DES & AES

19CSE311 Computer Security

Jevitha KP
Department of CSE

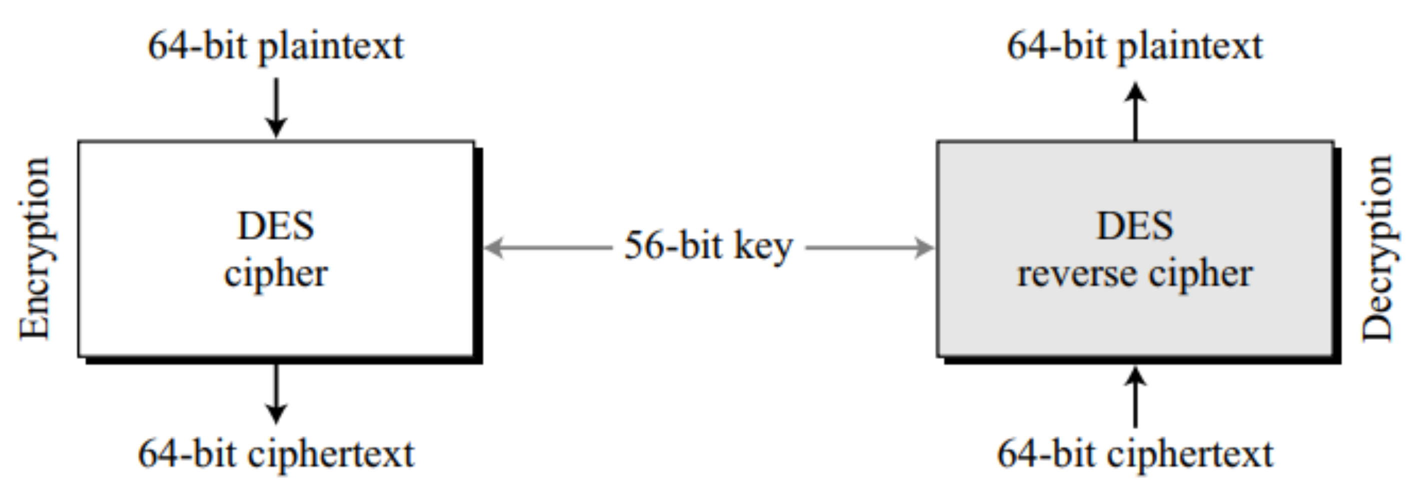
DES

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

- **Encryption** : DES takes a 64-bit plaintext and creates a 64-bit ciphertext;

- **Decryption** : DES takes a 64-bit ciphertext and creates a 64-bit block of plaintext.

