الجامعة المستنصرية /كلية التربية / قسم علوم الحاسبات

4th Class

Computers & Data Security

أمنية الحاسوب والبيانات

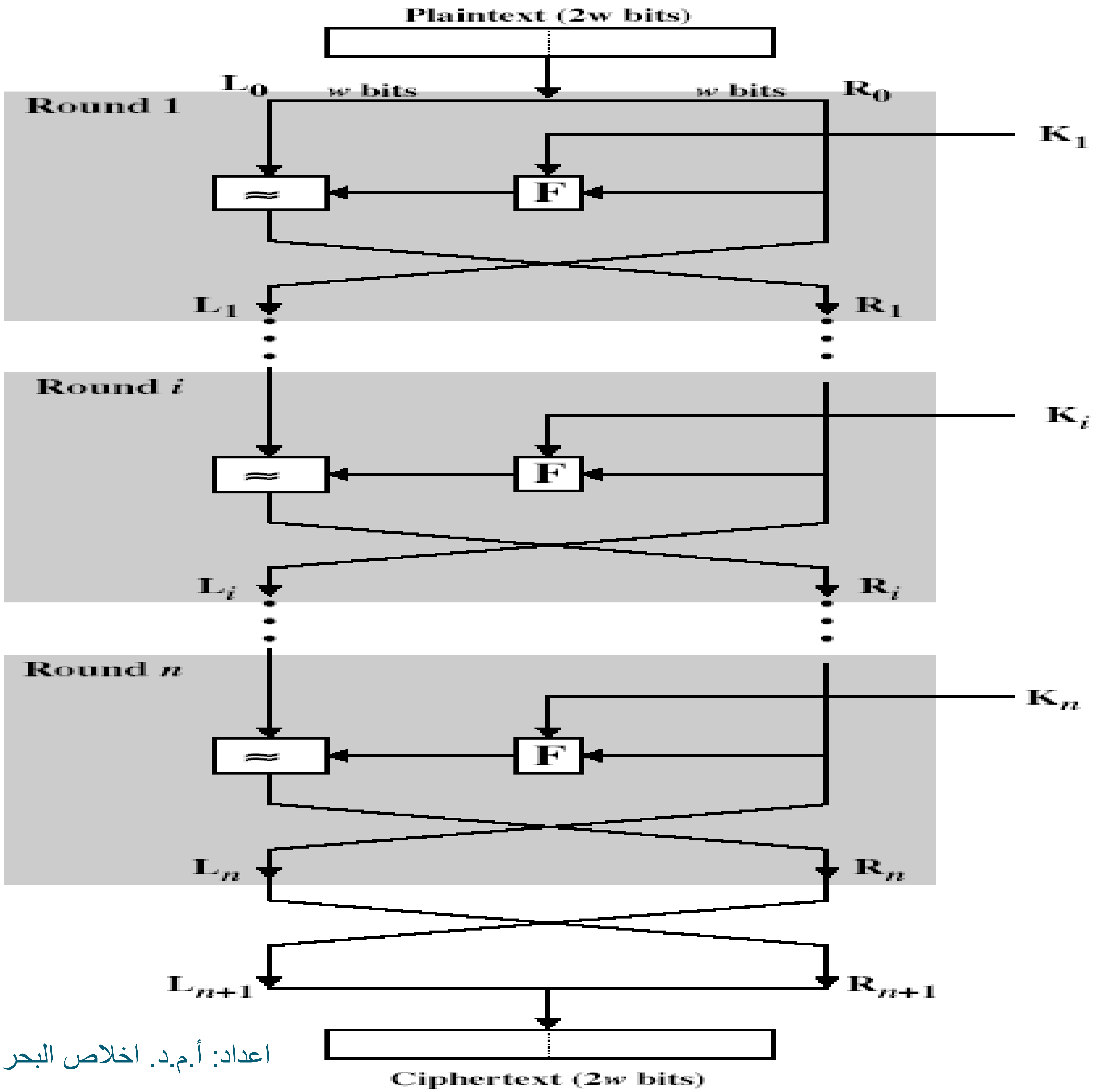
أستاذ المادة

أ.م . د . اخلاص عباس البحراني

Block Cipher algorithms

- **Product Cipher** - An encryption scheme that "uses multiple ciphers in which the cipher text of one cipher is used as the clear text of the next cipher". Usually, substitution ciphers and transposition ciphers are used alternatively to construct a product cipher.
- **Iterated Block Cipher** - A block cipher that "iterates a fixed number of times of another block cipher, called round function, with a different key, called round key, for each iteration".
- Most symmetric block ciphers are based on a **Feistel Cipher Structure.**
- **Feistel Cipher** - An iterate block cipher that Process through multiple rounds which
  - partitions input block into two halves
  - perform a substitution on left data half
  - based on round function of right half & sub key
  - then have permutation swapping halves

