Chapter 6

Data Encryption Standard (DES)

Chư ơ ng 6

Mục tiêu

Để xem lại lịch sử ngắn gọn của DES

Để xác định cấu trúc cơ bản của DES

Để mô tả chi tiết các phần tử xây dựng của DES

Để mô tả quá trình tạo khóa tròn

Để phân tích DES

6-1 GIỚI THIỆU

Tiêu chuẩn mã hóa dữ liệu (DES) là mật mã khối khóa đối xứng đư ợc xuất bản bởi Viện Tiêu chuẩn và Công nghệ Quốc gia (NIST).

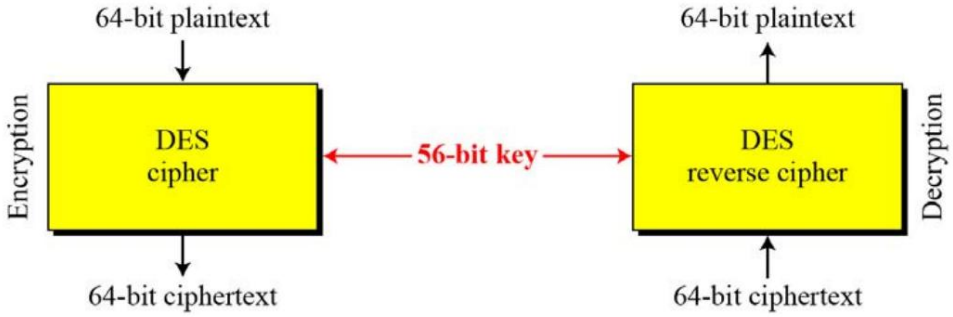
Các chủ đề đư ợc thảo luận trong phần này: 6.1.1

6.1.1 Lịch sử

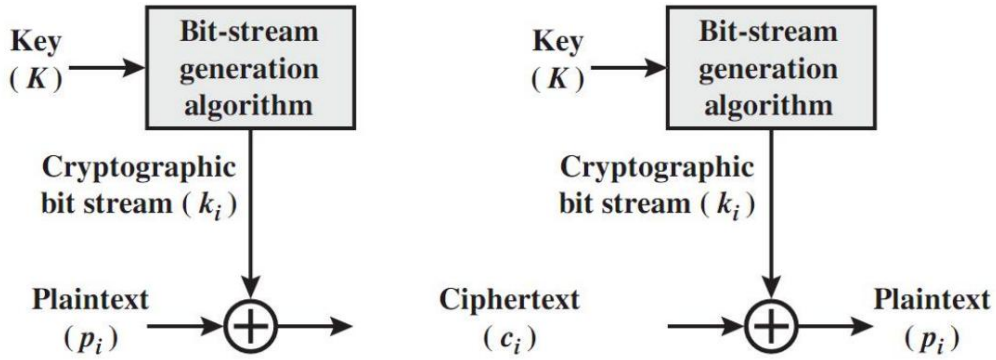
Năm 1973, NIST công bố yêu cầu đề xuất hệ thống mật mã khóa đối xứng quốc gia. Một đề xuất từ IBM, một bản sửa đổi của dự án có tên Lucifer, đã đư ợc chấp nhận dư ới tên DES. DES đư ợc xuất bản trong Cục Đăng ký Liên bang vào tháng 3 năm 1975 dư ới dạng bản dự thảo của Tiêu chuẩn Xử lý Thông tin Liên bang (FIPS).

DES là một mật mã khối, như trong Hình 6.1.

Hình 6.1 Mã hóa và giải mã bằng DES

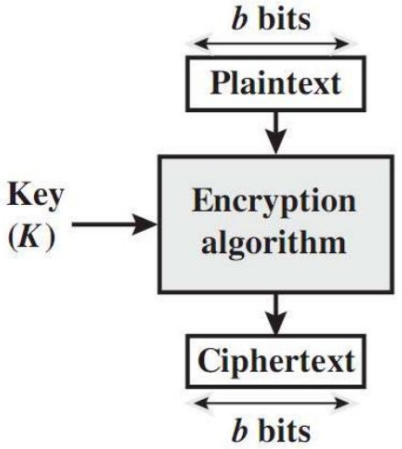


Mật mã dòng và mật mã khối



(a) Mật mã dòng bằng cách sử dụng bộ tạo dòng bit thuật toán

Mật mã dòng và mật mã khối

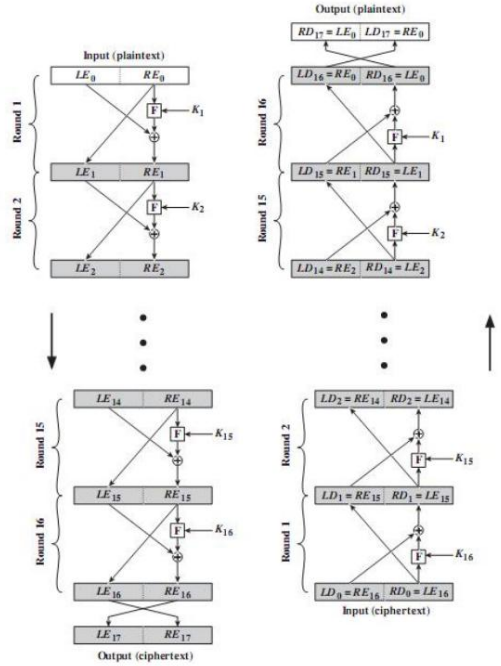


(b) Mật mã khối

Feistel đề xuất sử dụng mật mã xen kẽ các phép thay thế và hoán vị, trong đó các thuật ngữ này được định nghĩa như sau:

- Thay thế: Mỗi phần tử hoặc nhóm phần tử bản rõ được thay thế duy nhất bằng một phần tử hoặc nhóm phần tử bản mã tương ứng.
- Hoán vị: Một chuỗi các phần tử bản rõ được thay thế bằng một hoán vị của dãy đó. Nghĩa là, không có phần tử nào được thêm vào, xóa đi hoặc thay thế trong chuỗi mà thứ tự xuất hiện của các phần tử trong chuỗi bị thay đổi.