- **Key:** Same 56-bit cipher key is used for both encryption and decryption

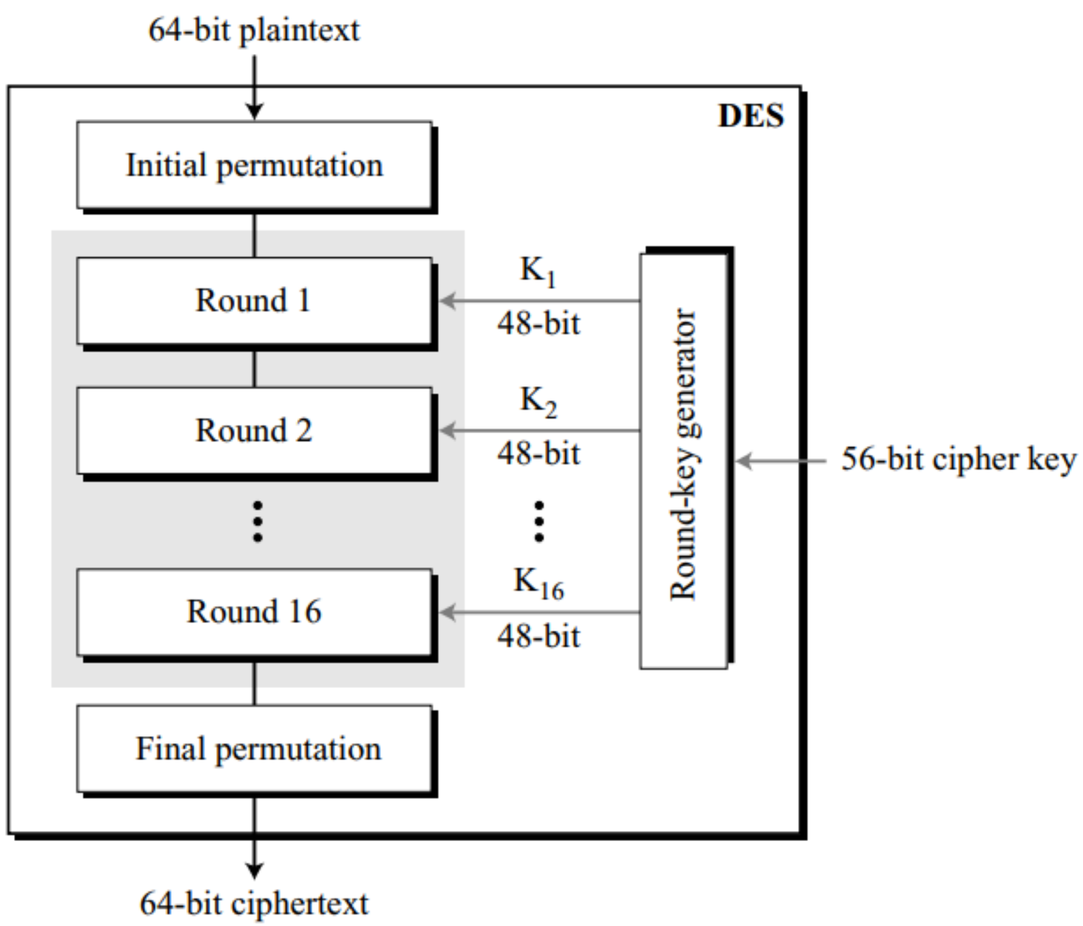


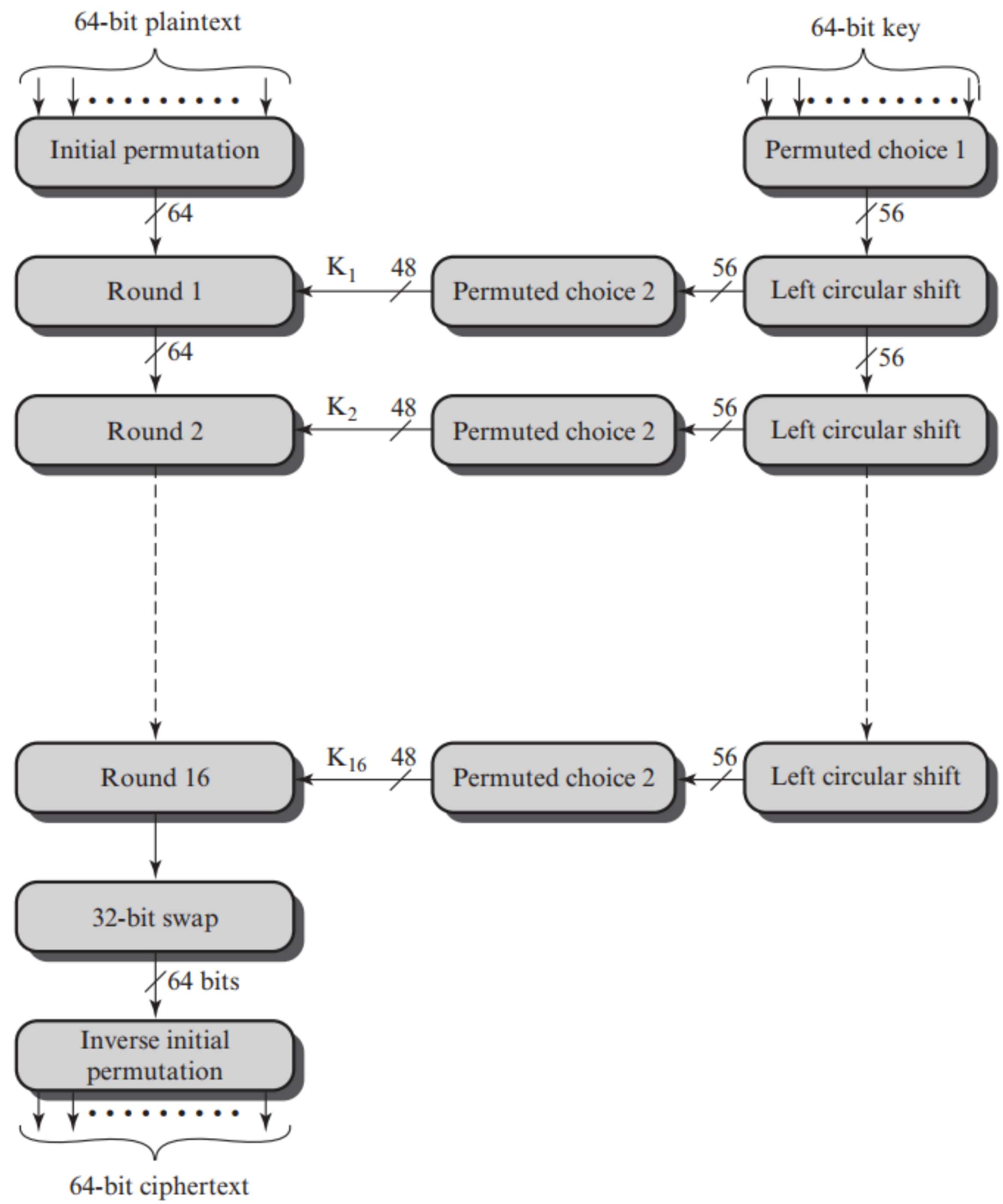
DES Encryption

- The encryption process is made of **two** permutations **(P-boxes)**, called **initial** and **final permutations**, and **sixteen Feistel rounds**.
- Each round uses a different **48-bit round key** generated from the cipher key according to a predefined algorithm
- The initial and final permutations are **straight P-boxes** that are inverses of each other.

DES

- **Key :** Function expects a 64-bit key as input.
- only 56 of these bits are ever used;
- the other 8 bits can be used as parity bits or set arbitrarily

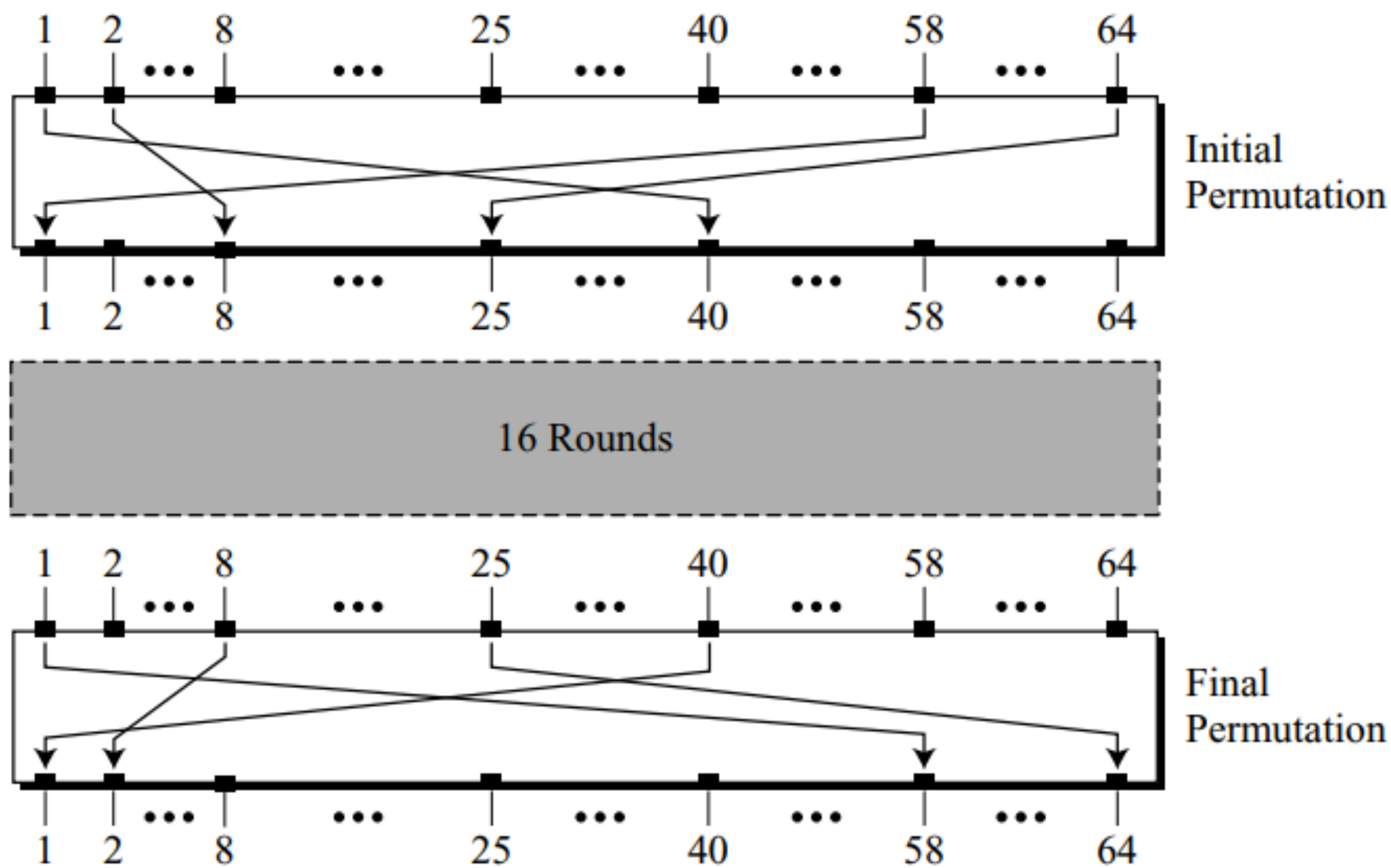




Initial and Final Permutations

- Initial and final permutations (P-boxes) takes a 64-bit input and permutes them according to a predefined rule.
- These permutations are keyless straight permutations that are the **inverse of each other**.
- They have no cryptography significance in DES.

Initial and Final Permutation (IP and FP)