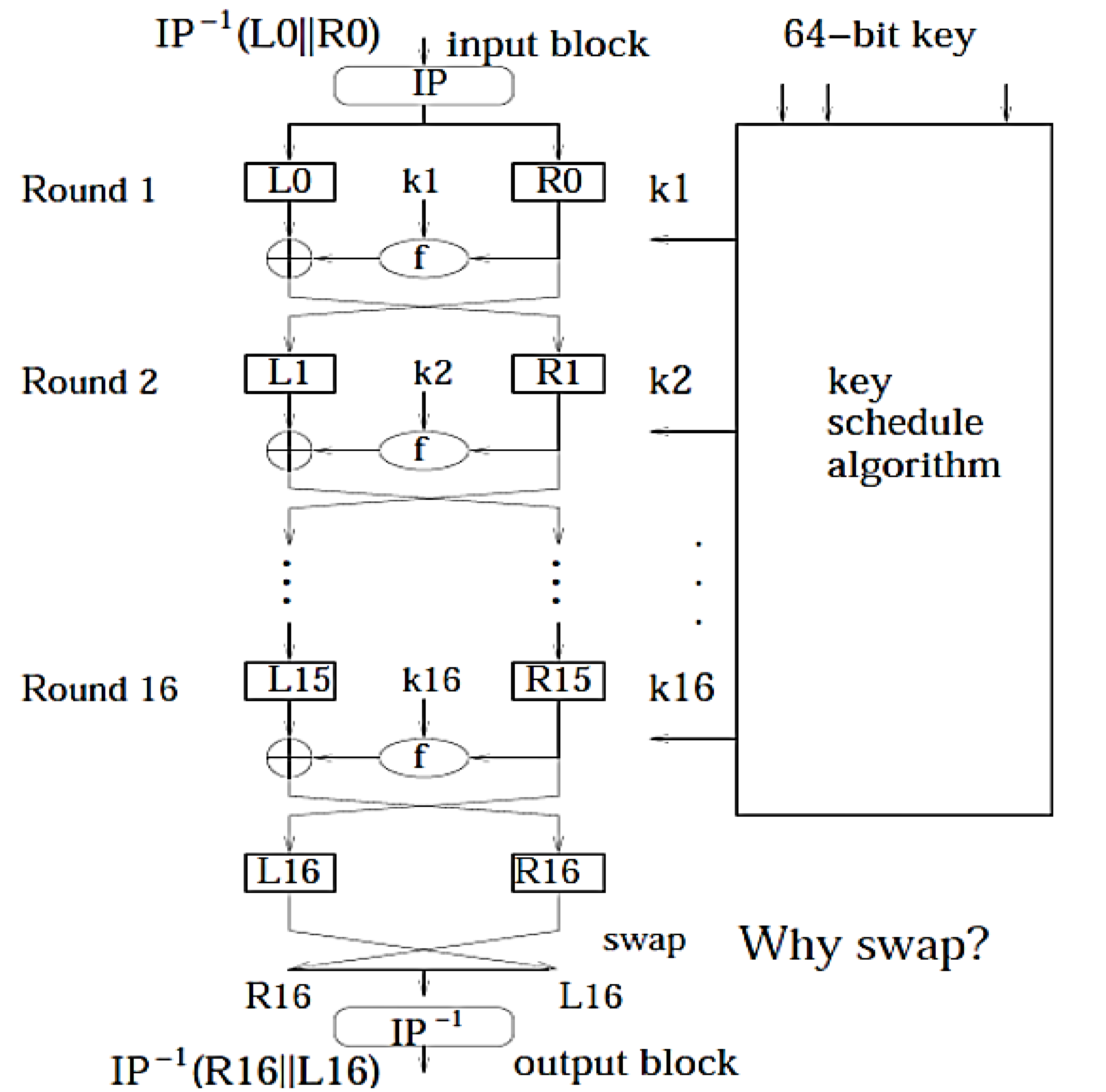
Feistel Cipher Structure



Data Encryption Standard (DES)

- A 16-round Feistel cipher with block size of 64 bits.
- Published in 1977, standardized in 1979.
- Key: 64 bit quantity = 8-bit parity+56-bit key
  - every eighth bit is used for parity checking and is ignored.
- It encrypts 64-bit data, and uses 56-bit key with 16 48-bit sub-keys.
- DES is a symmetric algorithm: The same algorithm and key are used for both encryption and decryption (except for minor differences in the key schedule).
- DES is a single combination of these techniques (a substitution followed by a permutation) on the text, based on the key.
- All security rests within the key.
- The algorithm is nothing more than a combination of the two basic techniques of encryption: **confusion** and diffusion.
  - **Diffusion** – dissipates statistical structure of plaintext over bulk of ciphertext(spreading of the effect of a change in the plaintext to many parts of the ciphertext).
  - **Confusion** – makes relationship between ciphertext and key as complex as possible (difficulty in determining how a change in the plaintext will affect the ciphertext).