- Khối bản rõ đư ợc chia thành hai nửa L0 và R0
- Hai nửa dữ liệu trải qua nhiều vòng xử lý và sau đó kết hợp lại để tạo ra khối bản mã
- Mỗi vòng tôi có đầu vào Li-1 và Ri-1 có nguồn gốc từ vòng trư ớc, cũng như một khóa con Ki có nguồn gốc từ tổng thể K. Nói chung, các khóa con Ki khác nhau và khác với K.



Mã hóa và giải mã Feistel (16 vòng

Thus, we have $LD_1 = RE_{15}$ and $RD_1 = LE_{15}$. Therefore, the output of the first round of the decryption process is $RE_{15} \| LE_{15}$, which is the 32-bit swap of the input to the sixteenth round of the encryption. This correspondence holds all the way through the 16 iterations, as is easily shown. We can cast this process in general terms. For the ith iteration of the encryption algorithm,

$$LE_i = RE_{i-1}$$
$$RE_i = LE_{i-1} \oplus \mathrm{F}(RE_{i-1}, K_i)$$

Rearranging terms:

$$RE_{i-1} = LE_i$$
$$LE_{i-1} = RE_i \oplus \mathrm{F}(RE_{i-1}, K_i) = RE_i \oplus \mathrm{F}(LE_i, K_i)$$

the encryption process

$$LE_{16} = RE_{15}$$
$$RE_{16} = LE_{15} \oplus \mathrm{F}(RE_{15}, K_{16})$$

On the decryption side,

$$LD_1 = RD_0 = LE_{16} = RE_{15}$$
$$RD_1 = LD_0 \oplus \mathrm{F}(RD_0, K_{16})$$
$$= RE_{16} \oplus \mathrm{F}(RE_{15}, K_{16})$$
$$= [LE_{15} \oplus \mathrm{F}(RE_{15}, K_{16})] \oplus \mathrm{F}(RE_{15}, K_{16})$$

The XOR has the following properties:

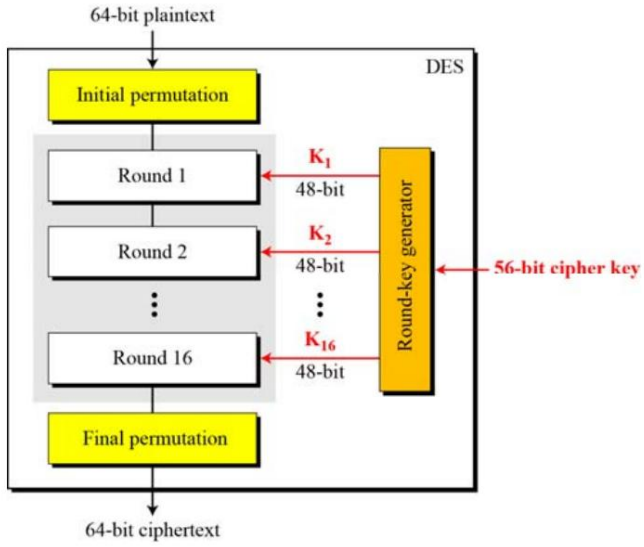
$$[A \oplus B] \oplus C = A \oplus [B \oplus C]$$
$$D \oplus D = 0$$
$$E \oplus 0 = E$$

CẤU TRÚC 6-2 DES

Quá trình mã hóa đư ợc thực hiện bằng hai hoán vị (Hộp P), mà chúng tôi gọi là ban đầu và cuối cùng hoán vị, và mư ời sáu vòng Feistel.

Các chủ đề đư ợc thảo luận trong phần này:

Hình 6.2 Cấu trúc chung của DES



6.2.1 Tiếp tục

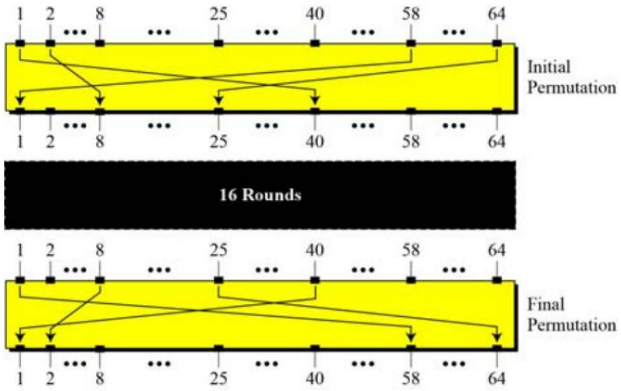
Quá trình mã hóa được thực hiện bằng hai hoán vị (hộp P), mà chúng tôi gọi là hoán vị ban đầu và cuối cùng, và mười sáu vòng Feistel. Mỗi vòng sử dụng một khóa tròn 48 bit khác nhau được tạo từ khóa mật mã theo thuật toán được xác định trước

