Chapter two part two
Modern type of cryptography.

- **Block Ciphers**
  - Block Cipher Principles
  - Simplified DES
  - The Data Encryption Standard (DES)
  - Triple Data Encryption Standard (3-DES)
  - Advanced Encryption Standard (AES)

Block Ciphers and Stream Ciphers

A **stream cipher** is one that encrypts a digital data stream one bit or one byte at a time. Examples of classical stream ciphers are the auto keyed Vigenère cipher and the Vernam cipher.

A **block cipher** is an encryption/decryption scheme in which a block of plaintext is treated as a whole and used to produce a cipher text block of equal length.

Typically, a block size of 64 or 128 bits is used.

Many block ciphers have **a *Feistel structure.*** Such a structure consists of a number of identical rounds of processing.

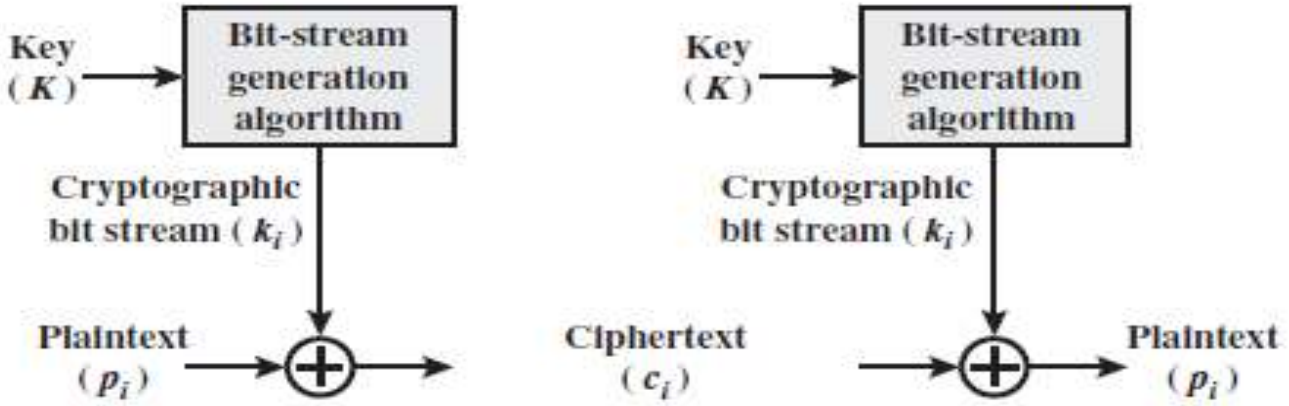
In each round, a **substitution** is performed on one half of the data being processed, followed by a permutation that interchanges the two halves. The original key is expanded so that a different key is used for each round.

Substitution: Each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements.

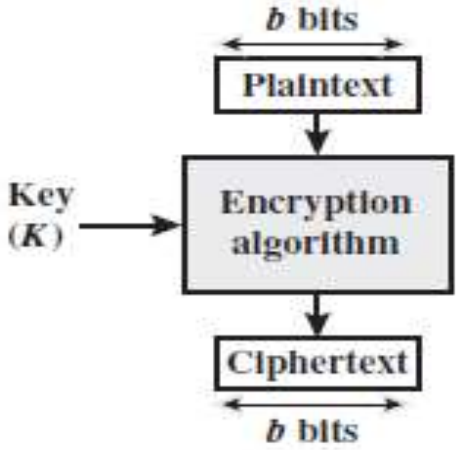
- **Permutation:** A sequence of plaintext elements is replaced by a permutation of that sequence. That is, no elements are added or deleted or replaced in the sequence, rather the order in which the elements appear in the sequence is changed.

- **Problem of block cipher**

- **Small block size:** are vulnerable to a statistical analysis of the plaintext.
- For a **n -bit** ideal block cipher, the length of the key defined in this fashion is bits **$n*2^n$**. For a 64-bit block, which is a desirable length to statistical attacks, the required key length is **$64*2^{64}$** bits.



(a) Stream cipher using algorithmic bit-stream generator



(b) Block cipher

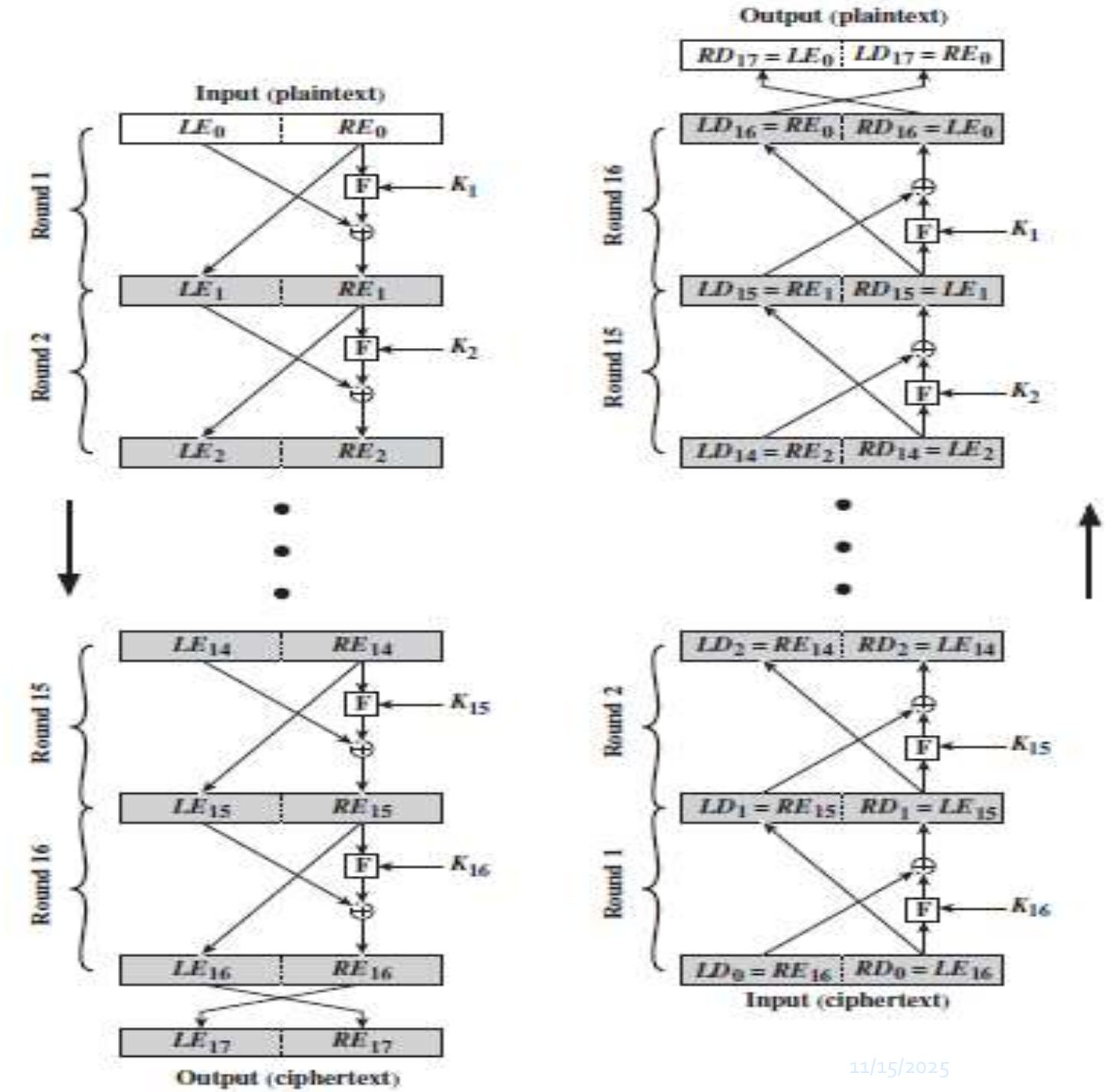
Stream Cipher and Block Cipher

Feistel Cipher Structure

- A block cipher operates on a plaintext block of *n bits to produce a ciphertext* block of *n bits. There are possible different plaintext blocks and, for* the encryption to be reversible (i.e., for decryption to be possible)
- In particular, Feistel proposed the use of a cipher that alternates **substitutions and permutations**

- In fact, Feistel's is a practical application of a proposal by Claude Shannon to develop a product cipher that **alternates *confusion and diffusion***
- **Diffusion:** Structure of the plaintext is dissipated into long-range statistics of the cipher text. This is achieved by having each plaintext digit affect the value of many cipher text digits; this is equivalent to having each cipher text digit be affected by many plaintext digits. An example of diffusion is to encrypt a message **m =m1,m2,m3** of characters with an averaging operation:

- **Confusion :**is seeks to make the relationship between the statistics of the cipher text and the value of the encryption key as complex as possible, again to thwart attempts to discover the key. Thus, even if the attacker can get some handle on the statistics of the cipher text, the way in which the key was used to produce that **cipher text is so complex** as to make it **difficult to deduce the key**. This is achieved by the use of a complex substitution algorithm.

