11464: INFORMATION SYSTEMS SECURITY

12/5/2021

Chapter 4: Modern Encryption Techniques

Modern Encryption Techniques

By

Mustafa Al-Fayoumi

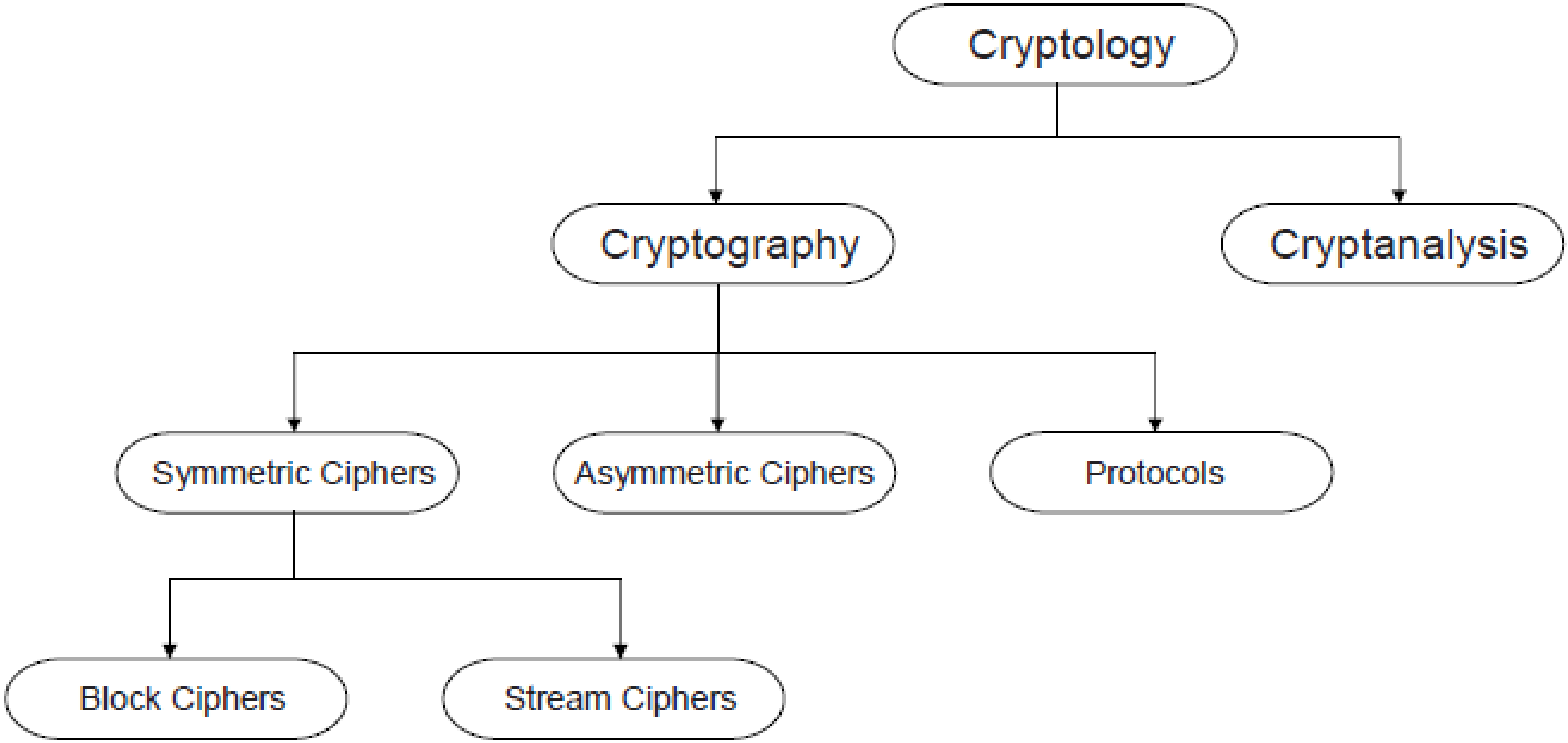
Objectives

- To distinguish between traditional and modern symmetric-key ciphers.
- To introduce modern block ciphers and discuss their characteristics.
- To explain why modern block ciphers need to be designed as substitution ciphers.
- To introduce components of block ciphers such as P-boxes and S-boxes.
- To discuss product ciphers and distinguish between two classes of product ciphers: Feistel and non-Feistel ciphers.

Outline

- Block Cipher Princeple
  - Stream cipher
  - Block cipher
  - The Feistel Cipher
- The Data Encryption Standard (DES)
  - DES Encryption
  - DES Decryption
  - The strength of DES
  - Simplified DES
- Advanced Encryption Standard (AES)
  - AES Structure
  - Detailed Structure

Classification the field of Cryptography



Types of symmetric cipher Operations

- **Substitution cipher:** The encryption process systematically manipulate a symbol (or a group of symbols) in the plaintext to produce a different symbol (or group of symbols), which becomes the ciphertext.
  - The substituted symbols in the ciphertext appear in exactly the same order as the original versions in the plaintext.
- **Transposition cipher:** The encryption process ‘scrambles’ the order of the symbols of the plaintext in some systematic way.
    - Using this approach, the symbols remain unchanged between plaintext and ciphertext, but the ordering of those symbols changes
- **Product cipher:** Combination between transposition cipher and substitution cipher.

Block Cipher

A block of plaintext is treated as a whole and used to produce a ciphertext block of equal length

Typically a block size of 64 or 128 bits is used

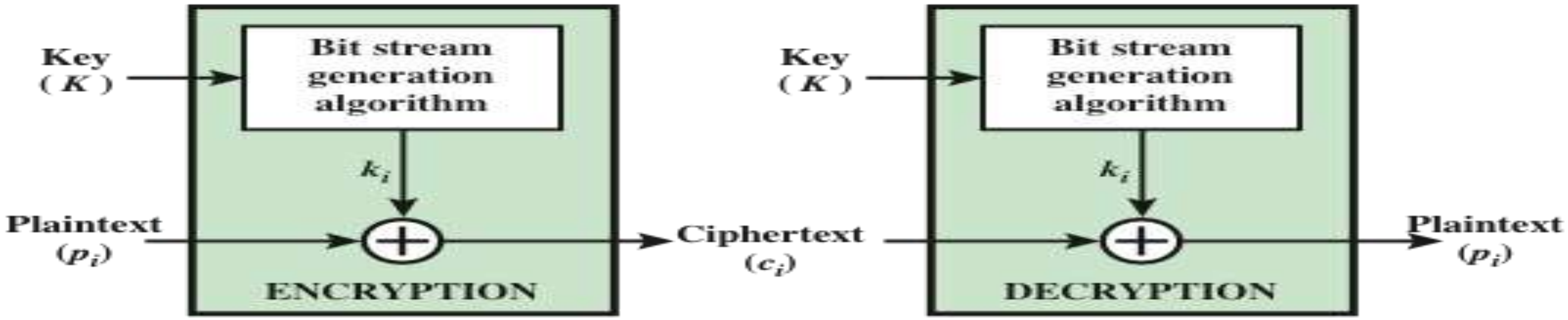
As with a stream cipher, the two users share a symmetric encryption key

The majority of network-based symmetric cryptographic applications make use of block ciphers

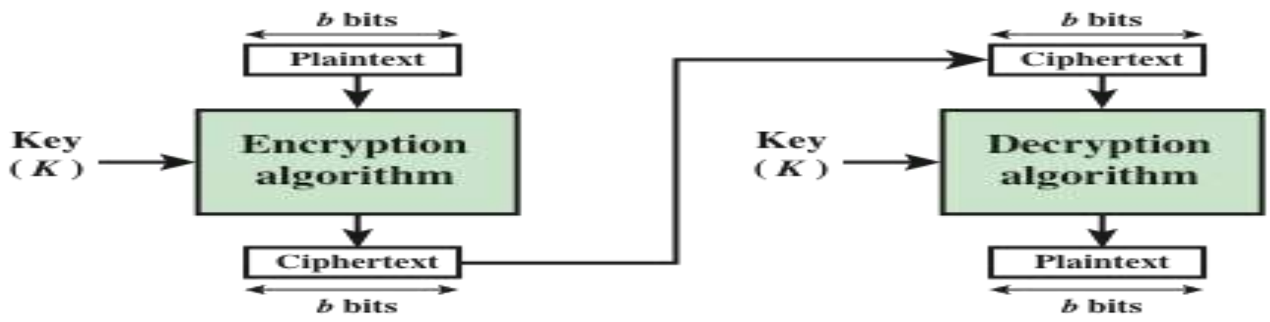
Block vs Stream Ciphers

- Block ciphers process messages into blocks, each of which is then en/decrypted
- like a substitution on very big characters
  - 64-bits or more
- Stream ciphers process messages a bit or byte at a time when en/decrypting
- many current ciphers are block ciphers

Stream Cipher and Block Cipher



(a) Stream Cipher Using Algorithmic Bit Stream Generator



(b) Block Cipher

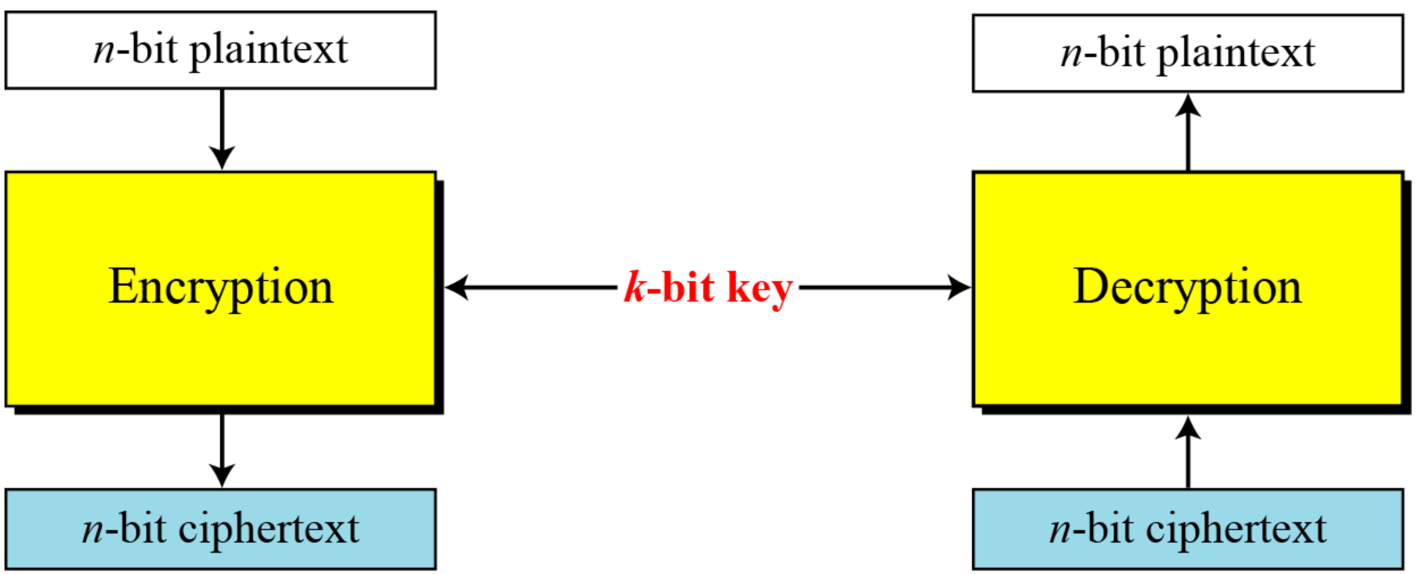
Figure 3.1 Stream Cipher and Block Cipher

10 Product Cipher

Modern Block Ciphers

- A symmetric-key modern block cipher encrypts an n-bit block of plaintext or decrypts an n-bit block of ciphertext. The encryption or decryption algorithm uses a k-bit key.

Figure 4.1 ***A modern block cipher***



Example

- How many padding bits must be added to a message of 100 characters if 8-bit ASCII is used for encoding and the block cipher accepts blocks of 64 bits?