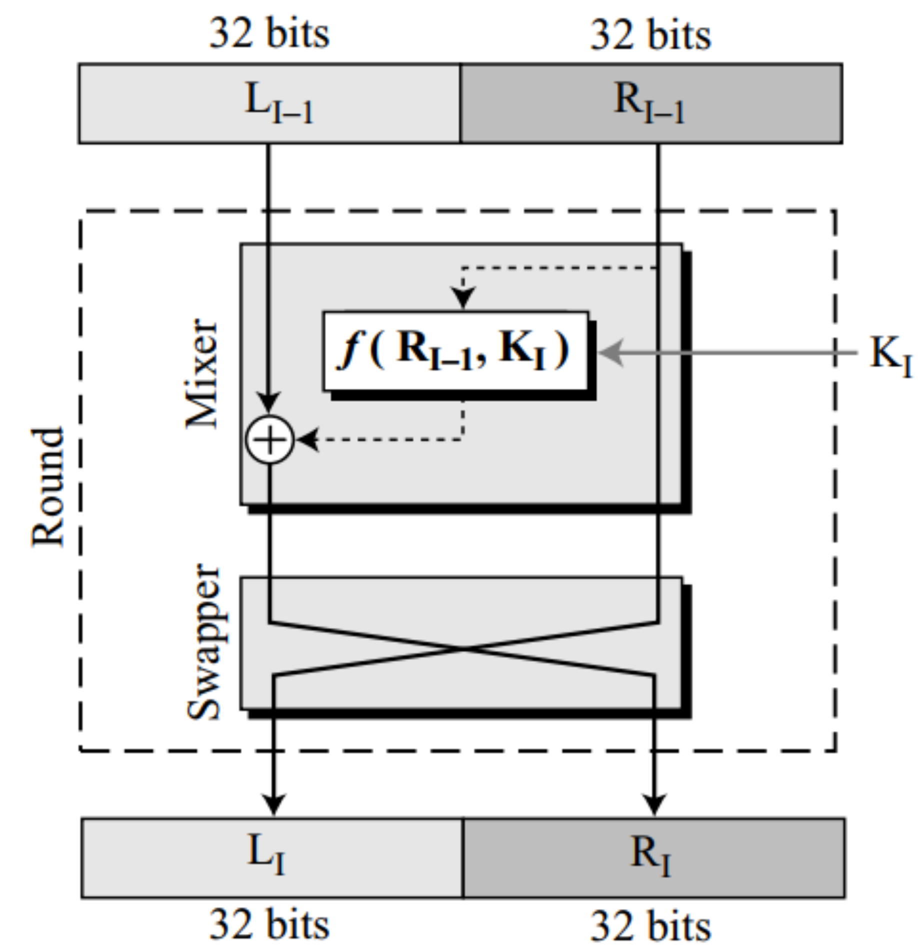


Initial and Final Permutation (IP and FP)

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

Rounds

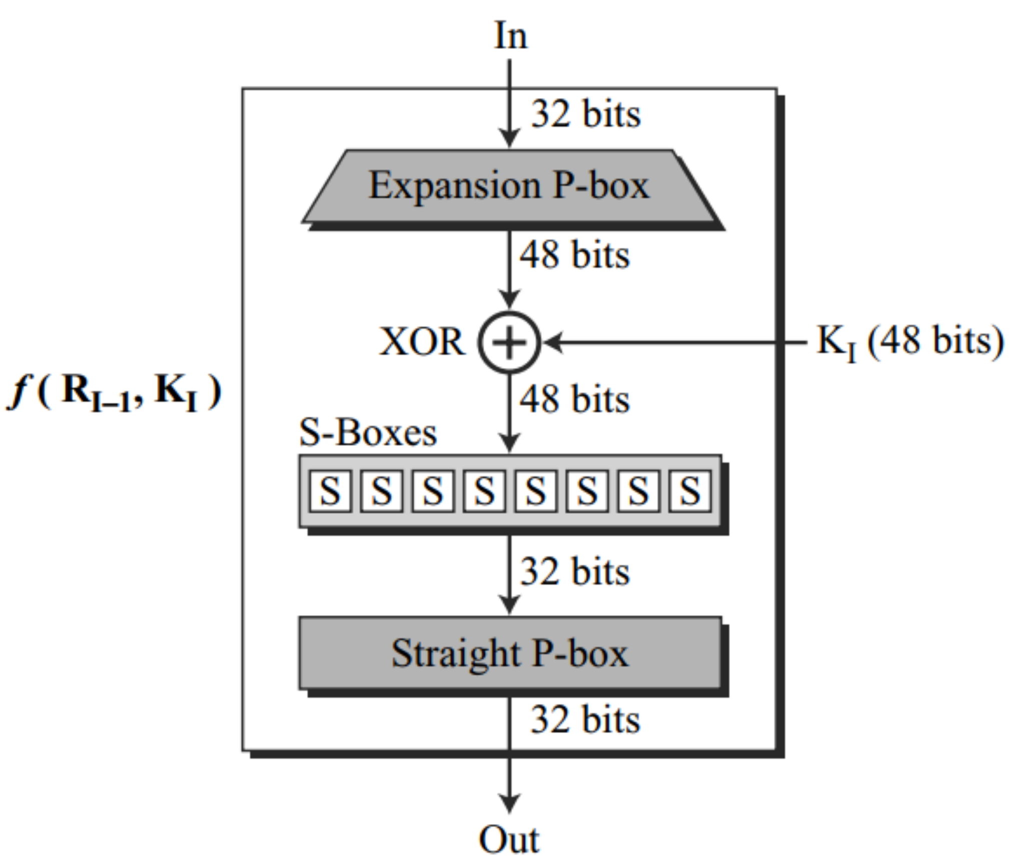
- DES uses 16 rounds.
- Each round of DES is a Feistel cipher
- The round takes LI−1 and RI−1 from previous round (or the initial permutation box) and creates LI and RI, which go to the next round (or final permutation box).
- Each round has **two** cipher elements (mixer and swapper).
- Each of these elements is **invertible**.
- The **swapper** is invertible, which swaps the left half of the text with the right half.
- The mixer is invertible because of the **XOR operation.**
- All noninvertible elements are collected inside the function **f (RI−1, KI)**