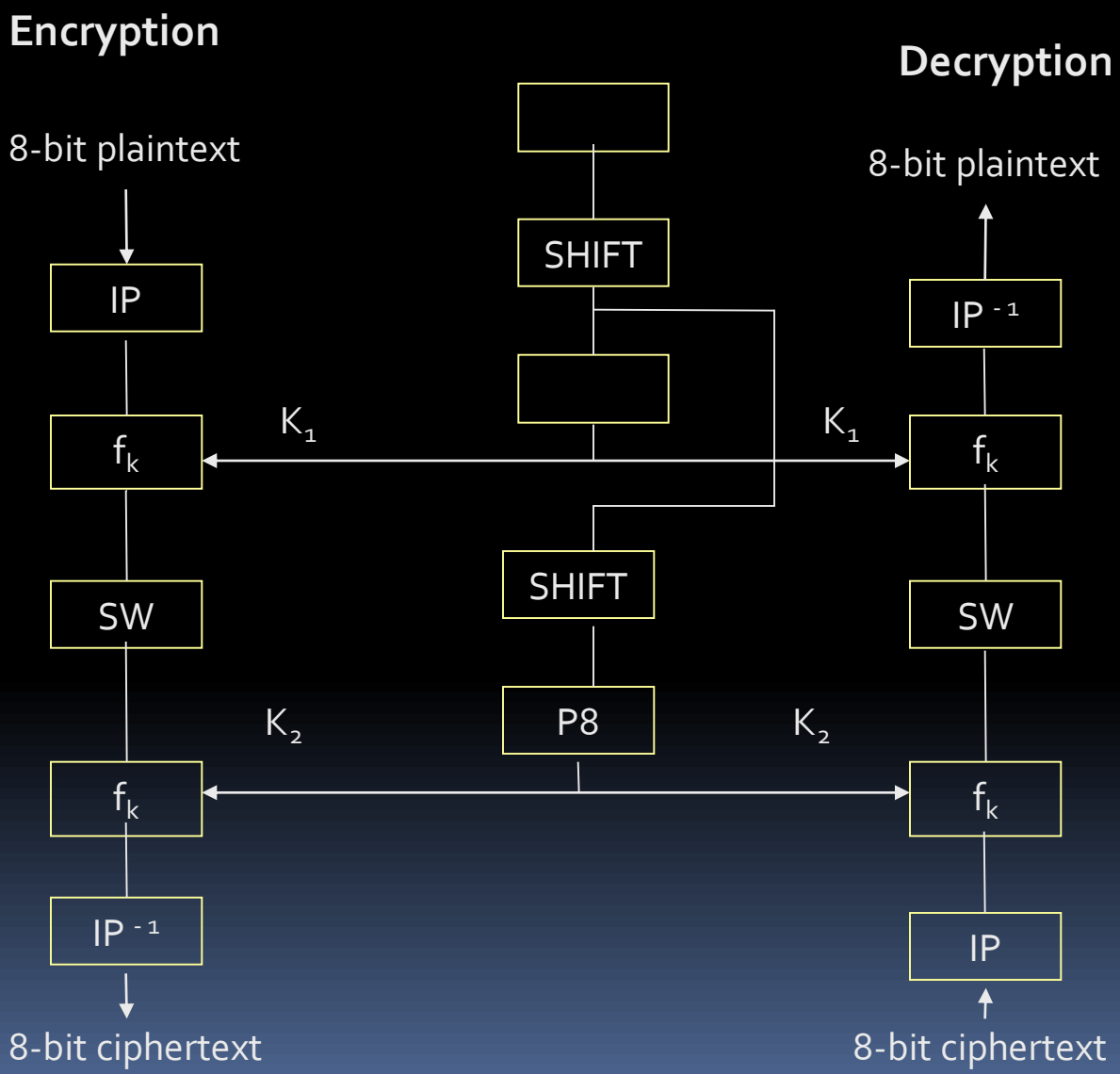

Input (plaintext)
LE_0 RE_0
Round 1
F
K_1
LE_1 RE_1
Round 2
F
K_2
LE_2 RE_2
LE_{14} RE_{14}
Round 15
F
K_{15}
LE_{15} RE_{15}
Round 16
F
K_{16}
LE_{16} RE_{16}
LE_{17} RE_{17}
Output (ciphertext)
Output (plaintext)
$RD_{17} = LE_0$ $LD_{17} = RE_0$
$LD_{16} = RE_0$ $RD_{16} = LE_0$
Round 16
F
K_1
$LD_{15} = RE_1$ $RD_{15} = LE_1$
Round 15
F
K_2
$LD_{14} = RE_2$ $RD_{14} = LE_2$
$LD_2 = RE_{14}$ $RD_2 = LE_{14}$
Round 2
F
K_{15}
$LD_1 = RE_{15}$ $RD_1 = LE_{15}$
Round 1
F
K_{16}
$LD_0 = RE_{16}$ $RD_0 = LE_{16}$
Input (ciphertext)

Feistel Cipher Structure

- **Block size:** larger block sizes mean greater security
- **Key Size:** larger key size means greater security
- **Number of rounds:** multiple rounds offer increasing security
- **Subkey generation algorithm:** greater complexity will lead to greater difficulty of cryptanalysis.
- **Fast software Encryption/Decryption:** the speed of execution of the algorithm becomes a concern
- **Ease of analysis:** if the algorithm can be concisely and clearly explained, it is easier to analyze that algorithm for cryptanalytic vulnerabilities

What is Simplified DES

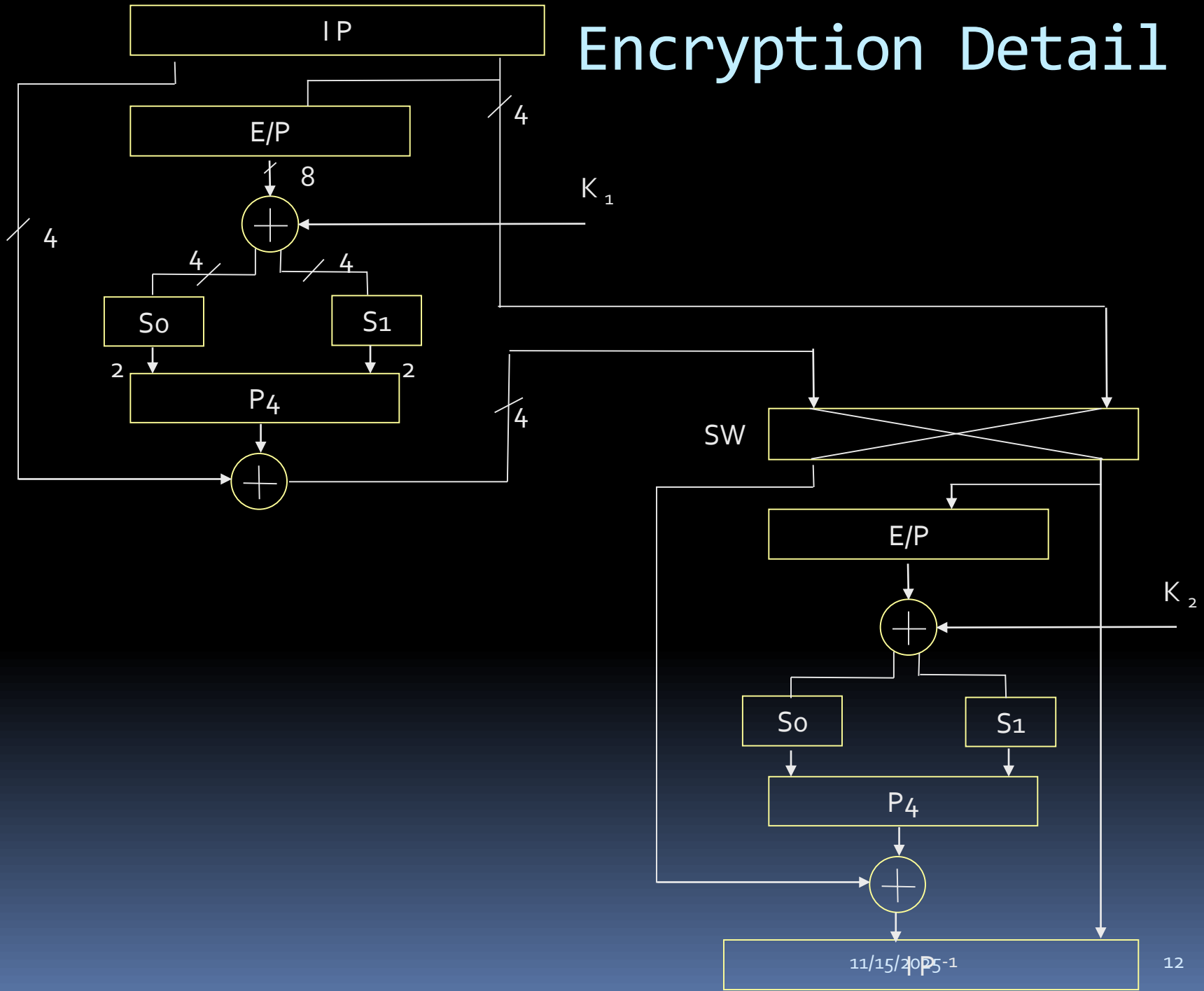
- Developed 1996 as a teaching tool
  - Santa Clara University
    - Prof. Edward Schaefer
  - Takes an 8-bit block plaintext, a 10 –bit key and produces an 8-bit block of ciphertext
  - Decryption takes the 8-bit block of ciphertext, the same 10-bit key and produces the original 8-bit block of plaintext

