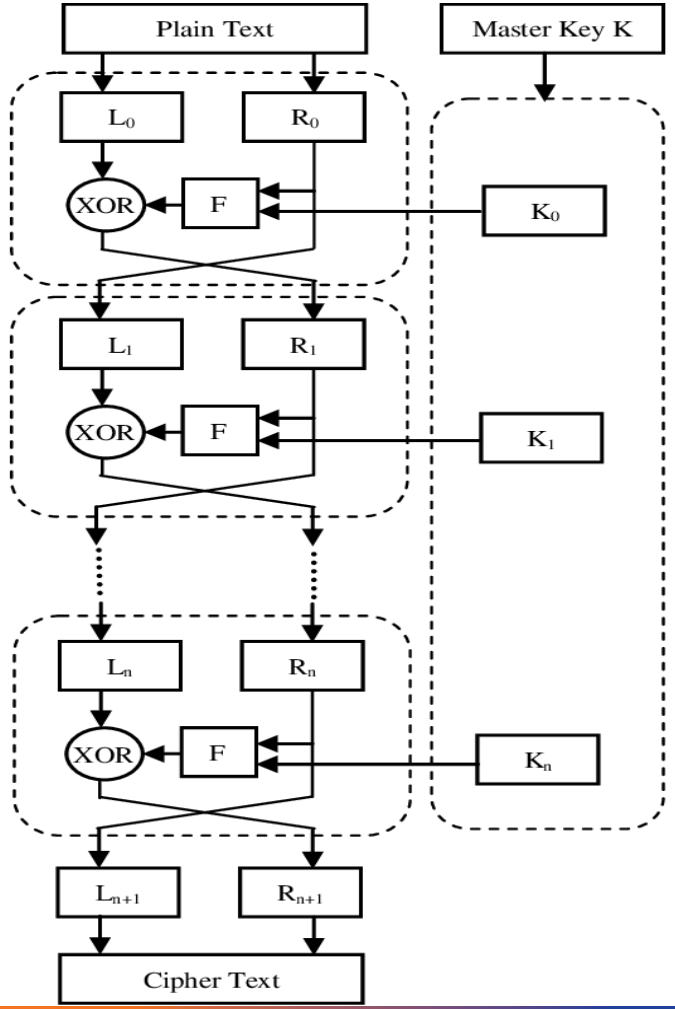
ĐẠI NAM UNIVERSITY

Bài 4

THUẬT TOÁN DES

Giảng viên: TS. Trần Quý Nam
(namtq@dainam.edu.vn)



NHẮC LẠI BÀI CŨ

Mã
khối
Feistel

OUTLINE

1. Mã TinyDES
2. Mã hoá DES
3. Double / Triple DES
4. Thực hành

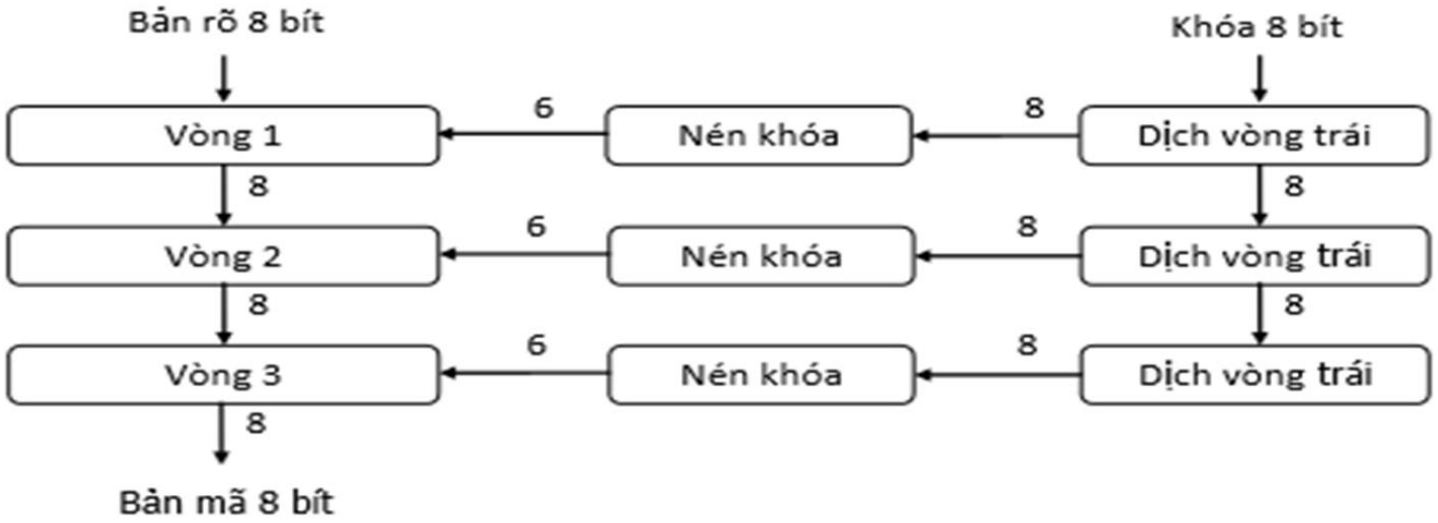
Mã TinyDES

Mã TinyDES

- Mã TinyDES có các tính chất sau:
  - Thuộc hệ mã Feistel gồm 3 vòng
  - Kích thước của khối là 8 bít
  - Kích thước khóa là 8 bít
  - Mỗi vòng của TinyDES dùng khóa con có kích thước 6 bít được trích ra từ khóa chính.

Mã TinyDES

Hình dưới đây minh họa các vòng của mã TinyDES



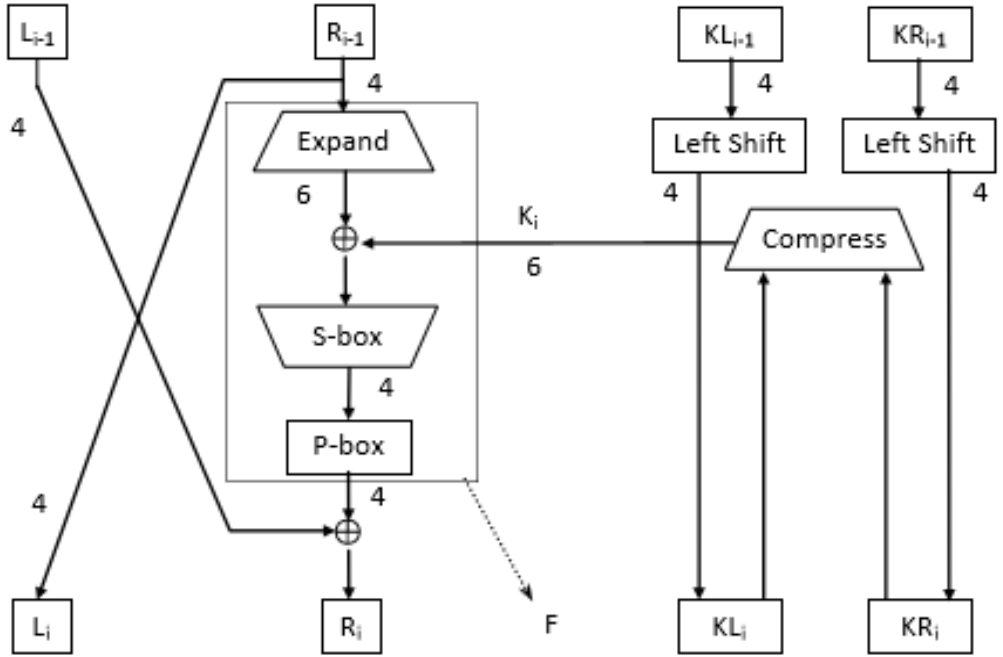
Hình . Các vòng Feistel của mã TinyDES

Sơ đồ mã TinyDES trên gồm hai phần, phần thứ nhất là các vòng Feistel, phần thứ hai là thuật toán sinh khóa con.