Solution

- Encoding 100 characters using 8-bit ASCII results in an 800-bit message. The plaintext must be divisible by 64. If | M | and |Pad| are the length of the message and the length of the padding,

$$|M| + |Pad| = 0 \bmod 64 \quad \rightarrow \quad |Pad| = -800 \bmod 64 \quad \rightarrow \quad 32 \bmod 64$$

Block Cipher Primitives: Confusion and Diffusion

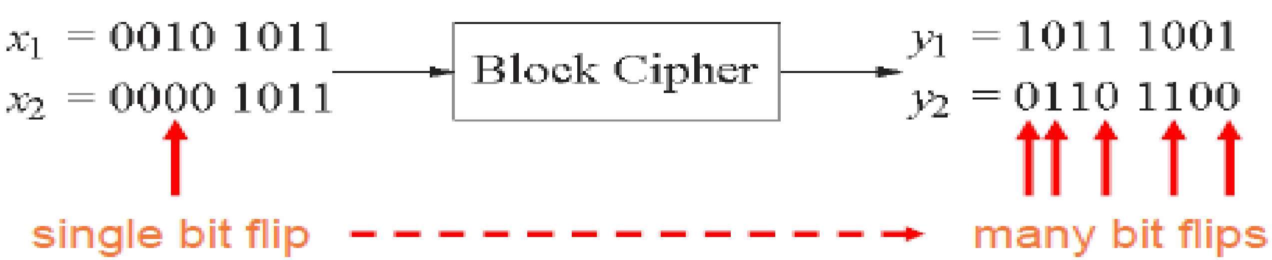
- in 1949 Shannon introduced idea of substitution-permutation (S-P) networks
  - modern substitution-transposition product cipher
- these form the basis of modern block ciphers
- S-P networks are based on the two primitive cryptographic operations we have seen before:
  - *substitution* (S-box)
  - *permutation* (P-box) (transposition)
- Terms introduced by Claude Shannon to capture the two basic building blocks for any cryptographic system
  - Shannon's concern was to thwart cryptanalysis based on statistical analysis
    - Assume the attacker has some knowledge of the statistical characteristics of the plaintext
  - cipher needs to completely obscure statistical properties of original message
  - provide *confusion* and *diffusion* of message

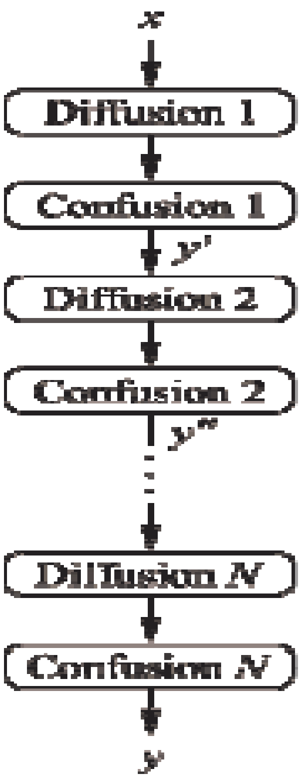
Diffusion and Confusion

- **Claude Shannon**: There are two primitive operations with which strong encryption algorithms can be built:
  - **diffusion** – dissipates statistical structure of plaintext over bulk of ciphertext (the influence of one plaintext symbol (bit) is spread over many cipher text symbols (bits)
  - A simple diffusion element is the **bit permutation**, which is frequently used within DES.
  - **confusion** – makes relationship between ciphertext and key as complex as possible (relationship between the ciphertext and the key is obscured)
  - Today, a common element for achieving confusion is **substitution**, which is found in both AES and DES.
- Both operations by themselves cannot provide security. The idea is to concatenate confusion and diffusion elements to build so called *product ciphers*.

Product Ciphers

- Most of today's block ciphers are *product ciphers* as they consist of rounds which are applied repeatedly to the data.
- Can reach excellent diffusion: **changing of one bit of plaintext results** *on average* in the **change of half the output bits.**
- **Example:**





Feistel Cipher

- Proposed the use of a cipher that alternates substitutions and permutations

Substitutions
- Each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements

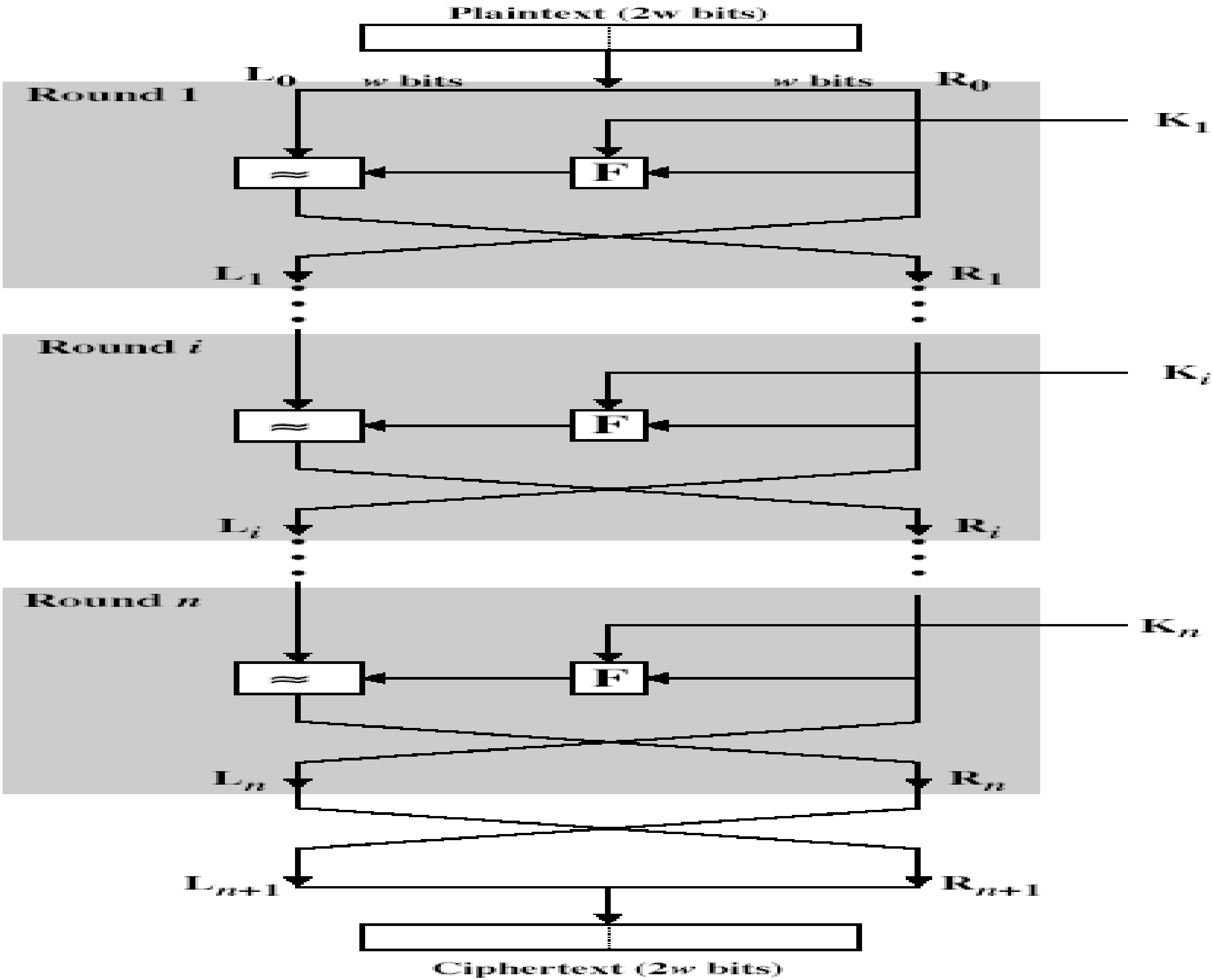
Permutation
- No elements are added or deleted or replaced in the sequence, rather the order in which the elements appear in the sequence is changed

- Is a practical application of a proposal by Claude Shannon to develop a product cipher that alternates confusion and diffusion functions

- Is the structure used by many significant symmetric block ciphers currently in use

Feistel Cipher Structure

- Horst Feistel devised the **feistel cipher**
  - implements Shannon’s substitution-permutation network concept
- partitions input block into two halves
  - process through multiple rounds which
  - perform a substitution on left data half
  - based on round function of right half & subkey
  - then have permutation swapping halves


Plaintext (2w bits)
L0
w bits
w bits
R0
Round 1
K1
≈
F
L1
R1
Round i
Ki
≈
F
Li
Ri
Round n
Kn
≈
F
Ln
Rn
Ln+1
Rn+1
Ciphertext (2w bits)

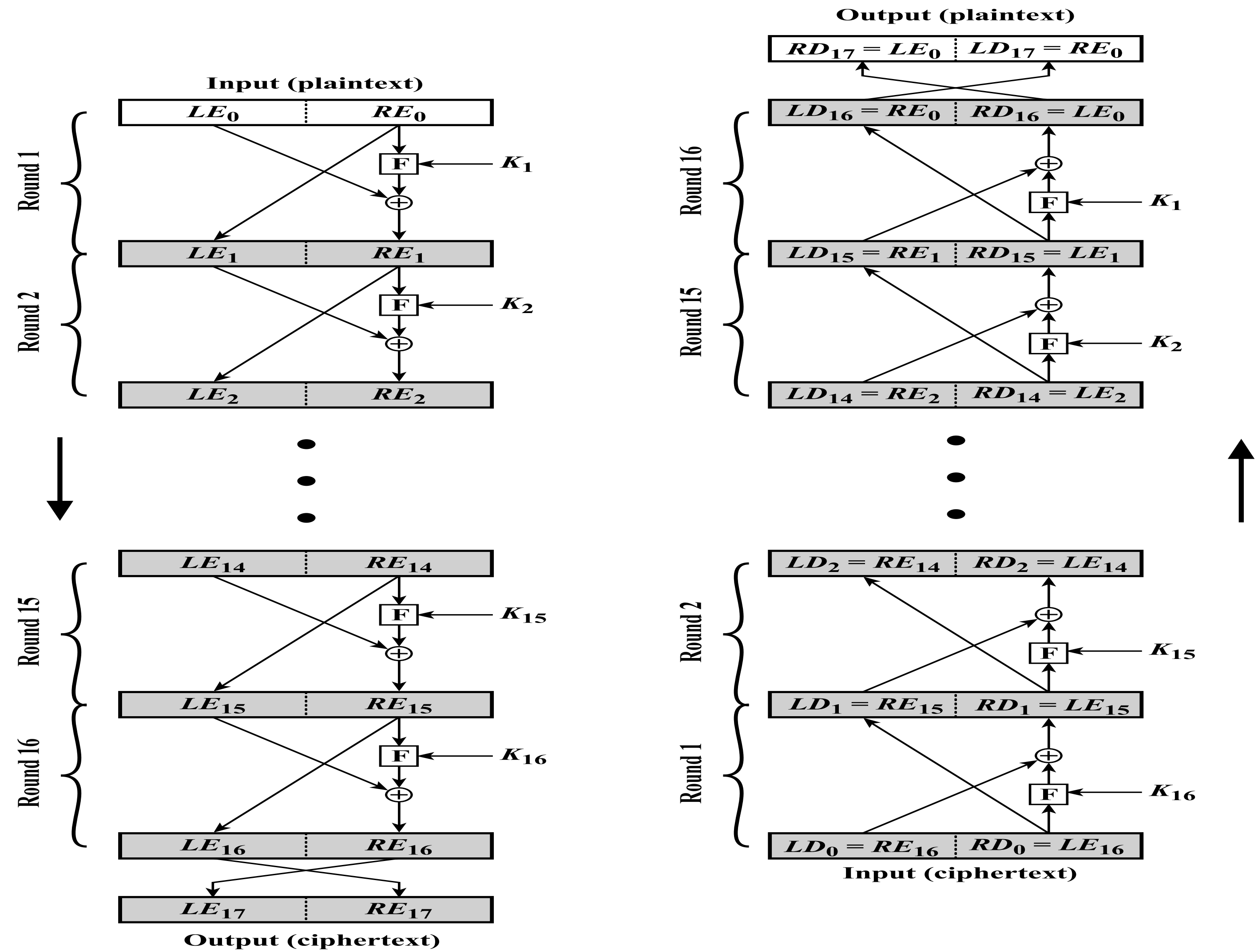


Figure 3.3 Feistel Encryption and Decryption (16 rounds)

Feistel Cipher

- n sequential rounds
- A substitution on the left half L_i
  - 1. Apply a round function F to the right half R_i and
  - 2. Take XOR of the output of (1) and L_i
- The round function is parameterized by the subkey K_i
  - K_i are derived from the overall key K

$$RE_1 = LE_0 \oplus F(RE_0, K_1)$$

Feistel Cipher

- Rounds can be expressed as:

$$L_i = R_{i-1},$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$$

- *For Encryption :*

$$LE_i = RE_{i-1}$$
$$RE_i = LE_{i-1} \oplus [RE_{i-1}, k_i]$$

- Example:
- $LE_1 = RE_0$
- $RE_1 = LE_0 \oplus F[RE_0, k_1]$

Feistel Cipher

- First, consider the encryption process, we see that

  $LE_{16} = RE_{15}$

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

- On the decryption process, we see that

  $LD_1 = RD_0 = LE_{16} = RE_{15}$

  $RD_1 = LD_0 \oplus F(RD_0, k_0)$

  $= RE_{16} \oplus F(RE_{15}, k_{16})$

  $= [LE \oplus F(RE_{15}, k_{16})] \oplus F(RE_{15}, k_{16})$

- In general terms, for i^{th} iteration of encryption algorithm

  $LE_i = RE_{i-1}$

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

  Rearrange the item

  $RE_{i-1} = LE_i$

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

Feistel Cipher Design Principles

- **block size**
  - increasing size improves security, but slows cipher
- **key size**
  - increasing size improves security, makes exhaustive key searching harder, but may slow cipher
- **number of rounds**
  - increasing number improves security, but slows cipher
- **subkey generation**
  - greater complexity can make analysis harder, but slows cipher
- **round function**
  - greater complexity can make analysis harder, but slows cipher
- **fast software en/decryption & ease of analysis**
  - are more recent concerns for practical use and testing

