TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN ĐIỆN TỬ - VIỄN THÔNG

BỘ MÔN ĐIỆN TỬ HÀNG KHÔNG VŨ TRỤ

Môn học:

LÝ THUYẾT MẬT MÃ
CRYPTOGRAPHY THEORY
ET3310

Giảng viên: PGS.TS. Đỗ Trọng Tuấn
Email: dotrongtuan@gmail.com

Mục tiêu học phần

Cung cấp kiến thức cơ bản về mật mã đảm bảo an toàn và bảo mật thông tin:

- ✓ Các phương pháp mật mã khóa đối xứng; Phương pháp mật mã khóa công khai;
- ✓ Các hệ mật dòng và vấn đề tạo dãy giả ngẫu nhiên;
- ✓ Lược đồ chữ ký số Elgamal và chuẩn chữ ký số ECDSA;
- ✓ Độ phức tạp xử lý và độ phức tạp dữ liệu của một tấn công cụ thể vào hệ thống mật mã;
- ✓ Đặc trưng an toàn của phương thức mã hóa;
- ✓ Thám mã tuyến tính, thám mã vi sai và các vấn đề về xây dựng hệ mã bảo mật cho các ứng dụng.

Nội Dung

1. Chương 1. Tổng quan
2. Chương 2. Mật mã khóa đối xứng
3. **Chương 3. Hệ mật DES**
4. Chương 4. Hàm băm và chữ ký số
5. Chương 5. Dãy giả ngẫu nhiên và hệ mật dòng
6. Chương 6. Kỹ thuật quản lý khóa

Tài liệu tham khảo

1. A. J. Menezes, P. C. Van Oorschot, S. A. Vanstone, *Handbook of applied cryptography*, CRC Press 1998.
2. B. Schneier, *Applied Cryptography*. John Wiley Press 1996.
3. M. R. A. Huth, *Secure Communicating Systems*, Cambridge University Press 2001.
4. W. Stallings, *Network Security Essentials, Applications and Standards*, Prentice Hall. 2000.

Nhiệm vụ của Sinh viên

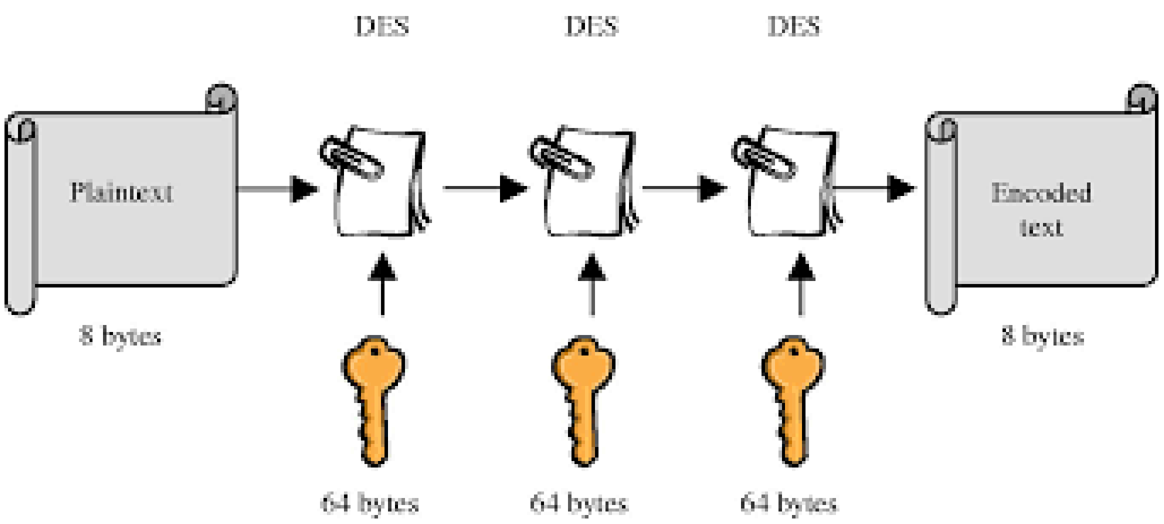
1. Chấp hành nội quy lớp học
2. Thực hiện đầy đủ bài tập
3. Nắm vững ngôn ngữ lập trình Matlab

Chương 3. Hệ mật DES

3.1. Giới thiệu sơ lược hệ mật DES

3.2. Cấu trúc hệ mật DES

3.3. Thám mã hệ mật DES



3.1. Sơ lược hệ mật DES

The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).

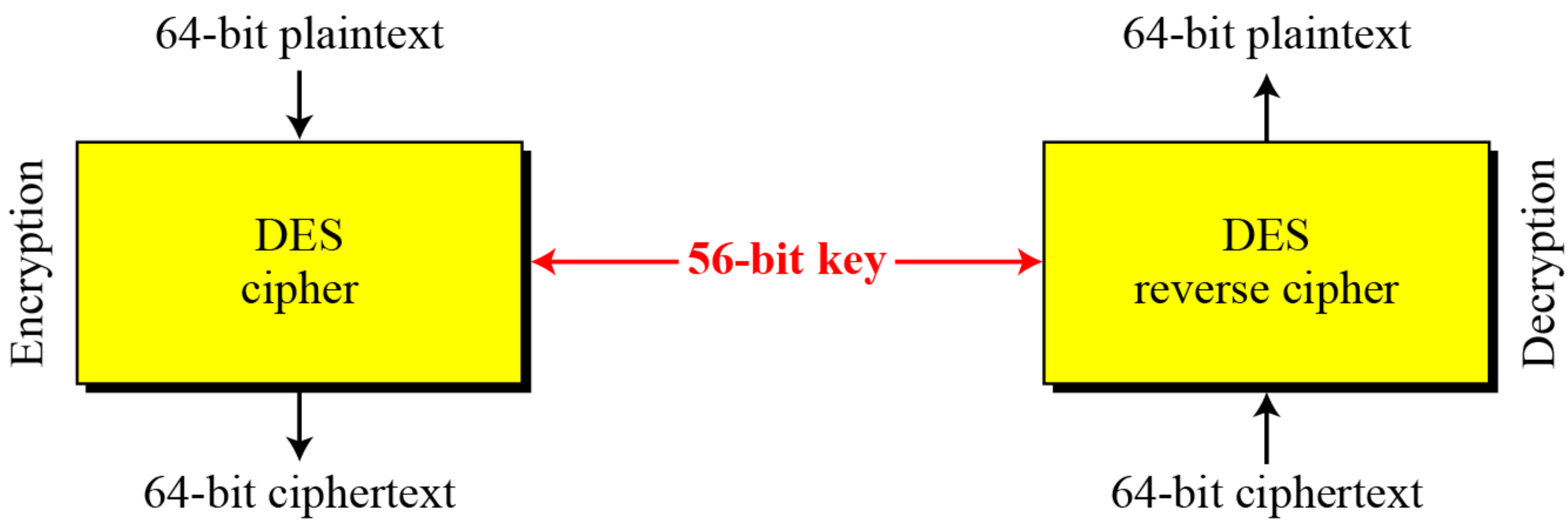
In 1973, NIST published a request for proposals for a national symmetric-key cryptosystem. A proposal from IBM, a modification of a project called Lucifer, was accepted as DES. DES was published in the Federal Register in March 1975 as a draft of the Federal Information Processing Standard (FIPS).