Chúng ta sẽ lần lượt đi vào chi tiết của từng phần.

Mã TinyDES

Hình bên minh họa một vòng Feistel của TinyDES



Hình Cấu trúc một vòng của mã TinyDES

Trong TinyDES, hàm F của Feistel là:

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

Mã TinyDES

❖Trong đó hàm Expand vừa mở rộng vừa hoán vị R_{i-1} từ 4 bít lên 6 bít.

❖Hàm S-boxes biến đổi một số 6 bít đầu vào thành một số 4 bít đầu ra.

❖Hàm P-box là một hoán vị 4 bít.

❖Mô tả của các hàm trên là như sau:

- Expand: gọi 4 bít của R_{i-1} là: **$b_0b_1b_2b_3$**.
- Hàm Expand hoán vị và mở rộng 4 bít thành 6 bít cho ra kết quả: **$b_2b_3b_1b_2b_1b_0$**. Ví dụ: **$R_0 = 0110$** Expand(R_0) = **101110**
- S-box: Gọi **$b_0b_1b_2b_3b_4b_5$** là 6 bít đầu vào của S-box, ứng với mỗi trường hợp của 6 bít đầu vào sẽ có 4 bít đầu ra. Việc tính các bít đầu ra dựa trên bảng sau:

Mã TinyDES

b_0b_5 \ $b_1b_2b_3b_4$	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1110	0100	1101	0001	0010	1111	1011	1000	0011	1010	0110	1100	0101	1001	0000	0111
01	0000	1111	0111	0100	1110	0010	1101	0001	1010	0110	1100	1011	1001	0101	0011	1000
10	0100	0001	1110	1000	1101	0110	0010	1011	1111	1100	1001	0111	0011	1010	0101	0000
11	1111	1100	1000	0010	0100	1001	0001	0111	0101	1011	0011	1110	1010	0000	0110	1101

Mã TinyDES

b_0b_5 \ $b_1b_2b_3b_4$	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7
1	0	F	7	4	E	2	D	1	A	6	C	B	9	5	3	8
2	4	1	E	8	D	6	2	B	F	C	9	7	3	A	5	0
3	F	C	8	2	4	9	1	7	5	B	3	E	A	0	6	D

Ví dụ: X = 101010. Tra bảng ta có S-box(X) = 0110.

Mã TinyDES

P-box:

Thực hiện hoán vị 4 bít đầu $b_0b_1b_2b_3$ cho ra kết quả $b_2b_0b_3b_1$.

Thuật toán sinh khóa con của TinyDES

Khóa K 8 bít ban đầu được chia thành 2 nửa trái phải KL_0 và KR_0 , mỗi nửa có kích thước 4 bít.

- Tại vòng thứ nhất KL_0 và KR_0 được dịch vòng trái 1 bít để có được KL_1 và KR_1.
- Tại vòng thứ hai KL_1 và KR_1 được dịch vòng trái 2 bít để có được KL_2 và KR_2.
- Tại vòng tại vòng thứ 3 KL_2 và KR_2 được dịch vòng trái 1 bít để có KL_3 và KR_3.

Mã TinyDES

Cuối cùng khóa K_i của mỗi vòng được tạo ra bằng cách hoán vị và nén (compress) 8 bít của: KL_i và KR_i

$(k_0k_1k_2k_3k_4k_5k_6k_7)$

Thành kết quả gồm 6 bít : $k_5k_1k_3k_2k_7k_0$.

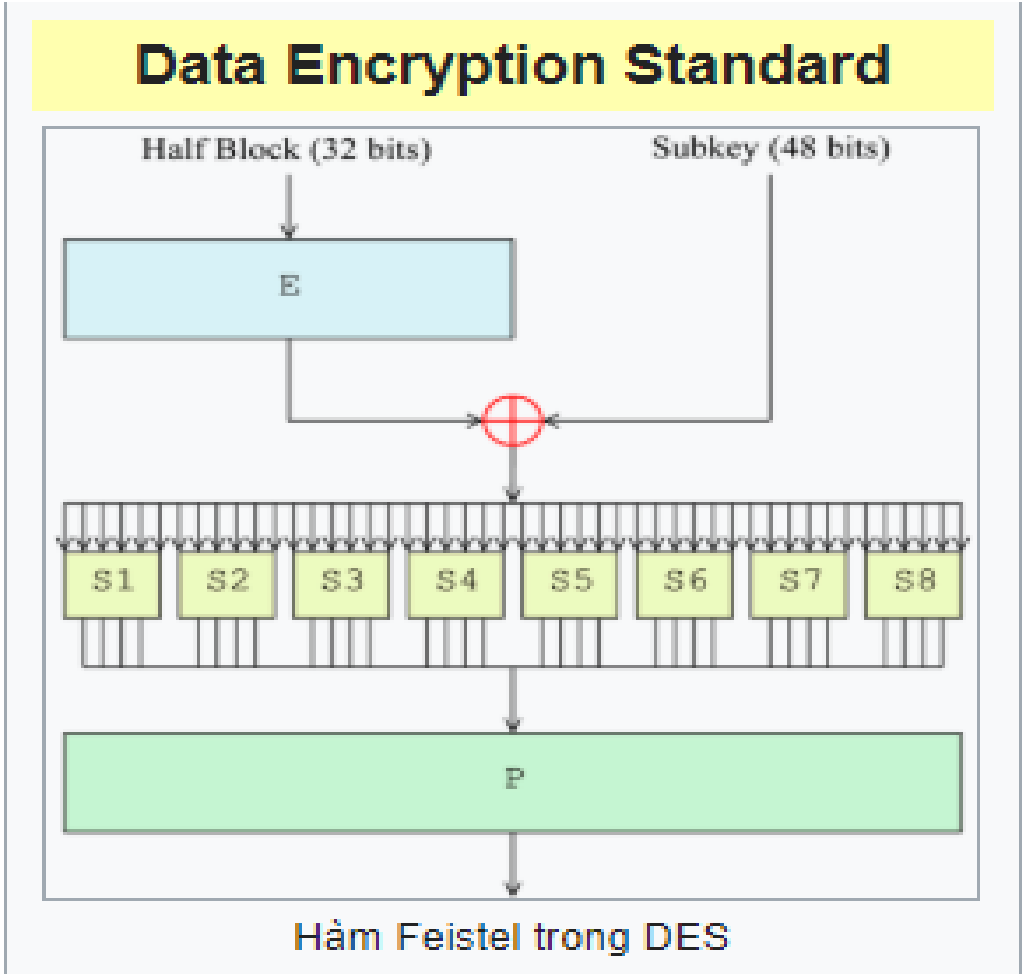
Mã DES

GIỚI THIỆU

- **DES (Data Encryption Standard)** là một thuật toán mã hoá khối (block cipher) được phát triển vào đầu năm 1970 bởi 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.
- DES là một thuật toán mã hoá đối xứng (symmetric-key algorithm), sử dụng cùng một khoá để mã hoá và giả mã dữ liệu.
- DES hoạt động trên các khối dữ liệu có kích thước đố định (64 bit) và sử dụng khoá có độ dài 56 bit

GIỚI THIỆU

- DES hoạt động trên các khối dữ liệu có kích thước đố định 64 bit.
- Mỗi khối được mã hoá bằng một khoá 56 bit + 8 bit để kiểm tra tính chẵn lẻ.
- DES sử dụng cấu trúc Feistel: chia khối dữ liệu thành 2 nửa (mỗi nửa 32 bit) và thực hiện 16 vòng để biến đổi dữ liệu.
- Mỗi vòng sử dụng một phiên bản của khoá mã hoá để thực hiện các phép toán trên dữ liệu.



