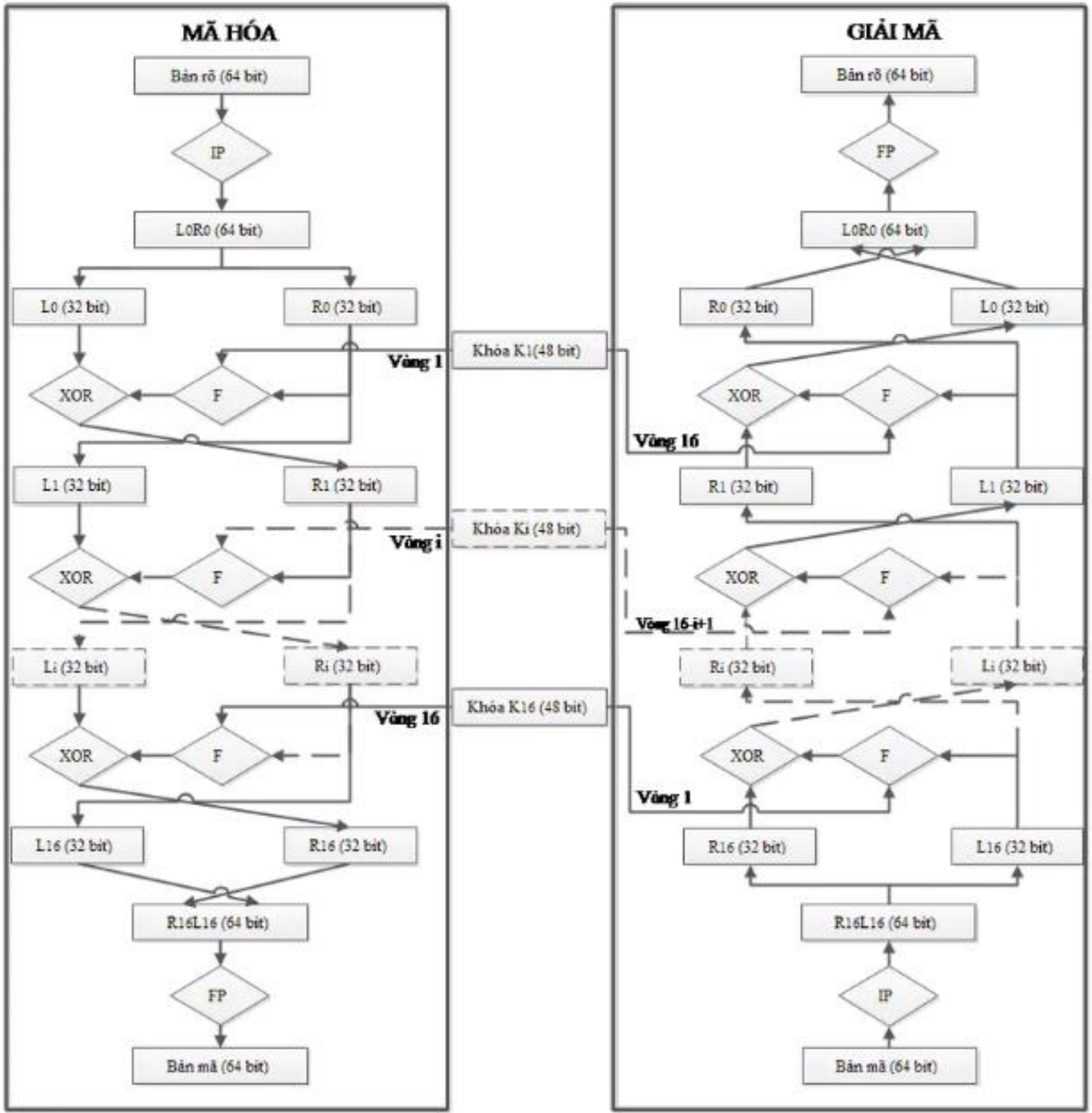
Bản mã (ciphertext) của "ILOVEYOU"

Khóa 0x133457799BBCDDFF

=> d2e5e5d5f6f1c2a2 (dạng hex).

