

GLOBAL  
EDITION

# Cryptography and Network Security

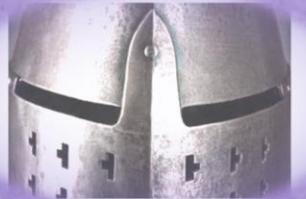
*Principles and Practice*

SEVENTH EDITION

William Stallings



Pearson



# *Chapter 4*

---

## Block Ciphers and the Data Encryption Standard

# Stream Cipher

Encrypts a digital data stream one bit or one byte at a time

## Examples:

- Autokeyed Vigenère cipher
- Vernam cipher

In the ideal case, a one-time pad version of the Vernam cipher would be used, in which the keystream is as long as the plaintext bit stream

If the cryptographic keystream is random, then this cipher is unbreakable by any means other than acquiring the keystream

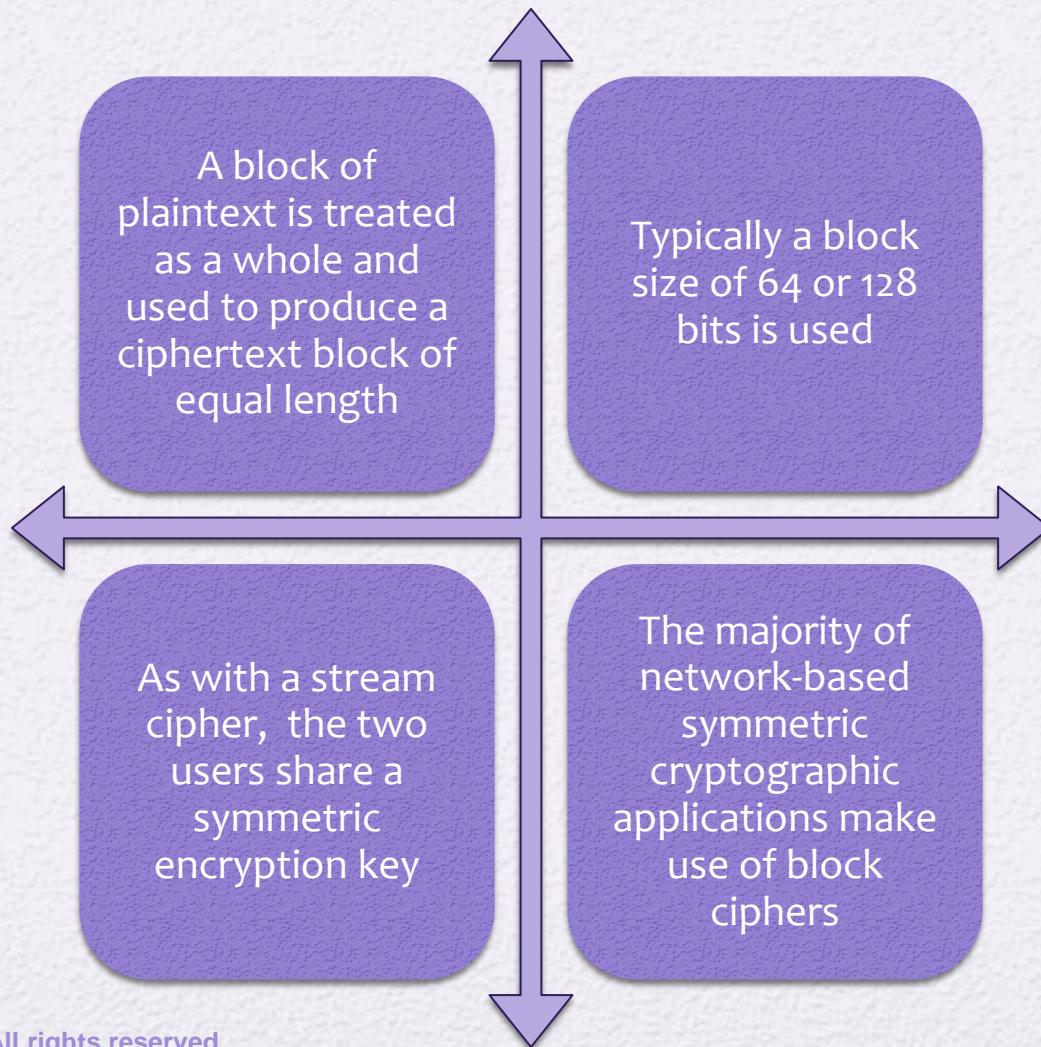
- Keystream must be provided to both users in advance via some independent and secure channel
- This introduces insurmountable logistical problems if the intended data traffic is very large