Feistel Cipher Decryption

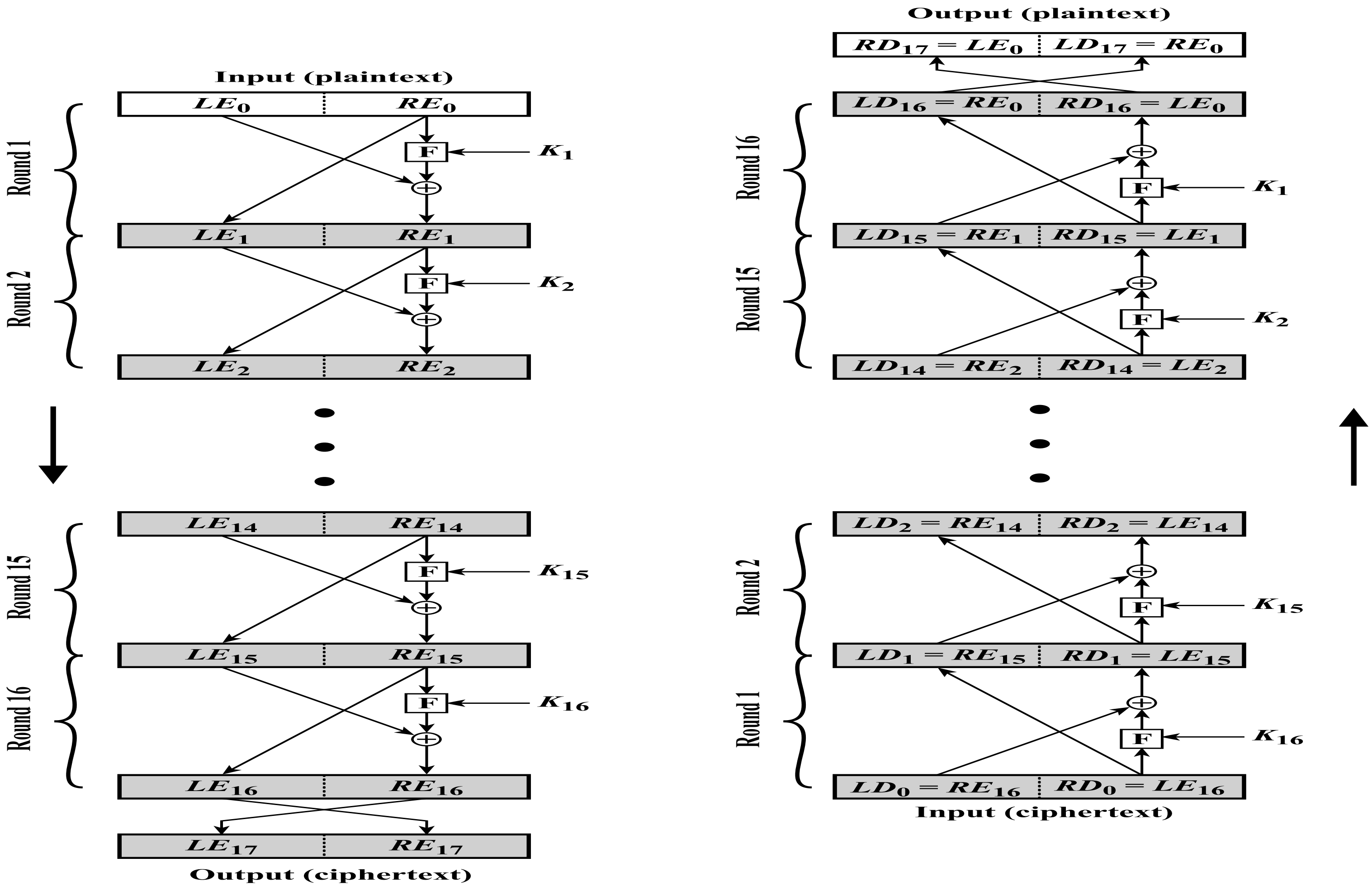


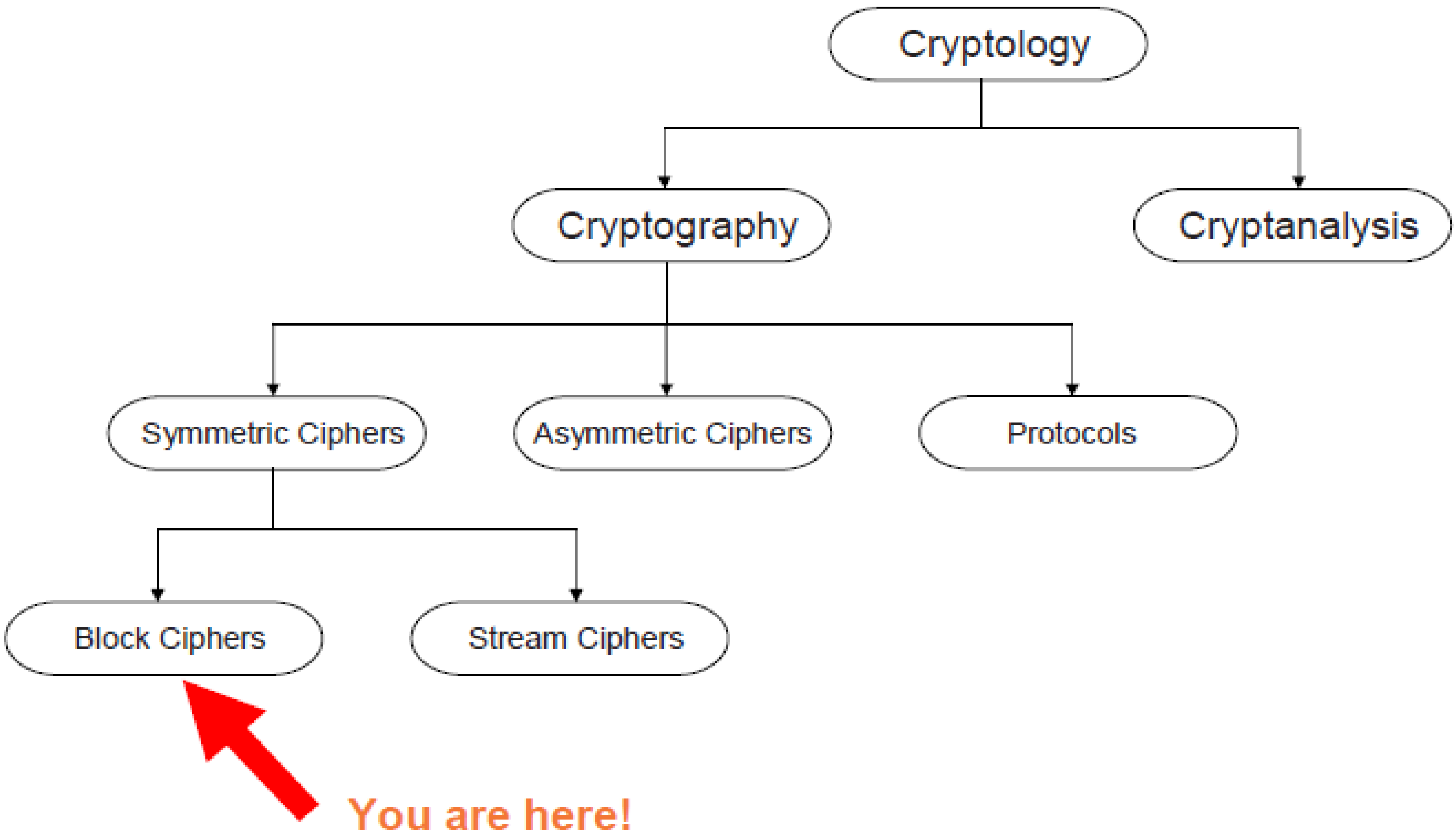
Figure 3.3 Feistel Encryption and Decryption (16 rounds)

Data Encryption Standard (DES)

Objectives

- To review a short history of DES
- To define the basic structure of DES
- To describe the details of building elements of DES
- To describe the round keys generation process
- To analyze DES

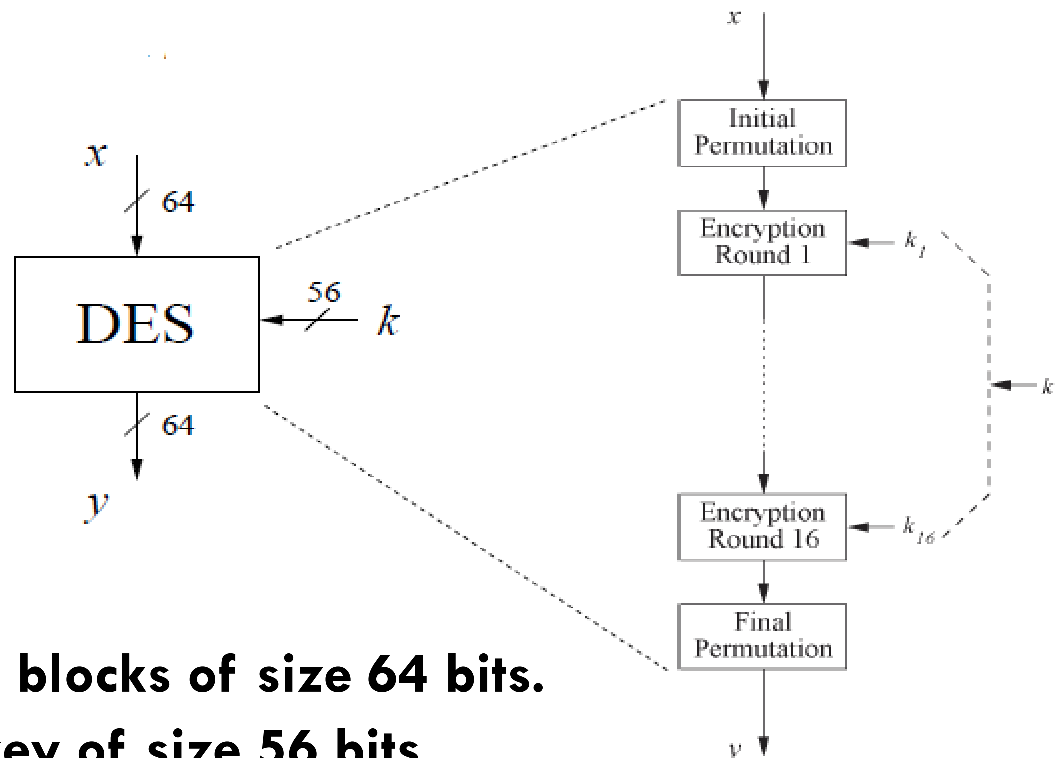
Classification of DES in the Field of Cryptology



DES Facts

- Data Encryption Standard (DES) encrypts **blocks of size 64 bit.**
- Developed by **IBM** based on the cipher *Lucifer* under influence of the *National Security Agency* (NSA), the design criteria for DES have not been published
- **Standardized 1977** by the **National Bureau of Standards** (NBS) today called *National Institute of Standards and Technology* (NIST)
- Most popular **block cipher** for most of the last 30 years.
- By far best studied symmetric algorithm.
- Nowadays considered insecure due to the small **key length of 56 bit.**
- **But: 3DES yields very secure cipher,** still widely used today.
- Replaced by the *Advanced Encryption Standard* (**AES**) in 2000

Overview of the DES Algorithm



- **Encrypts blocks of size 64 bits.**
- **Uses a key of size 56 bits.**
- Symmetric cipher: uses same key for encryption and decryption
- Uses 16 rounds which all perform the identical operation
- Different subkey in each round derived from main key

DES STRUCTURE

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

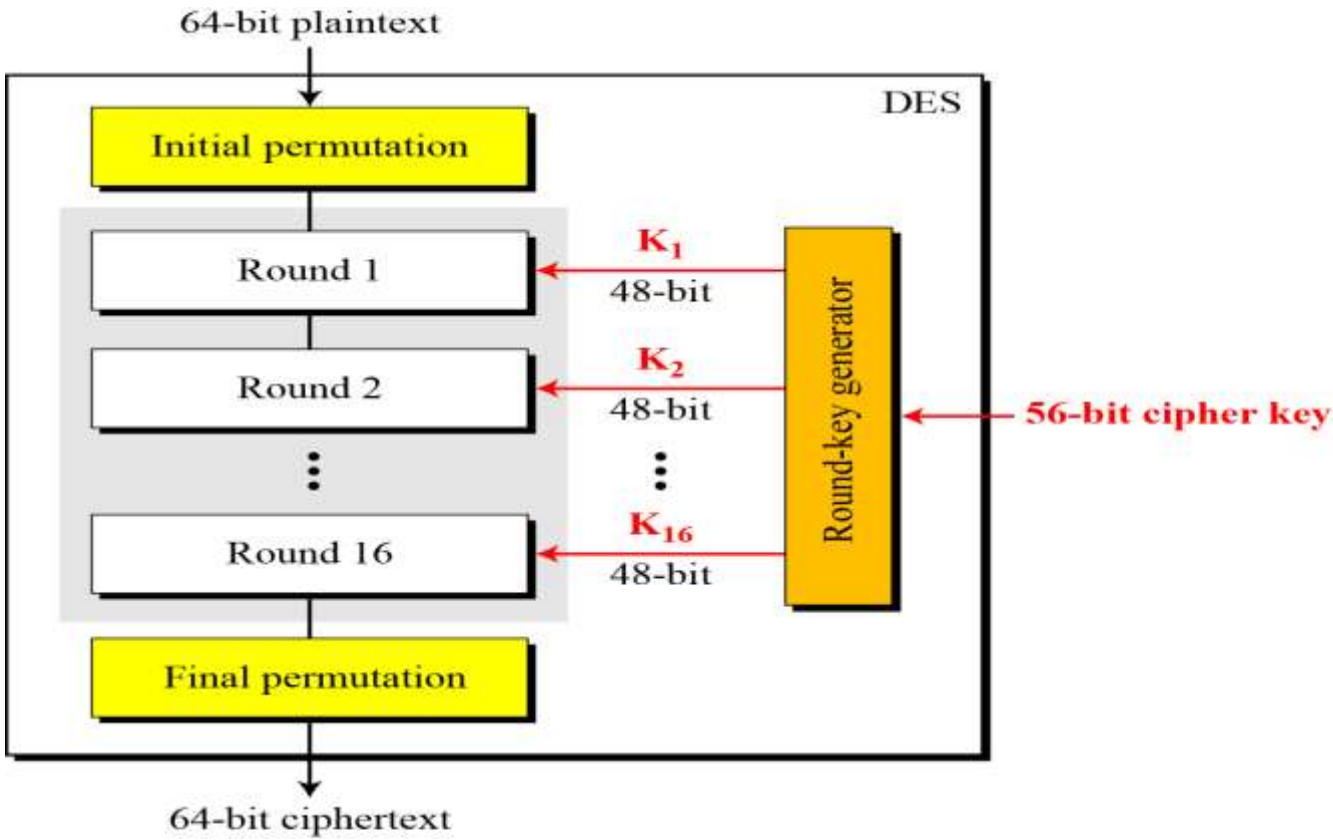
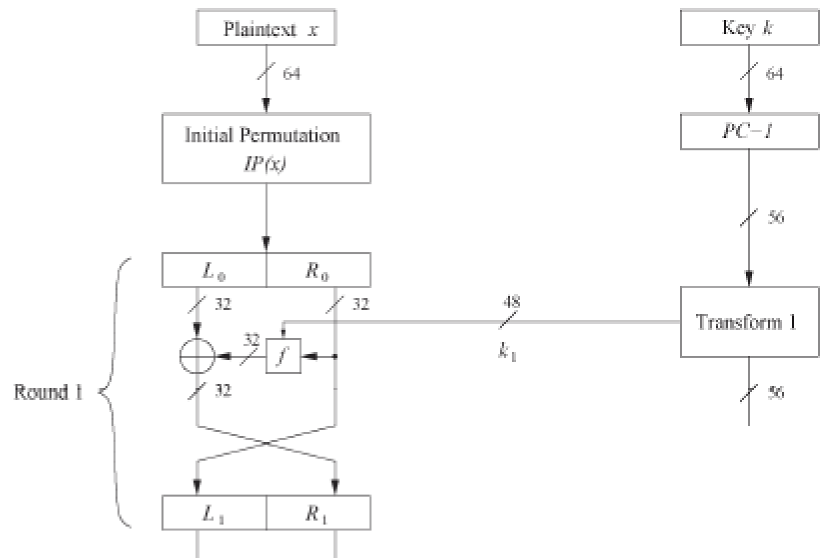