Hàm Feistel trong DES

GIỚI THIỆU

Là một phương pháp mật mã hóa được FIPS (Tiêu chuẩn Xử lý Thông tin Liên bang Hoa Kỳ) chọn làm chuẩn chính thức vào năm 1977.

GIỚI THIỆU

Mã DES có các tính chất sau:

- 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.*
- Kích thước khóa là 56 bít
- Mỗi vòng của DES dùng khóa con có kích thước 48 bít được trích ra từ khóa chính.

GIỚI THIỆU

❖ Cho đến năm 2001, DES không còn đủ mạnh để bảo vệ thông tin và nó được thay thế bởi thuật Advanced Encryption Standard (AES) do hai nhà mật mã học người Bỉ, Joan Daemen và Vincent Rijmen phát triển.

DES

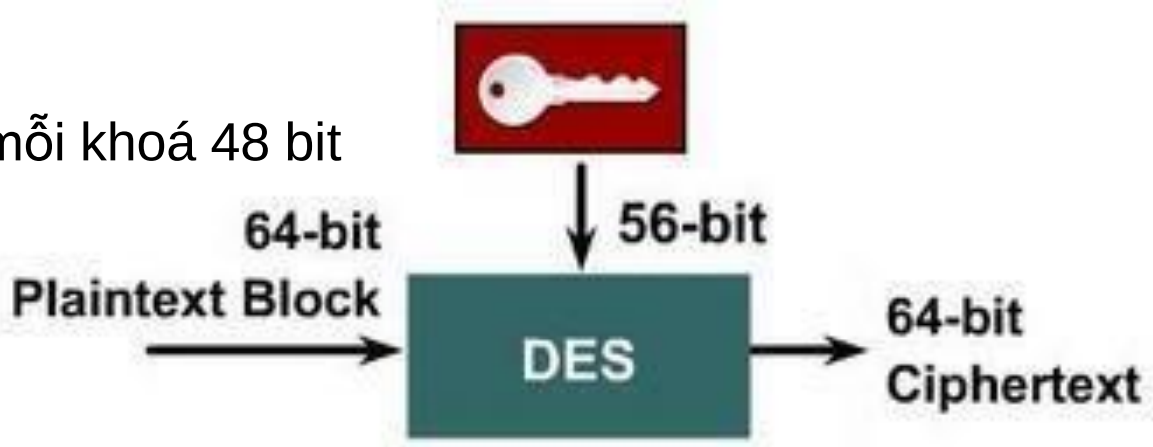
Demonstration

https://simewu.com/des/

DES

Đặc điểm

- Kích thước khối = 64 bit
- Kích thước khoá = 56 bit (64 bit, 8 bit được sử dụng để làm bit kiểm tra chẵn lẻ nhằm kiểm soát lỗi)
- Số vòng = 16
- 16 khoá con, mỗi khoá 48 bit



MÔ TẢ THUẬT TOÁN DES

- DES nhận vào một thông điệp M với 64 bit, một khóa K 56 bit và cho ra một bảng mã C là 64 bit.
- Bước 1: áp dụng một phép hoán vị(bit khởi tạo IP vào M cho ra M': M'=IP(M).
- Bước 2, chia M' thành hai phần: nửa trái L0 = 32 bit và nửa phải R_0=32 bit.
- Bước 3, thi hành các phép toán sau với i = 1, 2,…16 (có 16 vòng).

  Li = Ri-1

  Ri = Li-1 $\oplus$ f(Ri-1, Ki)

- Cuối cùng hoán vị với phép hoán vị(IP^{-1}) để được bản mã cuối cùng C.

CÁC BƯỚC CHI TIẾT HĐ DES

❖ **Bước 1: Chuẩn bị dữ liệu và khóa**

- Dữ liệu vào: Chia dữ liệu thành các khối 64-bit.
- Khóa: DES sử dụng khóa 64-bit, nhưng thực tế chỉ có 56-bit là được dùng (8-bit còn lại là bit kiểm tra chẵn lẻ).

❖ **Bước 2: Hoán vị ban đầu (Initial Permutation - IP)**

- Hoán vị dữ liệu theo một bảng cố định để phân tán các bit trong khối dữ liệu.

CÁC BƯỚC HĐ THUẬT TOÁN DES

❖ **Bước 3: Chia dữ liệu thành 2 phần**

- Sau IP, dữ liệu được chia thành hai phần bằng nhau:

  L0: 32-bit bên trái

  R0: 32-bit bên phải

- Sau đó, dữ liệu sẽ đi qua **16 vòng Feistel**.

CÁC BƯỚC HĐ THUẬT TOÁN DES

❖ **Bước 4: 16 vòng Feistel.** Mỗi vòng Feistel gồm các bước:

1. Mở rộng R (Expansion - E): Biến R (32-bit) thành 48-bit bằng cách nhân đôi một số bit.
2. XOR với khóa con: Mỗi vòng sử dụng một khóa con 48-bit, được trích xuất từ khóa chính 56-bit bằng thuật toán sinh khóa.
3. Hộp S (Substitution - S-Box): Chuyển đổi 48-bit thành 32-bit thông qua 8 bảng S-Box.
4. Hoán vị P (Permutation - P-Box): Xáo trộn 32-bit theo một bảng hoán vị cố định.
5. Kết hợp với L: Giá trị mới được XOR với phần L của vòng trước đó.

➔ **Sau 16 vòng, ta thu được L16 và R16.**

CÁC BƯỚC HĐ THUẬT TOÁN DES

❖ Bước 5: Hoán vị cuối cùng (Final Permutation - FP)