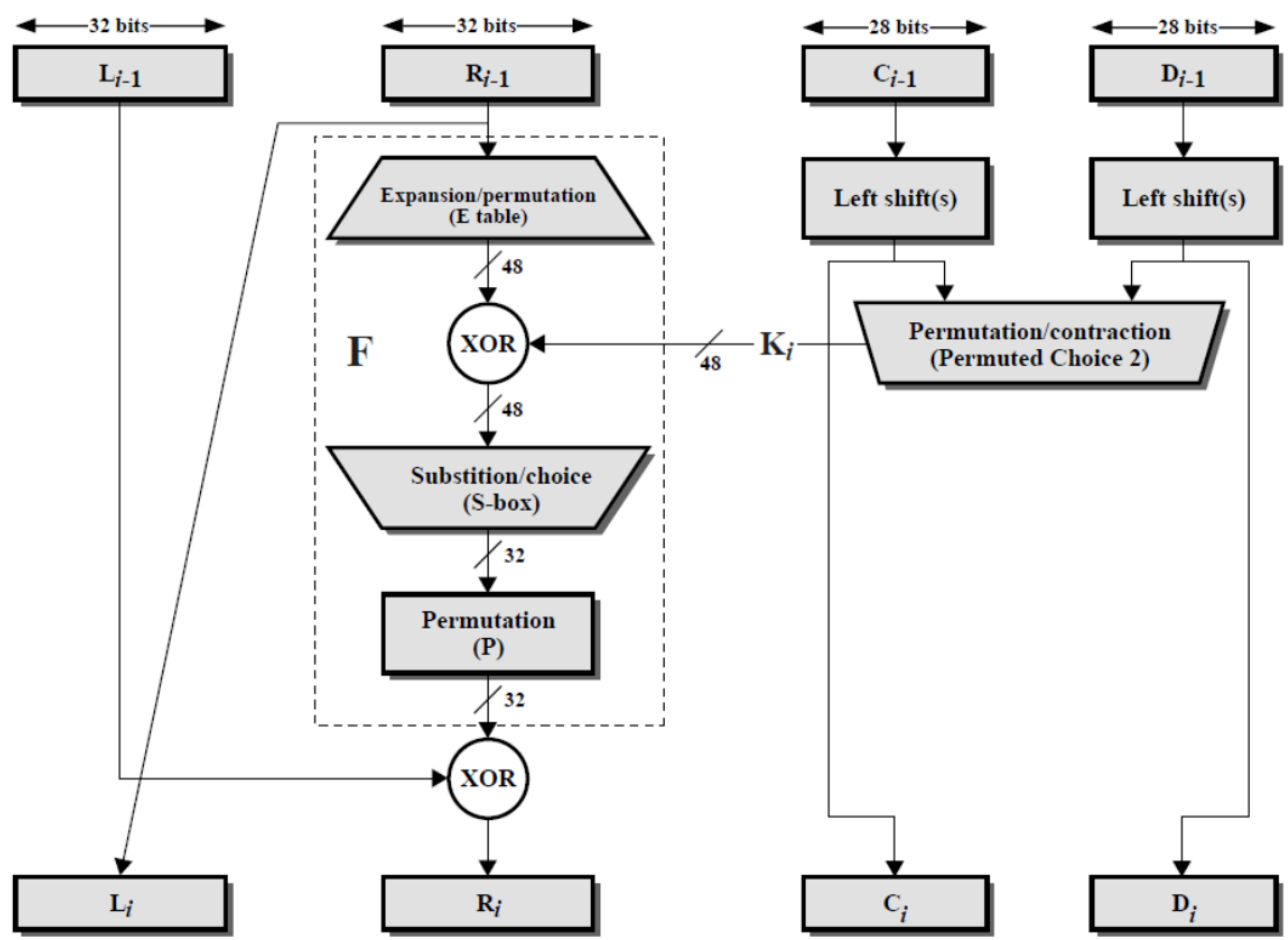


DES Function

- The heart of DES is the DES function.
- The DES function applies a **48-bit key** to the **rightmost 32 bits (RI−1) to produce a 32-bit output.**
- This function is made up of four sections:
    - an expansion P-box,
    - a whitener (that adds key),
    - a group of S-boxes, and
    - a straightP-box

DES Function

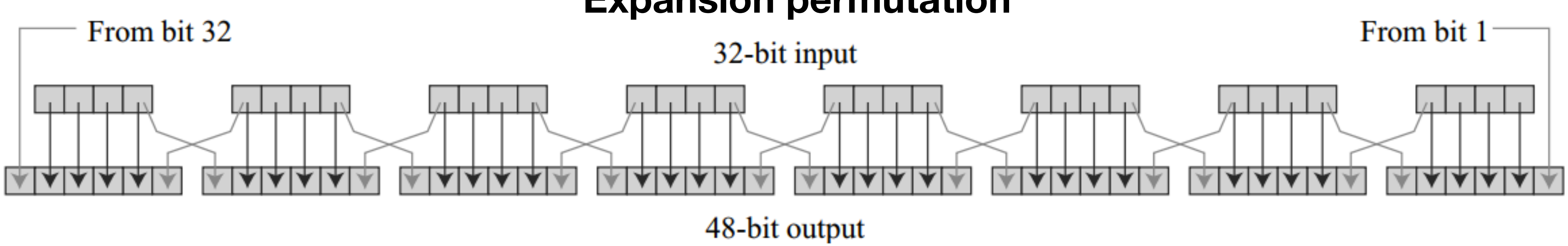




Expansion P-box

- Since RI−1 is a 32-bit input and KI is a 48-bit key, we first need to expand RI−1 to 48 bits.
- RI−1 is divided into 8 4-bit sections.
- Each 4-bit section is then expanded to 6 bits.
- This expansion permutation follows a predetermined rule.
- For each section, input bits 1, 2, 3, and 4 are copied to output bits 2, 3, 4, and 5, respectively. Output bit 1 comes from bit 4 of the previous section;
- output bit 6 comes from bit 1 of the next section.
- If sections 1 and 8 can be considered adjacent sections, the same rule applies to bits 1 and 32.
- Following figure shows the input and output in the expansion permutation.

Expansion permutation



Expansion P-box

- Although the relationship between the input and output can be defined mathematically, DES uses a table to define this P-box.
- Note that the number of output ports is 48, but the value range is only 1 to 32.
- Some of the inputs go to more than one output.

Expansion P-box table

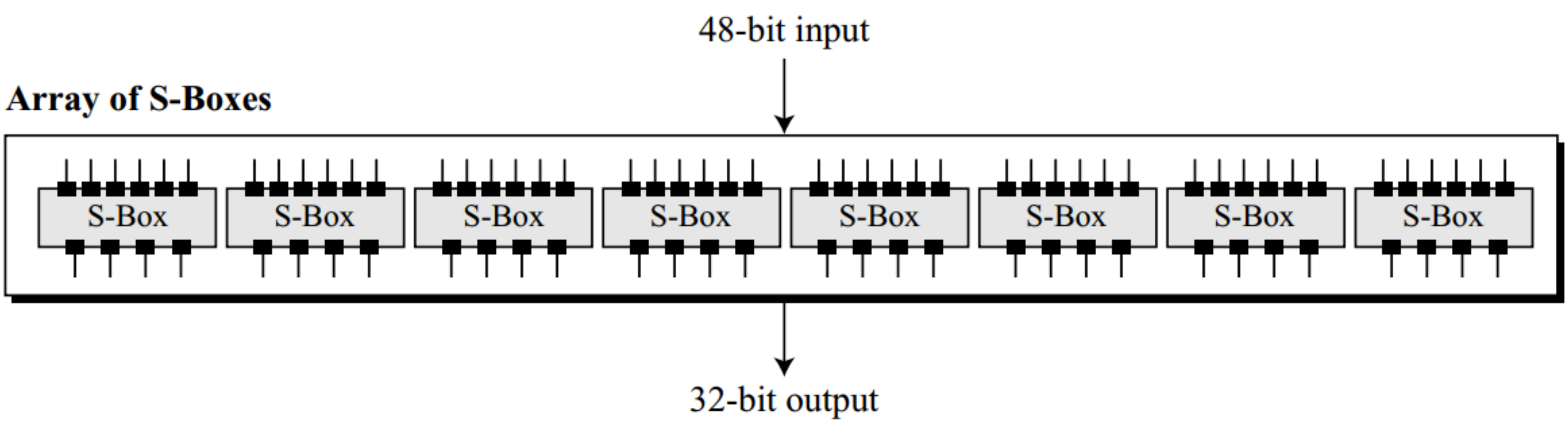
32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	01

Whitener (XOR)

- After the expansion permutation, DES uses the XOR operation on the expanded right section and the round key.
- Both the right section and the key are 48-bits in length.
- The round key is used only in this operation

S-Boxes

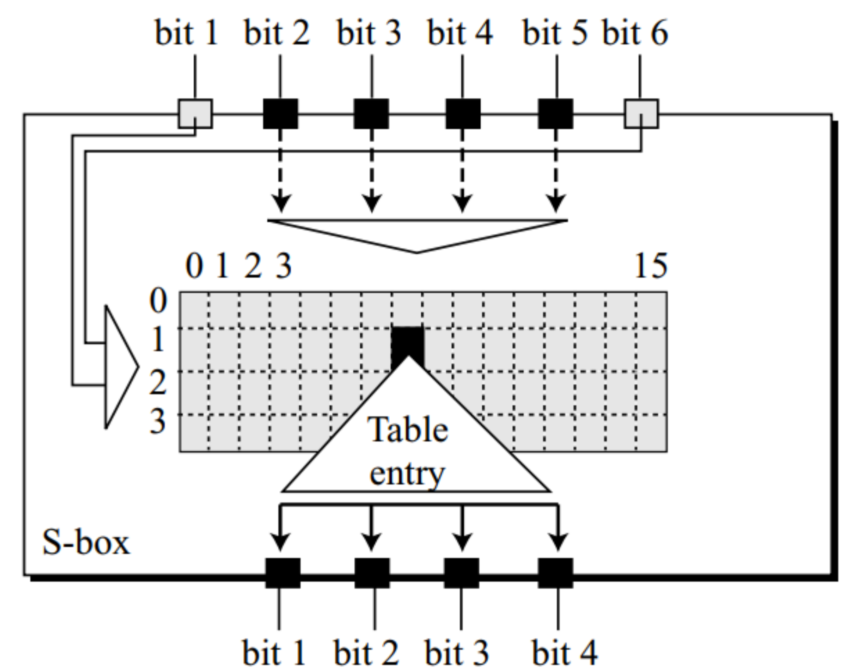
- The S-boxes do the real mixing (confusion).
- DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output.
- The 48-bit data from the second operation is divided into eight 6-bit chunks, and each chunk is fed into a box.
- The result of each box is a 4-bit chunk; when these are combined the result is a 32-bit text.