Figure 7.2 General structure of DES

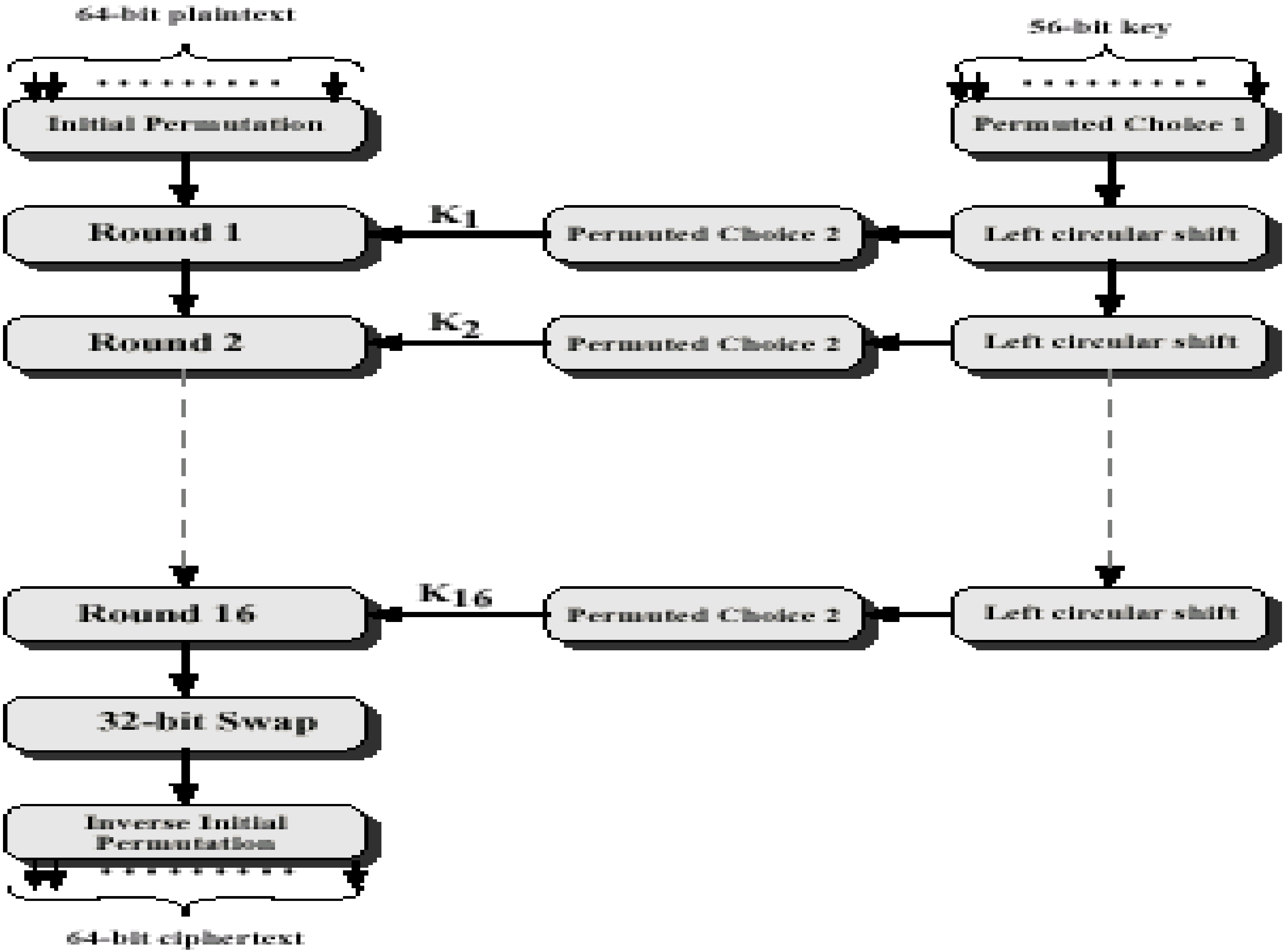
DES STRUCTURE

- DES structure is a *Feistel network*
- Advantage: encryption and decryption differ only in keyschedule



- Bitwise initial permutation, then 16 rounds
  1. Plaintext is split into 32-bit halves L_i and R_i
  2. R_i is fed into the function *f*, the output of which is then XORed with L_i
  3. Left and right half are swapped
- Rounds can be expressed as:

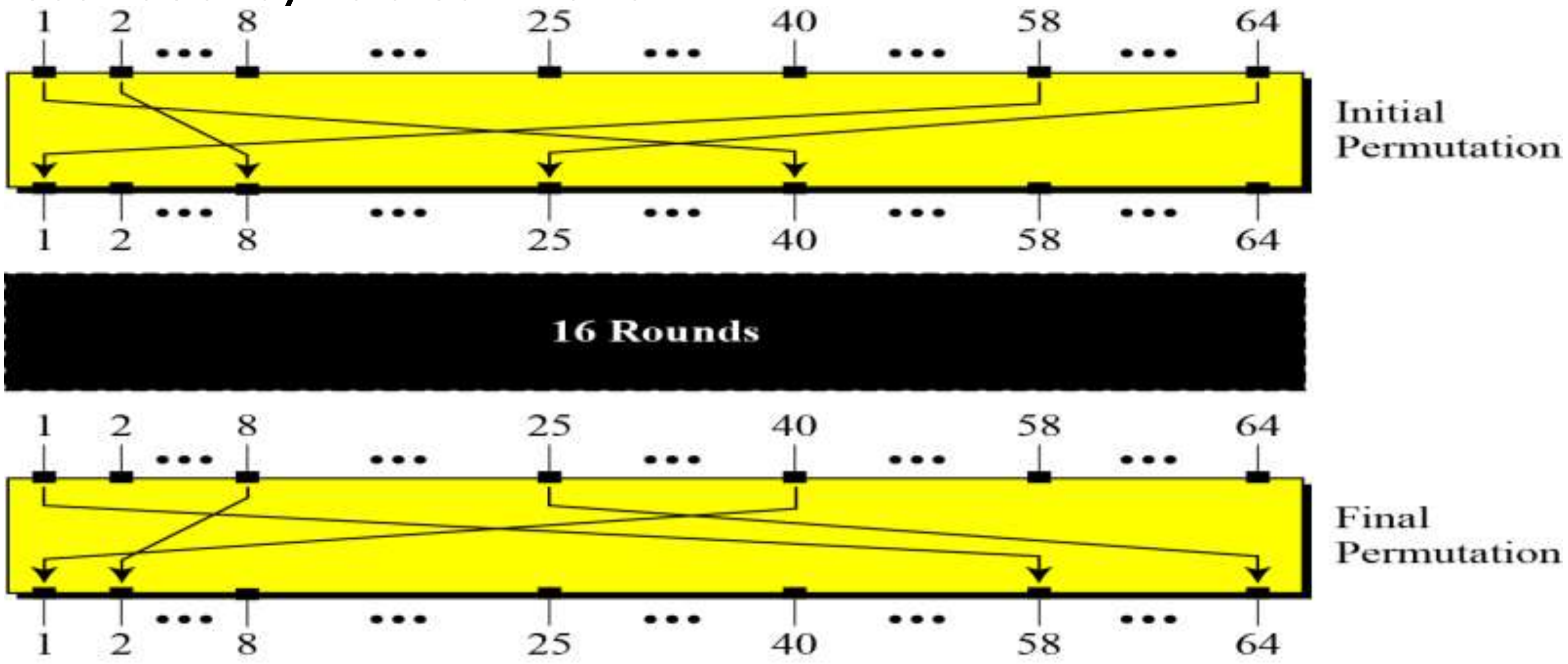
$$L_i = R_{i-1},$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$$


64-bit plaintext
56-bit key
Initial Permutation
Permuted Choice 1
Round 1
K1
Permuted Choice 2
Left circular shift
Round 2
K2
Permuted Choice 2
Left circular shift
Round 16
K16
Permuted Choice 2
Left circular shift
32-bit Swap
Inverse Initial Permutation
64-bit ciphertext

Internal Structure of DES

Initial and Final Permutation

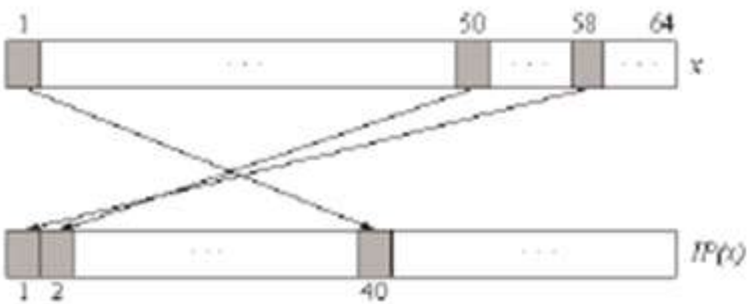
- Bitwise Permutations.
- Inverse operations.
- Described by tables *IP* and IP^{-1}.



Initial and Final Permutation

Initial Permutation

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



Final Permutation

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

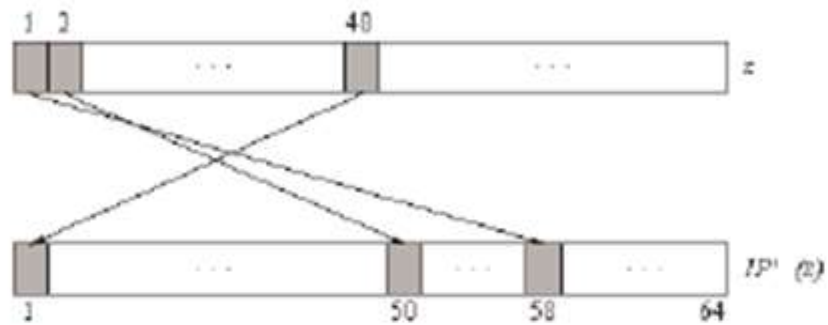


Table 7.1 Initial and final permutation tables

1_{1}	0_{2}	0_{3}	1_{4}	1_{5}	0_{6}	1_{7}	0_{8}
0_{9}	1_{10}	1_{11}	0_{12}	1_{13}	0_{14}	0_{15}	1_{16}
1_{17}	1_{18}	0_{19}	1_{20}	1_{21}	1_{22}	1_{23}	1_{24}
1_{25}	1_{26}	0_{27}	0_{28}	1_{29}	0_{30}	1_{31}	1_{32}
0_{33}	0_{34}	1_{35}	0_{36}	0_{37}	1_{38}	0_{39}	0_{40}
0_{41}	1_{42}	1_{43}	0_{44}	0_{45}	1_{46}	0_{47}	1_{48}
0_{49}	1_{50}	1_{51}	1_{52}	0_{53}	0_{54}	1_{55}	0_{56}
0_{57}	0_{58}	1_{59}	1_{60}	0_{61}	0_{62}	1_{63}	1_{64}

0	1	1	0	1	1	1	0
1	1	0	0	0	1	0	1
0	0	1	1	0	1	0	0
1	0	1	0	1	1	1	0
0	0	0	0	1	1	0	1
1	1	1	1	0	0	1	0
0	0	0	0	1	1	1	1
1	1	0	0	1	1	0	1