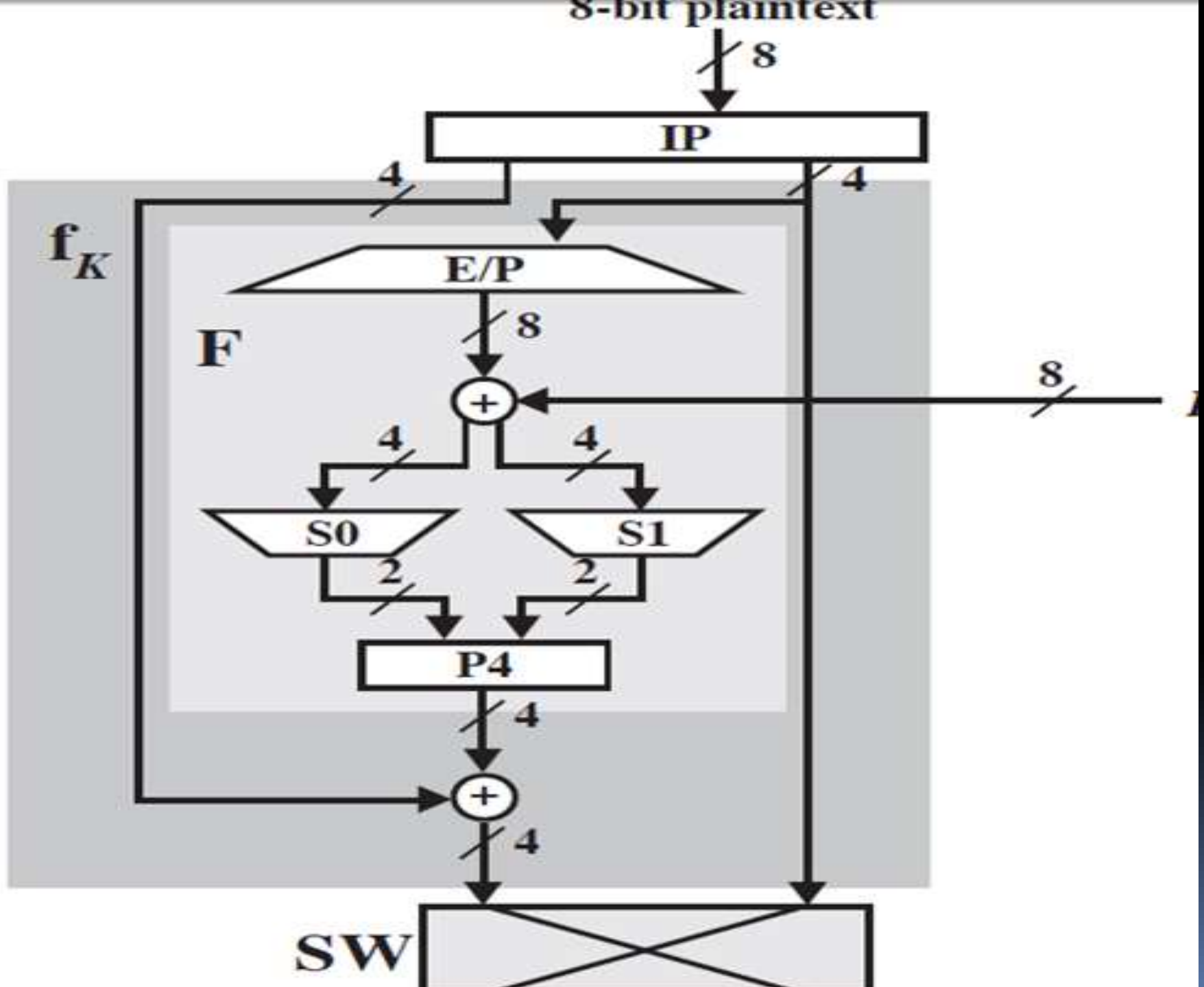

Encryption
Decryption
8-bit plaintext
8-bit plaintext
IP
SHIFT
IP - 1
K1
K1
fk
fk
SW
SHIFT
SW
K2
P8
K2
fk
fk
IP - 1
IP
8-bit ciphertext
8-bit ciphertext

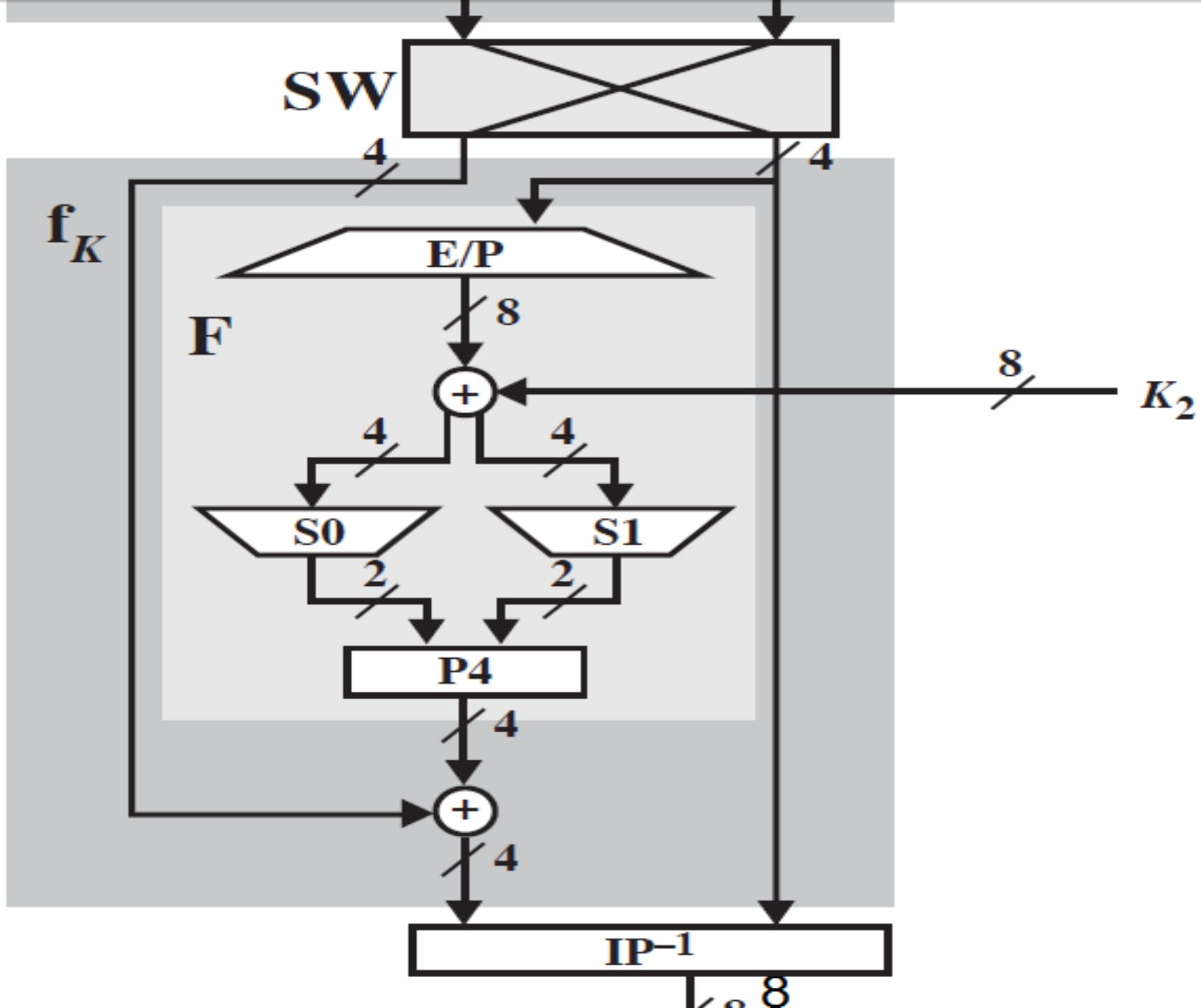
Five Functions to Encrypt

- IP – an initial permutation
- f_k - a complex, 2-input function
- SW – a simple permutation that swaps the two snybles
- f_k - a complex, 2-input function; again
- IP – inverse permutation of the initial permutation