S-Boxes

- The substitution in each box follows a pre-determined rule based on a 4-row by 16-column table.
- The combination of bits 1 and 6 of the input defines one of four rows;
- the combination of bits 2 through 5 defines one of the sixteen columns

S-box rule



S-Boxes

- Because each S-box has its own table, we need eight tables(S1 to S8), to define the output of these boxes.
- The values of the inputs (row number and column number) and the values of the outputs are given as decimal numbers.
- These need to be changed to binary

Sample S-Boxes

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 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

S-Box lookup

- The input to S-box 1 is 100011. What is the output?
- 1st bit and 6th bit is 11 => decimal = 3
- 2,3,4,5 bits is 0001 => decimal = 1

S-Box lookup

- 100011
- First and the sixth bit => 11 in binary, which is 3 in decimal.
- Remaining bits are 0001 in binary => 1 in decimal.
- We look for the value in row 3, column 1in S-box 1 table.
- The result is 12 in decimal => binary is **1100**.

Straight Permutation

- The last operation in the DES function is a straight permutation with a 32-bit input and a 32-bit output.
- The input/output relationship for this operation is shown
- It follows the same general rule as previous permutation tables

Straight permutation table

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

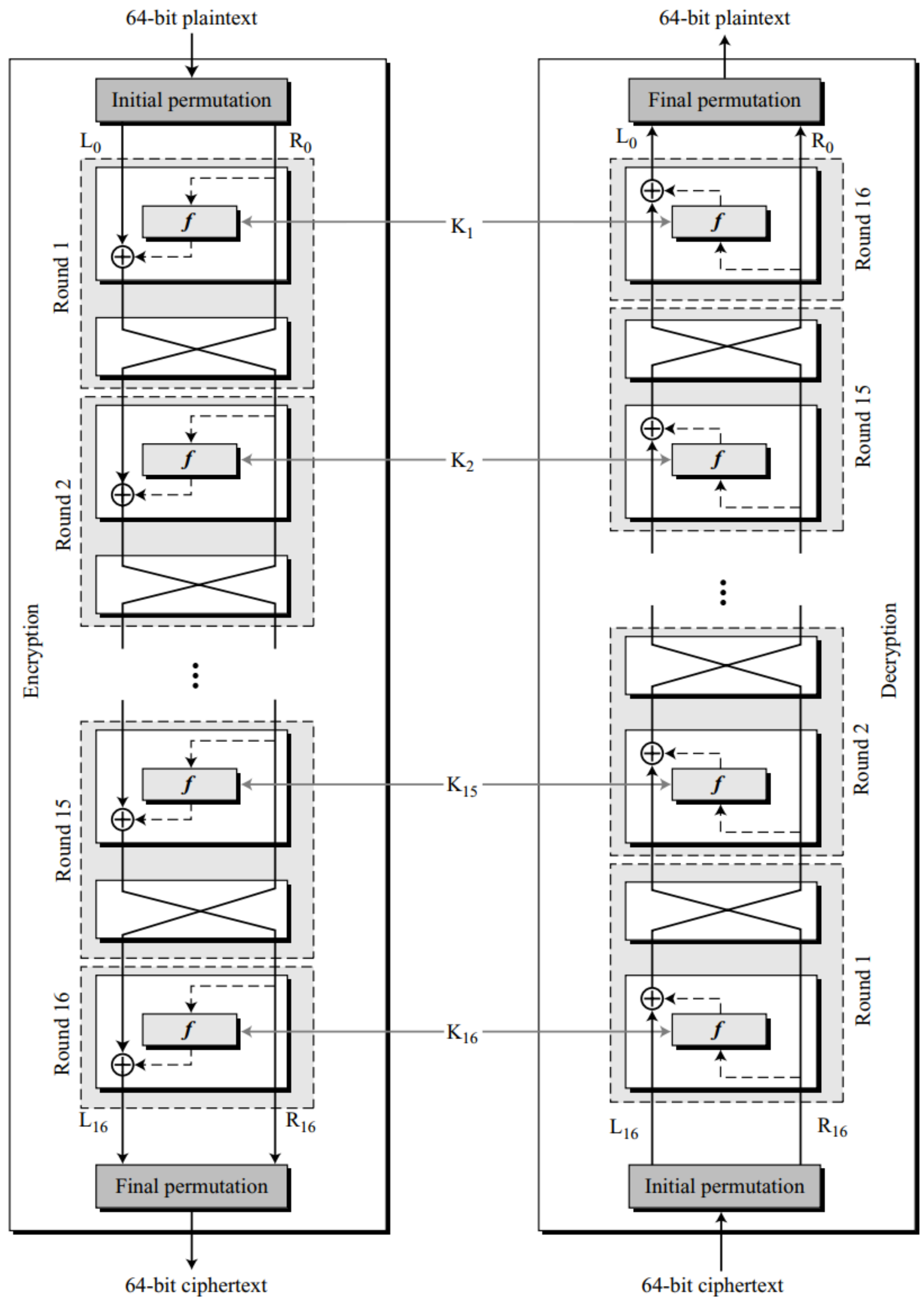
Cipher and Reverse Cipher

- Using **mixers and swappers**, we can create the cipher and reverse cipher, each having 16 rounds.
- The **cipher is used at the encryption** site; the **reverse cipher is used at the decryption** site.
- The whole idea is to make the cipher and the reverse cipher algorithms similar.

Cipher and Reverse Cipher

- **First Approach**
- One approach is to make the last round (round 16) **different** from the others; it **has only a mixer** and **no swapper**.
- Although the rounds are not aligned, the **elements (mixer or swapper) are aligned.**
- Mixer and Swapper is a **self-inverse**
- The **final and initial permutations are also inverses** of each other.
- The left section of the plaintext at the encryption site, L0, is enciphered as L16 at the encryption site;
- L16 at the decryption is deciphered as L0 at the decryption site.
- Same with R0 and R16.

Cipher and Reverse Cipher



Cipher and Reverse Cipher

- **Alternative Approach**

- In the first approach, round 16 is different from other rounds; there is no swapper in this round.

- This is needed to make the last mixer in the cipher and the first mixer in the reverse cipher aligned.