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

1_{1}	0_{2}	0_{3}	1_{4}	1_{5}	0_{6}	1_{7}	0_{8}
0_{9}	1_{10}	1_{11}	0_{12}	1_{13}	0_{14}	0_{15}	1_{16}
1_{17}	1_{18}	0_{19}	1_{20}	1_{21}	1_{22}	1_{23}	1_{24}
1_{25}	1_{26}	0_{27}	0_{28}	1_{29}	0_{30}	1_{31}	1_{32}
0_{33}	0_{34}	1_{35}	0_{36}	0_{37}	1_{38}	0_{39}	0_{40}
0_{41}	1_{42}	1_{43}	0_{44}	0_{45}	1_{46}	0_{47}	1_{48}
0_{49}	1_{50}	1_{51}	1_{52}	0_{53}	0_{54}	1_{55}	0_{56}
0_{57}	0_{58}	1_{59}	1_{60}	0_{61}	0_{62}	1_{63}	1_{64}

0_{1}	1_{2}	1_{3}	0_{4}	1_{5}	1_{6}	1_{7}	0_{8}
1_{9}	1_{10}	0_{11}	0_{12}	0_{13}	1_{14}	0_{15}	1_{16}
0_{17}	0_{18}	1_{19}	1_{20}	0_{21}	1_{22}	0_{23}	0_{24}
1_{25}	0_{26}	1_{27}	0_{28}	1_{29}	1_{30}	1_{31}	0_{32}
0_{33}	0_{34}	0_{35}	0_{36}	1_{37}	1_{38}	0_{39}	1_{40}
1_{41}	1_{42}	1_{43}	1_{44}	0_{45}	0_{46}	1_{47}	0_{48}
0_{49}	0_{50}	0_{51}	0_{52}	1_{53}	1_{54}	1_{55}	1_{56}
1_{57}	1_{58}	0_{59}	0_{60}	1_{61}	1_{62}	0_{63}	1_{64}

1	0	0	1	1	0	1	0
0	1	1	0	1	0	0	1
1	1	0	1	1	1	1	1
1	1	0	0	1	0	1	1
0	0	1	0	0	1	0	0
0	1	1	0	0	1	0	1
0	1	1	1	0	0	1	0
0	0	1	1	0	0	1	1

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

Initial and Final Permutation

Example 7.1

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

0x0000 0080 0000 0002

Solution

Only bit 25 and bit 64 are 1s; the other bits are 0s. In the final permutation, bit 25 becomes bit 64 and bit 63 becomes bit 15. The result is

0x0002 0000 0000 0001

The f-Function

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

- The overall processing at each iteration:

  $L_i = R_{i-1} \otimes F(R_{i-1}, K_i)$

  $R_i = L_{i-1}$

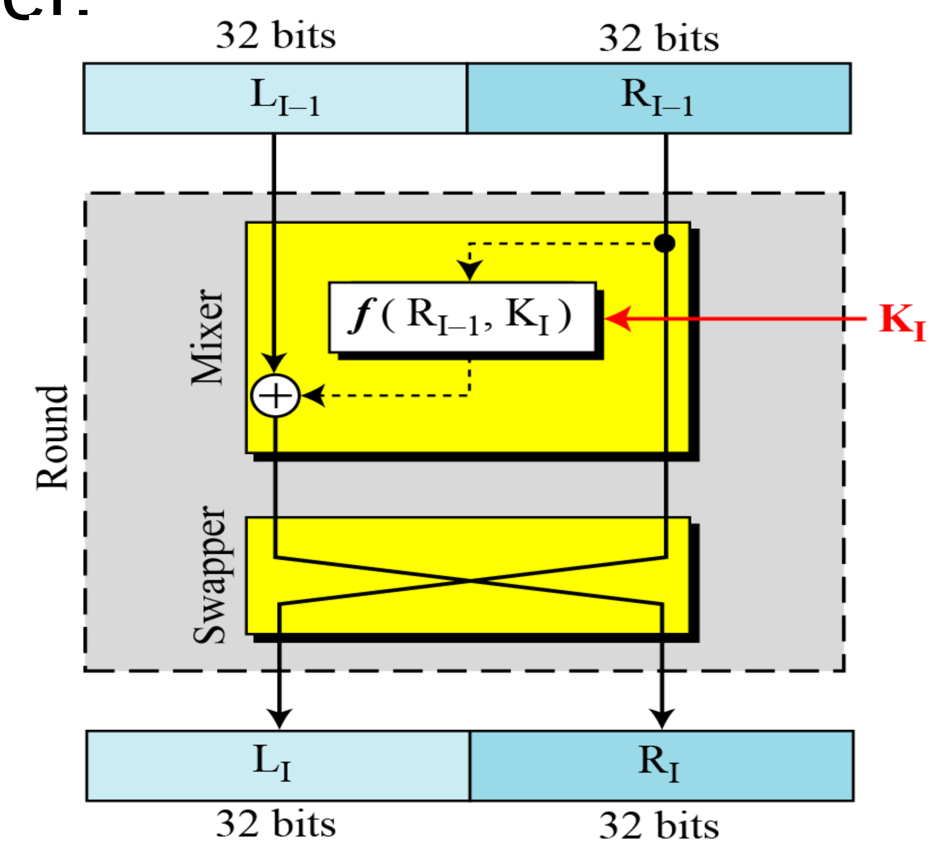
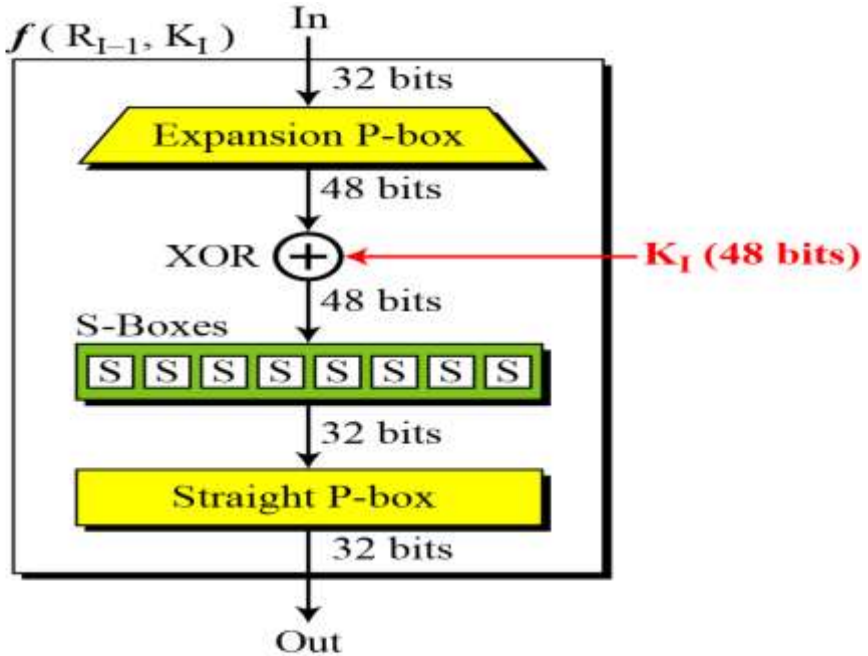


Figure 7.4 A round in DES (encryption site)

The f-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.

Figure 7.5
DES function

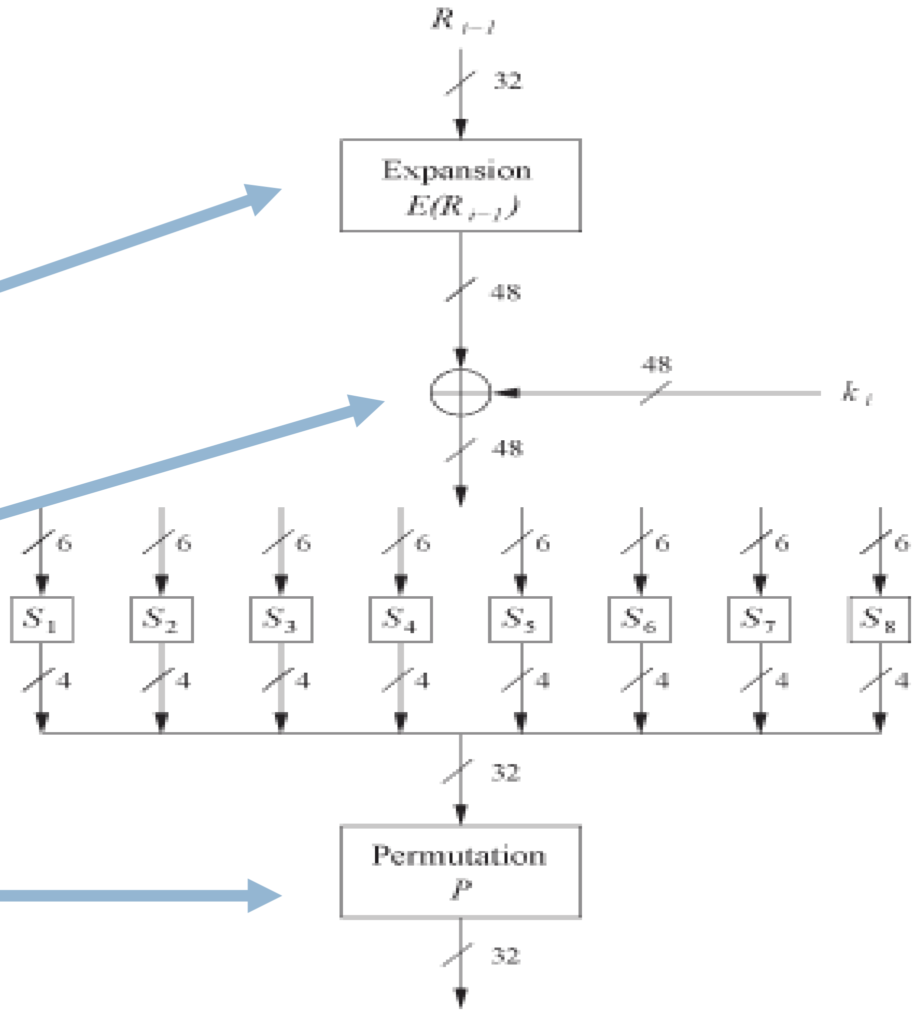


The f-Function

- **main operation of DES**
- *f*-Function inputs:
  Ri-1 and round key *ki*

- **4 Steps:**

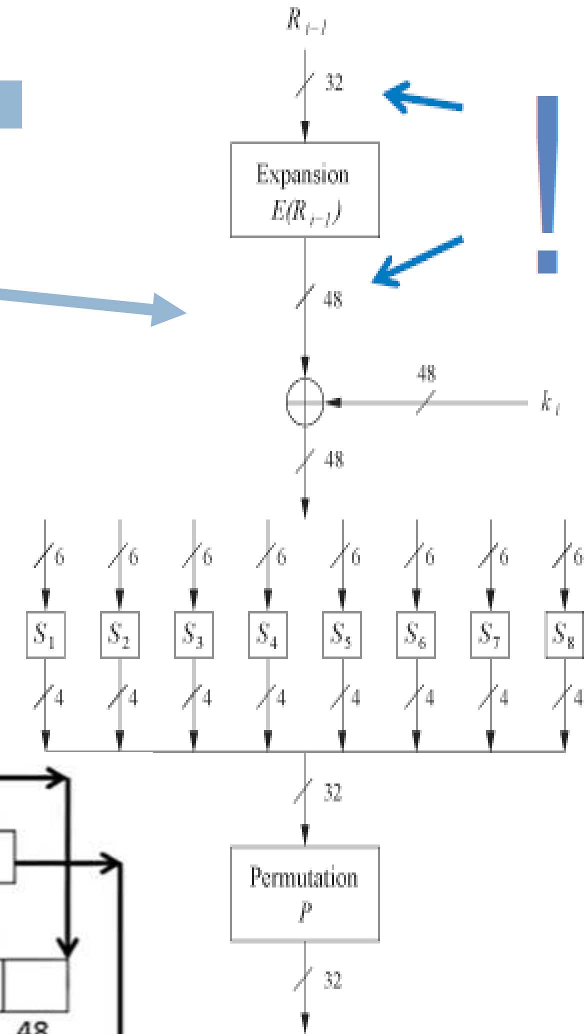
1. Expansion E
2. XOR with round key
3. S-box substitution
4. Permutation

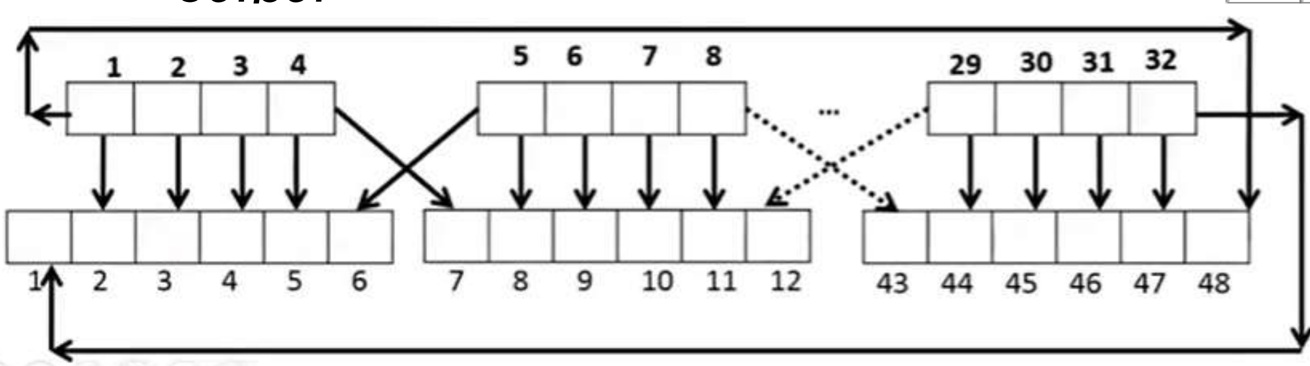


The Expansion Function E

1. Expansion E

- *Since R_{l-1} is a 32-bit input and K_l is a 48-bit key, we first need to expand R_{l-1} to 48 bits.*
- *Expansion permutation steps:*
  - *Each 4 bit block is expanded to 6-bit and produce 48-bit output*