Encryption Detail

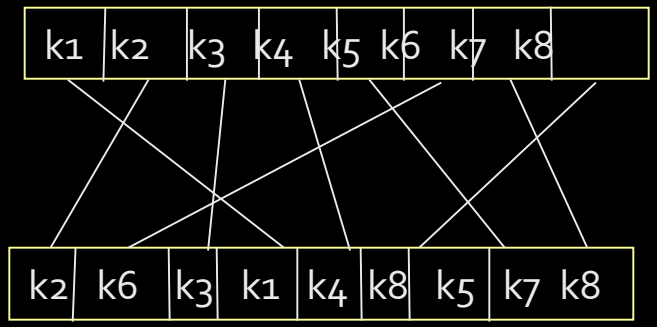



8-bit plaintext
8
IP
4
4
f_K
E/P
F
8
8
4
4
S0
S1
2
2
P4
4
4
SW


SW
4
4
f_K
E/P
F
8
8
K_2
4
4
S0
S1
2
2
P4
4
4
IP–1
8

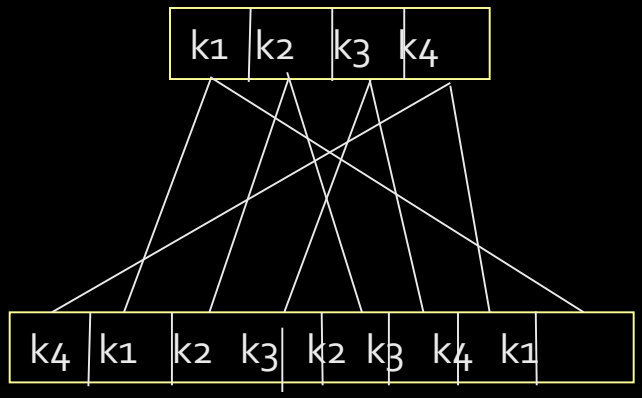
Initial Permutation (IP)

Move the bits of the original character around a little…

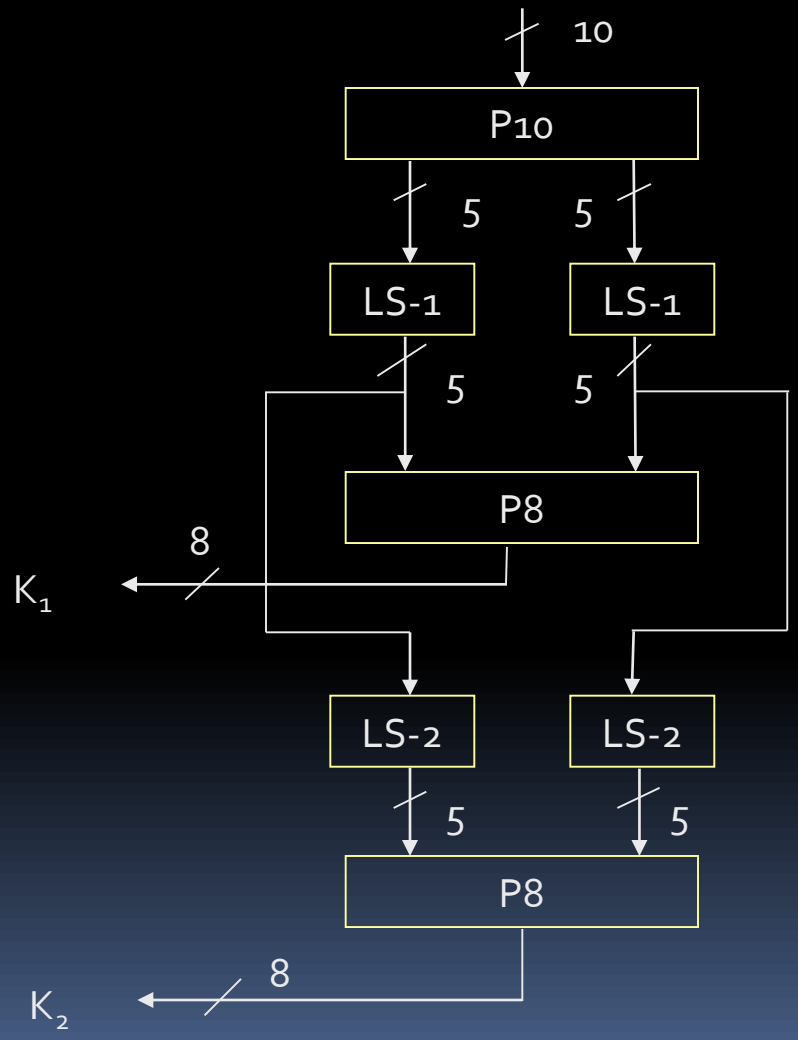


Expansion/Permutation (E/P)

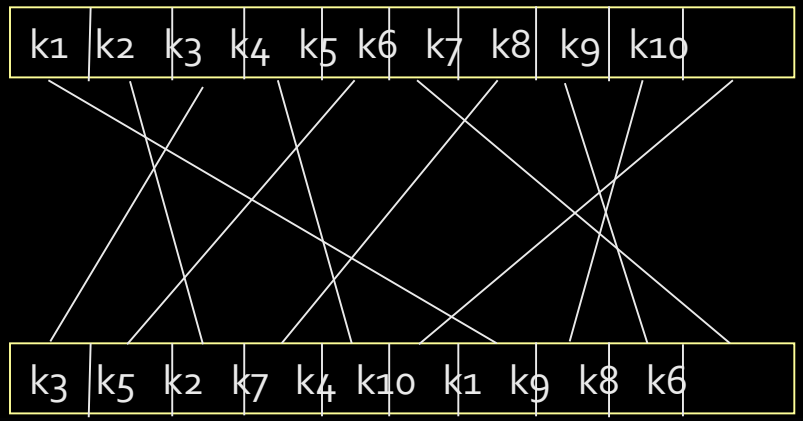
Expand 4 bits into 8 and permutate them…