Bảng 6.1 Bảng hoán vị đầu và cuối

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

6.2.1 Hoán vị ban đầu và cuối cùng

Hình 6.3 Các bước hoán vị ban đầu và cuối cùng trong DES



6.2.1 Tiếp theo

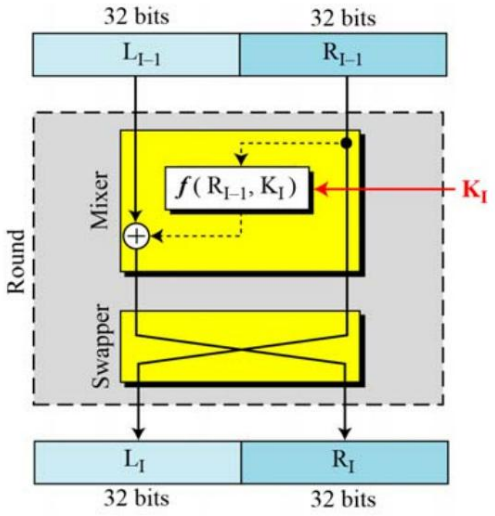
Ghi chú

Các hoán vị ban đầu và cuối cùng là các hộp P thẳng nghịch đảo nhau.

Chúng không có ý nghĩa mật mã trong DES.

6.2.2 Vòng đấu

DES sử dụng 16 vòng. Mỗi vòng của DES là một mật mã Feistel.



Hình 6.4
Một vòng ở DES

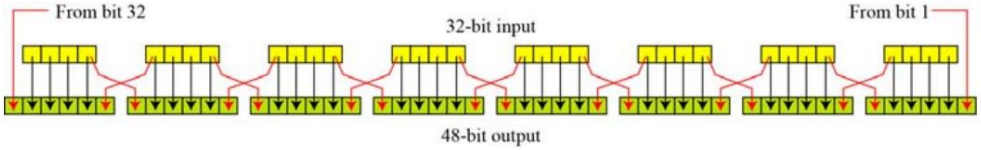
(trang web mã hóa)

6.2.2 Tiếp tục

Hộp P mở rộng Vì

RI 1 là đầu vào 32 bit và KI là khóa 48 bit nên trư ớc tiên chúng ta cần mở rộng RI 1 thành 48 bit.

Hình 6.6 Hoán vị mở rộng



6.2.2 Tiếp theo

Hàm DES

Trái tim của DES là hàm DES. Hàm DES áp dụng khóa 48 bit cho 32 bit ngoài cùng bên phải để tạo ra đầu ra 32 bit.

Hình 6.5
Hàm DES

6.2.2 Tiếp tục

Mặc dù mối quan hệ giữa đầu vào và đầu ra có thể đư ợc xác định bằng toán học, như ng DES sử dụng Bảng 6.2 để định nghĩa hộp P này.

Bảng 6.6 Bảng P-box mở rộng

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	31	31	32	01

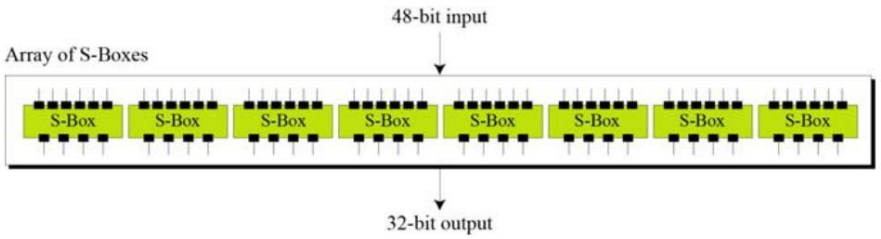
Chất làm trắng (XOR)

Sau khi hoán vị mở rộng, DES sử dụng thao tác XOR trên phần bên phải mở rộng và phím tròn. Lưu ý rằng cả phần bên phải và khóa đều có độ dài 48 bit. Cũng lưu ý rằng phím tròn chỉ được sử dụng trong thao tác này.

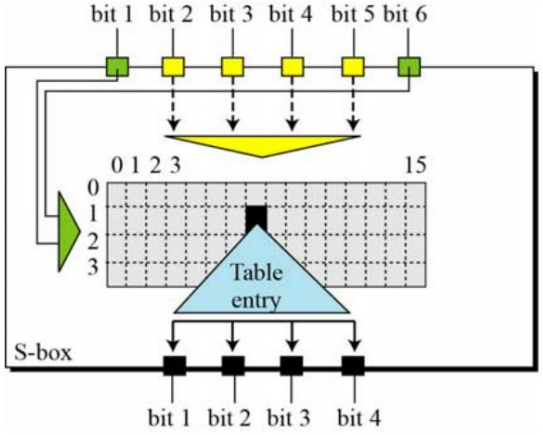
Hộp chữ S

Hộp S thực hiện việc trộn thực sự (nhầm lẫn). DES sử dụng 8 hộp S, mỗi hộp có đầu vào 6 bit và đầu ra 4 bit. Xem Hình 6.7.

Hình 6.7 Hộp S



Hình 6.8 Quy tắc hộp S



Bảng 6.3 cho thấy hoán vị của hộp S 1. Đối với các hộp còn lại, hãy xem sách giáo khoa.

Bảng 6.3 Hộp S 1

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	10	03	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

Ví dụ 6.3

Đầu vào của S-box 1 là 100011. Đầu ra là gì?

Giải pháp

Nếu chúng ta viết bit đầu tiên và bit thứ sáu cùng nhau, chúng ta sẽ nhận được 11 ở dạng nhị phân, tức là 3 ở dạng thập phân. Các bit còn lại là 0001 ở dạng nhị phân, là 1 ở dạng thập phân. Chúng ta tìm giá trị ở hàng 3, cột 1, trong Bảng 6.3 (Hộp S 1). Kết quả là 12 ở dạng thập phân, trong đó ở dạng nhị phân là 1100. Vì vậy, đầu vào 100011 mang lại đầu ra 1100.