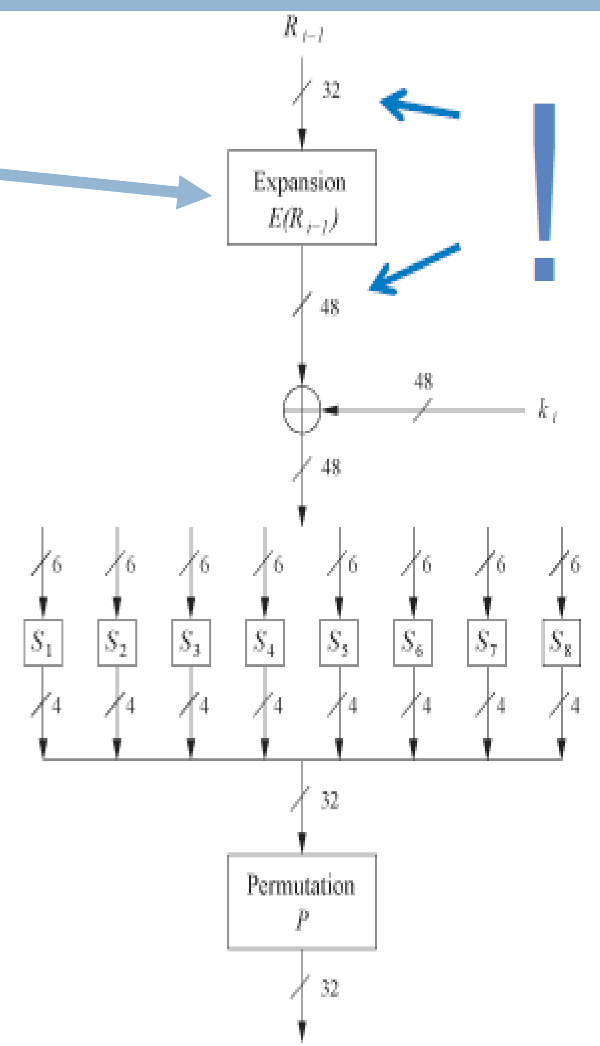




The Expansion Function E

1. Expansion E

- *Since R_{I-1} is a 32-bit input and K_I is a 48-bit key, we first need to expand R_{I-1} to 48 bits.*



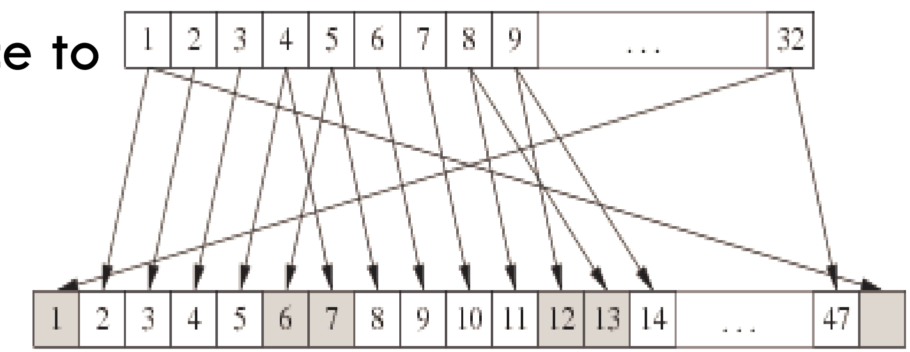
Added

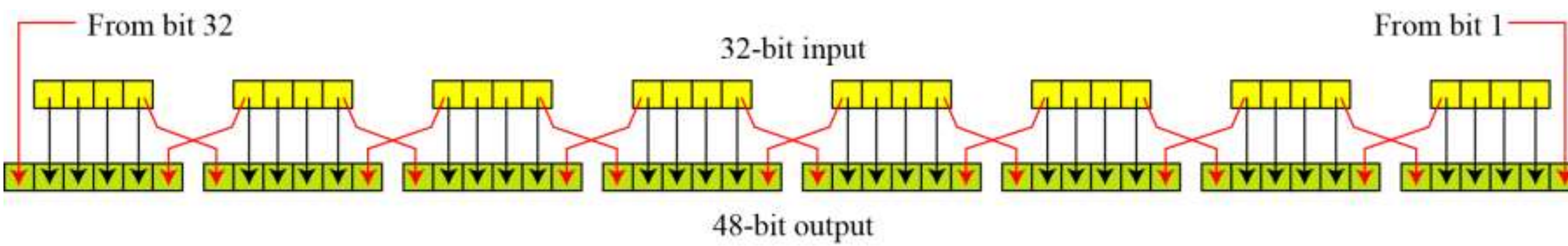
	E bit-selection table				
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Table 7.6 Expansion P-box table

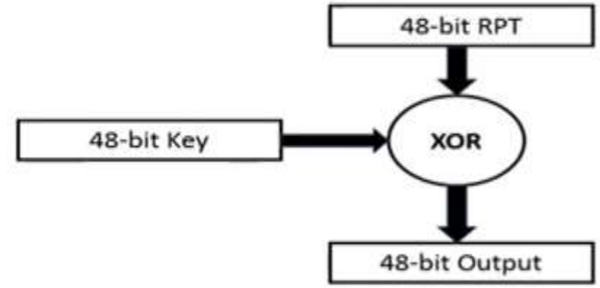
The Expansion Function E

- Although the relationship between the input and output can be defined mathematically, DES uses Table 7.6 to define this P-box.
- The 32-bits of R_i are permuted and 16 of them are repeated twice to obtain a 48 bit string.





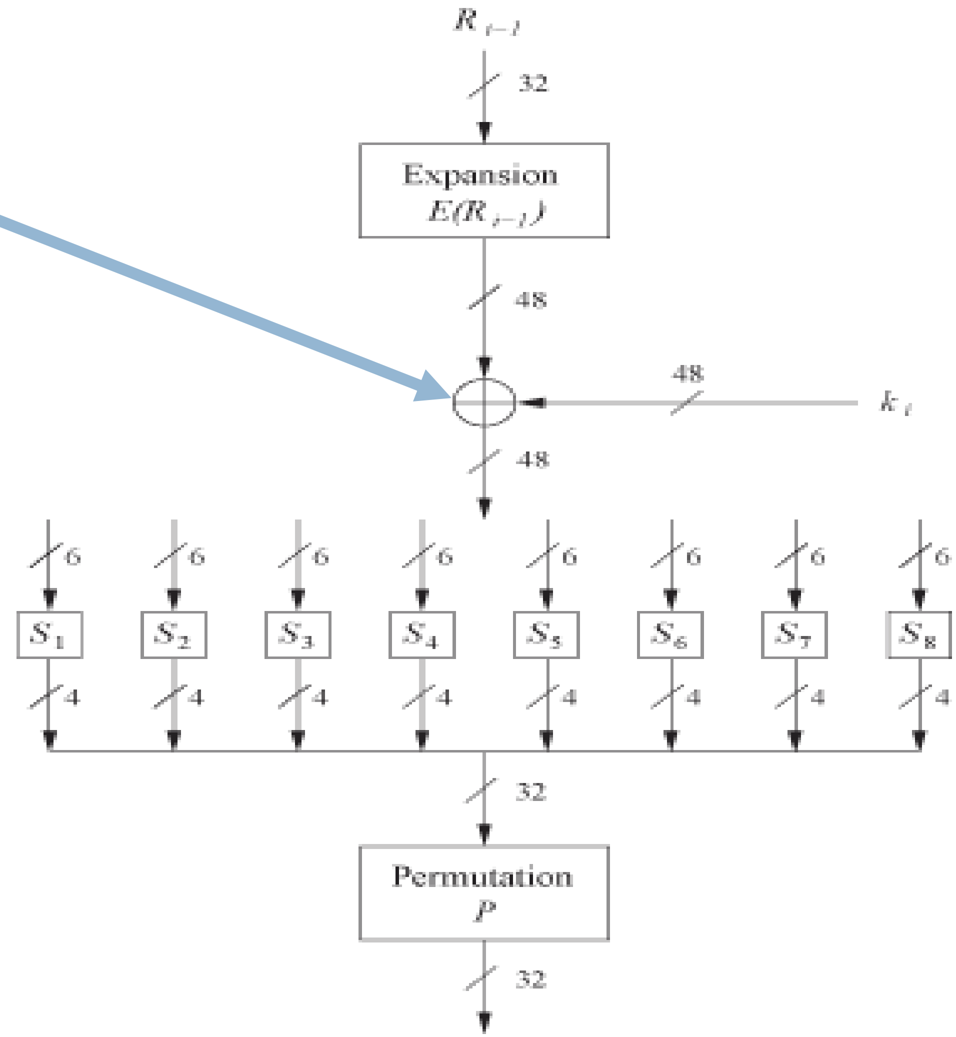
Add Round Key



2. XOR Round Key

- 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. Also note that the round key is used only in this operation.

- Round keys are derived from the main key in the DES keyschedule (in a few slides



0	0	1	0	1	1
1	0	0	0	0	1
0	0	1	1	0	1
1	0	1	0	1	1
0	0	0	0	1	1
1	1	1	1	0	0
0	0	0	0	1	1
1	1	0	0	1	1

Expansion E
8 x 6 = 48

$\oplus$

1	1	1	1	0	1
0	1	0	1	1	0
1	0	0	0	1	0
0	1	1	0	0	0
1	0	1	1	0	0
0	0	1	0	1	1
1	1	0	1	0	0
0	0	1	1	0	0

Round Key
8 x 6 = 48

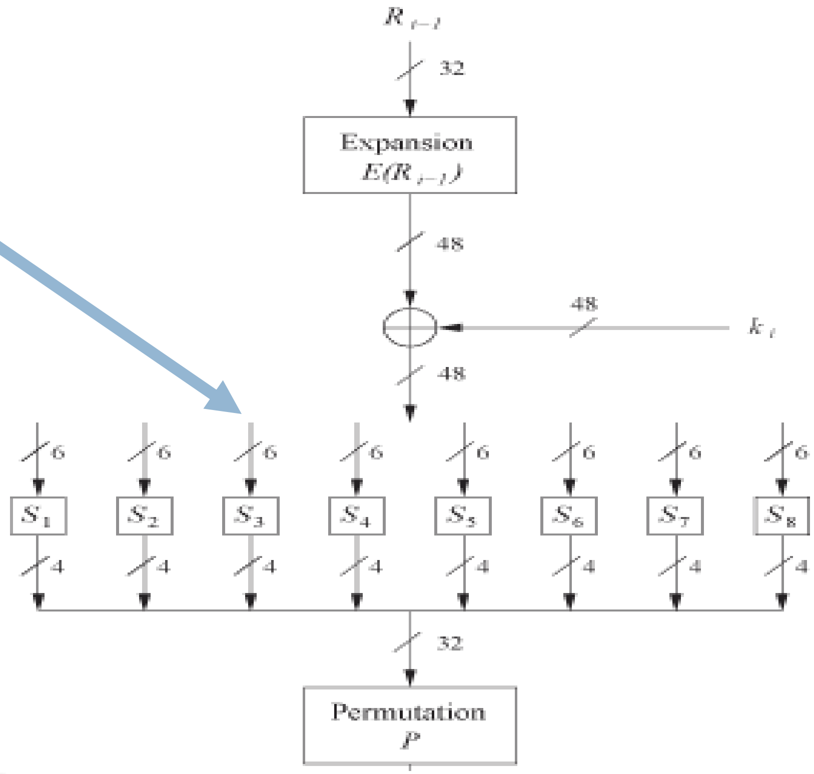
=

1	1	0	1	1	0
1	1	0	1	1	1
1	0	1	1	1	1
1	0	0	0	1	1
1	0	1	1	1	1
1	1	0	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

The DES S-Boxes

3. S-Box substitution

- The S-boxes do the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output. See Figure 7.7.
- Non-linear and resistant to differential cryptanalysis.
- Crucial element for DES security!