Key Generation

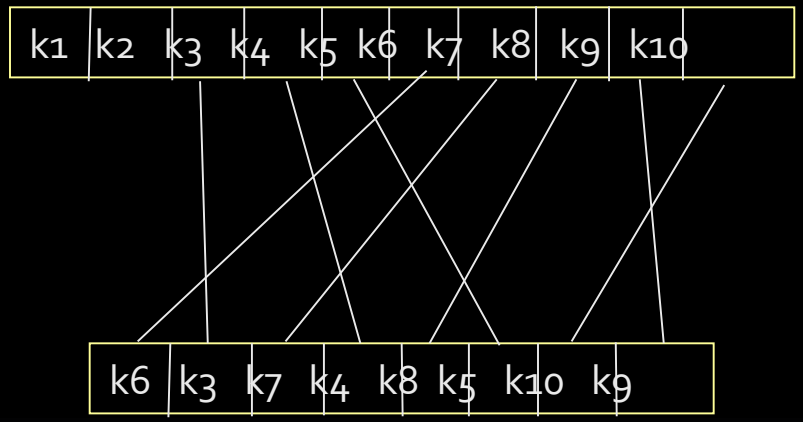


P10 Permutation



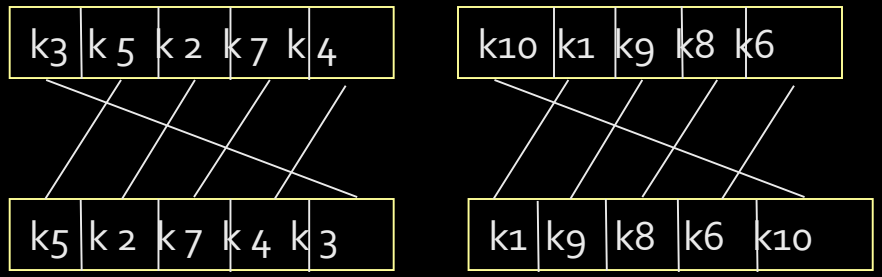
P8 Permutation

Permutate 10 into 8



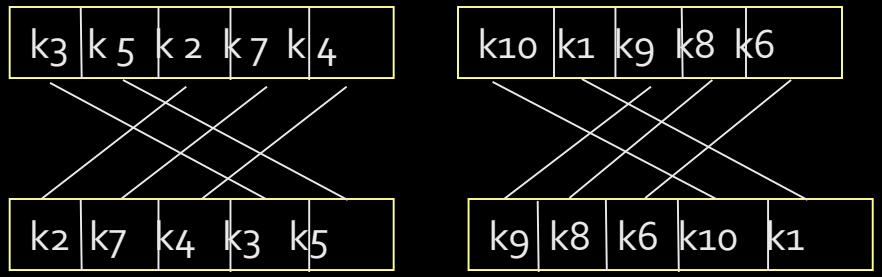
LS-1

Left circular shift 1 each 5 bit group



LS-2

Left circular shift 2 each 5 bit group



Substitution Boxes

S_0

1	0	3	2
3	2	1	0
0	2	1	3
3	1	3	2

S_1

0	1	2	3
2	0	1	3
3	0	1	0
2	1	0	3

Example

Using S-DES, encrypt and decrypt the string (10100010) using the key (0111111101) by hand. Show intermediate results after each function (IP, F_k, SW, F_k, IP^{-1}). Then decode the first 4 bits of the plaintext string to a letter and the second 4 bits to another letter where we encode A through P in base 2 (i.e., A = 0000, B = 0001,..., P = 1111).

Hint: As a midway check, after the application of SW, the string should be (00010011).

Let the 10 bit key be [1100011110]

P10									
3	5	2	7	4	10	1	9	8	6

P8							
6	3	7	4	8	5	10	9

Find k1 and k2

Example: given

Input

0 1 1 0 1 1 0 1

IP							
2	6	3	1	4	8	5	7

E/P							
4	1	2	3	2	3	4	1

K1							
1	0	1	0	0	1	0	0

S0 =

	c0	c1	c2	c3
r0	1	0	3	2
r1	3	2	1	0
r2	0	2	1	3
r3	3	1	3	2

S1 =

	c0	c1	c2	c3
r0	0	1	2	3
r1	2	0	1	3
r2	3	0	1	0
r3	2	1	0	3

P4			
2	4	3	1

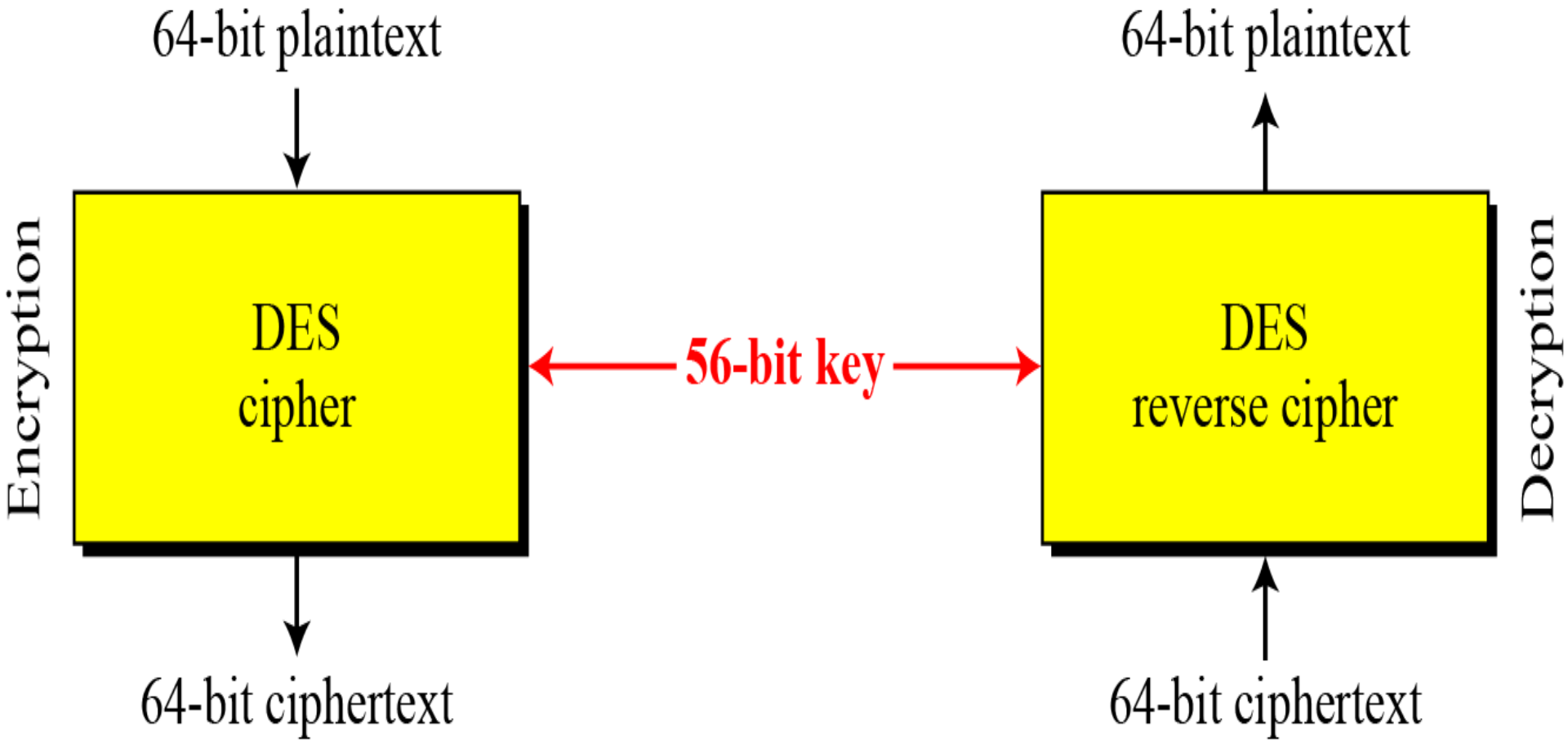
K2

0	**1**	**0**	**0**	**0**	**0**	**1**	**1**

IP^{-1}							
4	1	3	5	7	2	8	6

DES is a block cipher, as shown in Figure 6.1.