- **We can make all 16 rounds the same by including one swapper to the 16th round and add an extra swapper after that (two swappers cancel the effect of each other).**

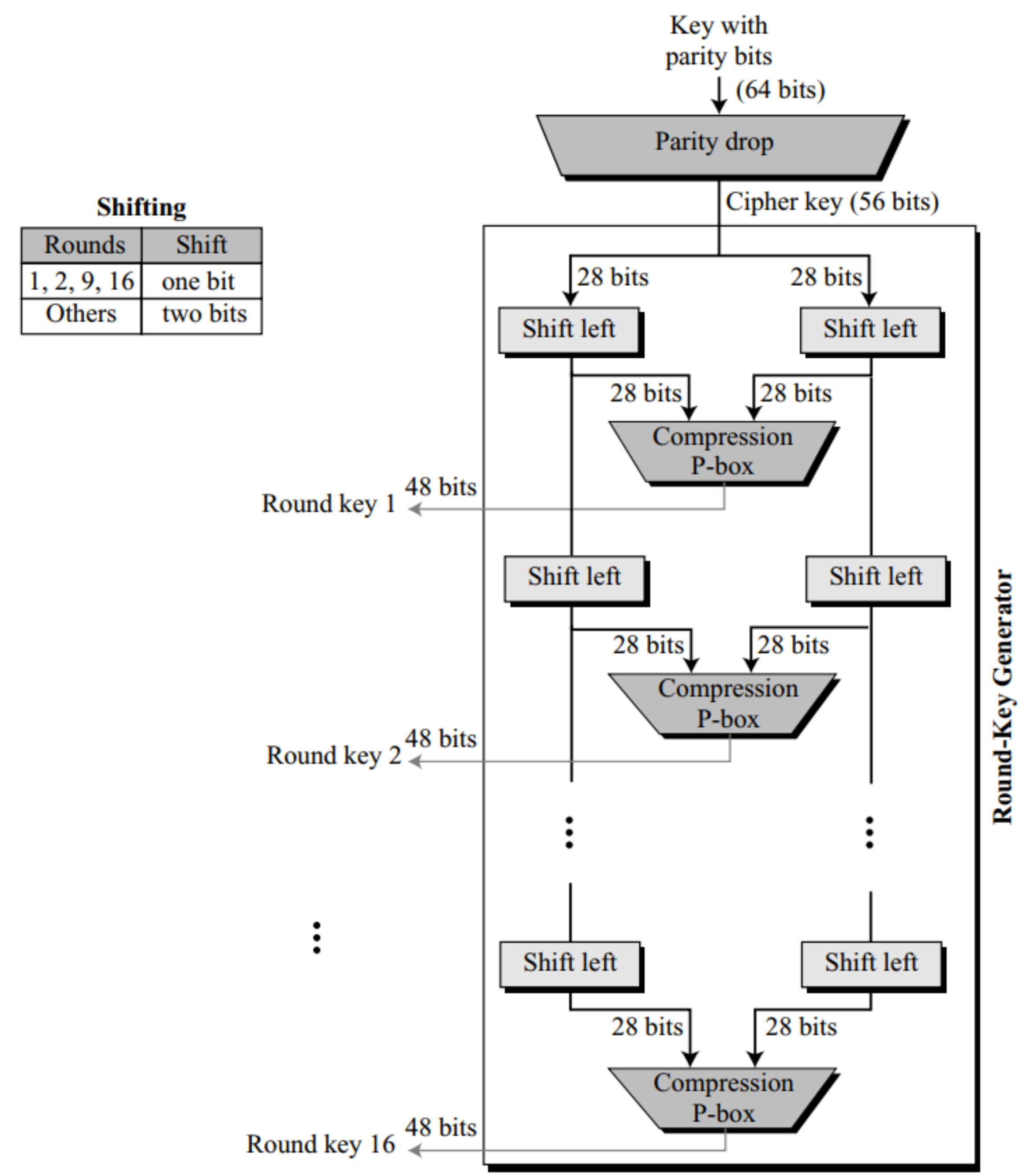
Round Keys

- Round keys (K1 to K16) should be applied in the reverse order.
- At the encryption site, round 1 uses K1 and round 16 uses K16;
- At the decryption site, **round 1 uses K16** and **round 16 uses K1**

Key Generation

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

- The cipher key is normally given as a **64-bit key** in which **8 extra bits are the parity bits**, which are dropped before the actual key-generation process



Key Generation

- **Parity Drop**
  - The preprocess before key expansion is a **compression permutation** called **parity bit drop**.
  - It drops the parity bits (bits 8, 16, 24, 32, …, 64) from the 64-bit key and permutes the rest of the bits according to Parity Drop table
  - The remaining 56-bit value is the **actual cipher key** which is used to generate round keys.

Parity-bit drop table

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

Key Generation

- **Shift Left**
- After the straight permutation, the key is divided into two 28-bit parts.
- Each part is shifted left (circular shift) one or two bits.
- In rounds 1, 2, 9, and 16, shifting is one bit; in the other rounds, it is two bits.
- The two parts are then combined to form a 56-bit part.

Number of bit shifts

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 Generation

- **Compression Permutation**

- The compression permutation (P-box) changes the 56 bits to 48 bits, which are used as a key for a round.

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

Properties - Confusion and Diffusion

- Two desired properties of a block cipher are the avalanche effect(diffusion) and the completeness

- **Avalanche effect (Diffusion)** means a small change in the plaintext (or key) should create a significant change in the ciphertext.

- DES has been proved to be strong with regard to this property

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

- The diffusion and confusion produced by P-boxes and S-boxes in DES, show a very strong **completeness effect.**

Avalanche effect Example

- Two plaintext blocks differ only in 1 bit (the rightmost bit)
- Ciphertext blocks differ in 29 bits.
- This means that changing approximately 1.5 percent of the plaintext creates a change of approximately 45 percent in the ciphertex

Plaintext: 0000000000000000 Key: 22234512987ABB23
Ciphertext: 4789FD476E82A5F1

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

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

DES Weaknesses

- **S-boxes**
- At least three weaknesses are mentioned in the literature for S-boxes.
  - 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.
  - Two specifically chosen inputs to an S-box array can create the same output.
  - It is possible to obtain the same output in a single round by changing bits in only three neighboring S-boxes.

DES Weaknesses

- **P-boxes**
  - One mystery and one weakness were found in the design of P-boxes:
  - It is not clear why the designers of DES used the initial and final permutations; these have no security benefits.
    - In the expansion permutation (inside the function), the first and fourth bits of every 4-bit series are repeated

Weakness in the Cipher Key

- **Key Size**
  - 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

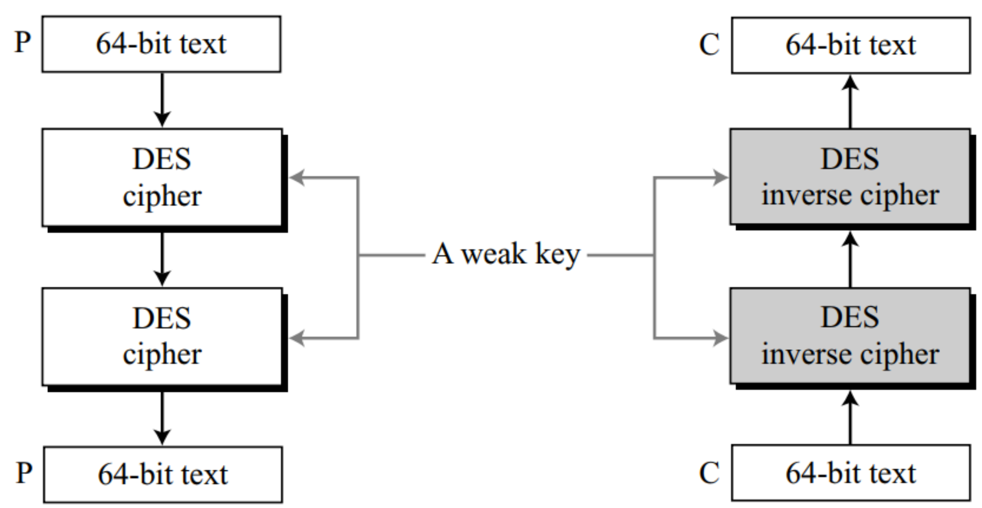
Weakness in the Cipher Key

- **Weak Keys**
  - Four out of 2^{56} possible keys are called **weak keys.**
  - A weak key is the one that, after parity drop operation , **consists either of all 0s, all 1s, or half 0s and half 1s.**
  - The round keys created from any of these weak keys are the **same** and have the **same pattern** as the cipher key.
  - For example, the sixteen round keys created from the **first key is all made of 0s**; the one from the second is made of **half 0s and half 1s.**
  - Because Key-generation algorithm first divides the cipher key into two halves - Shifting or permutation of a block does not change the block if it is made of all 0s or all 1s.