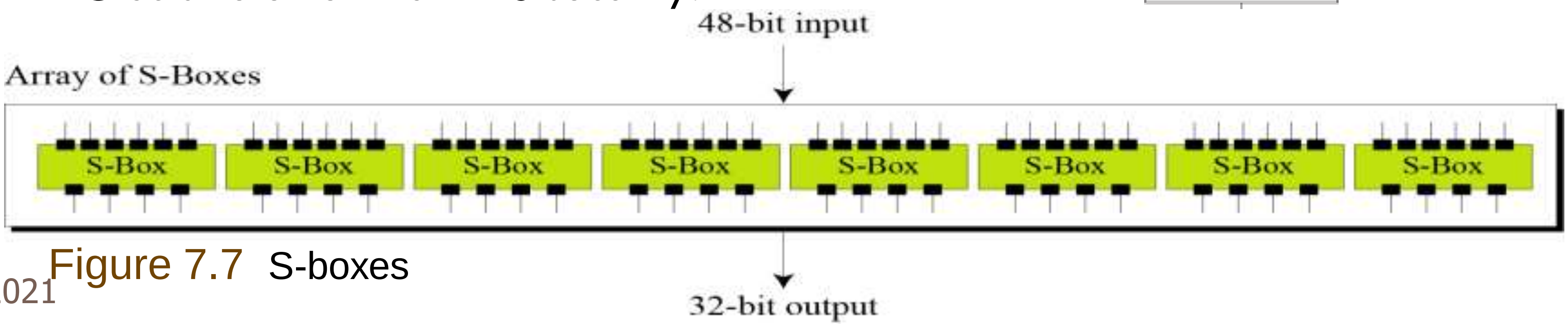


Figure 7.7 S-boxes

1	1	0	1	1	0	S_1
1	1	0	1	1	1	S_2
1	0	1	1	1	1	S_3
1	0	0	0	1	1	S_4
1	0	1	1	1	1	S_5
1	1	0	1	1	1	S_6
1	1	0	1	1	1	S_7
1	1	1	1	1	1	S_8

$8 \times 6 = 48$

0	1	1	1
1	1	0	0
0	0	0	0
1	1	1	1
1	0	0	0
0	1	1	1
1	1	1	1
1	0	0	1

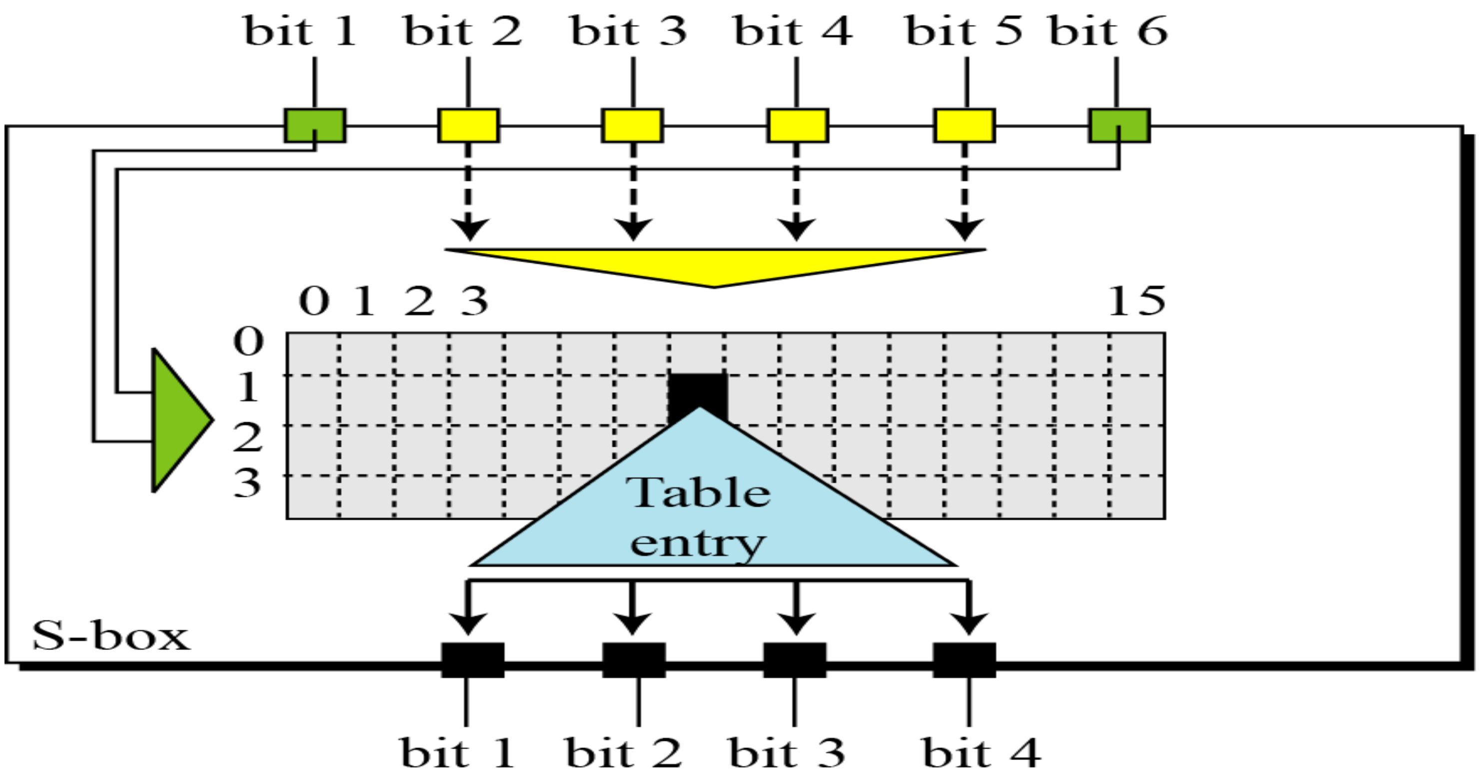
$8 \times 4 = 32$

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	01	10	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_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

The DES S-Boxes

- Figure 7.8 S-box rule



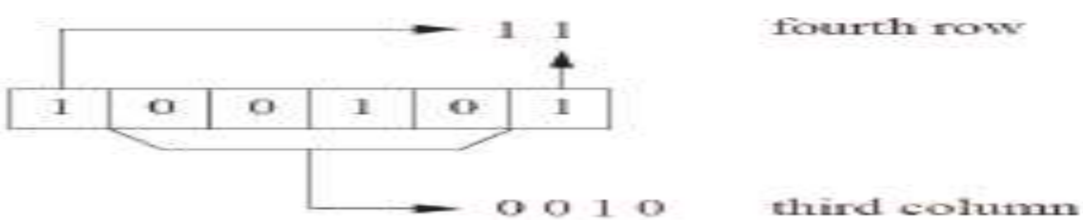
The DES S-Boxes

Example 6.3

- The input to S-box 1 is 100101. What is the output?

Solution

If we write the first and the sixth bits together, we get 11 in binary, which is 3 in decimal. The remaining bits are 0010 in binary, which is 2 in decimal. We look for the value in row 3, column 2, in Table (S-box 1). The result is 08 in decimal, which in binary is 1000. So the input 100101 yields the output 1000.



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	01	10	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

Example 6.4

- The input to S-box 8 is 000000. What is the output?
- Solution
- If we write the first and the sixth bits together, we get 00 in binary, which is 0 in decimal. The remaining bits are 0000 in binary, which is 0 in decimal. We look for the value in row 0, column 0, in Table 6.10 (S-box 8). The result is 13 in decimal, which is 1101 in binary. So the input 000000 yields the output 1101.

S_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	00	14	09	02
2	07	11	04	01	09	12	14	02	00	06	10	13	15	03	05	08
3	02	01	14	07	04	10	08	13	15	12	09	00	03	05	06	11

DES: S Boxes (1-4)

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	01	10	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_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

S_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_4	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	07	13	14	03	00	06	09	10	01	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

DES: S Boxes (5-8)

S_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_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	06	00	08	13

S_7	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	04	11	02	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

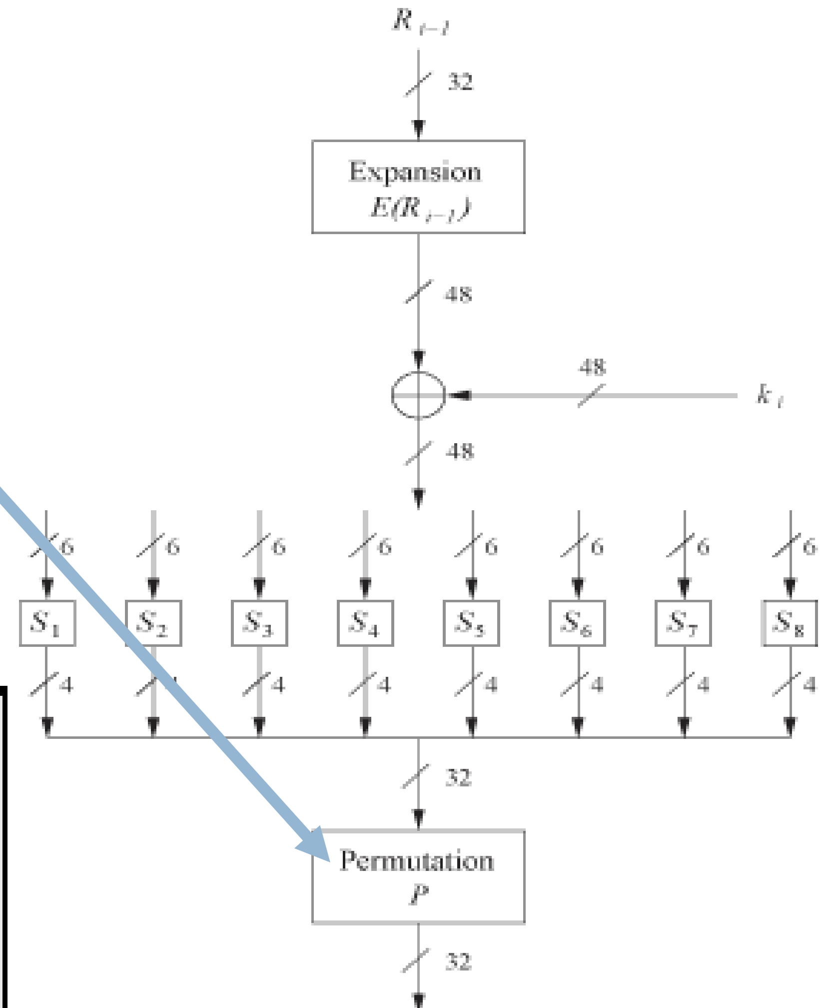
S_8	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	13	02	08	04	06	15	11	01	10	09	03	14	05	00	12	07
1	01	15	13	08	10	03	07	04	12	05	06	11	00	14	09	02
2	07	11	04	01	09	12	14	02	00	06	10	13	15	03	05	08
3	02	01	14	07	04	10	08	13	15	12	09	00	03	05	06	11

The Permutation P

4. Permutation P

- Bitwise permutation.
- Introduces diffusion.
- Output bits of one S-Box effect several S-Boxes in next round
- Diffusion by E, S-Boxes and P guarantees that after Round 5 every bit is a function of each key bit and each plaintext bit.

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

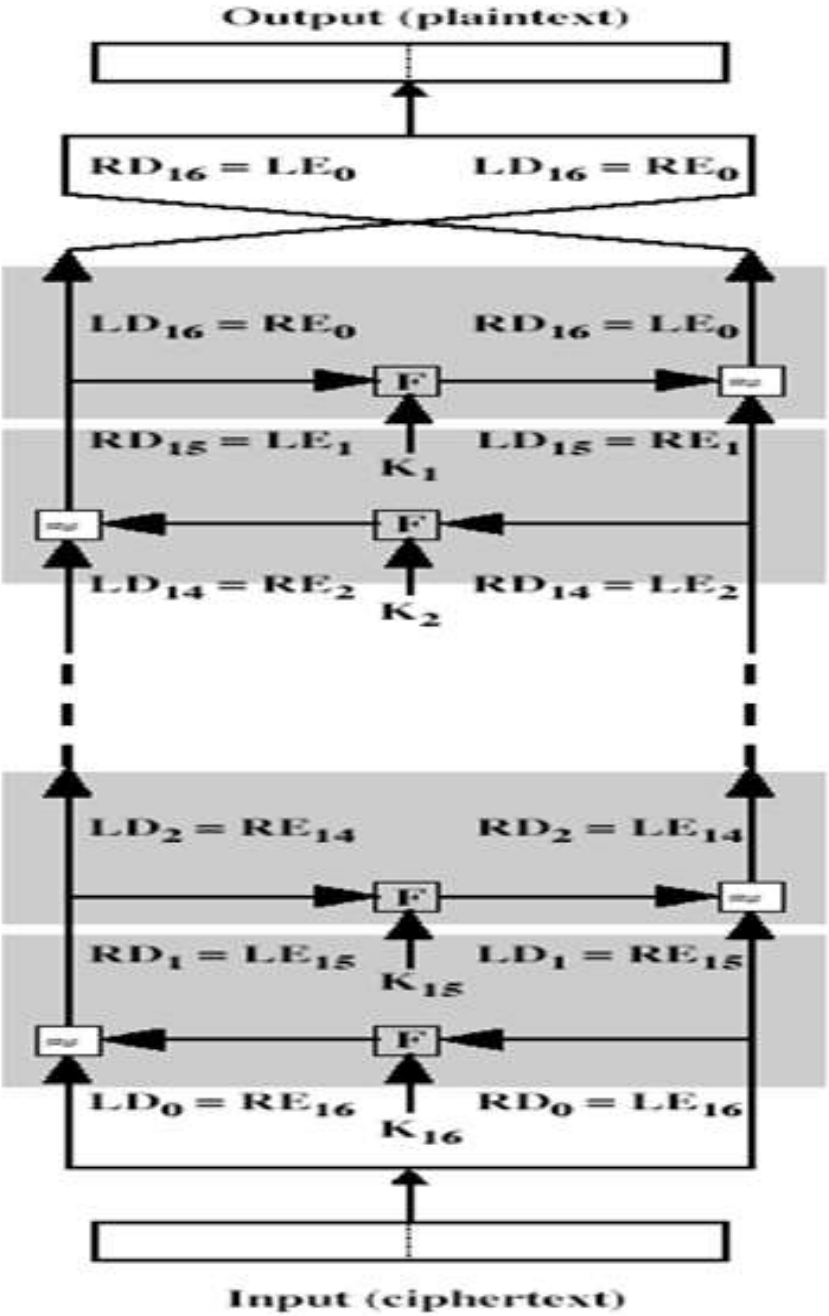


Cipher and Reverse Cipher

- When 16 rounds are finished, L and R are swapped and merged, then becomes a 64-bit “pre-output”
- Apply permutation IP^{-1} on the 64-bits to become the final cipher output.