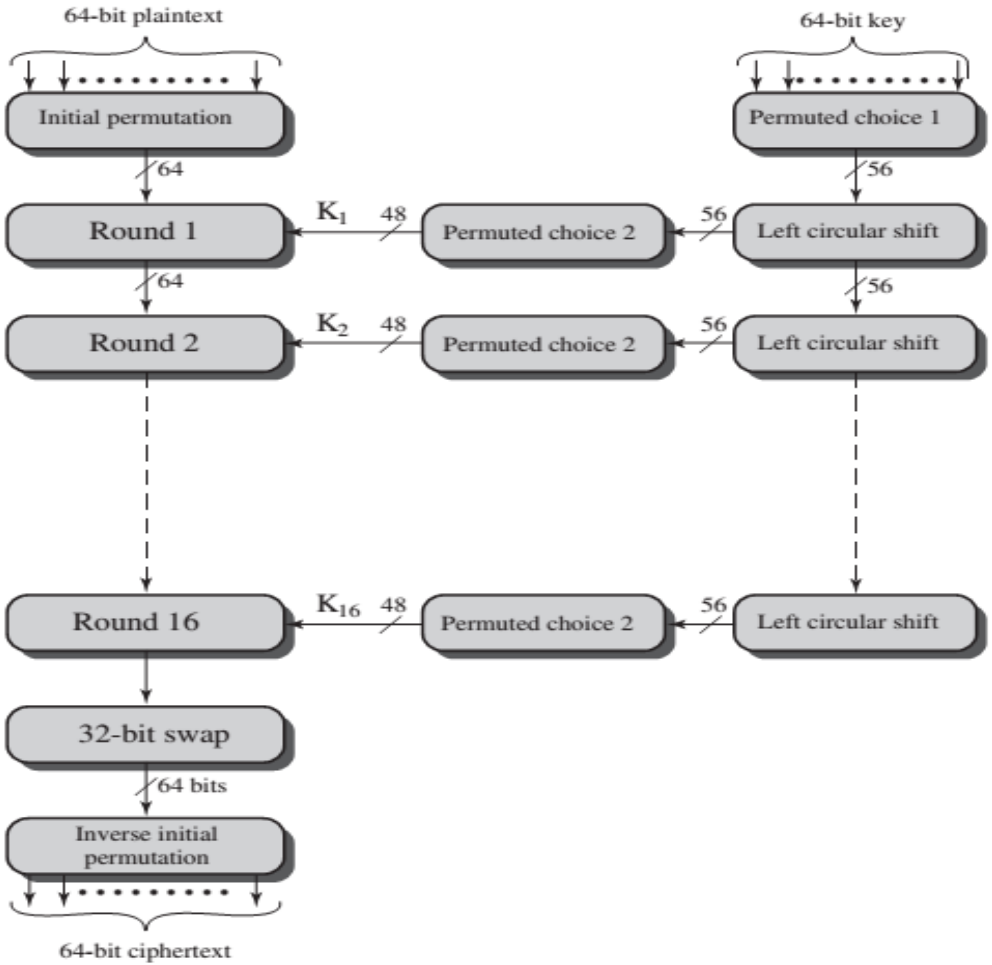
- Kết hợp L16 và R16, rồi thực hiện một hoán vị cuối cùng để tạo ra khối mã hóa 64-bit.

ĐÁNH GIÁ THUẬT TOÁN

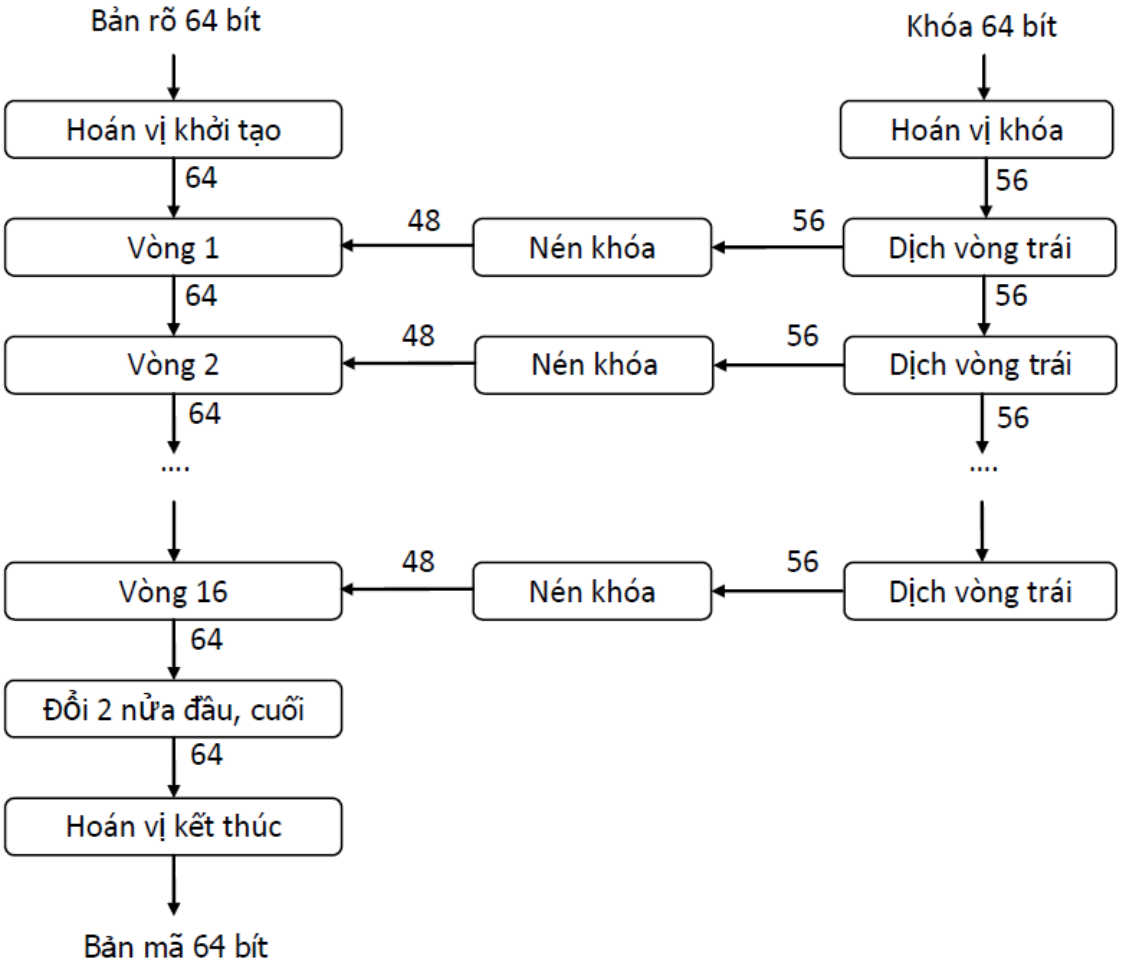
The Use of 56-Bit Keys

With a key length of 56 bits, there are 2^{56} possible keys, which is approximately 7.2×10^{16} keys. Thus, on the face of it, a brute-force attack appears impractical. Assuming that, on average, half the key space has to be searched, a single machine performing one DES encryption per microsecond would take more than a thousand years to break the cipher.

However, the assumption of one encryption per microsecond is overly conservative. As far back as 1977, Diffie and Hellman postulated that the technology existed to build a parallel machine with 1 million encryption devices, each of which could perform one encryption per microsecond [DIFF77]. This would bring the average search time down to about 10 hours. The authors estimated that the cost would be about $20 million in 1977 dollars.