Ví dụ 6.4

Đầu vào của S-box 8 là 000000. Đầu ra là gì?

Giải pháp

Nếu chúng ta viết bit đầu tiên và bit thứ sáu cùng nhau, chúng ta sẽ nhận được 00 ở dạng nhị phân, tức là 0 ở dạng thập phân. Các bit còn lại là 0000 ở dạng nhị phân, bằng 0 ở dạng thập phân. Chúng ta tìm giá trị ở hàng 0, cột 0, trong Bảng 6.10 (Hộp S 8). Kết quả là 13 ở dạng thập phân, tức là 1101 ở dạng nhị phân. Vì vậy, đầu vào 000000 mang lại đầu ra 1101.

Hoán vị thẳng

Bảng 6.11 Bảng hoán vị thẳng

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

Bằng cách sử dụng bộ trộn và bộ hoán đổi, chúng ta có thể tạo mật mã và mật mã ngược, mỗi bộ có 16 vòng.

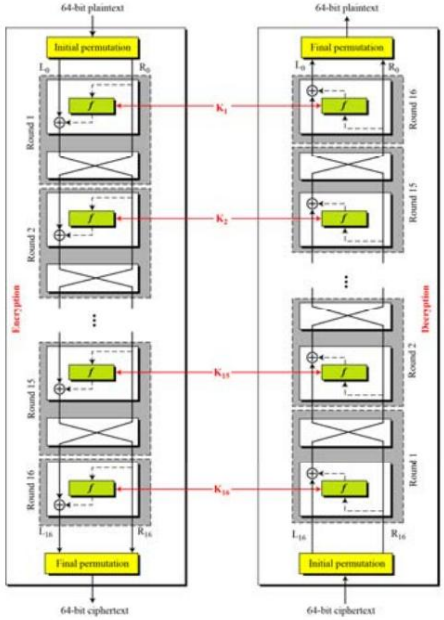
Cách tiếp cận đầu tiên

Để đạt được mục tiêu này, một cách tiếp cận là làm cho vòng cuối cùng (vòng 16) khác với các vòng khác; nó chỉ có một máy trộn và không có bộ trao đổi.

Ghi chú

Trong cách tiếp cận đầu tiên, không có người trao đổi ở vòng cuối cùng.

Hình 6.9 Mật mã DES và mật mã ngược cho phương pháp đầu tiên



Thuật toán 6.1 Mã giả cho mật mã DES

```
Cipher (plainBlock[64], RoundKeys[16, 48], cipherBlock[64])
{
    permute (64, 64, plainBlock, inBlock, InitialPermutationTable)
    split (64, 32, inBlock, leftBlock, rightBlock)
    for (round = 1 to 16)
    {
        mixer (leftBlock, rightBlock, RoundKeys[round])
        if (round!=16) swapper (leftBlock, rightBlock)
    }
    combine (32, 64, leftBlock, rightBlock, outBlock)
    permute (64, 64, outBlock, cipherBlock, FinalPermutationTable)
}
```


Thuật toán 6.1 Mã giả cho mật mã DES (Tiếp theo)

```
mixer (leftBlock[48], rightBlock[48], RoundKey[48])
{
    copy (32, rightBlock, T1)
    function (T1, RoundKey, T2)
    exclusiveOr (32, leftBlock, T2, T3)
    copy (32, T3, rightBlock)
}
swapper (leftBlock[32], rigthBlock[32])
{
    copy (32, leftBlock, T)
    copy (32, rightBlock, leftBlock)
    copy (32, T, rightBlock)
}
```


Thuật toán 6.1 Mã giả cho mật mã DES (Tiếp theo)

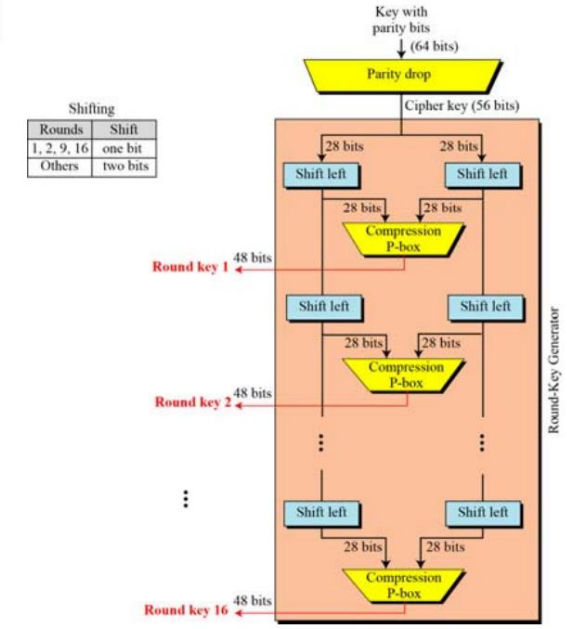
```
function (inBlock[32], RoundKey[48], outBlock[32])
{
    permute (32, 48, inBlock, T1, ExpansionPermutationTable)
    exclusiveOr (48, T1, RoundKey, T2)
    substitute (T2, T3, SubstituteTables)
    permute (32, 32, T3, outBlock, StraightPermutationTable)
}
```

Thuật toán 6.1 Mã giả cho mật mã DES (Tiếp theo)

```
substitute (inBlock[32], outBlock[48], SubstitutionTables[8, 4, 16])
{
    for (i = 1 to 8)
    {
        row ← 2 × inBlock[i × 6 + 1] + inBlock [i × 6 + 6]
        col ← 8 × inBlock[i × 6 + 2] + 4 × inBlock[i × 6 + 3] +
              2 × inBlock[i × 6 + 4] + inBlock[i × 6 + 5]

        value = SubstitutionTables [i][row][col]

        outBlock[[i × 4 + 1] ← value / 8;      value ← value mod 8
        outBlock[[i × 4 + 2] ← value / 4;      value ← value mod 4
        outBlock[[i × 4 + 3] ← value / 2;      value ← value mod 2
        outBlock[[i × 4 + 4] ← value
    }
}
```

Hình 6.10
Tạo khóa

Phương pháp tiếp cận

thay thế Chúng ta có thể thực hiện tất cả 16 vòng giống nhau bằng cách thêm một ngư ời hoán đổi vào vòng thứ 16 và thêm một ngư ời hoán đổi bổ sung sau đó (hai ngư ời hoán đổi sẽ hủy tác dụng của nhau).