3.1. Sơ lược hệ mật DES

- Published by NIST in 1977
- A variation of IBM's Lucifer algorithm developed by Horst Feistel
- For commercial and *unclassified* government applications
- 8 octet (64 bit) key.
  Each octet with 1 odd parity bit ⇒ 56-bit key
- Efficient hardware implementation
- Used in most financial transactions
- Computing power goes up 1 bit every 2 years
- 56-bit was secure in 1977 but is not secure today
- Now we use DES three times ⇒ Triple DES = 3DES

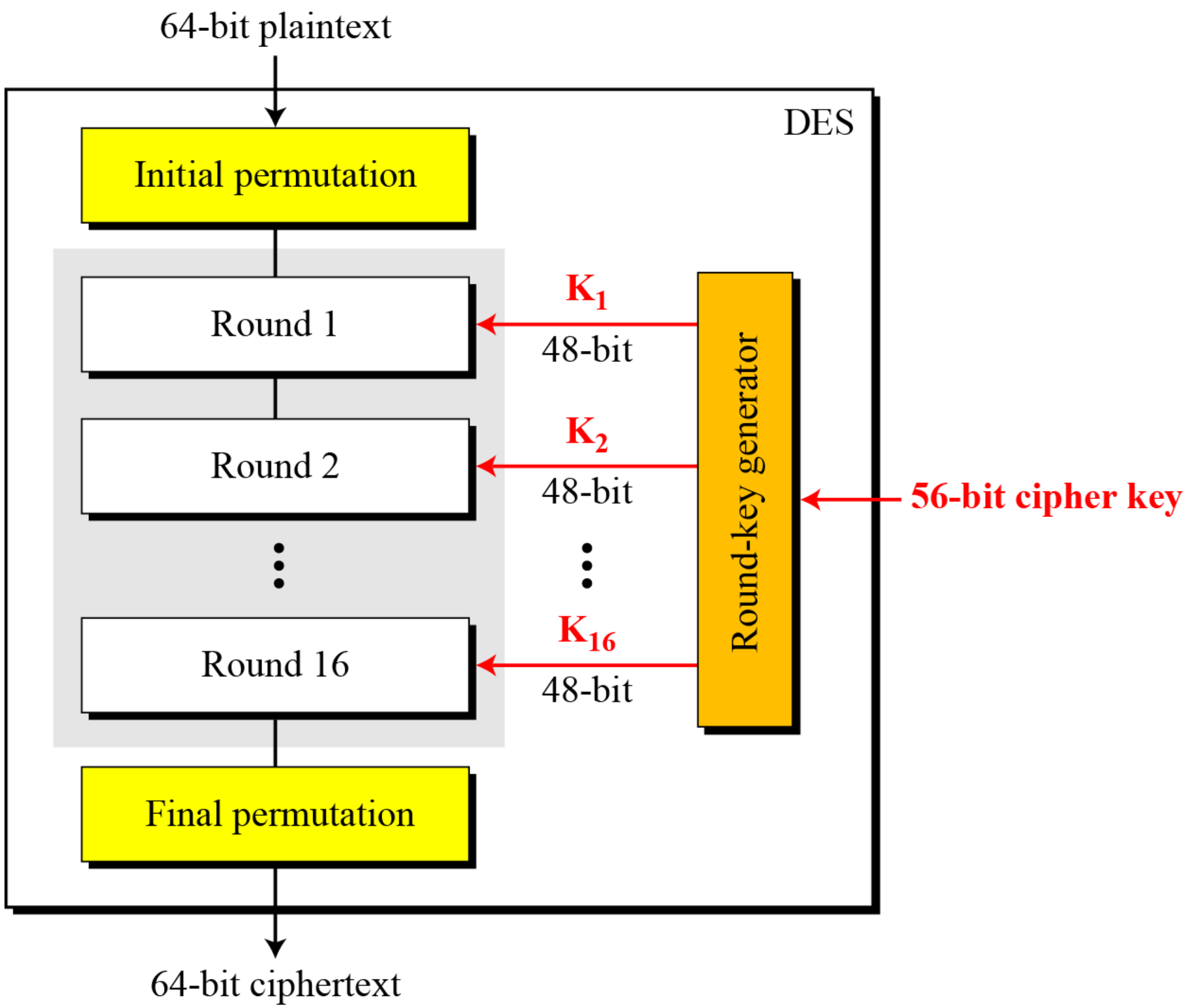
3.2. Cấu trúc hệ mật DES

DES is a block cipher



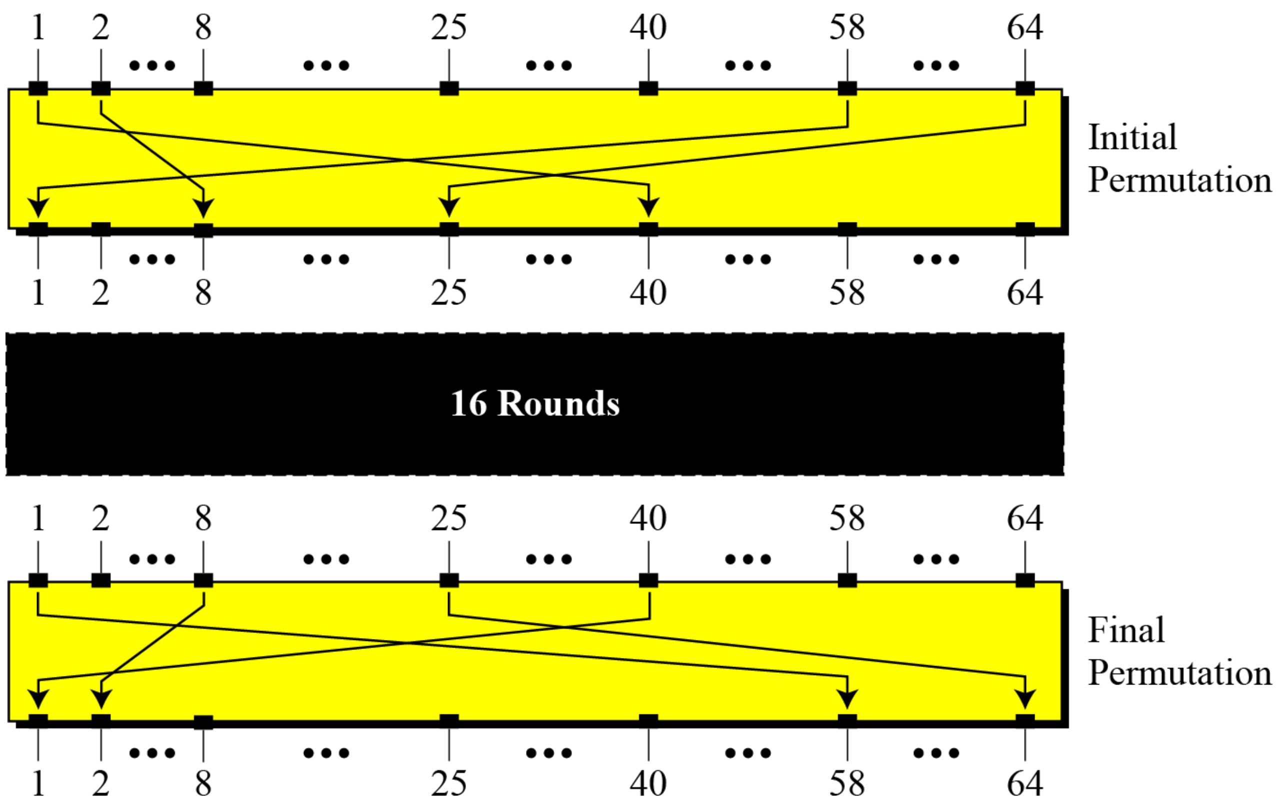
3.2. Cấu trúc hệ mật DES

The encryption process is made of two permutations (P-boxes), which we call initial and final permutations, and sixteen Feistel rounds.