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



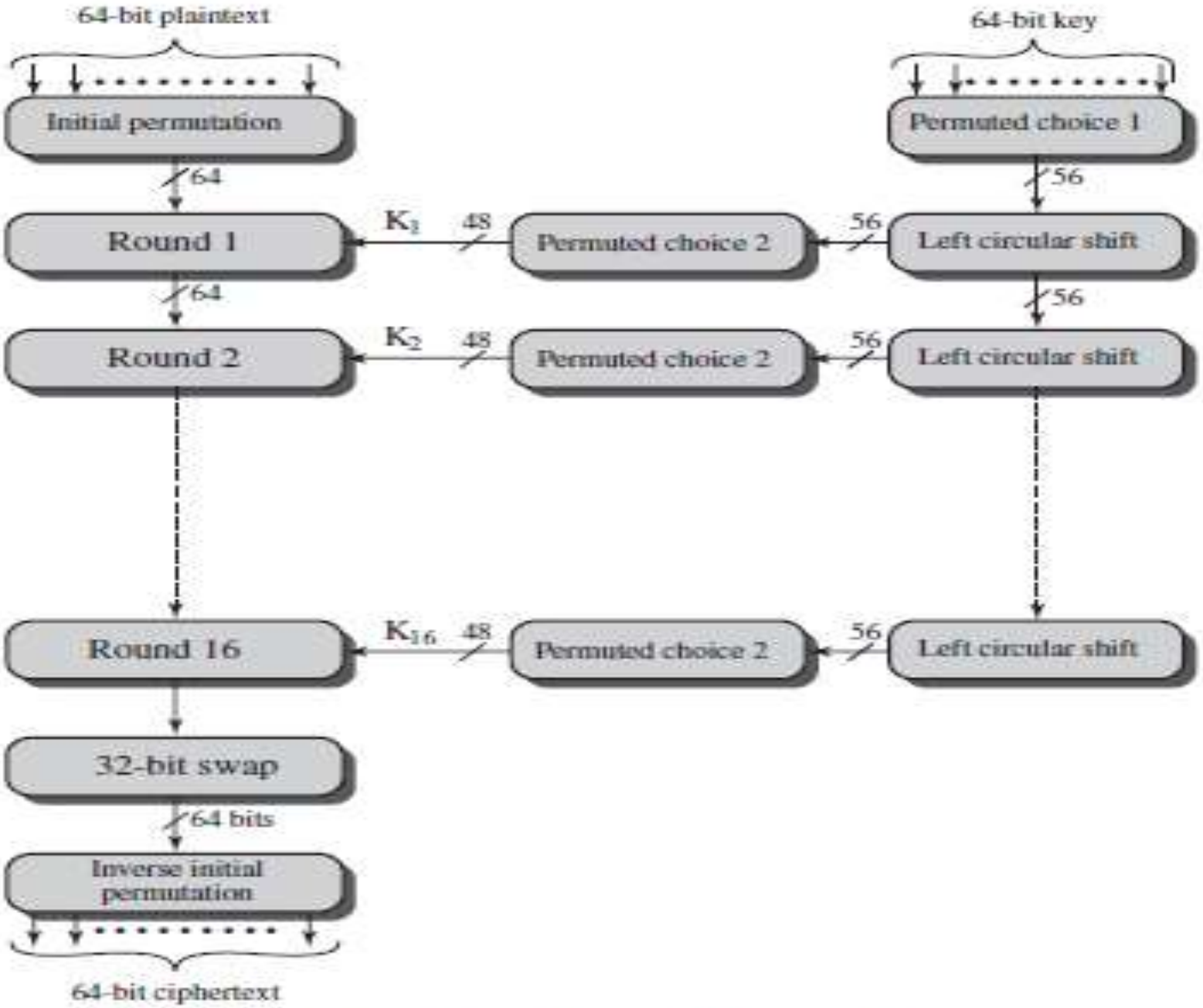


Figure 3.5 General Depiction of DES Encryption Algorithm

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

General structure of DES

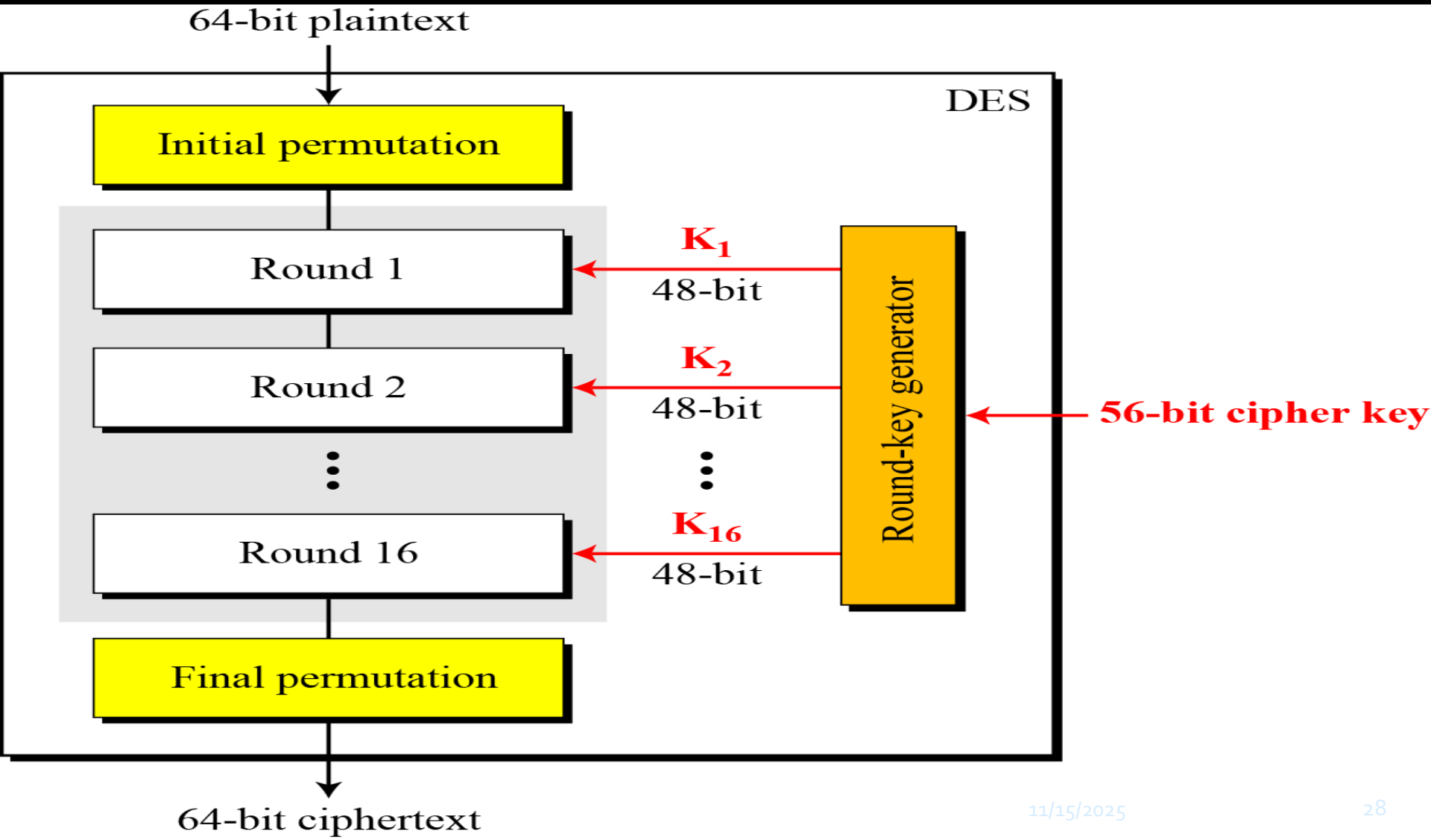
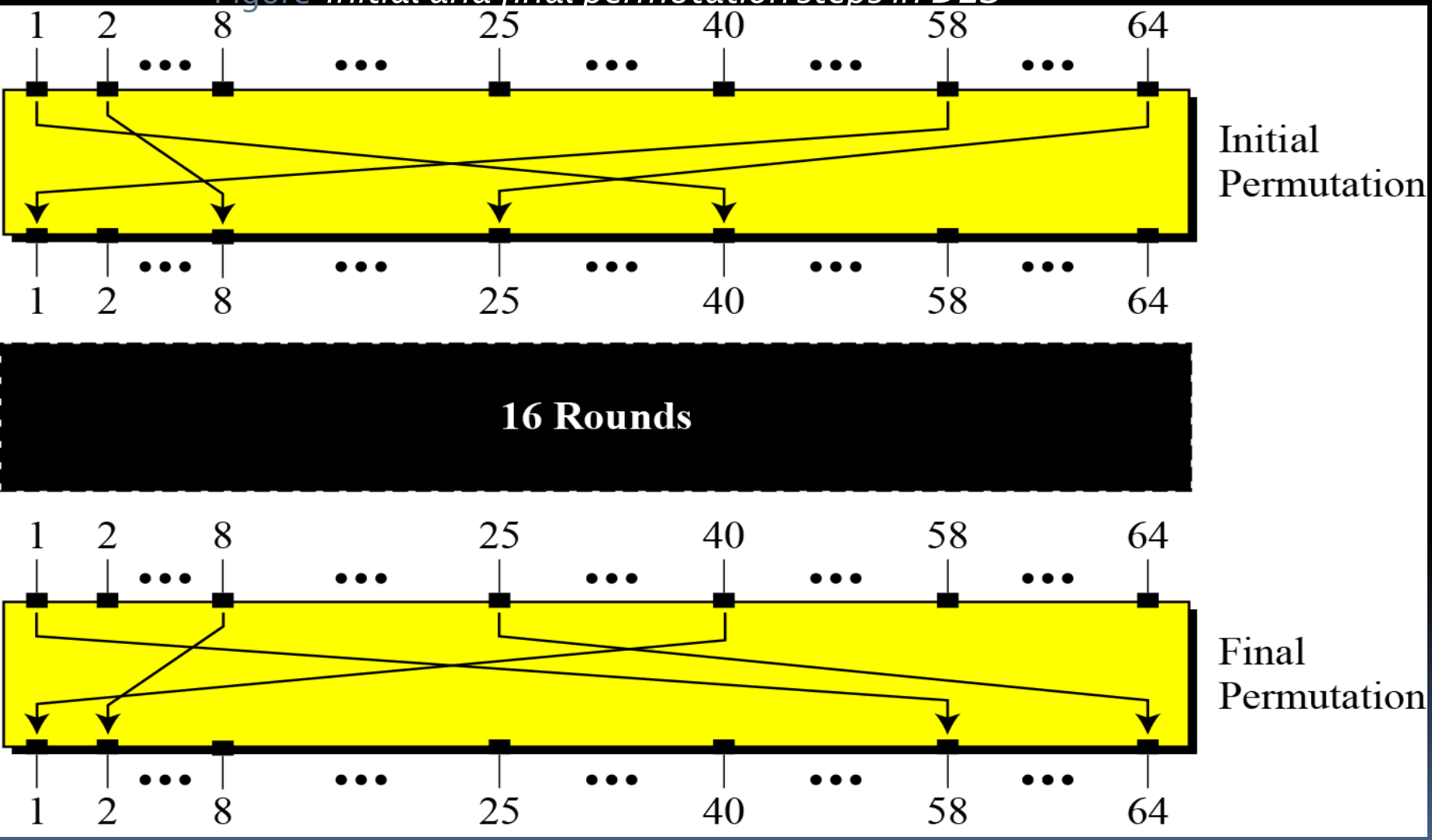