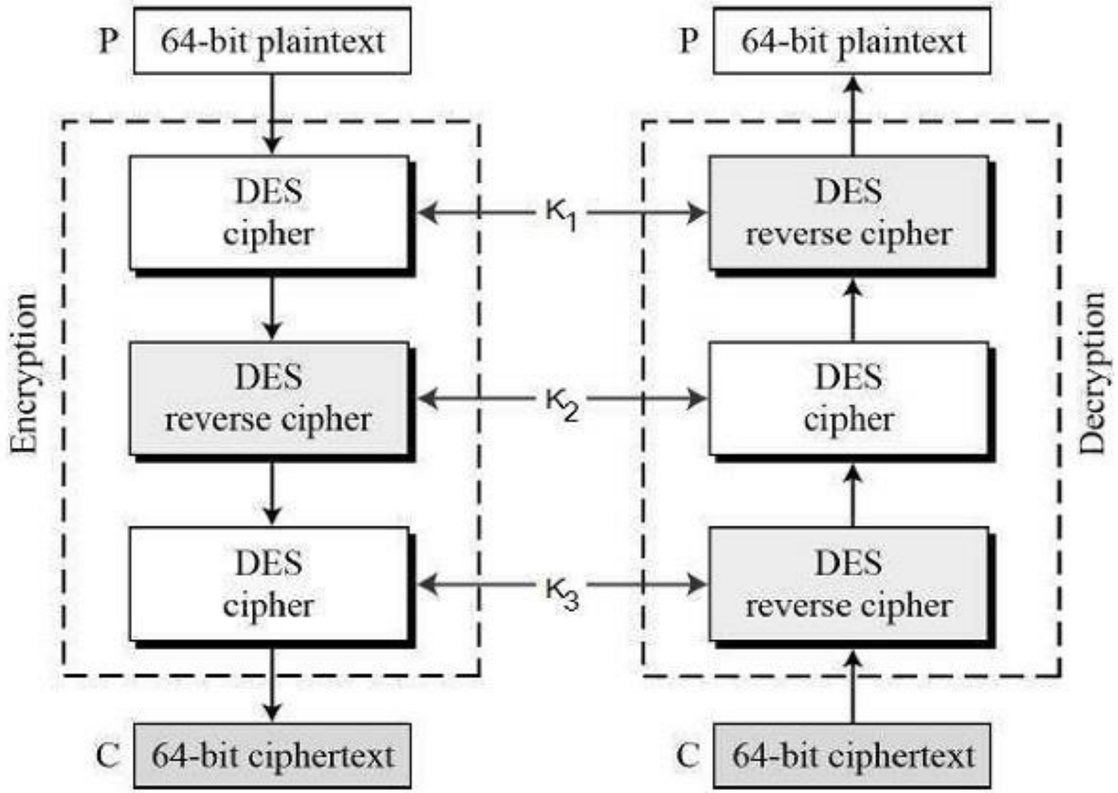
11010010 11100101 11100101 11010101
11110110 11110001 11000010 10100010

Giải mã


MÃ HÓA
Bản rõ (64 bit)
IP
L0R0 (64 bit)
L0 (32 bit)
R0 (32 bit)
Vòng 1
XOR
F
L1 (32 bit)
R1 (32 bit)
Vòng i
XOR
F
Li (32 bit)
Ri (32 bit)
Vòng 16
XOR
F
L16 (32 bit)
R16 (32 bit)
R16L16 (64 bit)
FP
Bản mã (64 bit)
Khóa K1(48 bit)
Khóa Ki (48 bit)
Khóa K16 (48 bit)
GIẢI MÃ
Bản rõ (64 bit)
FP
L0R0 (64 bit)
R0 (32 bit)
L0 (32 bit)
Vòng 16
XOR
F
R1 (32 bit)
L1 (32 bit)
Vòng 16-i+1
XOR
F
Ri (32 bit)
Li (32 bit)
Vòng 1
XOR
F
R16 (32 bit)
L16 (32 bit)
R16L16 (64 bit)
IP
Bản mã (64 bit)

Các Biến Thể của DES

- Tổng quan về các biến thể (Triple DES).
- Mục đích và sự khác biệt so với DES gốc.
- Giới thiệu Triple DES: sử dụng ba khóa DES để cải thiện bảo mật.



- **Cơ chế**: Dữ liệu được mã hóa/giải mã qua 3 giai đoạn DES:
  - Mã hóa bằng khóa K1.
  - Giải mã bằng khóa K2.
  - Mã hóa lại bằng khóa K3.

$$C = E_{K3}(D_{K2}(E_{K1}(P)))$$

Khóa Key: Tổng cộng 168-bit (3 x 56-bit)

DES Hoạt Động Như Thế Nào Trong Bảo Mật Thông Tin

Ứng dụng thực tế của DES trong mã hóa dữ liệu (trước đây).
Ví dụ minh họa trong hệ thống ngân hàng hoặc truyền thông.

Các Vấn Đề Thiết Kế của DES

Điểm yếu: kích thước khóa ngắn (56-bit), dễ bị tấn công brute-force.
Hạn chế dẫn đến sự ra đời của Triple DES và AES.

Tổng kết


SUMMARY
BRIEF
FACTS
COMPENDIOUS
CONCISE
DESCRIPTIVE
COMPACT
BULLETIN
INFORMATION
ABRIDGMENT
ESSENCE
DIGEST
EXTRACT
EPITOME
SHORT
REVIEW
LENGTH
RESUME
IDEA

ĐẠI NAM
UNIVERSITY

Thank You