DES Algorithm



اعداد: أ.م.د. اخلاص البحراني

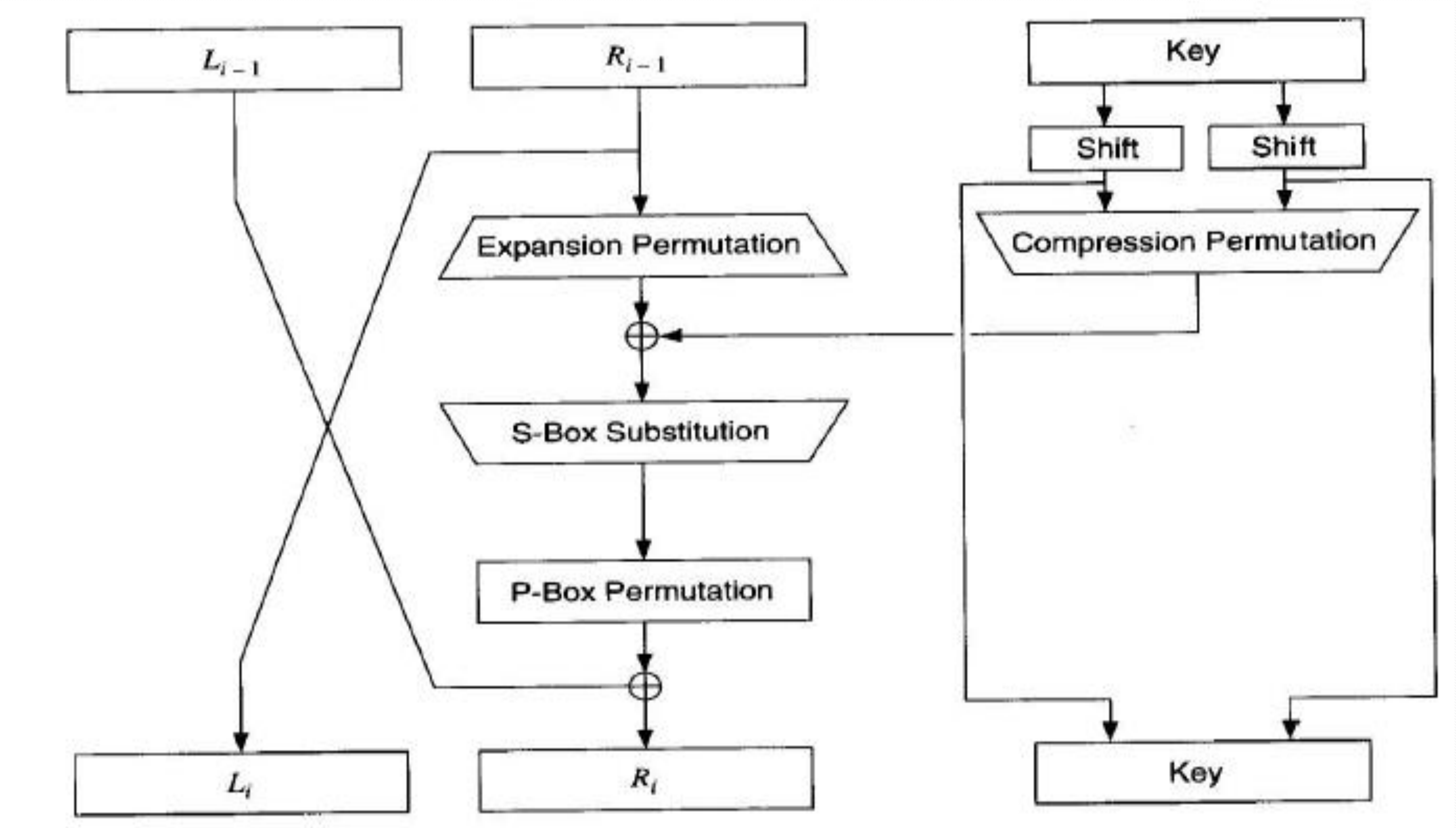
Outline of the Encryption Algorithm

- The basic process in enciphering a 64-bit data block using the DES consists of:
  - an initial permutation (IP)
  - 16 rounds of a complex key dependent calculation f
  - final permutation, being the inverse of IP
- In each round : -
  - the key bits are shifted, and then 48 bits are selected from the 56 bits of the key.
  - The right half of the data is expanded to 48 bits via an **expansion permutation**, combined with 48 bits of a shifted and permuted key via **an XOR**, sent through **8 S-boxes** producing 32 new bits, and **permuted** again.
  - These four operations make up **Function F**.
  - The output of Function F is then combined with the left half via another XOR.
  - The result of these operations becomes the new right half; the old right half becomes the new left half.
  - If Bi is the result of the ith iteration, Li and Ri are the left and right halves of Bi, Ki is the 48-bit key for round i, and F is the function that does all the substituting and permuting and XORing with the key, then a round looks like:

$$Li = R_{j-1}$$

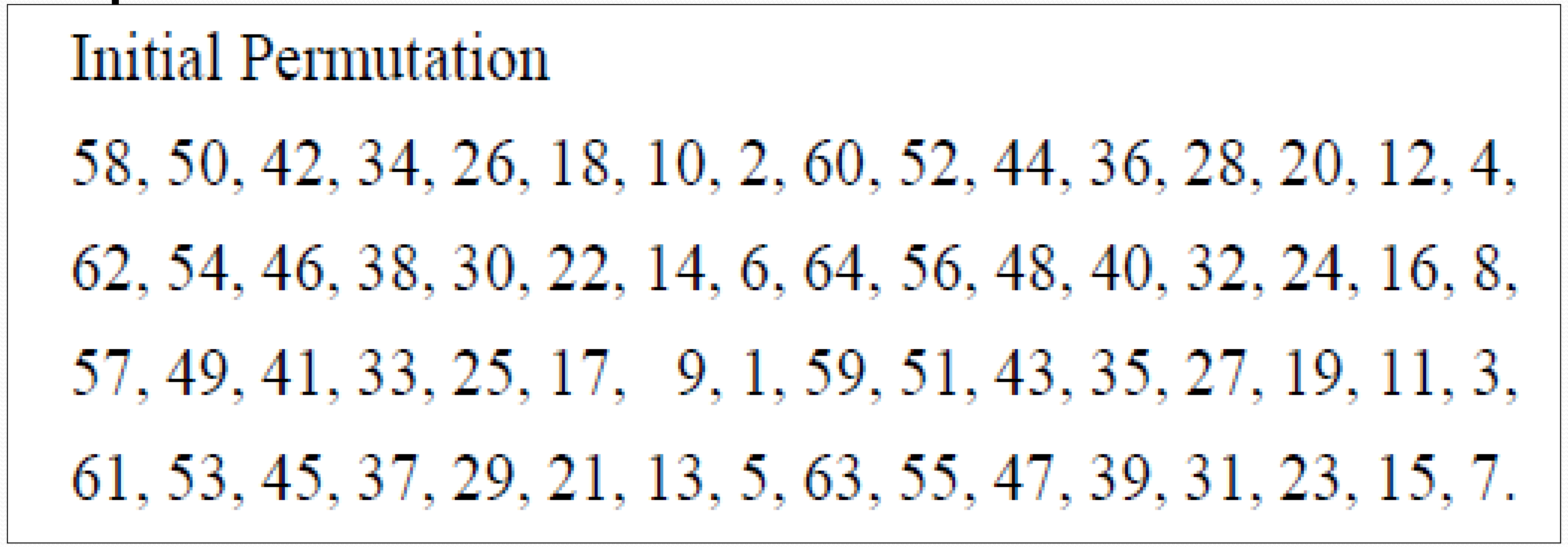
$$Ri = L_{i-1} \text{ Xor } f(R_{i-1}, K_i)$$