Tạo khóa

Bộ tạo khóa vòng tạo ra 16 khóa 48 bit từ khóa mật mã 56 bit.

Bảng 6.12 Bảng thả bit chẵn lẻ

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	63	55	47	39
31	23	15	07	62	54	46	38
30	22	14	06	61	53	45	37
29	21	13	05	28	20	12	04

Bảng 6.13 Số lư ợng dịch chuyển bit

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bit shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Bảng 6.14 Bảng nén khóa

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

Thuật toán 6.2 Thuật toán tạo khóa tròn

```
Key_Generator (keyWithParities[64], RoundKeys[16, 48], ShiftTable[16])
{
    permute (64, 56, keyWithParities, cipherKey, ParityDropTable)
    split (56, 28, cipherKey, leftKey, rightKey)
    for (round = 1 to 16)
    {
        shiftLeft (leftKey, ShiftTable[round])
        shiftLeft (rightKey, ShiftTable[round])
        combine (28, 56, leftKey, rightKey, preRoundKey)
        permute (56, 48, preRoundKey, RoundKeys[round], KeyCompressionTable)
    }
}
```


Thuật toán 6.2 Thuật toán tạo khóa tròn (Tiếp theo)

```
shiftLeft (block[28], numOfShifts)
{
    for (i = 1 to numOfShifts)
    {
        T ← block[1]
        for (j = 2 to 28)
        {
            block [j−1] ← block [j]
        }
        block[28] ← T
    }
}
```


Chúng tôi chọn một khối văn bản gốc ngẫu nhiên và một khóa ngẫu nhiên, đồng thời xác định khối văn bản mã hóa sẽ là gì (tất cả đều ở dạng thập lục phân):

Plaintext: 123456ABCD132536 Key: AABB09182736CCDD
CipherText: C0B7A8D05F3A829C

Bảng 6.15 Dấu vết dữ liệu cho Ví dụ 6.5

Plaintext: 123456ABCD132536			
*After initial permutation:*14A7D67818CA18AD After splitting: L_0=14A7D678 R_0=18CA18AD			
Round	*Left*	*Right*	*Round Key*
Round 1	18CA18AD	5A78E394	194CD072DE8C
Round 2	5A78E394	4A1210F6	4568581ABCCE
Round 3	4A1210F6	B8089591	06EDA4ACF5B5
Round 4	B8089591	236779C2	DA2D032B6EE3

Ví dụ 6.5 Tiếp theo

Bảng 6.15 Dấu vết dữ liệu cho Ví dụ 6.5 (Tiếp theo

Round 5	236779C2	A15A4B87	69A629FEC913
Round 6	A15A4B87	2E8F9C65	C1948E87475E
Round 7	2E8F9C65	A9FC20A3	708AD2DDB3C0
Round 8	A9FC20A3	308BEE97	34F822F0C66D
Round 9	308BEE97	10AF9D37	84BB4473DCCC
Round 10	10AF9D37	6CA6CB20	02765708B5BF
Round 11	6CA6CB20	FF3C485F	6D5560AF7CA5
Round 12	FF3C485F	22A5963B	C2C1E96A4BF3
Round 13	22A5963B	387CCDAA	99C31397C91F
Round 14	387CCDAA	BD2DD2AB	251B8BC717D0
Round 15	BD2DD2AB	CF26B472	3330C5D9A36D
Round 16	19BA9212	CF26B472	181C5D75C66D
After combination: 19BA9212CF26B472			
Ciphertext: C0B7A8D05F3A829C			*(after final permutation)*

Ví dụ 6.6

Chúng ta hãy xem Bob, ở đích đến, có thể giải mã văn bản mật mã nhận đư ợc từ Alice bằng cách sử dụng cùng một khóa như thế nào. Bảng 6.16 cho thấy một số điểm thú vị.

Ciphertext: C0B7A8D05F3A829C			
After initial permutation: 19BA9212CF26B472 After splitting: L_0=19BA9212 R_0=CF26B472			
Round	*Left*	*Right*	*Round Key*
Round 1	CF26B472	BD2DD2AB	181C5D75C66D
Round 2	BD2DD2AB	387CCDAA	3330C5D9A36D
. . .	. . .	. . .	. . .
Round 15	5A78E394	18CA18AD	4568581ABCCE
Round 16	14A7D678	18CA18AD	194CD072DE8C
After combination: 14A7D67818CA18AD			
Plaintext:123456ABCD132536			(after final permutation)

PHÂN TÍCH 6-3 DES

Các nhà phê bình đã sử dụng kính lúp mạnh để phân tích DES. Các thử nghiệm đã đư ợc thực hiện để đo độ mạnh của một số thuộc tính mong muốn trong mật mã khối.

Các chủ đề đư ợc thảo luận trong phần này: 6.3.1 Thuộc tính 6.3.2 Tiêu chí thiết kế 6.3.3 Điểm yếu của DES

Hai đặc tính mong muốn của mật mã khối là hiệu ứng tuyết lở và tính đầy đủ.