3.2. Cấu trúc hệ mật DES

Initial and final permutation steps in DES



3.2. Cấu trúc hệ mật DES

Initial and final permutation steps in DES

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

3.2. Cấu trúc hệ mật DES

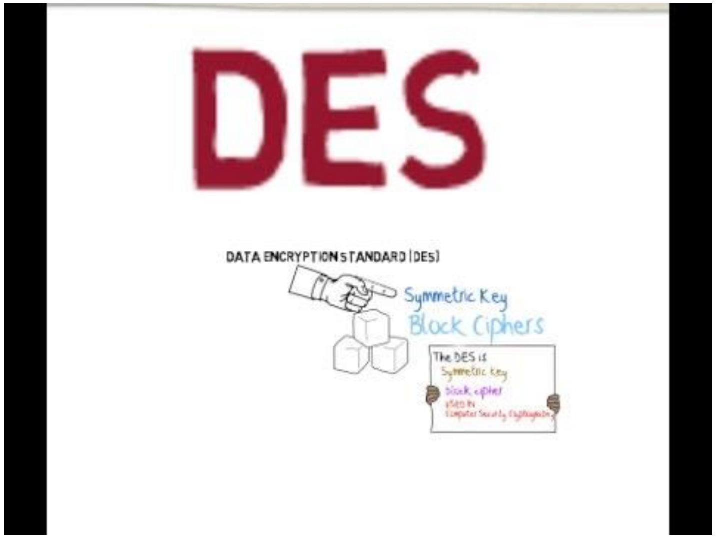
Ví dụ

Find the output of the final permutation box when the input is given in hexadecimal as:

0x0000 0080 0000 0002

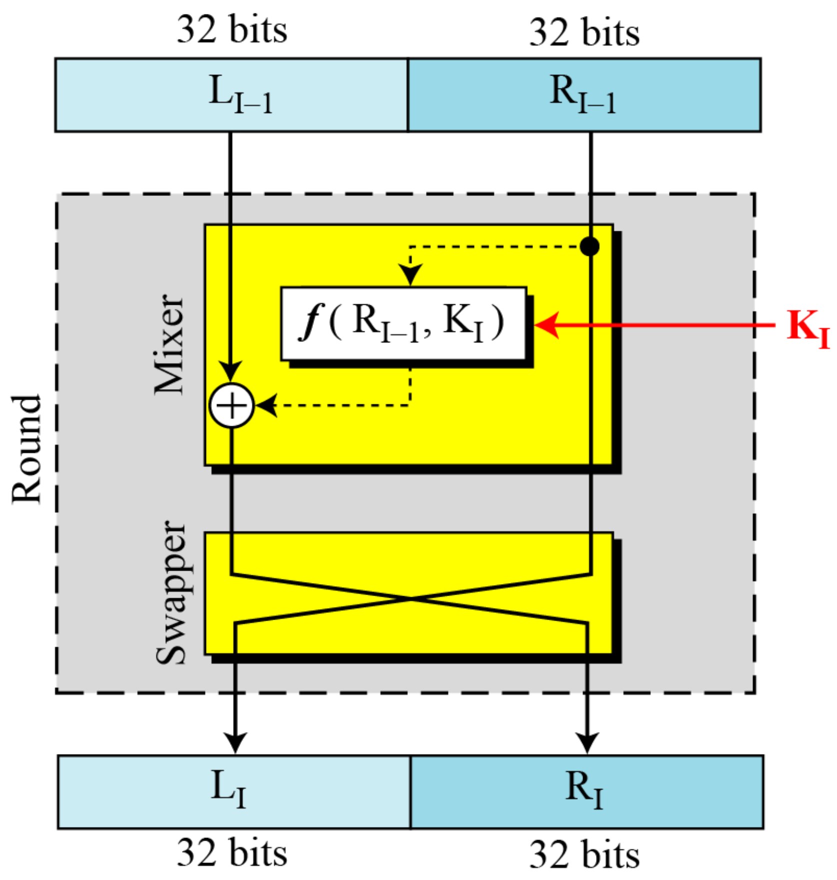
3.2. Cấu trúc hệ mật DES

The initial and final permutations are straight P-boxes that are inverses of each other.
They have no cryptography significance in DES.



3.2. Cấu trúc hệ mật DES

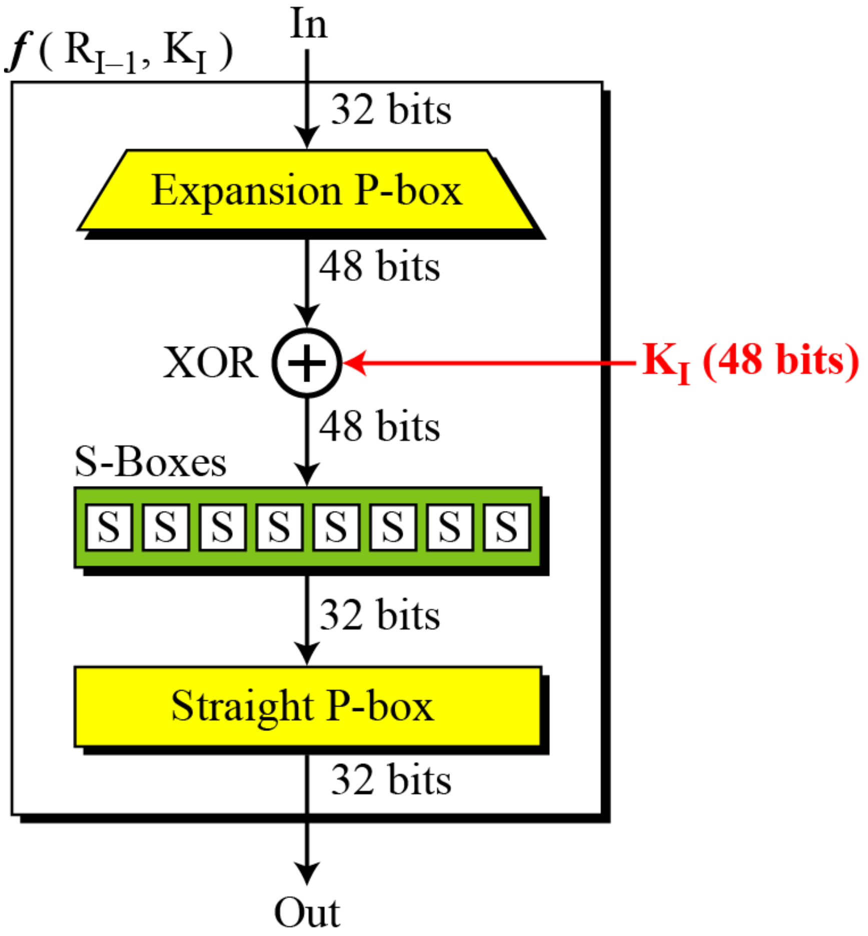
DES uses 16 rounds. Each round of DES is a Feistel cipher.