One round of DES



The Initial Permutation

- The initial permutation occurs before round 1.
- it transposes the input block as described in this Table



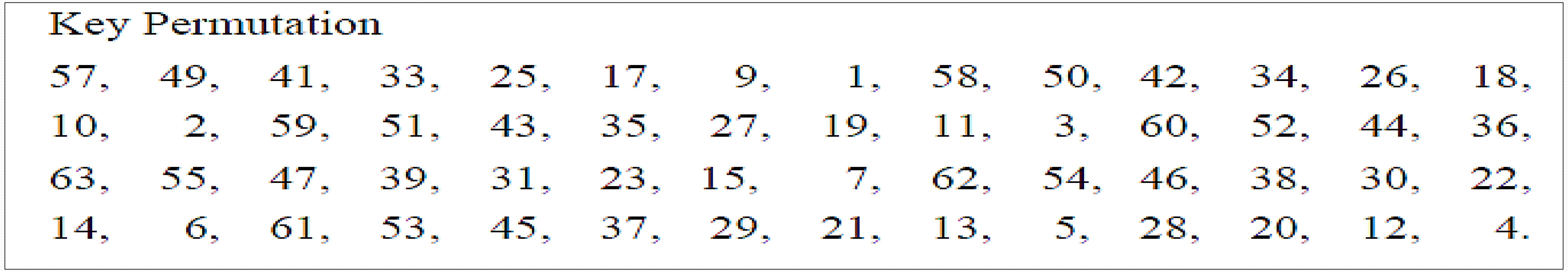
Initial Permutation

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.

- This table The initial permutation and the corresponding final permutation do not improve DES's security, just make DES more complex should be read left to right, top to bottom.
- For example, the initial permutation moves bit 58 of the plaintext to bit position 1, bit 50 to bit position 2, bit 42 to bit position 3, and so forth.
- Example: IP(675a6967 5e5a6b5a) = (ffb2194d 004df6fb)

The Key Transformation: -

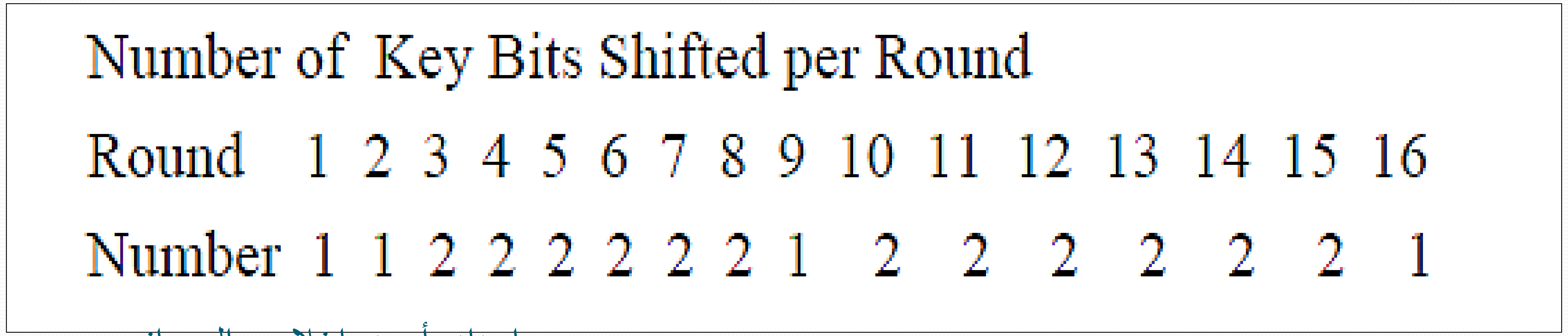
- Initially, the 64-bit DES key is reduced to a 56-bit key by ignoring every eighth bit using this table: -



Key Permutation

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.