Ví dụ 6.7

Để kiểm tra hiệu ứng tuyết lở trong DES, chúng ta hãy mã hóa hai khối văn bản gốc (có cùng khóa) chỉ khác nhau một bit và quan sát sự khác biệt về số lư ợng bit trong mỗi khối.

tròn.

Plaintext: 0000000000000000 Key: 22234512987ABB23
Ciphertext: 4789FD476E82A5F1

Plaintext: 0000000000000001 Key: 22234512987ABB23
Ciphertext: 0A4ED5C15A63FEA3

Ví dụ 6.7 Tiếp theo

Mặc dù hai khối văn bản gốc chỉ khác nhau ở bit ngoài cùng bên phải, nhưng các khối văn bản mã hóa khác nhau ở 29 bit. Điều này có nghĩa là việc thay đổi khoảng 1,5% bản rõ sẽ tạo ra sự thay đổi khoảng 45% trong bản mã.

Bảng 6.17 Số lượng khác biệt bit cho Ví dụ 6.7

Rounds	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*
Bit differences	1	6	20	29	30	33	32	29	32	39	33	28	30	31	30	29

S-Boxe

Thiết kế này tạo ra sự nhầm lẫn và khuếch tán các bit từ mỗi vòng này sang vòng tiếp theo.

Hộp P

Họ cung cấp sự khuếch tán của bit.

Số vòng DES sử dụng 16 vòng

mật mã Feistel. bản mã hoàn toàn là một hàm ngẫu nhiên của bản rõ và bản mã.

Hiệu ứng đầy đủ Hiệu ứng

hoàn chỉnh có nghĩa là mỗi bit của bản mã cần phụ thuộc vào nhiều bit trên bản rõ.

Trong vài năm qua, các nhà phê bình đã tìm thấy một số điểm yếu trong DES.

Điểm yếu trong thiết kế mật mã 1. Điểm yếu

trong hộp S 2. Điểm yếu trong hộp P

3. Điểm yếu trong khóa

Table 6.18 *Weak keys*

Keys before parities drop (64 bits)	*Actual key (56 bits)*
0101 0101 0101 0101	0000000 0000000
1F1F 1F1F 0E0E 0E0E	0000000 FFFFFFF
E0E0 E0E0 F1F1 F1F1	FFFFFFF 0000000
FEFE FEFE FEFE FEFE	FFFFFFF FFFFFFF

Ví dụ 6.8

Chúng ta hãy thử dùng khóa yếu đầu tiên trong Bảng 6.18 để mã hóa một khối hai lần. Sau hai lần mã hóa với
cùng một khóa, khối văn bản gốc ban đầu sẽ được tạo. Lưu ý rằng chúng tôi đã sử dụng thuật toán mã hóa hai lần, không phải một lần mã hóa mà sau đó là một lần giải mã khác.

Key: 0x0101010101010101
Plaintext: *0x1234567887654321* Ciphertext: 0x814FE938589154F7

Key: 0x0101010101010101
Plaintext: 0x814FE938589154F7 Ciphertext: *0x1234567887654321*

Hình 6.11 Mã hóa và giải mã kép bằng khóa yếu

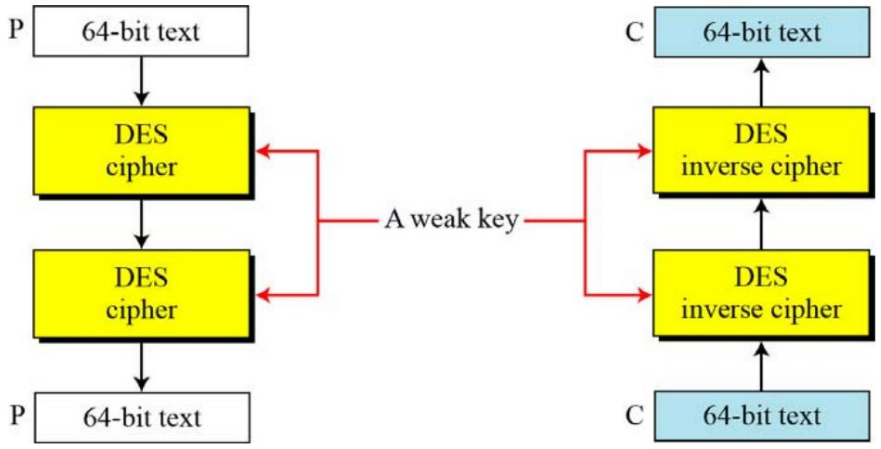
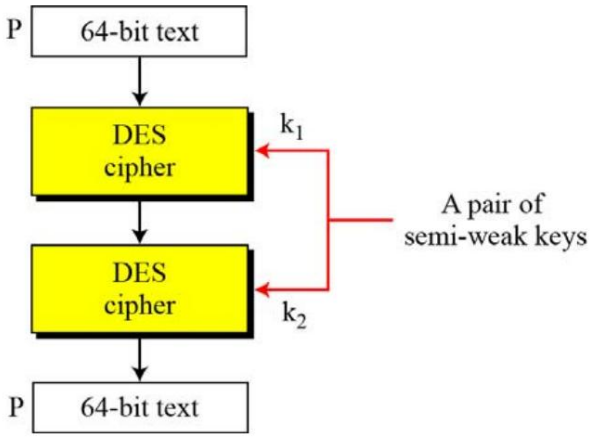