Weakness in the Cipher Key

- **Double encryption with Weak Keys**
  - If we encrypt a block with a weak key and subsequently **encrypt the result with the same weak** key, we get the **original** block.
  - The process creates the **same original block** if we **decrypt** the block twice.
  - In other words, **each weak key is the inverse of itself** **Ek(Ek(P)) = P**

Weakness in the Cipher Key



Weak Keys

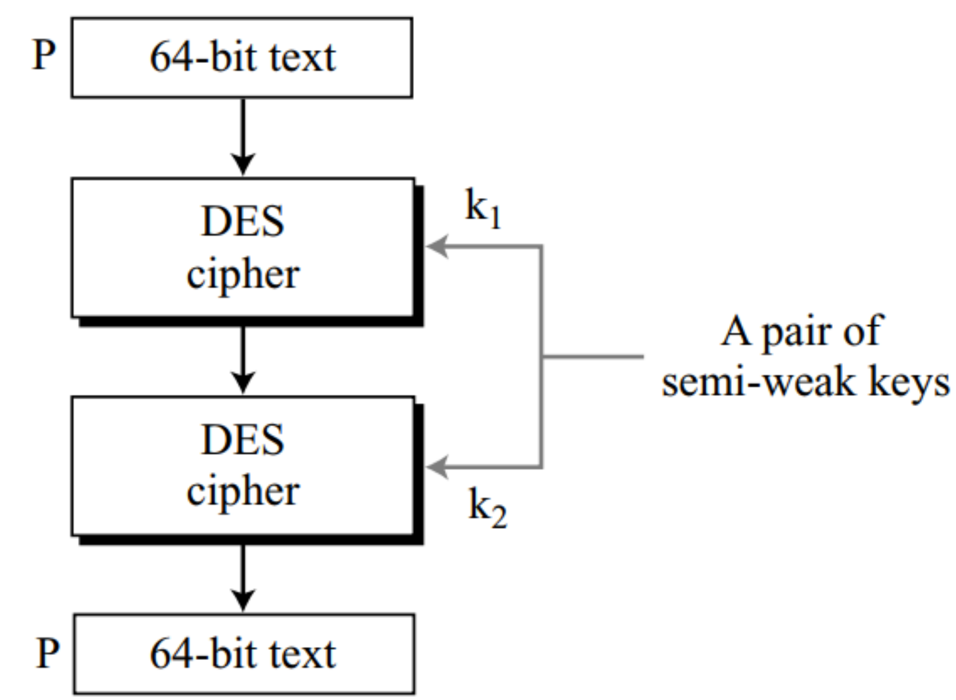
Keys before parities drop (64 bits)	*Actual key (56 bits)*
0101 0101 0101 0101	0000000 0000000
1F1F 1F1F 0E0E 0E0E	0000000 FFFFFFF
E0E0 E0E0 F1F1 F1F1	FFFFFFF 0000000
FEFE FEFE FEFE FEFE	FFFFFFF FFFFFFF

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

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

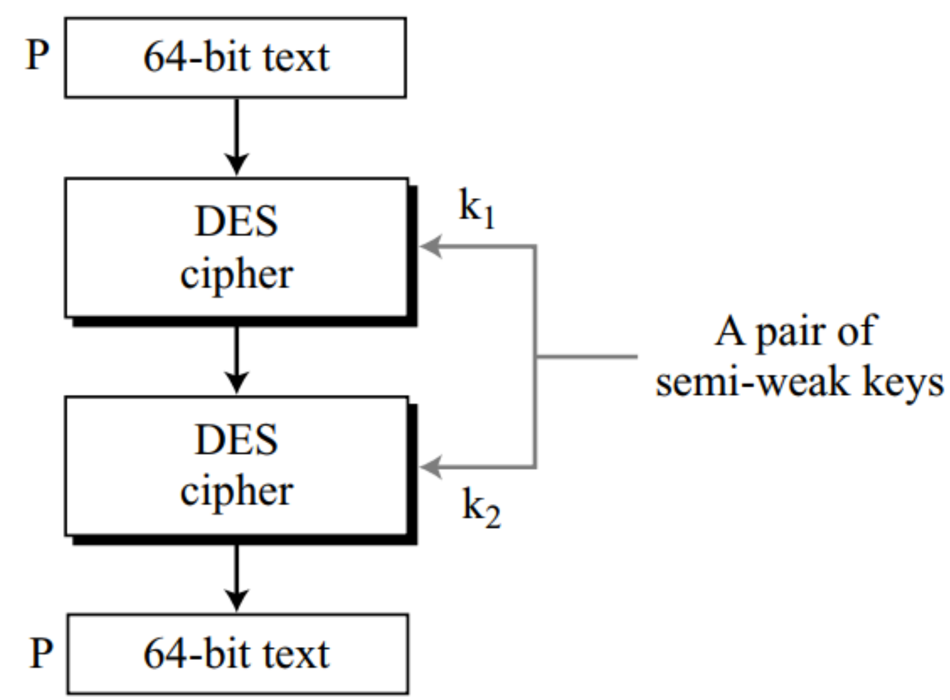
Weakness in the Cipher Key

- **Semi-weak Keys**
  - There are six key pairs that are called **semi-weak keys**.
  - A semi-weak key creates only **two different round keys** and **each of them** is **repeated eight times**.
  - The **round keys created from each pair are the same with different orders**.
  - This means that the keys are inverses of each other



Weakness in the Cipher Key

- **Possible Weak Keys**
  - There are also 48 keys that are called **possible weak keys.**
  - A **possible weak key** is a key that creates only four distinct round keys
  - the **sixteen round keys** are divided into **four groups** and each group is made of **four equal round keys**.



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.
- Eve can use only half of the possible keys (255) to perform brute-force attack.

Key Clustering

- **Key clustering** refers to the situation in which two or more different keys can create the **same ciphertext from the same plaintext.**

- Each pair of the semi-weak keys is a key cluster

- No more clusters have been found for the DES.

MULTIPLE DES

- The major criticism of DES regards its key length.
- With available technology and the possibility of parallel processing, a brute-force attack on DES is feasible.
- One solution to improve the security of DES is to abandon DES and design a new cipher - AES.
- Another solution is to use multiple (cascaded) instances of DES with multiple keys;

Double DES

- The first approach is to use **double DES (2DES)**.
- In this approach, we use two instances of DES ciphers for encryption and two instances of reverse ciphers for decryption.
- Each instance uses a different key, which means that the size of the key is now **doubled (112 bits)**.
- Double DES is vulnerable to a known-plain text attack

Triple DES

- To improve the security of DES, triple DES (3DES) was proposed.

- This uses three stages of DES for encryption and decryption.

- Two versions of triple DES are in use today:

    - triple DES with two keys and

    - triple DES with three keys

SECURITY OF DES

- Three interesting attacks on DES are :
  - brute-force,
  - differential cryptanalysis, and
  - linear cryptanalysis.