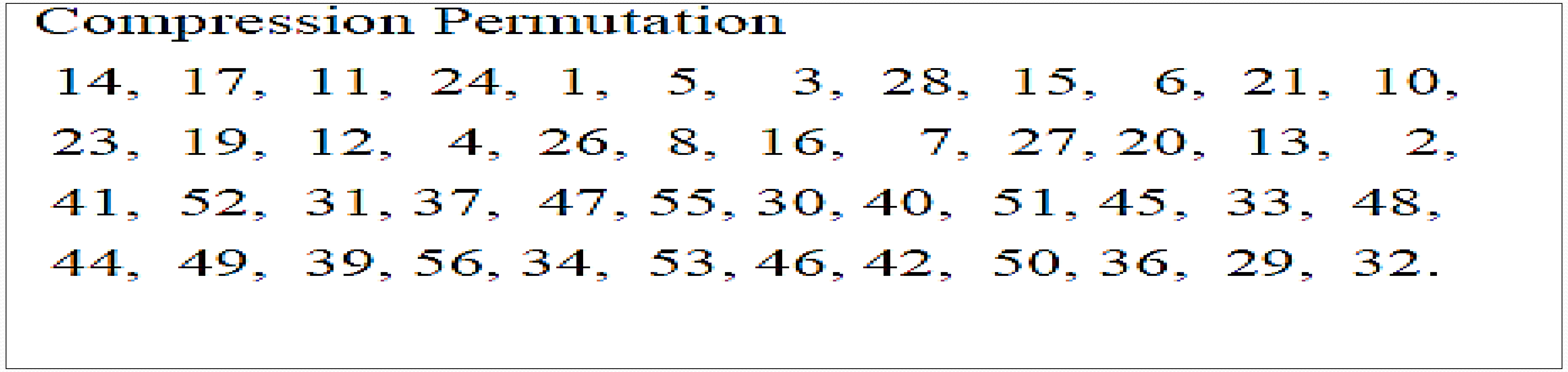
- Next the 56-bits key is reduced to a 48-bits subkey for each of the 16 rounds of DES. These subkeys, Ki, are determined in the following manner: -
  - splits the 56-bits key bits into 2 halves (C and D), each 28-bits
  - The halves C and D are circularly shifted left by either one or two bits, depending on the round. This shift is given in this table



Number of Key Bits Shifted per Round

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

- After being shifted, 48 out of the 56 bits are selected. This is done by an operation called **compression permutation**, it permutes the order of the bits as well as selects a subsets of bits.



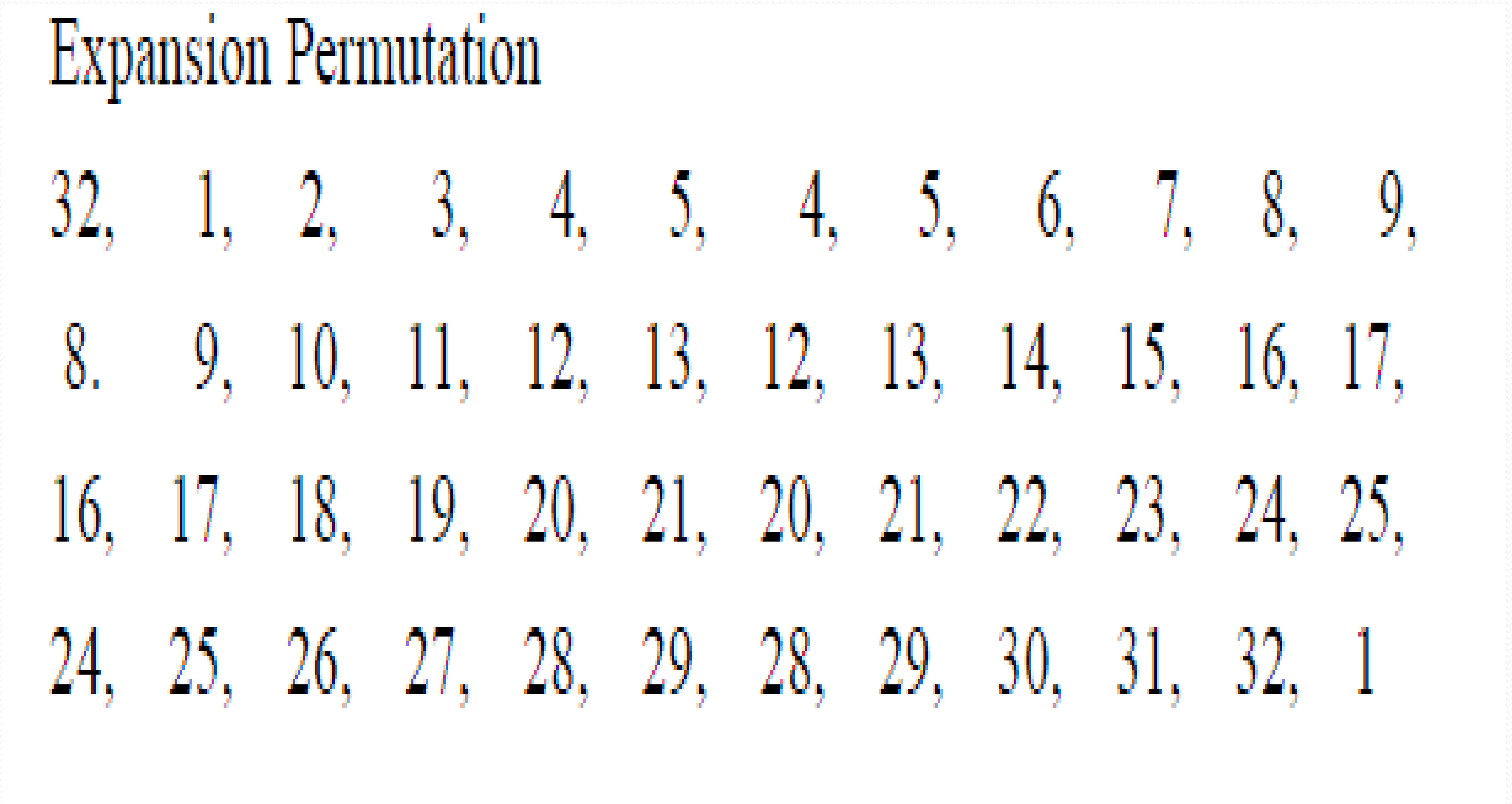
Compression Permutation

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.