3.2. Cấu trúc hệ mật DES

DES Function

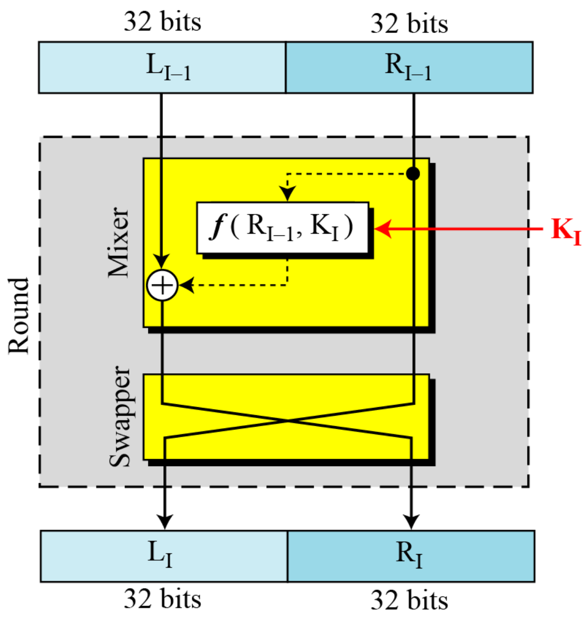
The heart of DES is the DES function. The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.



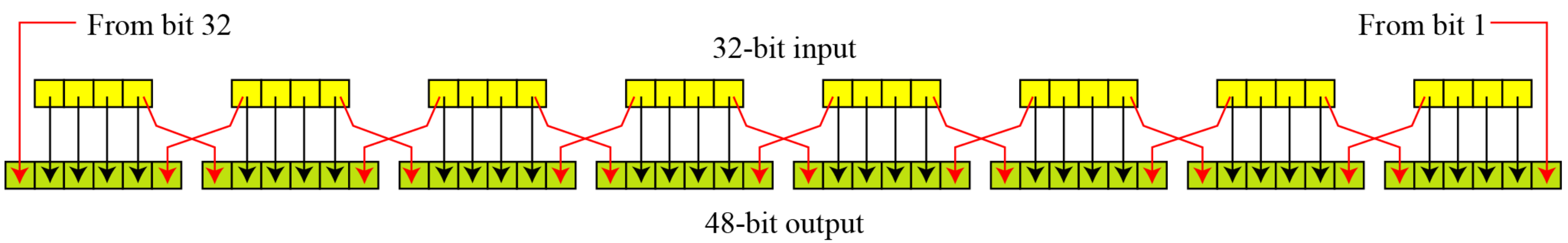
3.2. Cấu trúc hệ mật DES

Since R_{I-1} is a 32-bit input and K_I is a 48-bit key.

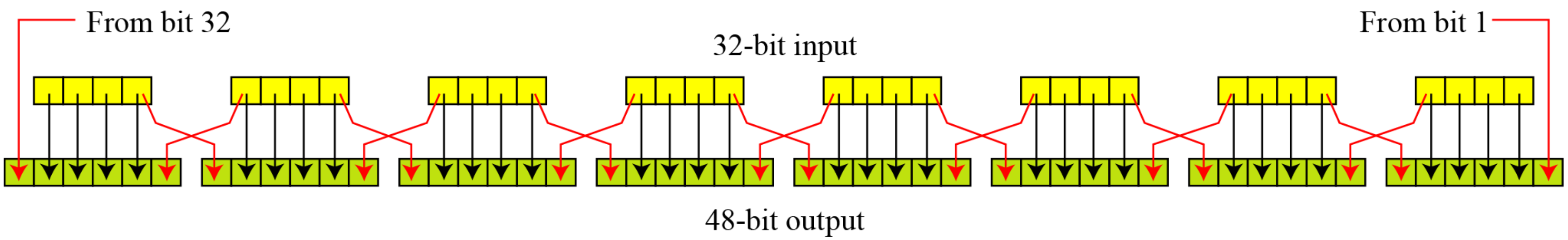
It needs to expand R_{I-1} to 48 bits.



Expansion P-box



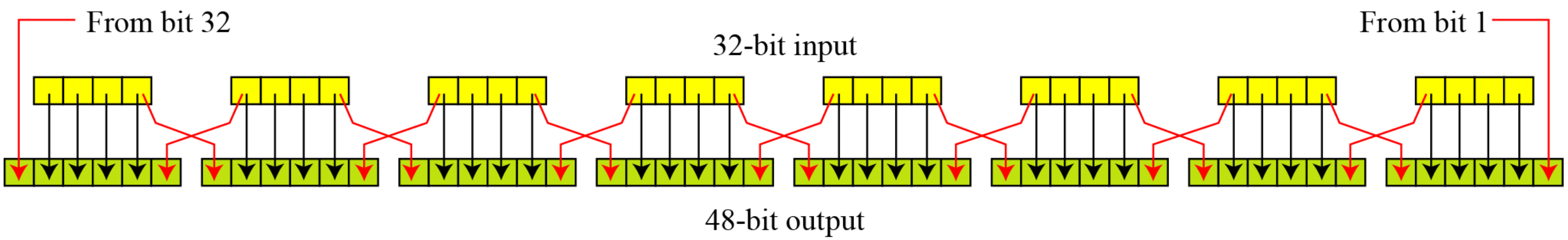
3.2. Cấu trúc hệ mật DES



DES uses this table to define this P-box

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

3.2. Cấu trúc hệ mật DES



Whitener (XOR)