Advanced Encryption Standard (AES)

- The Advanced Encryption Standard (AES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST) in December 2001

- The criteria defined by NIST for selecting AES fall into three areas: security, cost, and implementation

- AES is a non-Feistel cipher that encrypts and decrypts a data block of **128 bits.**

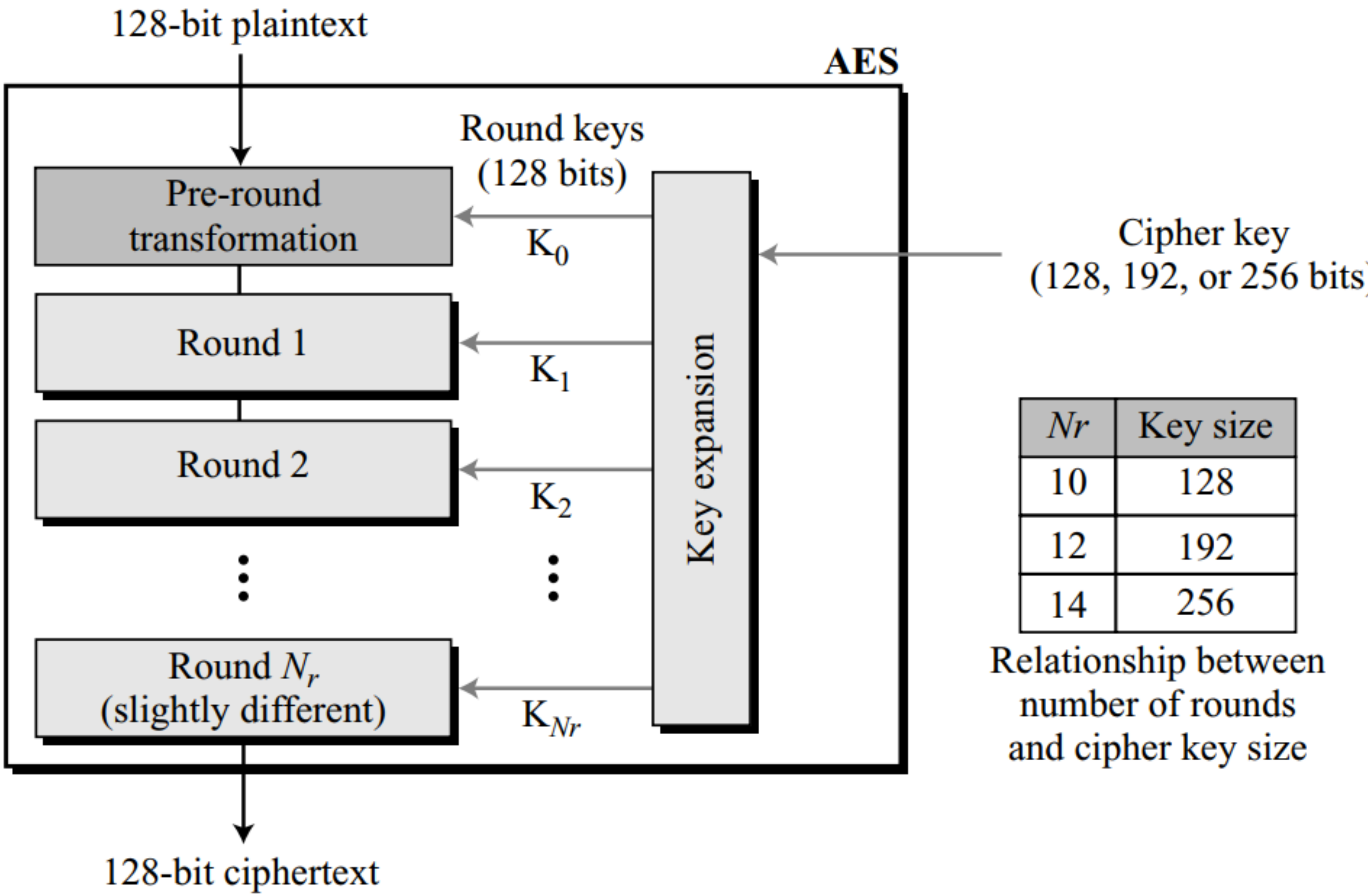
- It uses 10, 12, or 14 rounds.

- The key size, which can be **128, 192, or 256 bits,** depends on the number of rounds.

Advanced Encryption Standard (AES)

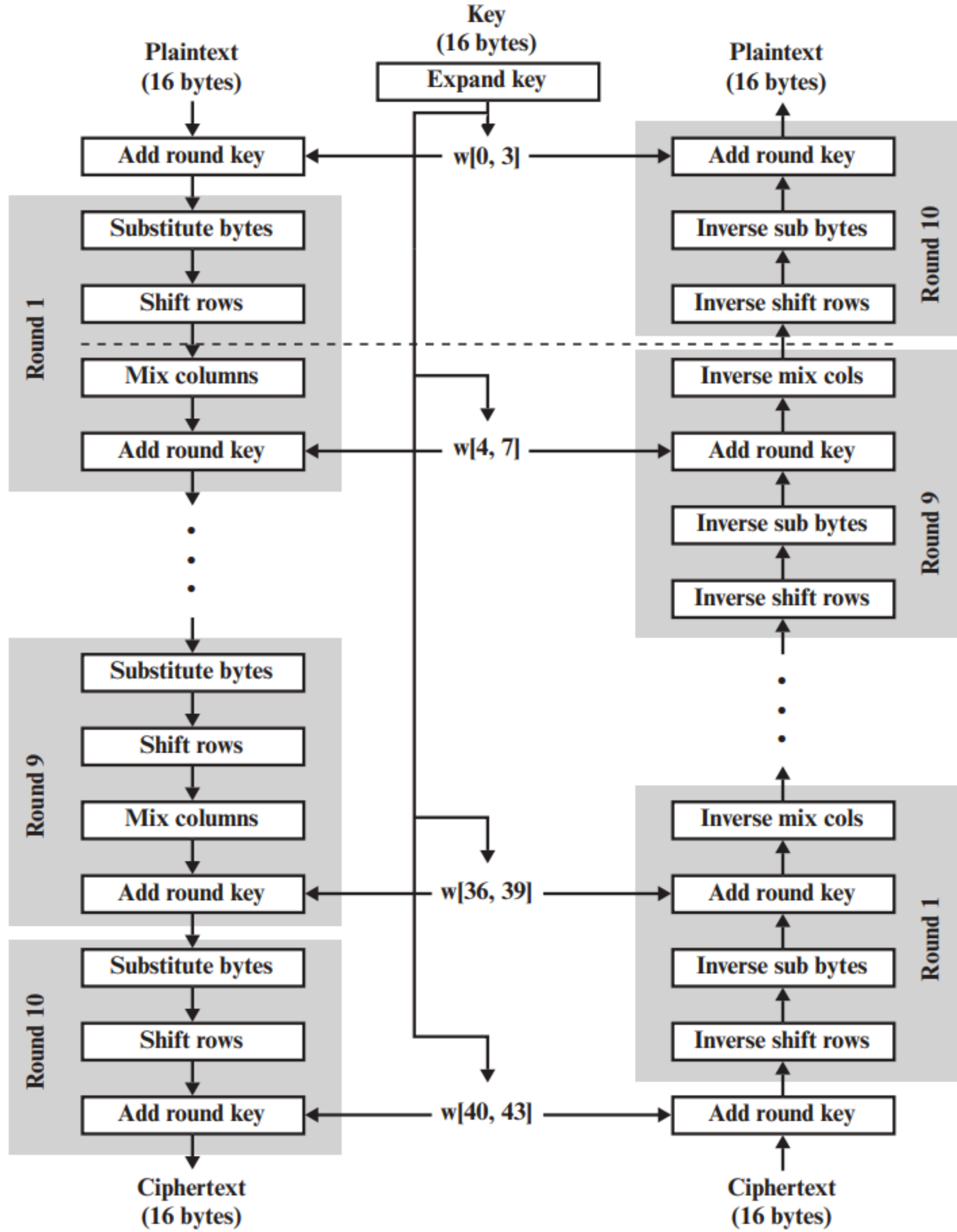
- The round keys are applied in the reverse order
- The figure also shows the relationship between the number of rounds and the key size,
- We can have three different AES versions based on **key size** - **AES-128, AES-192, and AES-256.**
- The round keys, which are created by the key-expansion algorithm are always 128 bits, the same size as the plaintext or ciphertext block

General design of AES encryption cipher



Nr	Key size
10	128
12	192
14	256

Relationship between number of rounds and cipher key size



(a) Encryption (b) Decryption