- Example: keyinit(5b5a5767, 6a56676e)

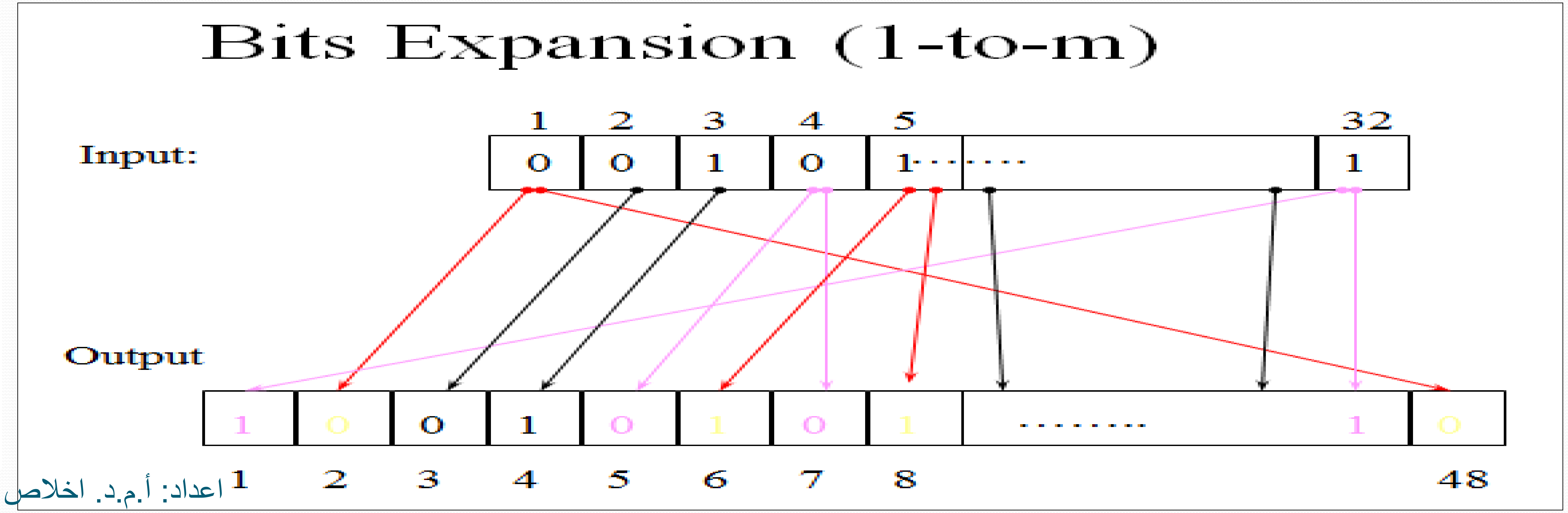
The Expansion Permutation: -

- This operation expands the right half of the data, Ri, from 32 bits to 48 bits.
- Because this operation changes the order of the bits as well as repeating certain bits, it is known as an expansion permutation
- This operation has two purposes:
  - It makes the right half the same size as the key for the XOR operation
  - and it provides a longer result that can be compressed during the substitution operation.
- For each 4-bit input block, the first and fourth bits each represent two bits of the output block, while the second and third bits each represent one bit of the output block as shown : -