VÒNG FEISTEL CỦA DES

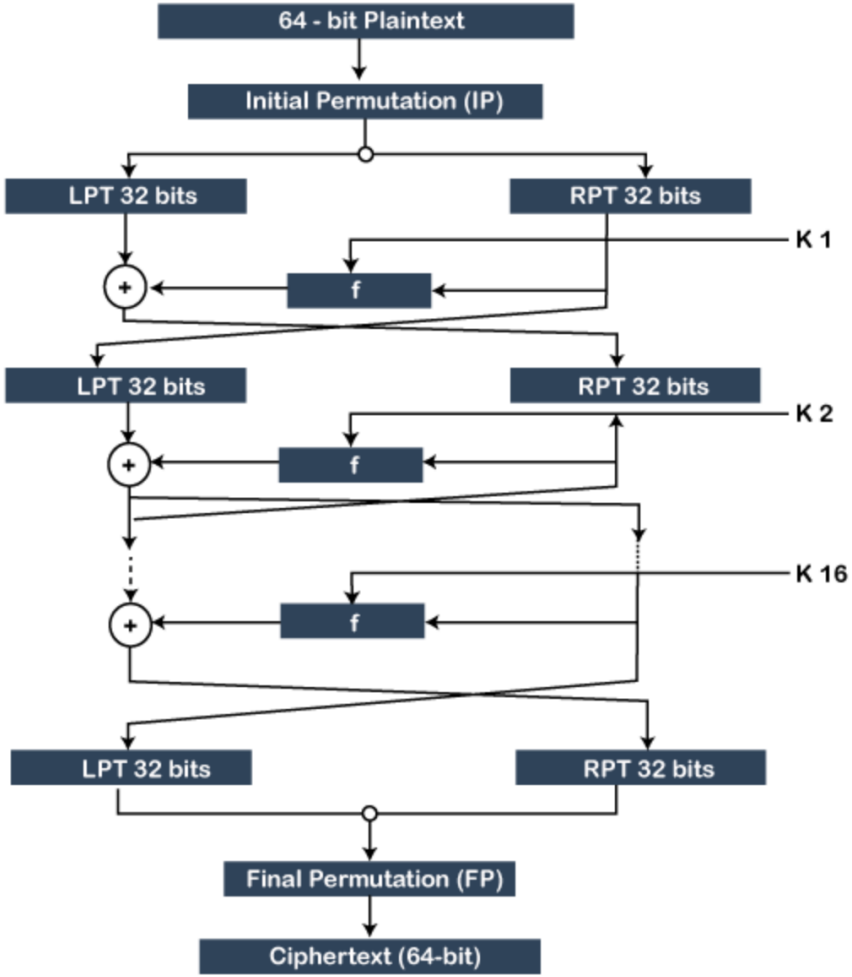


VÒNG FEISTEL CỦA DES

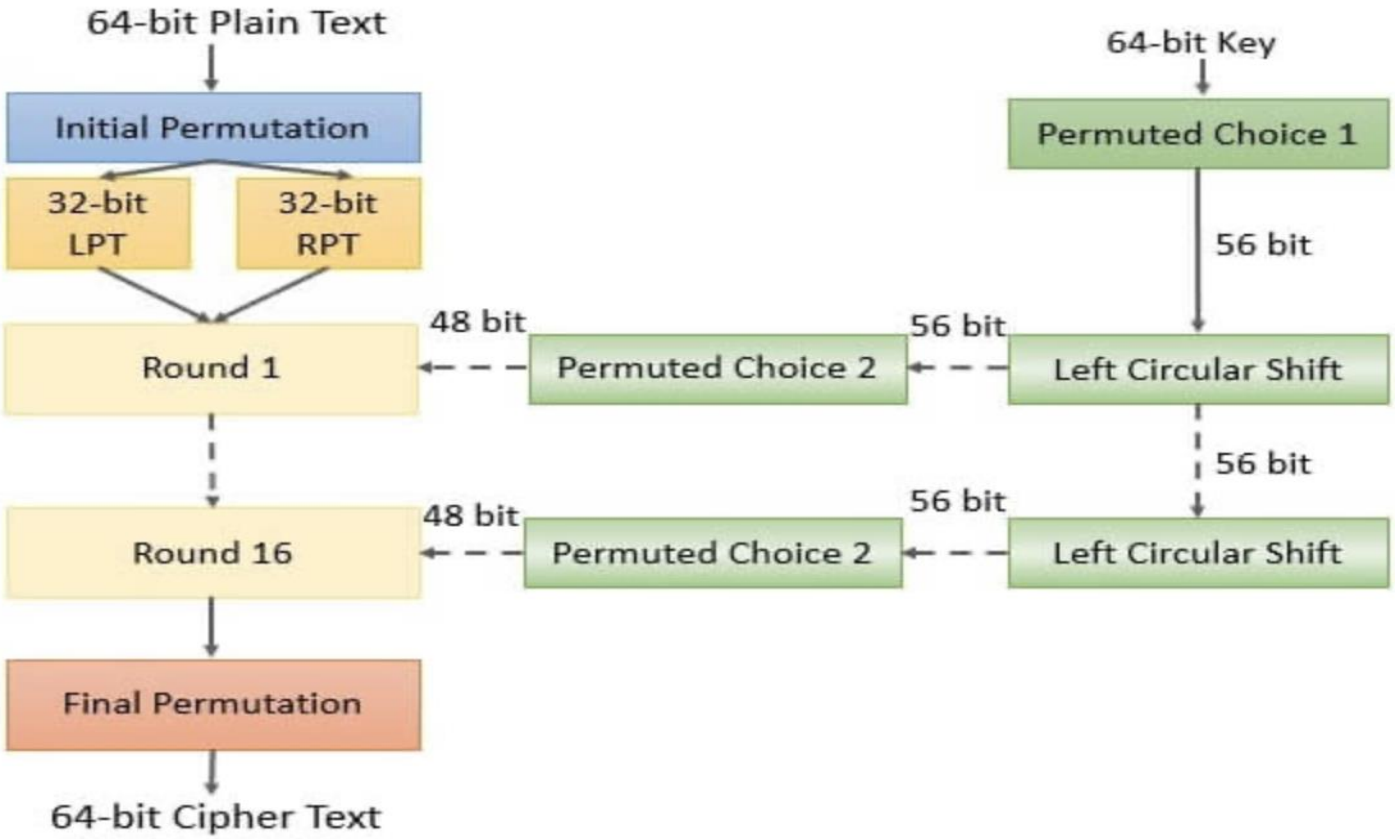
DES

Sơ đồ thuật toán

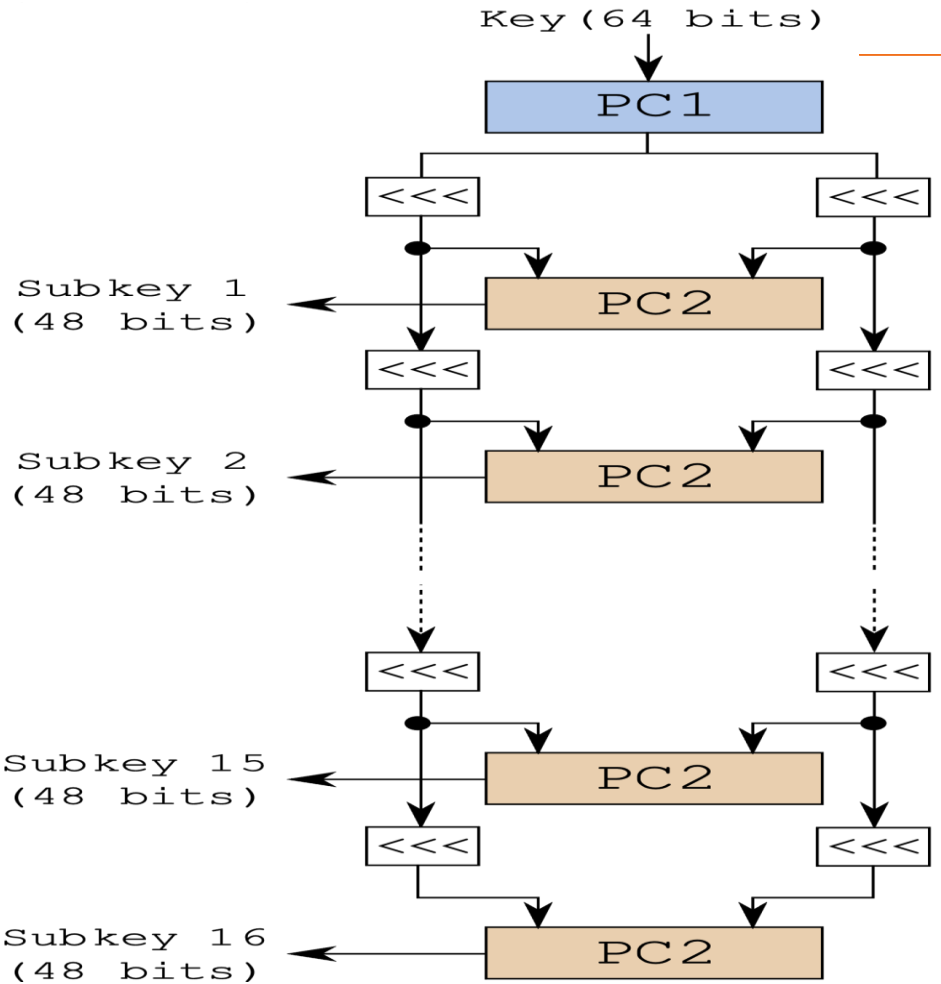
- $IP(x) = L_0R_0$
- $L_i = R_{i-1}$
- $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$
- $y = IP^{-1}(R_{16}L_{16})$