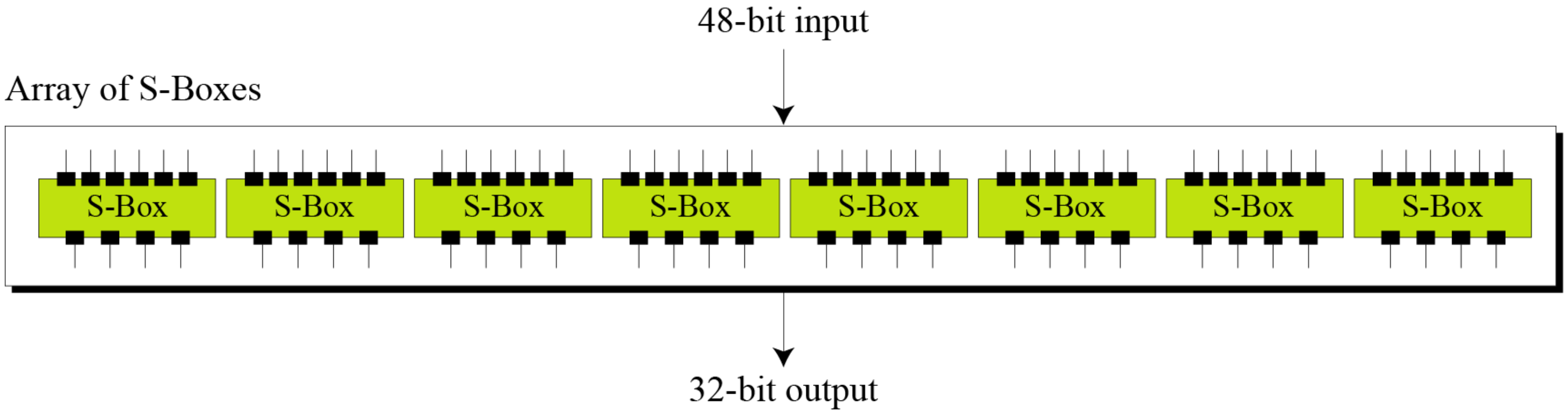
After the expansion permutation, DES uses the XOR operation on the expanded right section and the round key.

- ***Note that:***
  - ***Both the right section and the key are 48-bits in length.***
  - ***The round key is used only in this operation.***

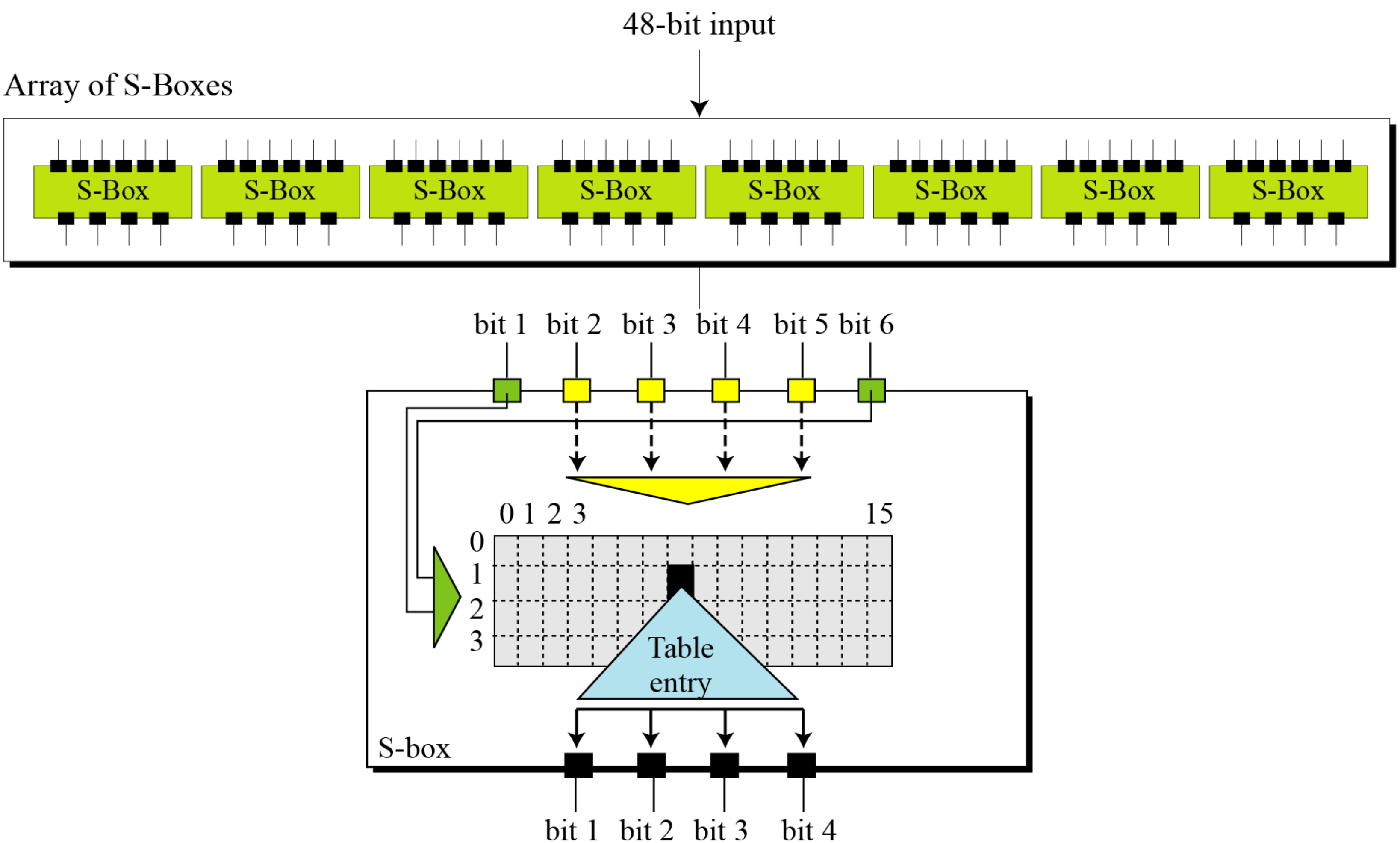
3.2. Cấu trúc hệ mật DES

S-Boxes

The S-boxes do the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output.



3.2. Cấu trúc hệ mật DES



3.2. Cấu trúc hệ mật DES

S-box 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

S-box 2

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

3.2. Cấu trúc hệ mật DES

S-box 3

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

S-box 4

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

3.2. Cấu trúc hệ mật DES

S-box 5

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

S-box 6

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

3.2. Cấu trúc hệ mật DES

S-box 7

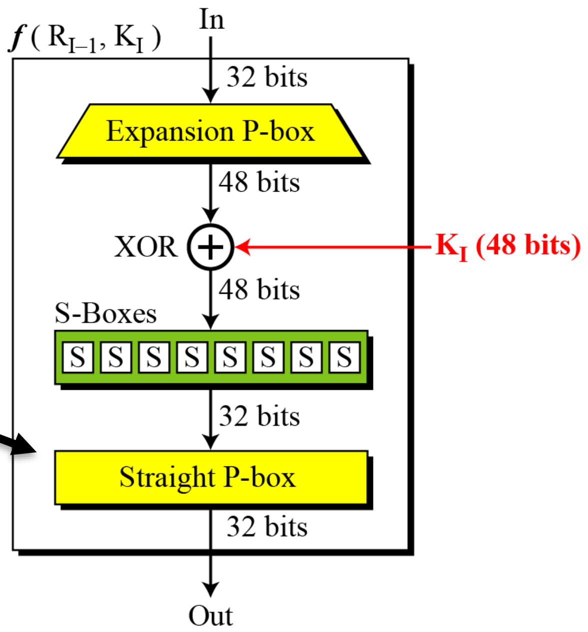
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	11	2	14	15	00	08	13	03	12	09	07	05	10	06	01
1	13	00	11	07	04	09	01	10	14	03	05	12	02	15	08	06
2	01	04	11	13	12	03	07	14	10	15	06	08	00	05	09	02
3	06	11	13	08	01	04	10	07	09	05	00	15	14	02	03	12

S-box 8

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

3.2. Cấu trúc hệ mật DES

Straight Permutation



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

3.2. Cấu trúc hệ mật DES

Using mixers and swappers, we can create the cipher and reverse cipher, each having 16 rounds.