Figure *Initial and final permutation steps in DES*



Initial and final permutation tables

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

Final Permutation							
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

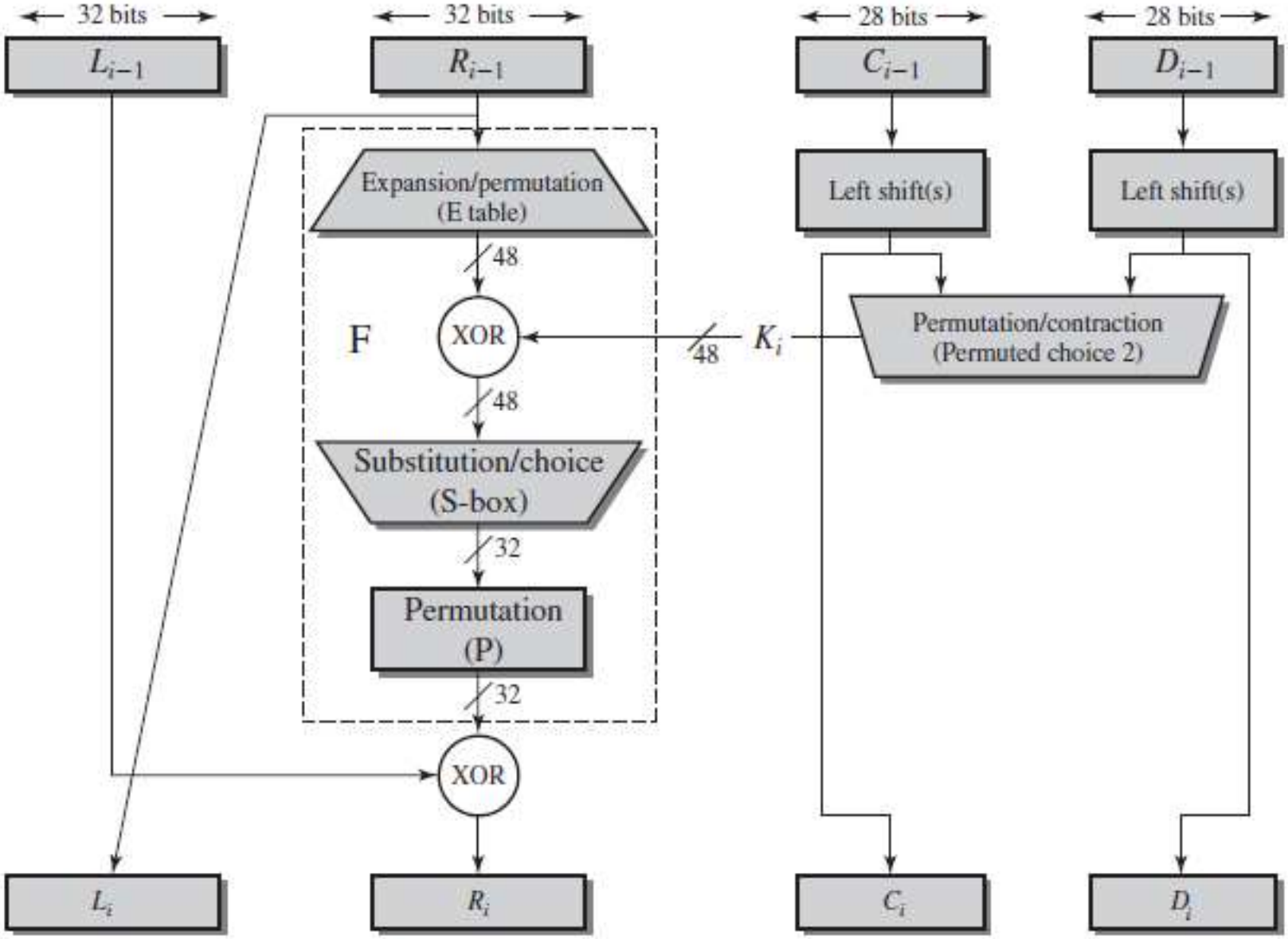


Figure 3.6 Single Round of DES Algorithm

(a) 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

(b) Inverse Initial 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

(c) Expansion Permutation (E)

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

(d) Permutation Function (P)

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

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

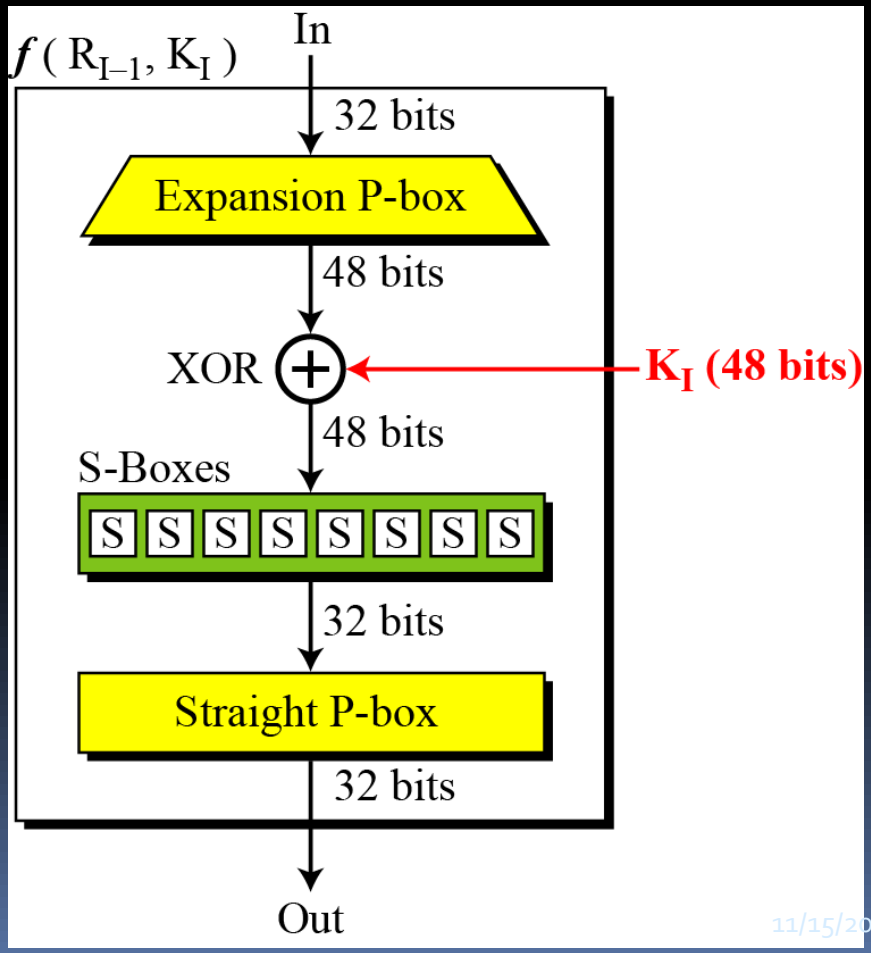
0x0002 0000 0000 0001

0x0000 0080 0000 0002

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.