HOẠT ĐỘNG CỦA DES



DES KEY SCHEDULE



DES KEY SCHEDULE

PC-1 (Permuted Choice 1)

- Đầu vào: khoá ban đầu 64 bit
- PC-1 chỉ chọn 56 bit từ khoá ban đầu
- Các bit được chọn và sắp xếp lại theo thứ tự được định nghĩa trong bảng PC-1
- Đầu ra: Hai nữa 28 bit:
  - C0: nửa trái
  - D0: nửa phải

C						
57	49	41	33	25	17	09
01	58	50	42	34	26	18
10	02	59	51	43	35	27
19	11	03	60	52	44	36
D						
63	55	47	39	31	23	15
07	62	54	46	38	30	22
14	06	61	53	45	37	29
21	13	13	28	20	12	04

PC1

DES KEY SCHEDULE

Dịch trái từng nửa C0, D0

- Khoá có thể được dịch chuyển sang trái 1 hoặc 2 bit tuỳ thuộc vào số vòng.
  - C1 = LeftShift(C0, số bit dịch)
  - D1 = LeftShift(D0, số bit dịch)
- Ghép C1 và D1 lại thành 1 một chuỗi 56 bit

No. of round	No. of left rotations
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

DES KEY SCHEDULE

PC-2 (Permuted Choice 2)

- Đầu vào: 56 bit kết quả sau khi dịch trái
- PC-2 chỉ chọn 48 bit từ 56 bit đầu vào
- Các bit được chọn và sắp xếp lại theo thứ tự được định nghĩa trong bảng PC-2
- Đầu ra: Khoá con Ki cho vòng lặp thứ i

14	17	11	24	01	05
03	28	15	06	21	10
23	19	12	04	26	08
16	07	27	20	13	02
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

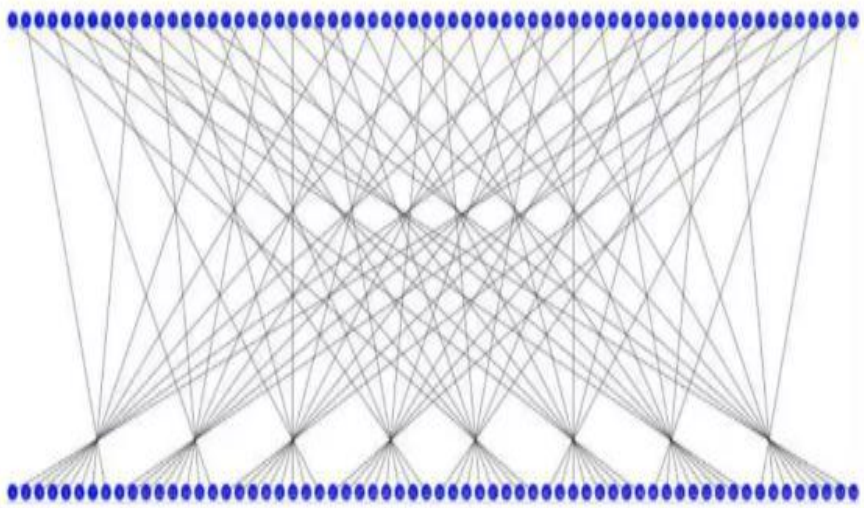
PC2

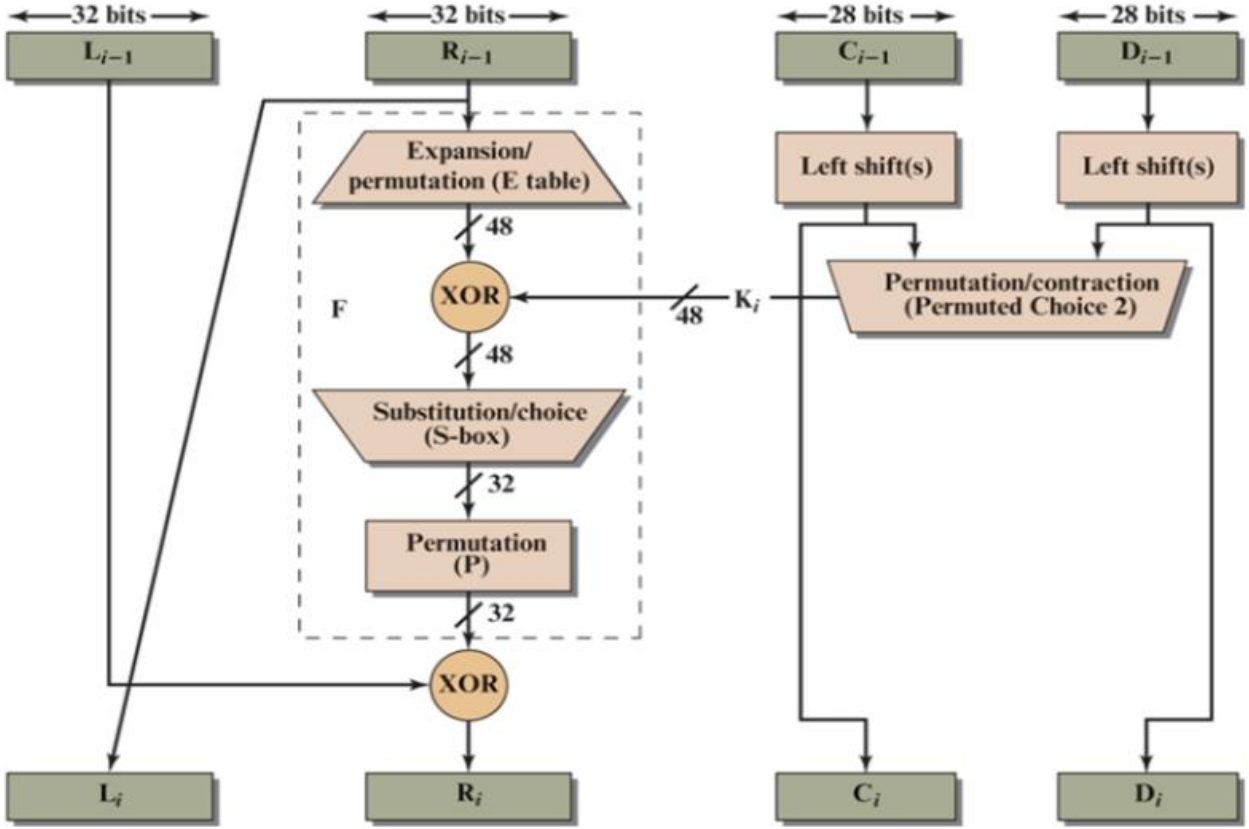
DES

Hoán vị khởi tạo (Initial Permutation)