Table 6.19 *Semi-weak keys*

First key in the pair	*Second key in the pair*
01FE 01FE 01FE 01FE	FE01 FE01 FE01 FE01
1FE0 1FE0 0EF1 0EF1	E01F E01F F10E F10E
01E0 01E1 01F1 01F1	E001 E001 F101 F101
1FFE 1FFE 0EFE 0EFE	FE1F FE1F FE0E FE0E
011F 011F 010E 010E	1F01 1F01 0E01 0E01
E0FE E0FE F1FE F1FE	FEE0 FEE0 FEF1 FEF1

Round key 1	9153E54319BD	6EAC1ABCE642
Round key 2	6EAC1ABCE642	9153E54319BD
Round key 3	6EAC1ABCE642	9153E54319BD
Round key 4	6EAC1ABCE642	9153E54319BD
Round key 5	6EAC1ABCE642	9153E54319BD
Round key 6	6EAC1ABCE642	9153E54319BD
Round key 7	6EAC1ABCE642	9153E54319BD
Round key 8	6EAC1ABCE642	9153E54319BD
Round key 9	9153E54319BD	6EAC1ABCE642
Round key 10	9153E54319BD	6EAC1ABCE642
Round key 11	9153E54319BD	6EAC1ABCE642
Round key 12	9153E54319BD	6EAC1ABCE642
Round key 13	9153E54319BD	6EAC1ABCE642
Round key 14	9153E54319BD	6EAC1ABCE642
Round key 15	9153E54319BD	6EAC1ABCE642
Round key 16	6EAC1ABCE642	9153E54319BD

Hình 6.12 Một cặp khóa bán yếu trong mã hóa và giải mã



Ví dụ 6.9

Xác suất chọn ngẫu nhiên một khóa yếu, nửa yếu hoặc có thể là khóa yếu là bao nhiêu?

Giải pháp

DES có miền khóa là 256. Tổng số khóa trên là 64 (4 + 12 + 48). Xác suất chọn đư ợc một trong các phím này là 8,8 × 10 16, gần như không thể.

Key Complement In the key domain (2^{56}), definitely half of the keys are *complement* of the other half. A **key complement** can be made by inverting (changing 0 to 1 or 1 to 0) each bit in the key. Does a key complement simplify the job of the cryptanalysis? It happens that it does. Eve can use only half of the possible keys (2^{55}) to perform brute-force attack. This is because

$$C = E(K, P) \quad \rightarrow \quad \overline{C} = E(\overline{K}, \overline{P})$$

In other words, if we encrypt the complement of plaintext with the complement of the key, we get the complement of the ciphertext. Eve does not have to test all 2^{56} possible keys, she can test only half of them and then complement the result.

Ví dụ 6.10

Chúng ta hãy kiểm tra khẳng định về các khoá bù. Chúng ta đã sử dụng một khóa tùy ý và bản rõ để tìm bản mã tư ơ ng ứng. Nếu chúng ta có phần bù khóa và bản rõ, chúng ta có thể thu đư ợc phần bù của bản mã trư ớc đó (Bảng 6.20).

Table 6.20 *Results for Example 6.10*

	Original	*Complement*
Key	1234123412341234	EDCBEDCBEDCBEDCB
Plaintext	12345678ABCDEF12	EDCBA987543210ED
Ciphertext	E112BE1DEFC7A367	1EED41E210385C98

6-4 Nhiều DES

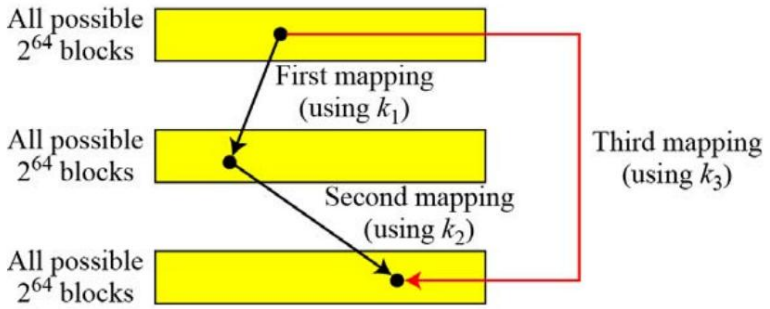
Những lời chỉ trích chính đối với DES liên quan đến độ dài khóa của nó. May mắn thay DES không phải là một nhóm. Điều này có nghĩa là chúng ta có thể sử dụng DES gấp đôi hoặc gấp ba để tăng kích thước khóa.

Các chủ đề được thảo luận trong phần này:

6-4 Tiếp tục

Một sự thay thế ánh xạ mọi đầu vào có thể có tới mọi đầu ra có thể có là một nhóm.

Hình 6.13 Thành phần của ánh xạ



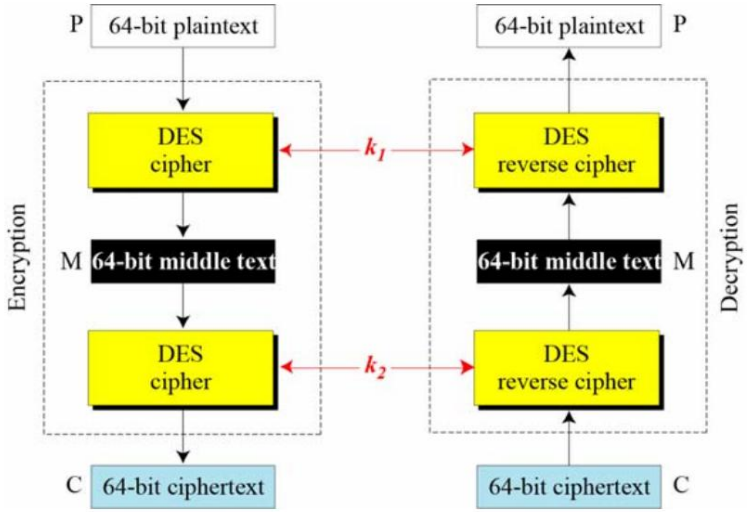
6.4.1 DES kép

Cách tiếp cận đầu tiên là sử dụng DES kép (2DES).