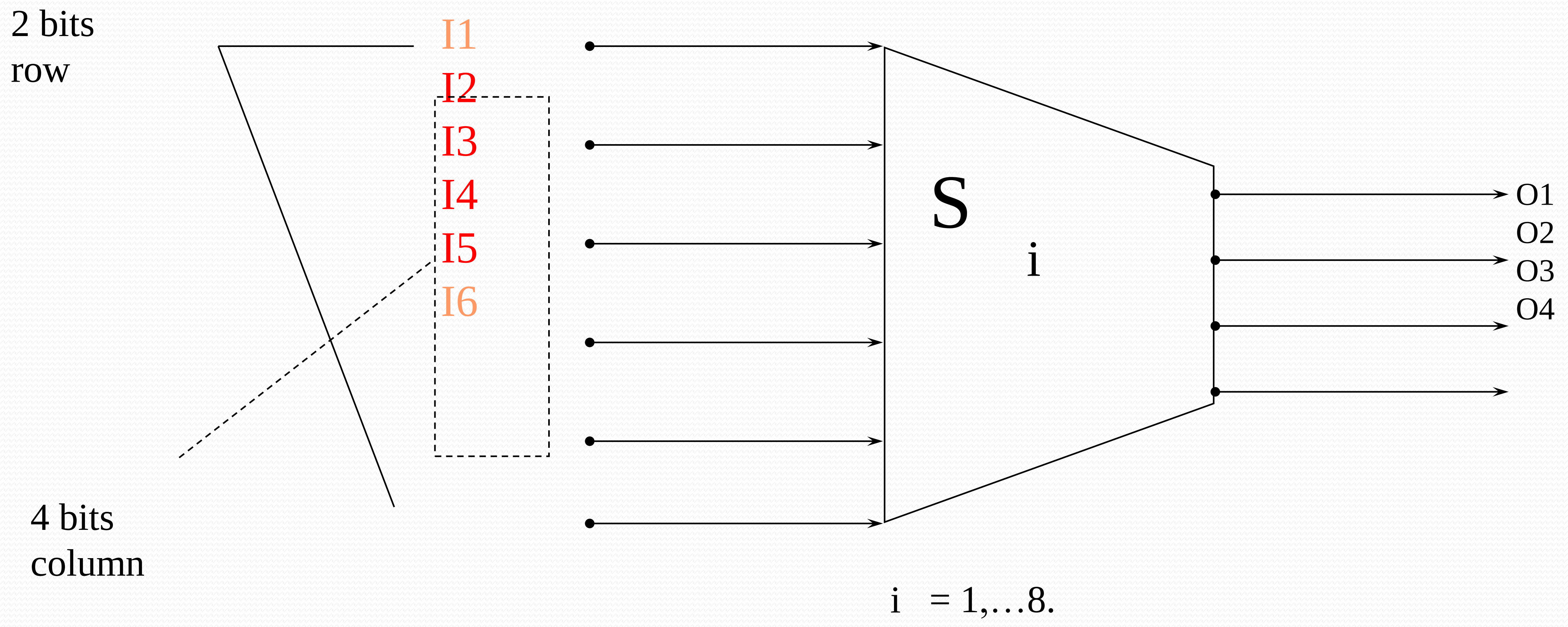
Expansion Permutation

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



The S-Box Substitution:-

- After the compressed key is XORed with the expanded block, the 48-bit result moves to a substitution operation.
- The substitutions are performed by eight substitution boxes, or S-boxes.
- There are eight different S-boxes.
- Each S-box has a 6-bit input and a 4-bit output.



- The 48 bits are divided into eight 6-bit sub-blocks.
- Each separate block is operated on by a separate S-box: The first block is operated on by S-box 1, the second block is operated on by S-box 2, and so on.
- Each S-box is a table of 4 rows and 16 columns. Each entry in the box is a 4-bit number.
- The 6 input bits of the S-box specify under which row and column number to look for the output. All eight S-boxes tables are :

S-box 1:

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

S-box 2:

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

S-box 3:

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

S-box 4:

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

S-box 5:

2,	12,	4,	1,	7,	10,	11,	6,	8,	5,	3,	15,	13,	0,	14,	9,
14,	11,	2,	12,	4,	7,	13,	1,	5,	0,	15,	10,	3,	9,	8,	6,
41,	2,	1,	11,	10,	13,	7,	8,	15,	9,	12,	5,	6,	3,	0,	14,
11,	8,	12,	7,	1,	14,	2,	13,	6,	15,	0,	9,	10,	4,	5,	3,

S-box 6:

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

S-box 7:

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

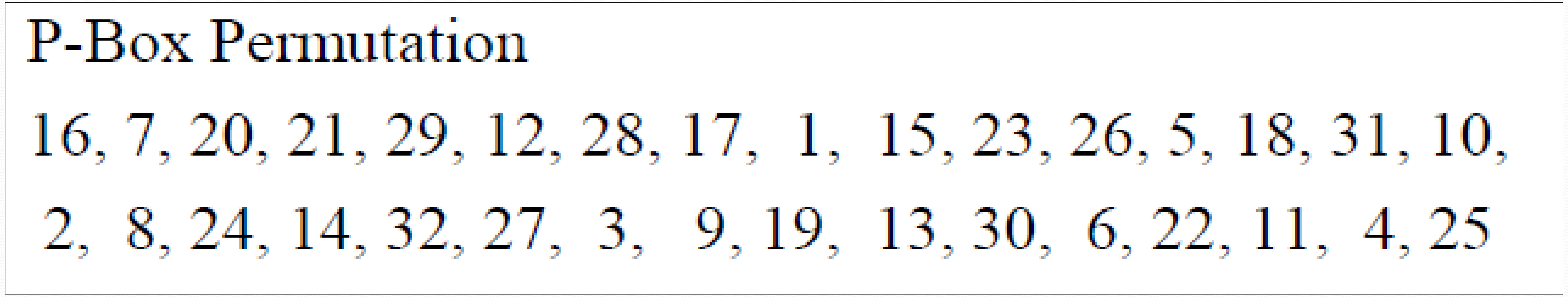
S-box 8:

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

- For example, assume that the input to the sixth S-box (i.e., bits 31 through 36 of the XOR function) is 110011.
- The first and last bits combine to form 11, which corresponds to row 3 of the sixth S-box.
- The middle 4 bits combine to form 1001, which corresponds to the column 9 of the same S-box.
- The entry under row 3, column 9 of S-box 6 is 14. (Remember to count rows and columns from 0 and not from 1.)
- The value 1110 is substituted for 110011.
- Example: S(18 09 12 3d 11 17 38 39) = 5fd25e03 (?)

The P-Box Permutation: -

- The 32-bit output of the S-box substitution is permuted according to a P-box.
- This permutation maps each input bit to an output position; no bits are used twice and no bits are ignored.