58	50	42	34	26	18	10	02
60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06
64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01
59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05
63	55	47	39	31	23	15	07

Initial Permutation Table



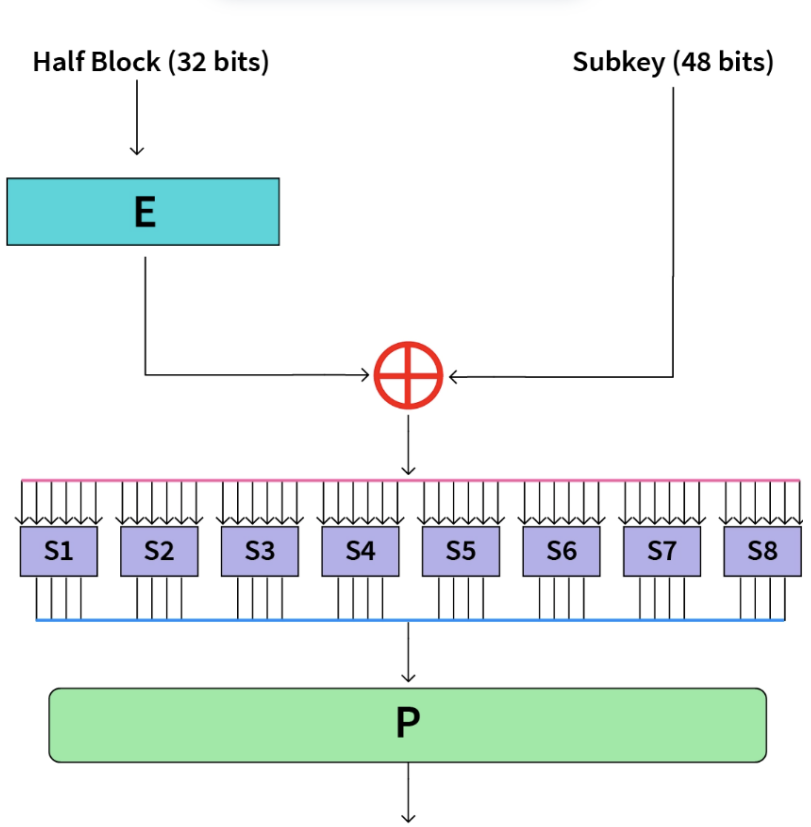


SINGLE ROUND OF DES

SINGLE ROUND OF DES

Hàm sinh khóa f(•):

- Hoán vị mở rộng E (Expansion Permuatation): 32 bit -> 48 bit
- Trộn khoá ⊕ (Key mixing): Áp dụng phép XOR dữ liệu với khoá con
- Thay thế S-box (Substitution S1, S2, ..., S8): 48 bit -> 32 bit
- Hoán vị P (Permutation): sắp xếp lại 32 bit
- XOR với Left



SINH 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.

SINGLE ROUND OF DES

Hoán vị mở rộng E

- Mở rộng 32 bit thành 48 bit để phù hợp với kích thước của khoá con
- 32 bit đầu vào được chia thành 8 khối 4 bit
- Mỗi khối 4 bit được mở rộng thành 6 bit bằng cách lấy thêm 2 bit từ các khối liền kề

32	01	02	03	04	05
04	05	06	07	08	09
08	09	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	01

E

SINGLE ROUND OF DES

Trộn khoá ⊕

- Thực hiện phép XOR từng bit của dữ liệu đã mở rộng với từng bit của khoá con

Input		Output
A	B	A xor B
0	0	0
0	1	1
1	0	1
1	1	0

SINGLE ROUND OF DES

Thay thế S-box

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

S_1

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

S_2

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

S_3

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

S_4

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

S_5

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

S_6

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

S_7

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

S_8

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

SINGLE ROUND OF DES

Hoán vị P

- Sắp xếp lại các bit sau khi thay thế để tăng tính ngẫu nhiên
- Sau đó thực hiện phép XOR với Left

16	07	20	21
29	12	28	17
01	15	23	26
05	18	31	10
02	08	24	14
32	27	03	09
19	13	30	06
22	11	04	25

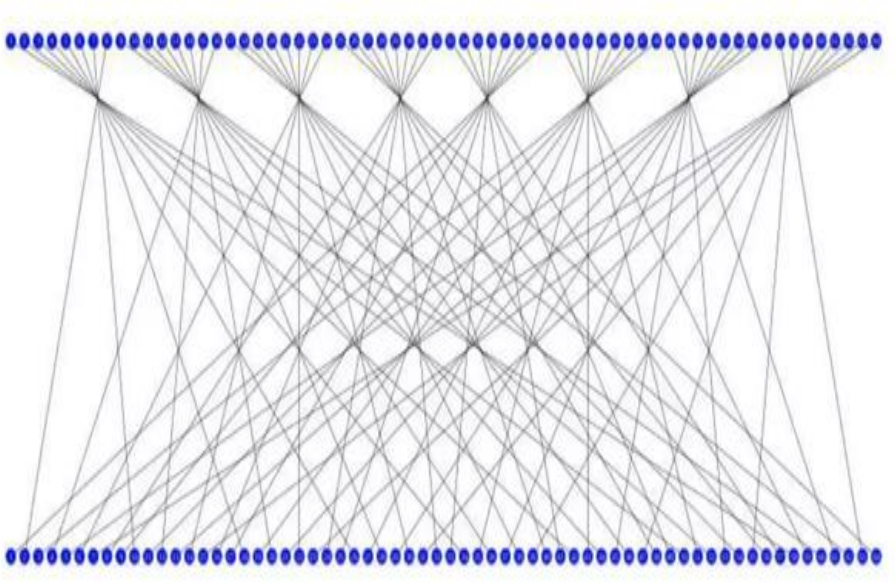
P

DES

Hoán vị kết thúc (Final Permutation)

$$\mathbf{FP} = \mathbf{IP}^{-1}$$

40	08	48	16	56	24	64	32
39	07	47	15	55	23	63	31
38	06	46	14	54	22	62	30
37	05	45	13	53	21	61	29
36	04	44	12	52	20	60	28
35	03	43	11	51	19	59	27
34	02	42	10	50	18	58	26
33	01	41	09	49	17	57	25



Timing Attacks

We discuss timing attacks in more detail in Part Three, as they relate to public-key algorithms. However, the issue may also be relevant for symmetric ciphers. In essence, a timing attack is one in which information about the key or the plaintext is obtained by observing how long it takes a given implementation to perform decryptions on various ciphertexts. A timing attack exploits the fact that an encryption or decryption algorithm often takes slightly different amounts of time on different inputs. [HEVI99] reports on an approach that yields the Hamming weight (number of bits equal to one) of the secret key. This is a long way from knowing the actual key, but it is an intriguing first step. The authors conclude that DES appears to be fairly resistant to a successful timing attack but suggest some avenues to explore. Although this is an interesting line of attack, it so far appears unlikely that this technique will ever be successful against DES or more powerful symmetric ciphers such as triple DES and AES.