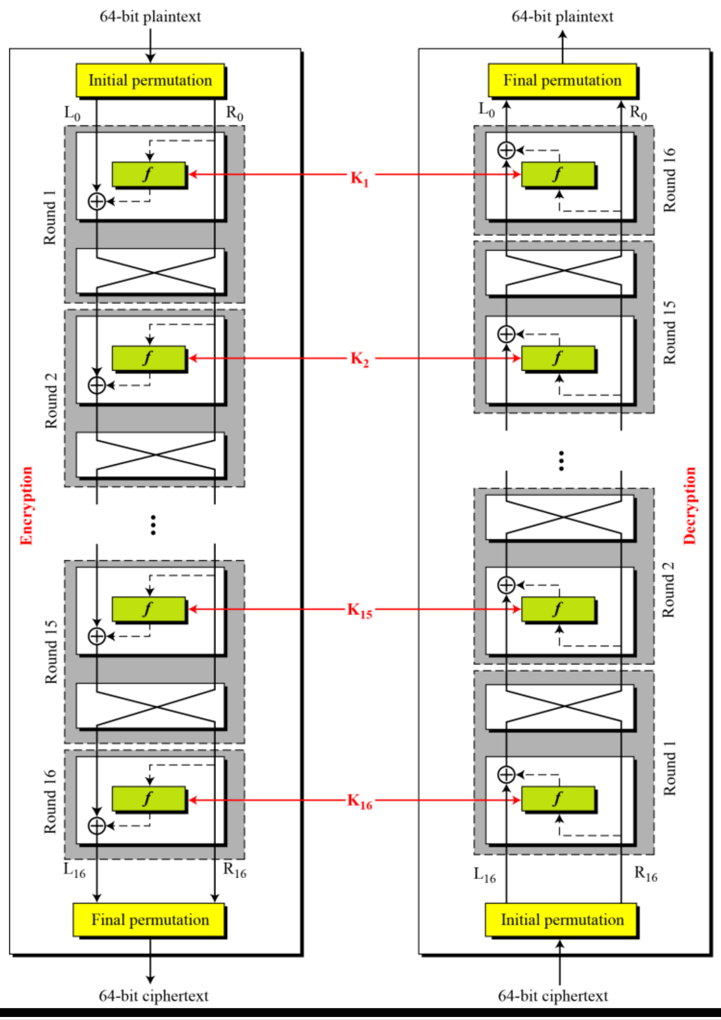
First Approach

To achieve this goal, one approach is to make the last round (round 16) different from the others; it has only a mixer and no swapper.

In the first approach, there is no swapper in the last round.

3.2. Cấu trúc hệ mật DES

*Using **mixers** and **swappers**, we can create the cipher and reverse cipher, each having 16 rounds.*



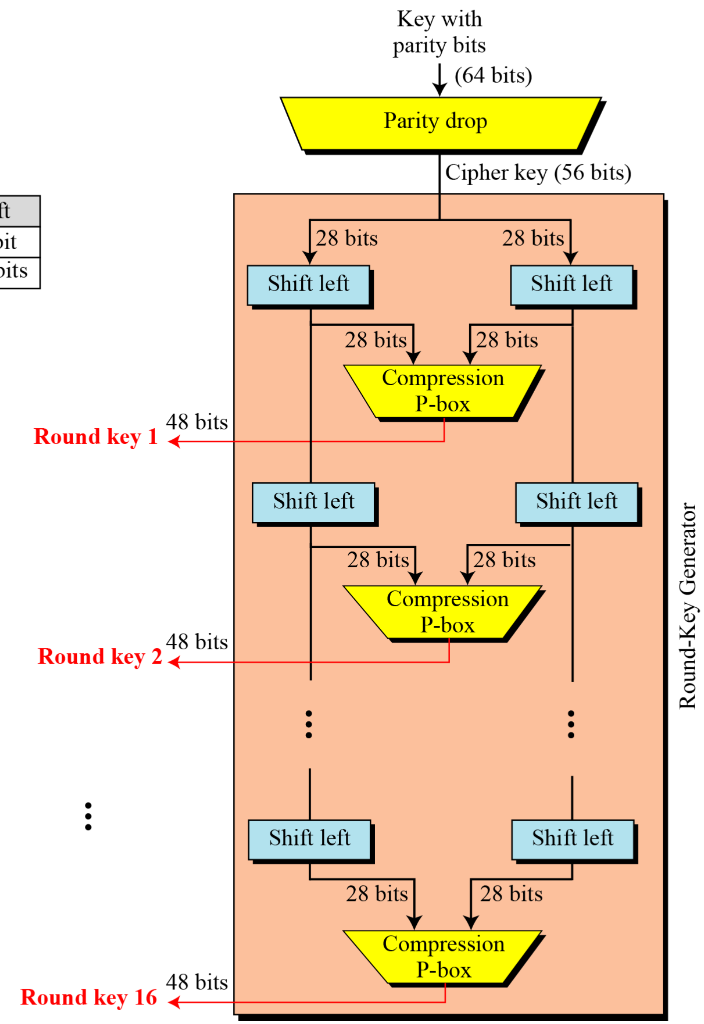
3.2. Cấu trúc hệ mật DES

Key Generation

The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key.