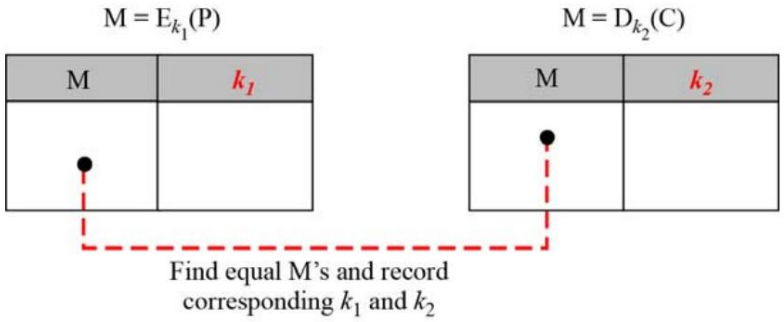
Tuy nhiên, việc sử dụng một cuộc tấn công bằng văn bản rõ ràng được gọi là tấn công gặp ở giữa chứng tỏ rằng DES kép cải thiện lỗ hổng này một chút (đến 257 thử nghiệm), nhưng không đáng kể (đến 2112).

6.4.1 Tiếp theo

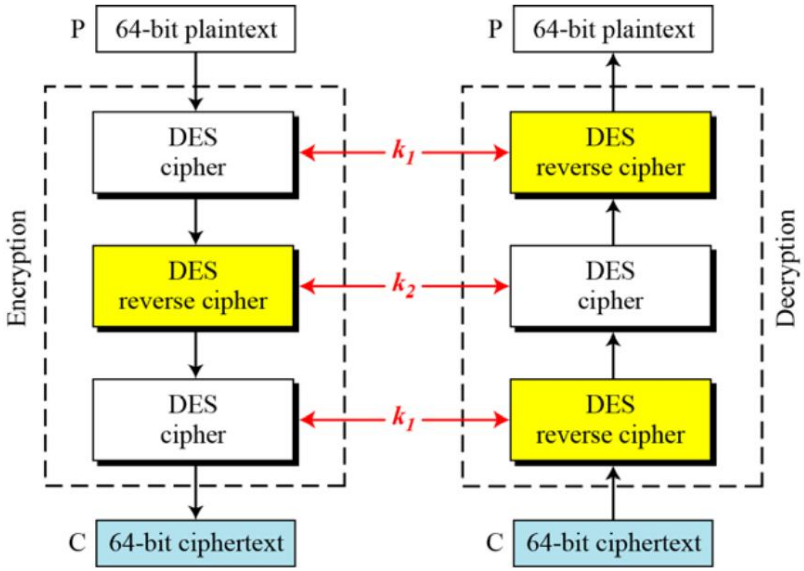
Hình 6.14 Cuộc tấn công giữa cuộc đối với DES kép



Hình 6.15 Các bảng cho cuộc tấn công gặp giữa



Hình 6.16 Triple DES với hai phím



Ba DES với ba khóa Khả năng tấn công bằng văn bản đã biết trên ba DES bằng hai khóa đã lôi kéo một số ứng dụng sử dụng ba DES với ba khóa. Triple DES với ba khóa đư ợc sử dụng trong nhiều ứng dụng như PGP (Xem Chư ơ ng 16).

6-5 Bảo mật của DES

DES, với tư cách là mật mã khối quan trọng đầu tiên, đã trải qua nhiều sự xem xét kỹ lư ỡng. Trong số các cuộc tấn công đã cố gắng, có ba cuộc tấn công đư ợc quan tâm: brute-force, phân tích mật mã vi sai và phân tích mật mã tuyến tính.

Các chủ đề đư ợc thảo luận trong phần này:

6.5.1 Tấn công Brute-Force 6.5.2 Phân tích mật mã vi sai 6.5.3 Phân tích mật mã tuyến tính

6.5.1 Tấn công vũ phu

Chúng ta đã thảo luận về điểm yếu của khóa mật mã ngắn trong DES. Kết hợp điểm yếu này với điểm yếu bổ sung khóa, rõ ràng là DES có thể bị phá bằng cách sử dụng 255 mã hóa.

6.5.2 Phân tích mật mã vi phân

Ngư ời ta tiết lộ rằng các nhà thiết kế của DES đã biết về kiểu tấn công này và đã thiết kế các hộp S và chọn 16 làm số vòng để làm cho DES có khả năng chống lại kiểu tấn công này một cách đặc biệt.

Ghi chú

Chúng tôi đư a ra một ví dụ về phân tích mật mã vi sai DES trong Phụ lục N.

6.5.3 Phân tích mật mã tuyến tính

Phân tích mật mã tuyến tính mới hơ n phân tích mật mã vi phân. DES dễ bị tổn thư ơ ng hơ n khi sử dụng phư ơ ng pháp phân tích tuyến tính hơ n là phân tích vi sai. Hộp S không có khả năng chống lại sự phân tích mật mã tuyến tính. Ngư ời ta đã chứng minh rằng DES có thể bị phá vỡ bằng cách sử dụng 243 cặp bản rõ đã biết. Tuy nhiên, từ quan điểm thực tế, việc tìm đư ợc nhiều cặp như vậy là rất khó xảy ra.

Ghi chú

Chúng tôi trình bày một ví dụ về giải mã tuyến tính DES trong Phụ lục N.