ĐỘ AN TOÀN DES

- Tấn công vét cạn khóa (Brute Force Attack): Vì khóa của mã DES có chiều dài là 56 bít nên để tiến hành brute-force attack, cần kiểm tra 2^{56} khóa khác nhau => không thể
- Phá mã DES theo phương pháp vi sai (differential cryptanalysis): Năm 1990 Biham và Shamir đã giới thiệu phương pháp phá mã vi sai. Tuy nhiên phương pháp phá mã này lại đòi hỏi phải có 2^{47} cặp bản rõ - bản mã được lựa chọn (chosen-plaintext) => không thể
- Phá mã DES theo phương pháp thử tuyến tính (linear cryptanalysis). Trong phương pháp này, cần phải biết trước 2^{43} cặp bản rõ-bản mã (known-plaintext) => không thể

SO SÁNH ĐỘ AN TOÀN DES

Key Size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 Decryptions/s	Time Required at 10^{13} Decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns = 1.125 years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	2^{127} ns = 5.3×10^{21} years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	2^{167} ns = 5.8×10^{33} years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	2^{191} ns = 9.8×10^{40} years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	2^{255} ns = 1.8×10^{60} years	1.8×10^{56} years
26 characters (permutation)	Monoalphabetic	$2! = 4 \times 10^{26}$	2×10^{26} ns = 6.3×10^{9} years	6.3×10^{6} years

DOUBLE DES

- Cách khắc phục yếu điểm kích thước khóa ngắn của DES là sử dụng mã hóa DES nhiều lần với các khóa khác nhau cho cùng một bản tin.
- Double DES là dùng DES hai lần với hai khóa khác nhau:

$$C = E(E(P, K1),K2)$$

- 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.

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 *K*1, *K*2 mà vẫn đảm bảo độ an toàn cần thiết.

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

TRIPLE DES

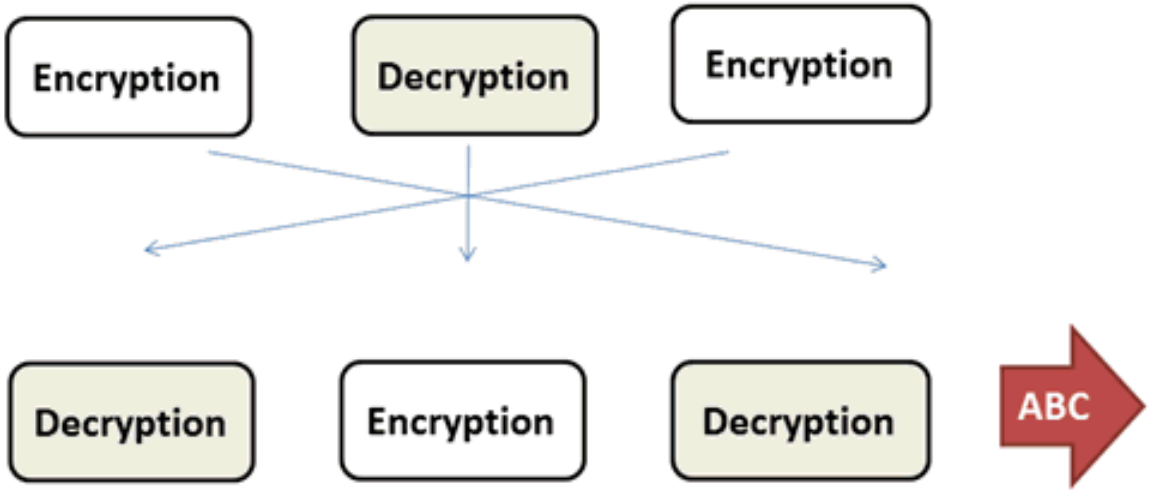
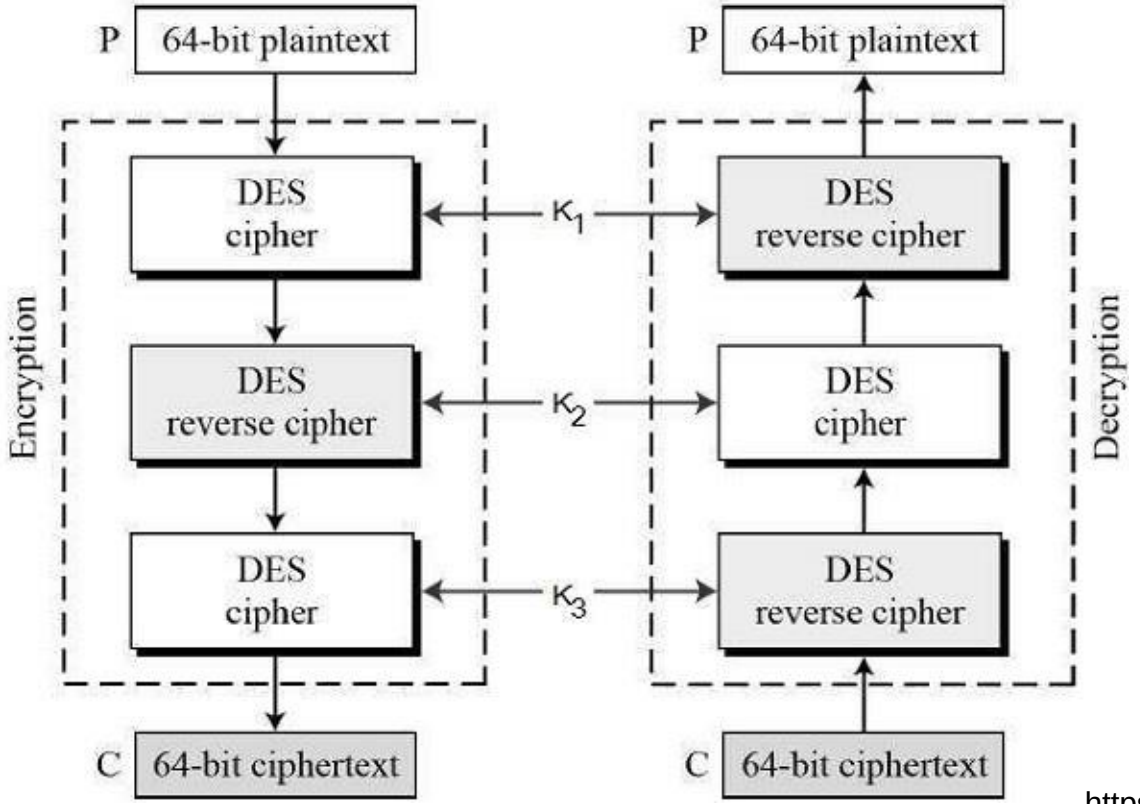


Fig : TripleDES Encryption

TRIPLE DES



https://www.geeksforgeeks.org/triple-des-3des/

Thực hành

Thực hành bài Lab 4

Tổng kết

- Mã TinyDES
- Mã DES
- Thực hành code C++

ĐẠI NAM
UNIVERSITY

Thank You