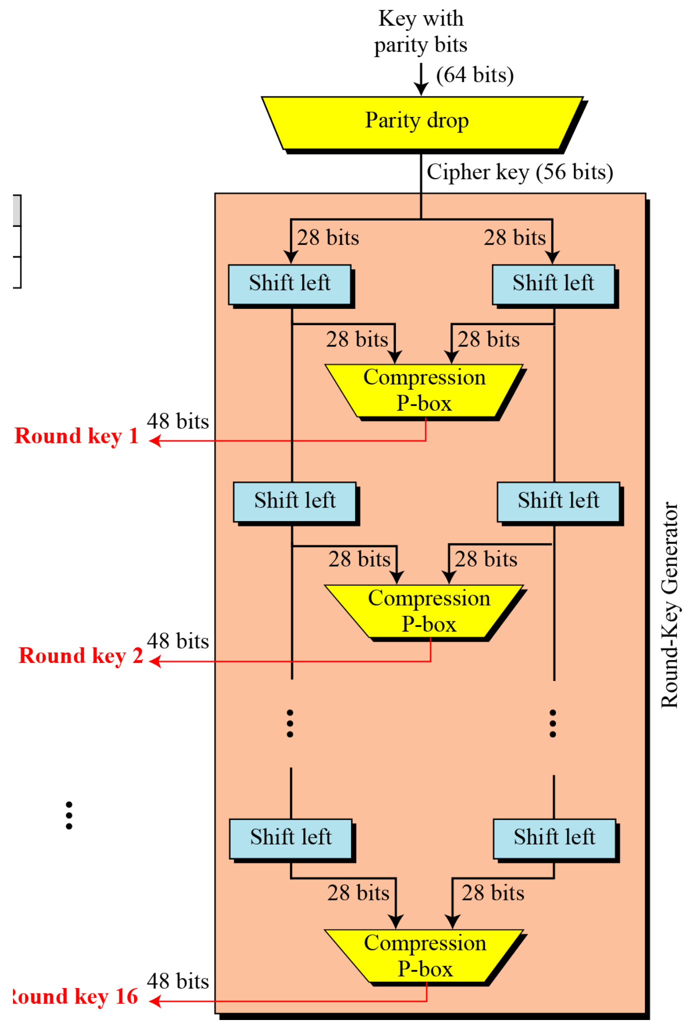
Shifting

Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits



3.2. Cấu trúc hệ mật DES

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



3.2. Cấu trúc hệ mật DES

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

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

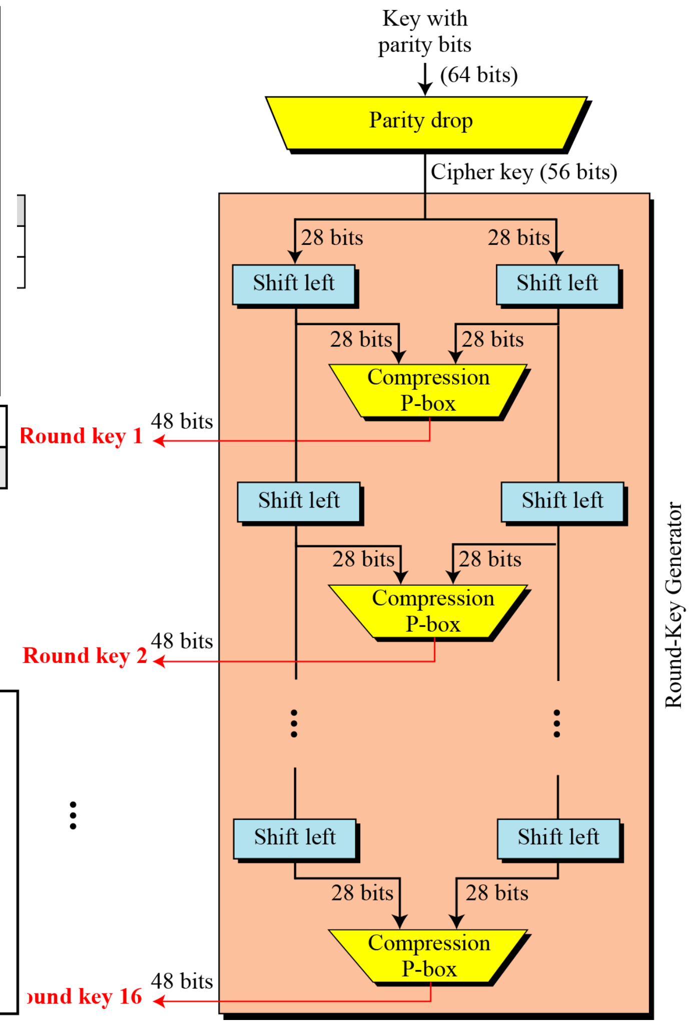
3.2. Cấu trúc hệ mật DES

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

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

Key-compression table

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



3.2. Cấu trúc hệ mật DES

Ví dụ

We choose a random plaintext block and a random key, and determine what the ciphertext block would be (all in hexadecimal):

Plaintext: 123456ABCD132536	Key: AABB09182736CCDD
CipherText: C0B7A8D05F3A829C	

3.2. Cấu trúc hệ mật DES

Plaintext: 123456ABCD132536 Key: AABB09182736CCDD
CipherText: C0B7A8D05F3A829C

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

3.2. Cấu trúc hệ mật DES

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)*

3.2. Cấu trúc hệ mật DES

At the destination, Bob can decipher the ciphertext received from Alice using the same key.

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)

3.3. Thám mã hệ mật DES

Two desired properties of a block cipher are the avalanche effect and the completeness.

Plaintext: 0000000000000000	Key: 22234512987ABB23
Ciphertext: 4789FD476E82A5F1	
Plaintext: 000000000000000<u>1</u>	Key: 22234512987ABB23
Ciphertext: 0A4ED5C15A63FEA3	

Completeness effect

Completeness effect means that each bit of the ciphertext needs to depend on many bits on the plaintext.

3.3. Thám mã hệ mật DES

During the last few years critics have found some weaknesses in DES.

Weaknesses in Cipher Design

1. Weaknesses in S-boxes
2. Weaknesses in P-boxes
3. Weaknesses in Key

3.3. Thám mã hệ mật DES

S-boxes At least three weaknesses are mentioned in the literature for S-boxes.

1. In S-box 4, the last three output bits can be derived in the same way as the first output bit by complementing some of the input bits.
2. Two specifically chosen inputs to an S-box array can create the same output.
3. It is possible to obtain the same output in a single round by changing bits in only three neighboring S-boxes.

P-boxes One mystery and one weakness were found in the design of P-boxes:

1. It is not clear why the designers of DES used the initial and final permutations; these have no security benefits.
2. In the expansion permutation (inside the function), the first and fourth bits of every 4-bit series are repeated.

3.3. Thám mã hệ mật DES

Key Size Critics believe that the most serious weakness of DES is in its key size (56 bits). To do a brute-force attack on a given ciphertext block, the adversary needs to check 2^{56} keys.

a. With available technology, it is possible to check one million keys per second. This means that we need more than two thousand years to do brute-force attacks on DES using only a computer with one processor.

b. If we can make a computer with one million chips (parallel processing), then we can test the whole key domain in approximately 20 hours. When DES was introduced, the cost of such a computer was over several million dollars, but the cost has dropped rapidly. A special computer was built in 1998 that found the key in 112 hours.

c. Computer networks can simulate parallel processing. In 1977 a team of researchers used 3500 computers attached to the Internet to find a key challenged by RSA Laboratories in 120 days. The key domain was divided among all of these computers, and each computer was responsible to check the part of the domain.