P-Box Permutation

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

- For example, bit 21 moves to bit 4. while bit 4 moves to bit 3 1.
- Finally, the result of the P-box permutation is XORed with the left half of the initial 64-bit block.
- Then the left and right halves are switched and another round begins.

The Final Permutation: -

- The final permutation is the inverse of the initial permutation.

Final Permutation

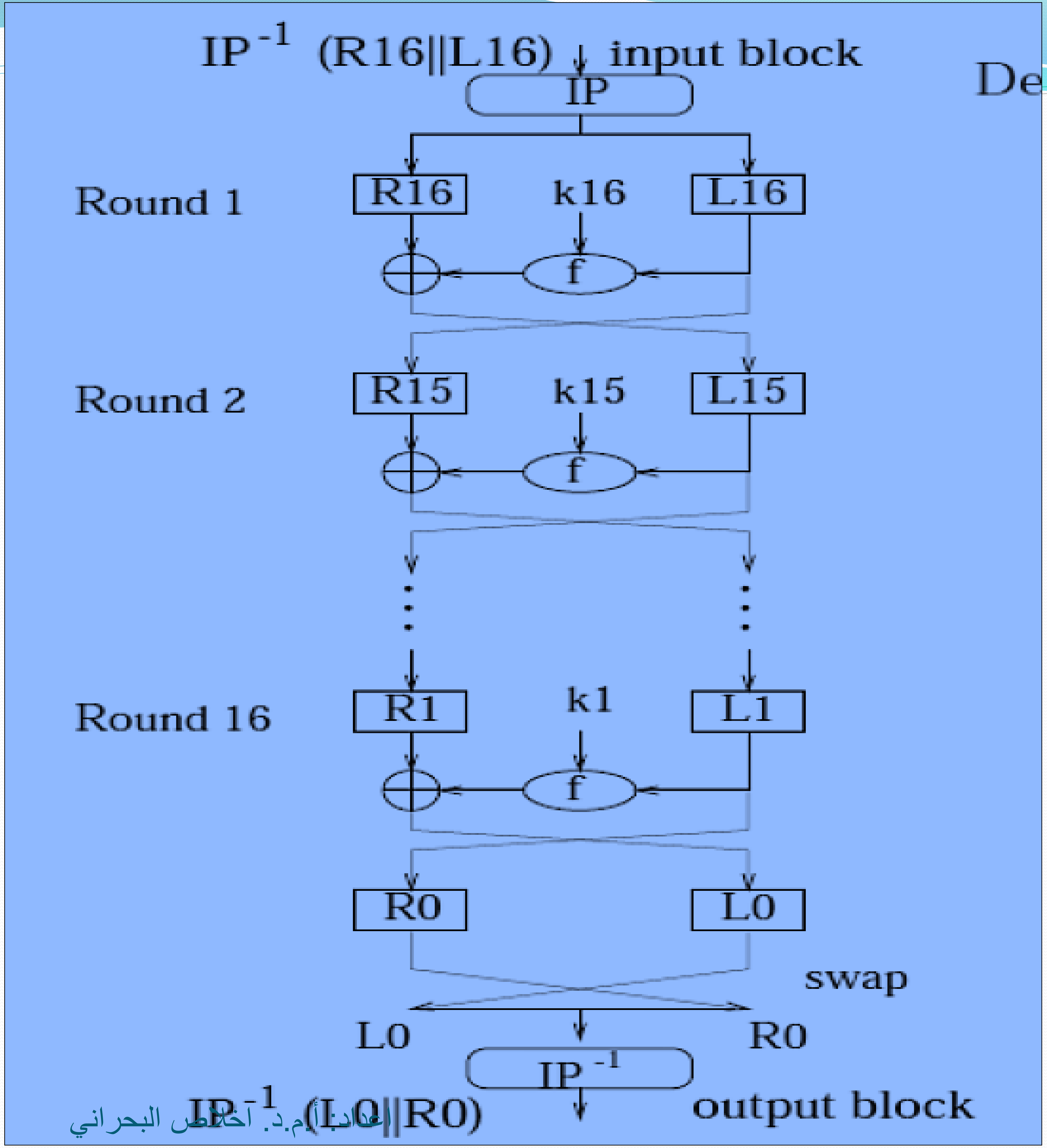
40, 8, 48, 16, 56, 24, 64, 32, 39, 7, 47, 15, 55, 23, 63,
31,
38, 6, 46, 14, 54, 22, 62 30, 37, 5, 45, 13, 53, 21, 61,
29,
36, 4, 44, 12, 52, 20, 60, 28, 35, 3, 43, 11, 51, 19, 59,
27,
34, 2, 42, 10, 50, 18, 58, 26, 33, 1, 41, 9, 49, 17, 57,
25.

- Note that the left and right halves are not exchanged after the last round of DES; instead the concatenated block R16L16 is used as the input to the final permutation.

Decrypting DES

- same function to encrypt or decrypt a block.
- The only difference is that the keys must be used in the reverse order. That is, if the encryption keys for each round are K1, K2, K3, . . . , K16, then the decryption keys are K16, K15, K14, . . . , K1,.
- The algorithm that generates the key used for each round is circular as well.
- The key shift is a right shift and the number of positions shifted is 0, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1.

De


IP -1 (R16||L16)
input block
IP
Round 1
R16
k16
L16
f
Round 2
R15
k15
L15
f
Round 16
R1
k1
L1
f
R0
L0
swap
L0
R0
IP -1
IP -1 (L0||R0)
output block