DES - Decryption

- The same algorithm as encryption. Almost all operations are the same as those of encryption.
- Only one is different: use the subkeys in descending order (reversed order). (k_{16} for round 1, key_{15} for round 2, etc....)

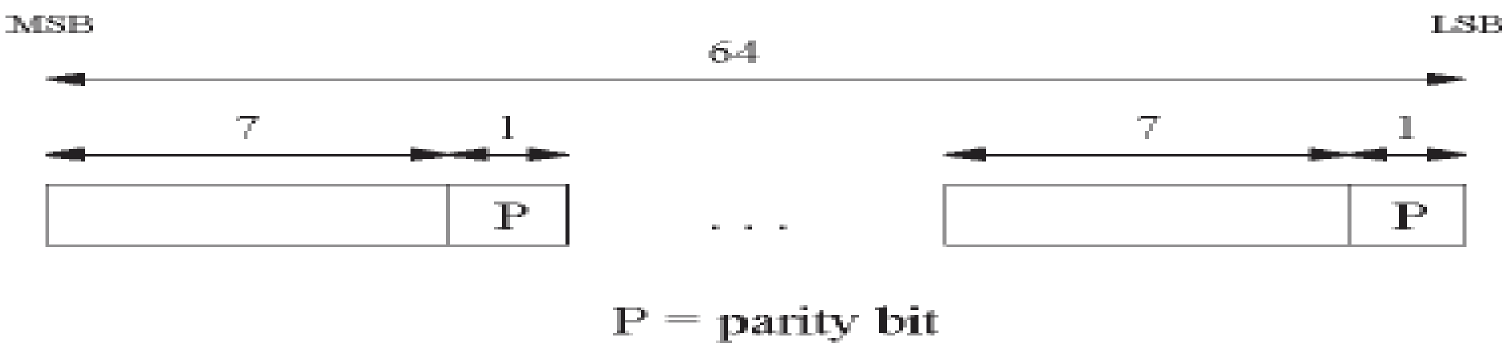


- The same algorithm as encryption.
- Reversed the order of key (Key_{16}, Key_{15}, … Key_{1}).
- For example:
  - IP undoes IP^{-1} step of encryption.
  - 1^{st} round with SK_{16} undoes 16^{th} encrypt round.

DES Key Schedule

DES: Key generation for each round

- Derives 16 round keys (or *subkeys*) k_i of 48 bits each from the original 56 bit key.
- The input key size of the DES is 64 bit: **56 bit key** and 8 bit parity:



- Parity bits are removed in a first permuted choice PC-1: (note that the bits 8, 16, 24, 32, 40, 48, 56 and 64 are not used at all)

Key Generation (Discard each 8th bit)

Discard these

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

$8 \times 7 = 56\ bits$

Shifting

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

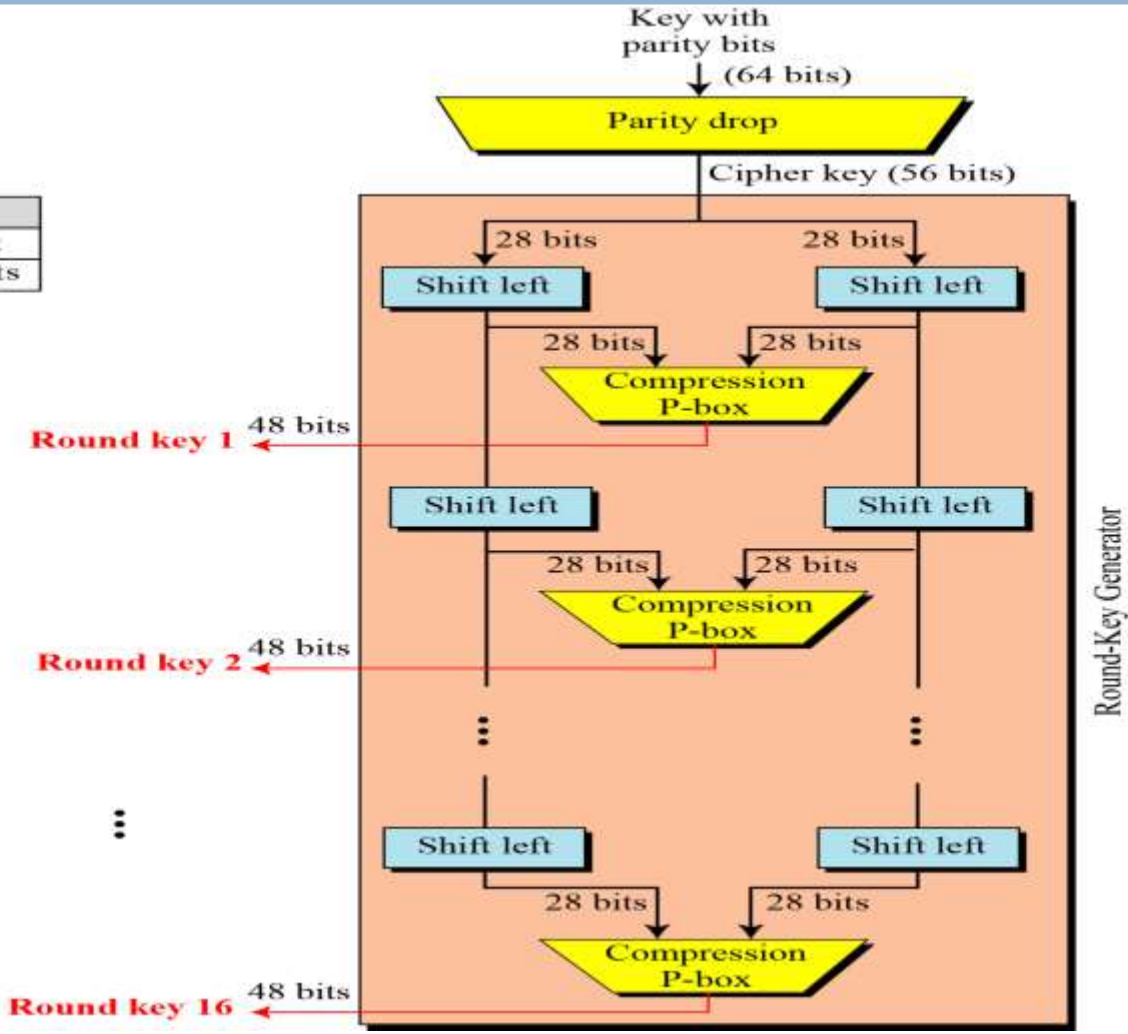


Figure 7.10
Key generation

Key Permuted Choice 1

- PC-1: Permutation of 56 bits

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

Table 6.12 Parity-bit drop table

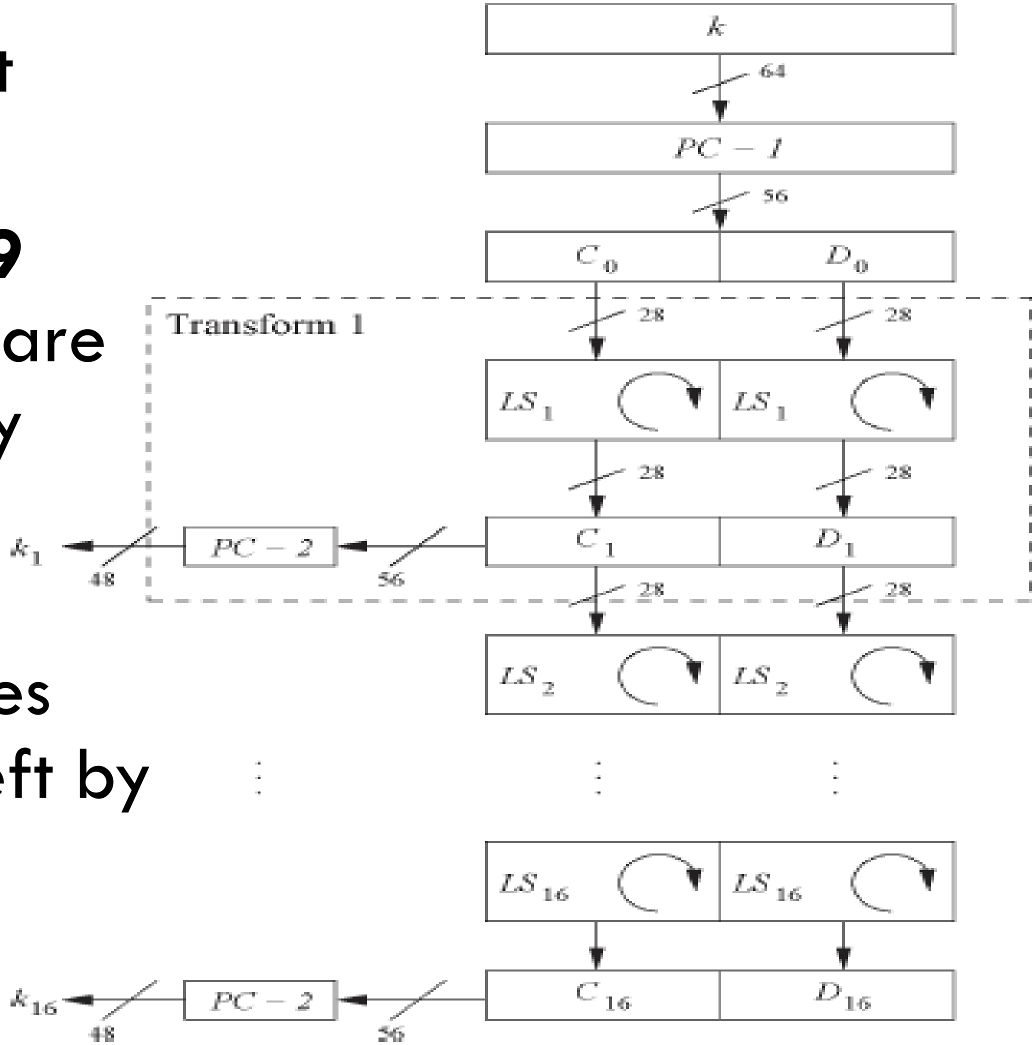
- Schedule of left shift

Round number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits rotated	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Table 7.13 Number of bits shifts

Key Permuted Choice 1

- Split key into 28-bit halves C_0 and D_0.
- In **rounds i = 1, 2, 9 ,16**, the two halves are each rotated left by **one bit.**
- In **all other rounds** where the two halves are each rotated left by **two bits.**



Key Permuted Choice 2

- *In each round i permuted choice* ***PC-2*** selects a permuted subset of 48 bits of *Ci* and *Di* as round key k_i, i.e. **each k_i is a permutation of k**!

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

The following bits are discarded

9	18	22	25	35	38	43	54

Key Shifting

- **Note:** The total number of rotations:

$4 \times 1 + 12 \times 2 = 28 \Rightarrow D_0 = D_{16}$ and $C_0 = C_{16}$!

Security of DES

- Critics have used a strong magnifier to analyze DES. Tests have been done to measure the strength of some desired properties in a block cipher.

Properties

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

Example 6.7

- To check the avalanche effect in DES, let us encrypt two plaintext blocks (with the same key) that differ only in one bit and observe the differences in the number of bits in each round.

Plaintext: 0000000000000000 Key: 22234512987ABB23
Ciphertext: 4789FD476E82A5F1

Plaintext: 000000000000000<u>**1**</u> Key: 22234512987ABB23
Ciphertext: 0A4ED5C15A63FEA3

Properties

- Avalanche effect in DES
  - If a small change in either the plaintext or the key, the ciphertext should change markedly.
- DES exhibits a strong avalanche effect.

(a) Change in Plaintext		(b) Change in Key	
Round	Number of bits that differ	Round	Number of bits that differ
0	1	0	0
1	6	1	2
2	21	2	14
3	35	3	28
4	39	4	32
5	34	5	30
6	32	6	32
7	31	7	35
8	29	8	34
9	42	9	40
10	44	10	38
11	32	11	31
12	30	12	33
13	30	13	28
14	26	14	26
15	29	15	34
16	34	16	35

Properties

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

Design Criteria

- S-Boxe
  - The design provides confusion and diffusion of bits from each round to the next.
- P-Boxes
  - They provide diffusion of bits.
- Number of Rounds
  - DES uses sixteen rounds of Feistel ciphers. the ciphertext is thoroughly a random function of plaintext and ciphertext.

DES Weaknesses

- During the last few years critics have found some weaknesses in DES.
- **After proposal of DES two major criticisms arose:**

  1. Key space is too small (2^{56} keys)

  2. S-box design criteria have been kept secret: Are there any hidden analytical attacks (*backdoors*), only known to the NSA?

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

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

DES Weaknesses

Example 7.8

- Let us try the first weak key in the following Table 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*