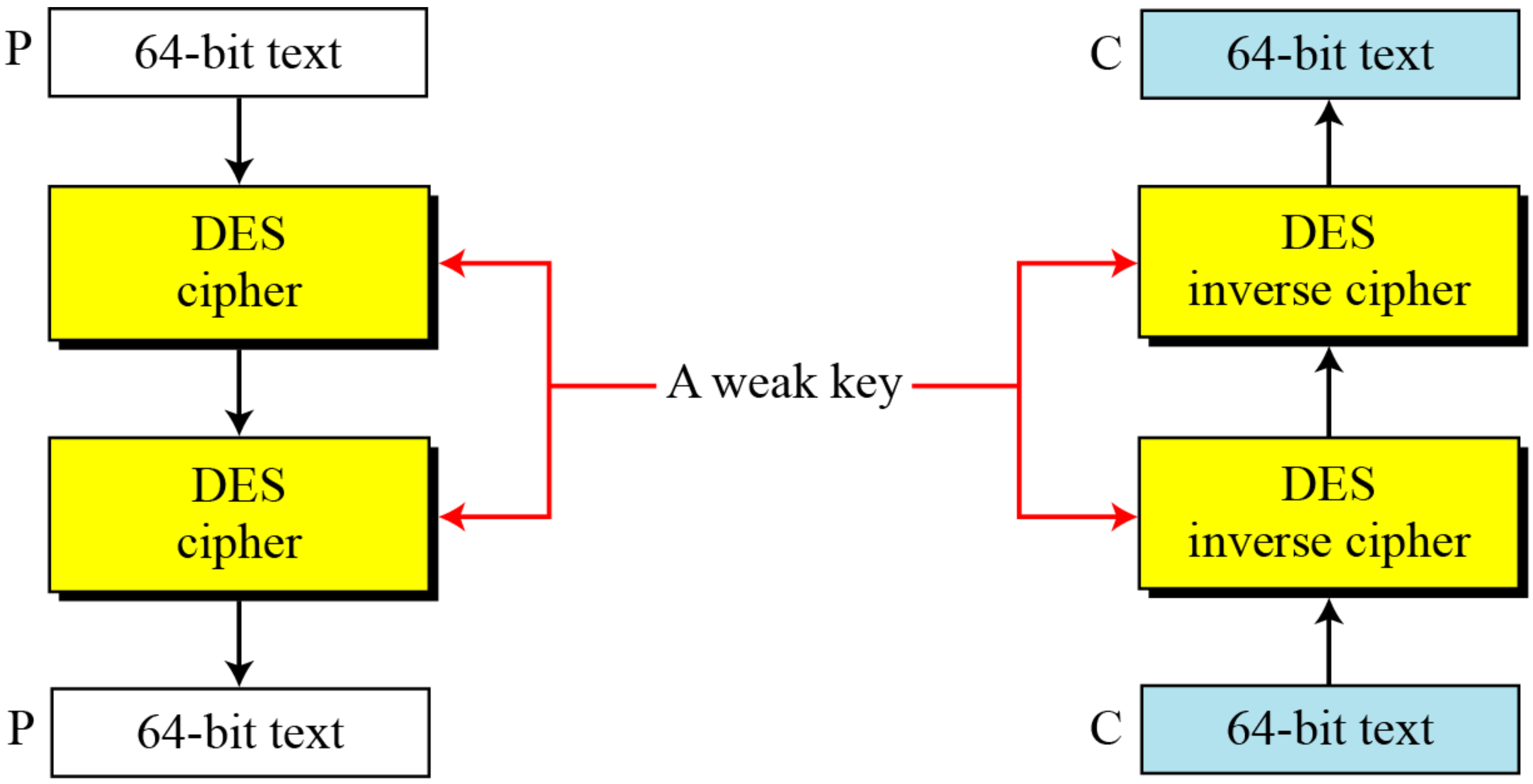
d. If 3500 networked computers can find the key in 120 days, a secret society with 42,000 members can find the key in 10 days.

3.3. Thám mã hệ mật DES

Trong 2^{56} trường hợp khóa K có 4 khóa có độ an toàn rất kém đó là các khóa toàn 0 hoặc 1

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

3.3. Thám mã hệ mật DES



3.3. Thám mã hệ mật DES

Let's try the first weak key to encrypt a block two times. After two encryptions with the same key the original plaintext block is created. Note that we have used the encryption algorithm two times, not one encryption followed by another decryption.

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

Weak key should be avoided

3.3. Thám mã hệ mật DES

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

Semi-weak Keys There are six key pairs that are called **semi-weak keys.** These six pairs are shown in Table (64-bit format before dropping the parity bits).

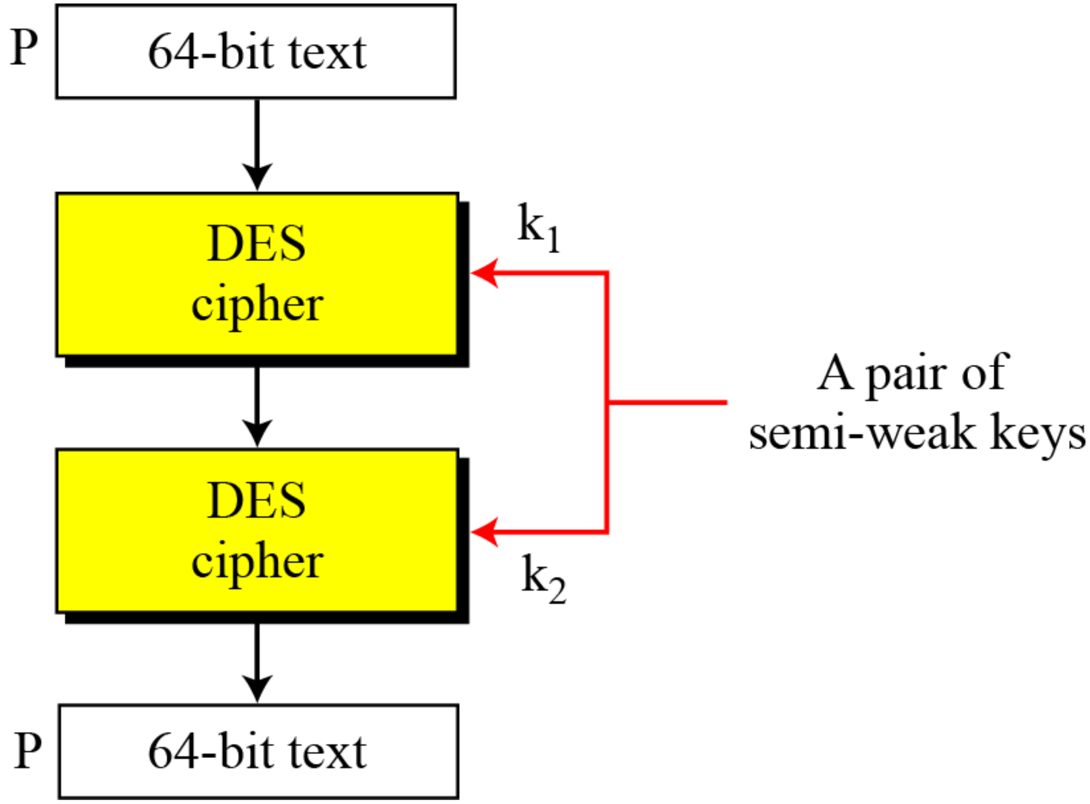
3.3. Thám mã hệ mật DES

A semi-weak key creates only two different round keys and each of them is repeated eight times. In addition, the round keys created from each pair are the same

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

3.3. Thám mã hệ mật DES

A pair of semi-weak keys in encryption and decryption



3.3. Thám mã hệ mật DES

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.

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