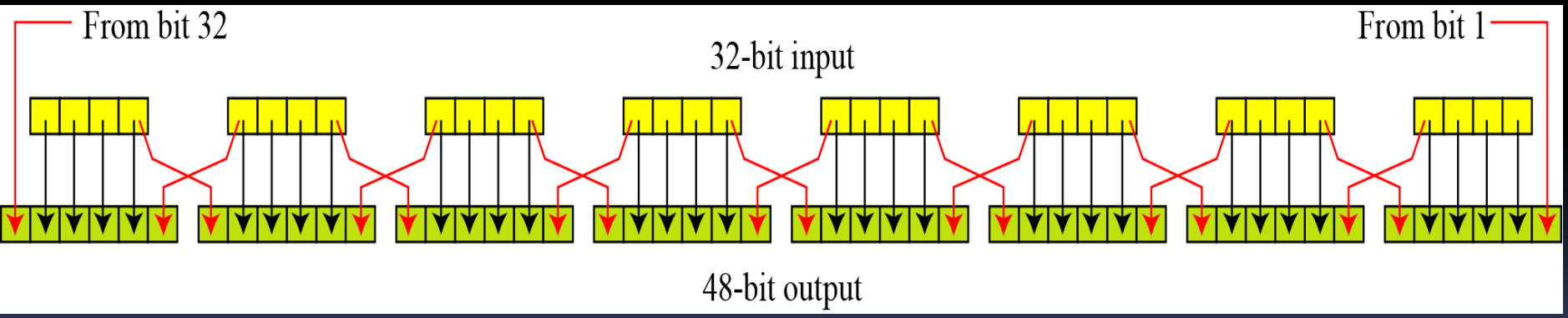
DES function



Expansion P-box

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



Although the relationship between the input and output can be defined mathematically, DES uses Table to define this P-box.

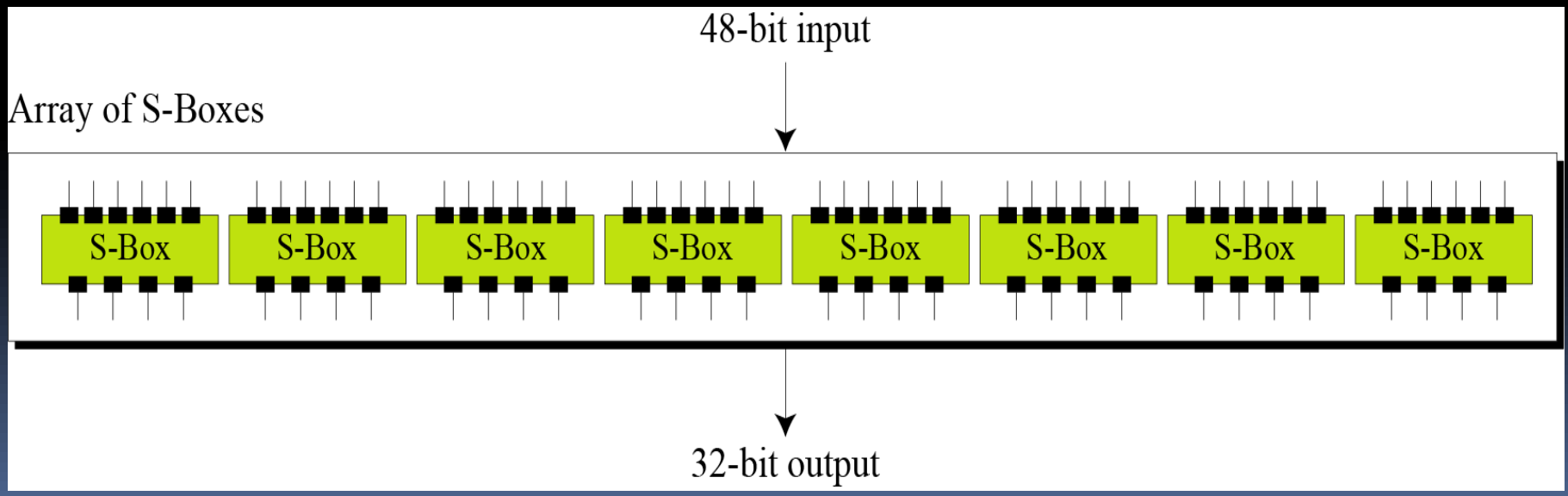
Expansion P-box table

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

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. See Figure

S-boxes



S-box rule

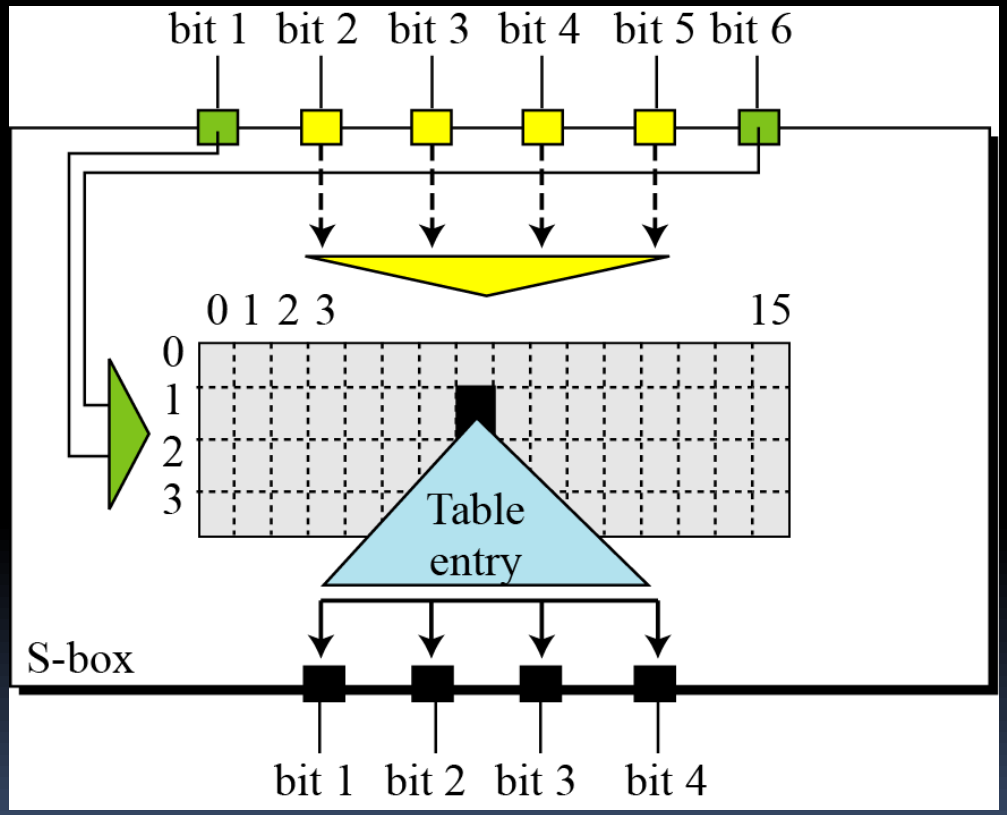

bit 1 bit 2 bit 3 bit 4 bit 5 bit 6
0 1 2 3
15
0
1
2
3
Table entry
S-box
bit 1 bit 2 bit 3 bit 4

Table shows the permutation for S-box 1. For the rest of the boxes see the textbook.

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

Example

The input to S-box 1 is 100011. 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 0001 in binary, which is 1 in decimal. We look for the value in row 3, column 1, in Table 6.3 (S-box 1). The result is 12 in decimal, which in binary is 1100. So the input 100011 yields the output 1100.

Continued

Example

The input to S-box 8 is oooooo. What is the output?

Solution

If we write the first and the sixth bits together, we get oo in binary, which is o in decimal. The remaining bits are oooo in binary, which is o in decimal. We look for the value in row o, column o, 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.

KEY GENERATION Returning to, we see that a 64-bit key is used as input to the algorithm. The bits of the key are numbered from 1 through 64; every eighth bit is ignored, as indicated by the lack of shading in below Table. The key is first subjected to a permutation governed by a table labeled Permuted Choice One then (The resulting 56-bit key is then treated as two 28-bit quantities, labeled

(a) Input Key

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

(b) Permuted Choice One (PC-1)

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

(c) Permuted Choice Two (PC-2)

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

Triple-DES

Triple-DES is just DES with two 56-bit keys applied.

Given a plaintext message, the first key is used to DES- encrypt the message. The second key is used to DES-decrypt the encrypted message.

(Since the second key is not the right key, this decryption just scrambles the data further.) The twice-scrambled message is then encrypted again with the first key to yield the final ciphertext. T his three-step procedure is called triple-DES. Triple-DES is just DES done three times with two keys used in a particular order. (Triple-DES can also be done with three separate keys instead of only two. In either case the resultant key space is about 2^112.)