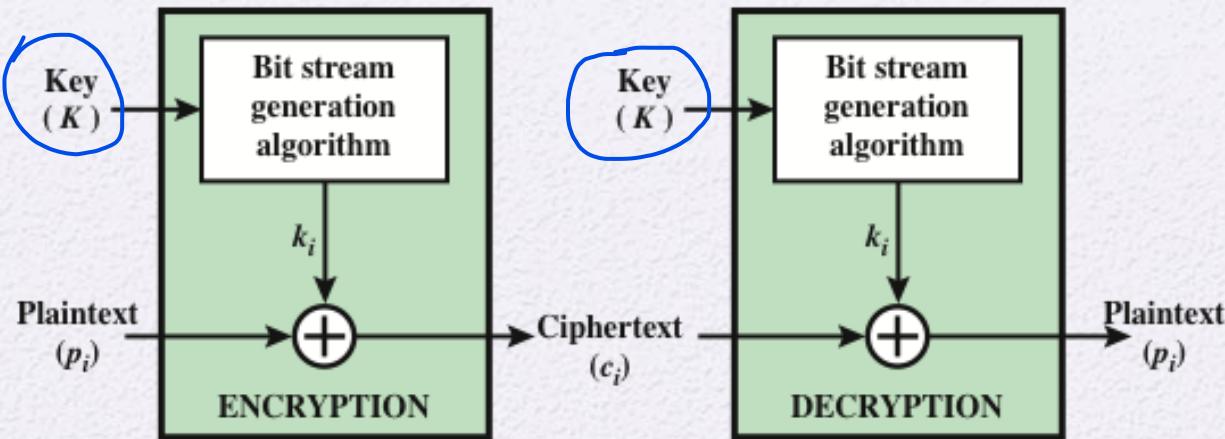
For practical reasons the bit-stream generator must be implemented as an algorithmic procedure so that the cryptographic bit stream can be produced by both users

It must be computationally impractical to predict future portions of the bit stream based on previous portions of the bit stream

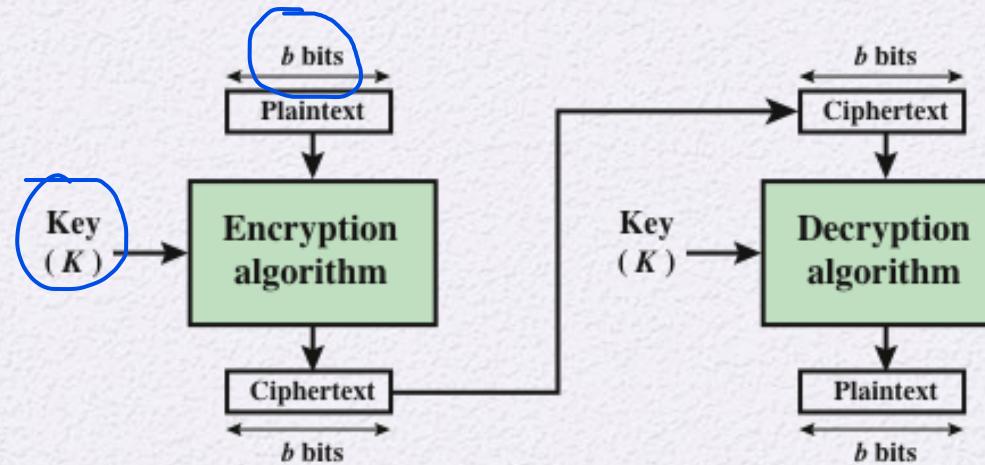
The two users need only share the generating key and each can produce the keystream

# Block Cipher





(a) Stream Cipher Using Algorithmic Bit Stream Generator



(b) Block Cipher

**Figure 4.1 Stream Cipher and Block Cipher**

Ideal block cipher:

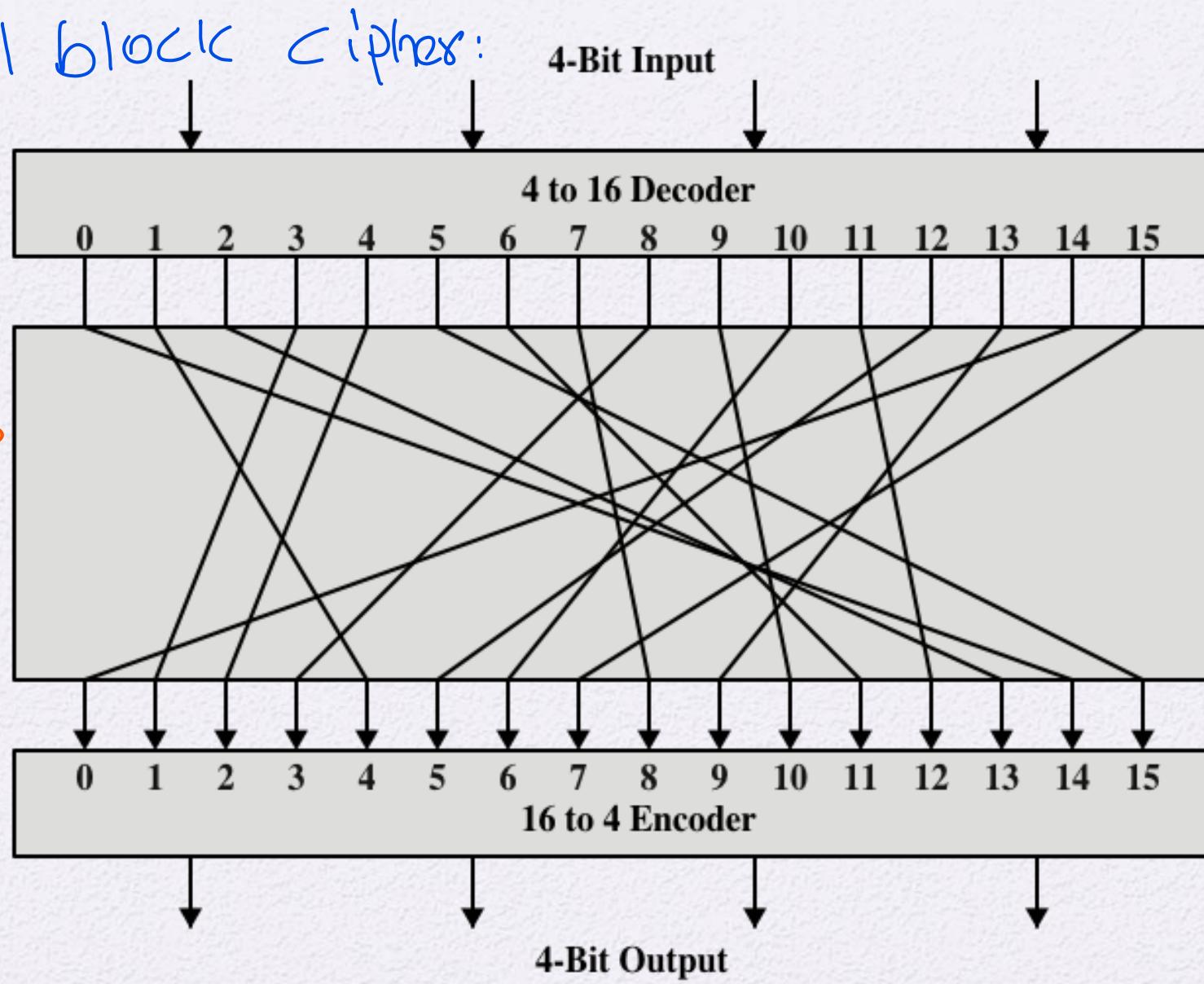


Figure 4.2 General  $n$ -bit- $n$ -bit Block Substitution (shown with  $n = 4$ )

Table 4.1

Encryption and Decryption Tables for Substitution Cipher of Figure

4.2

Plaintext	Ciphertext
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

Ciphertext	Plaintext
0000	1110
0001	0011
0010	0100
0011	1000
0100	0001
0101	1100
0110	1010
0111	1111
1000	0111
1001	1101
1010	1001
1011	0110
1100	1011
1101	0010
1110	0000
1111	0101

# Difficulties: Ideal Block Cipher

Netflix Pwle

- Smaller block sizes makes it vulnerable to statistical analysis
  - **Solution:** Larger block sizes
- Larger block sizes not feasible from an implementation and performance point of view.
- Key length an important aspect
  - 64-bit block to **thwart** statistical attacks
  - Key length for 64-bit blocks?

→ Ideal size after experiment

# Feistel Cipher

- Feistel 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

Substitutes  
with other  
letter

## Permutation

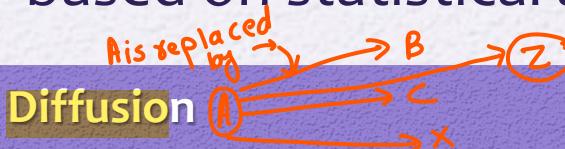
- No elements are added or deleted or replaced in the sequence, rather the order in which the elements appear in the sequence is changed

Change the  
Sequence.

- 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

# Diffusion and Confusion

- 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



- The statistical structure of the plaintext is dissipated into long-range statistics of the ciphertext
- This is achieved by having each plaintext digit affect the value of many ciphertext digits

## Confusion Secure & CipherText (Statistics pattern) & Key Complex

- Seeks to make the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible
- Even if the attacker can get some handle on the statistics of the ciphertext, the way in which the key was used to produce that ciphertext is so complex as to make it difficult to deduce the key

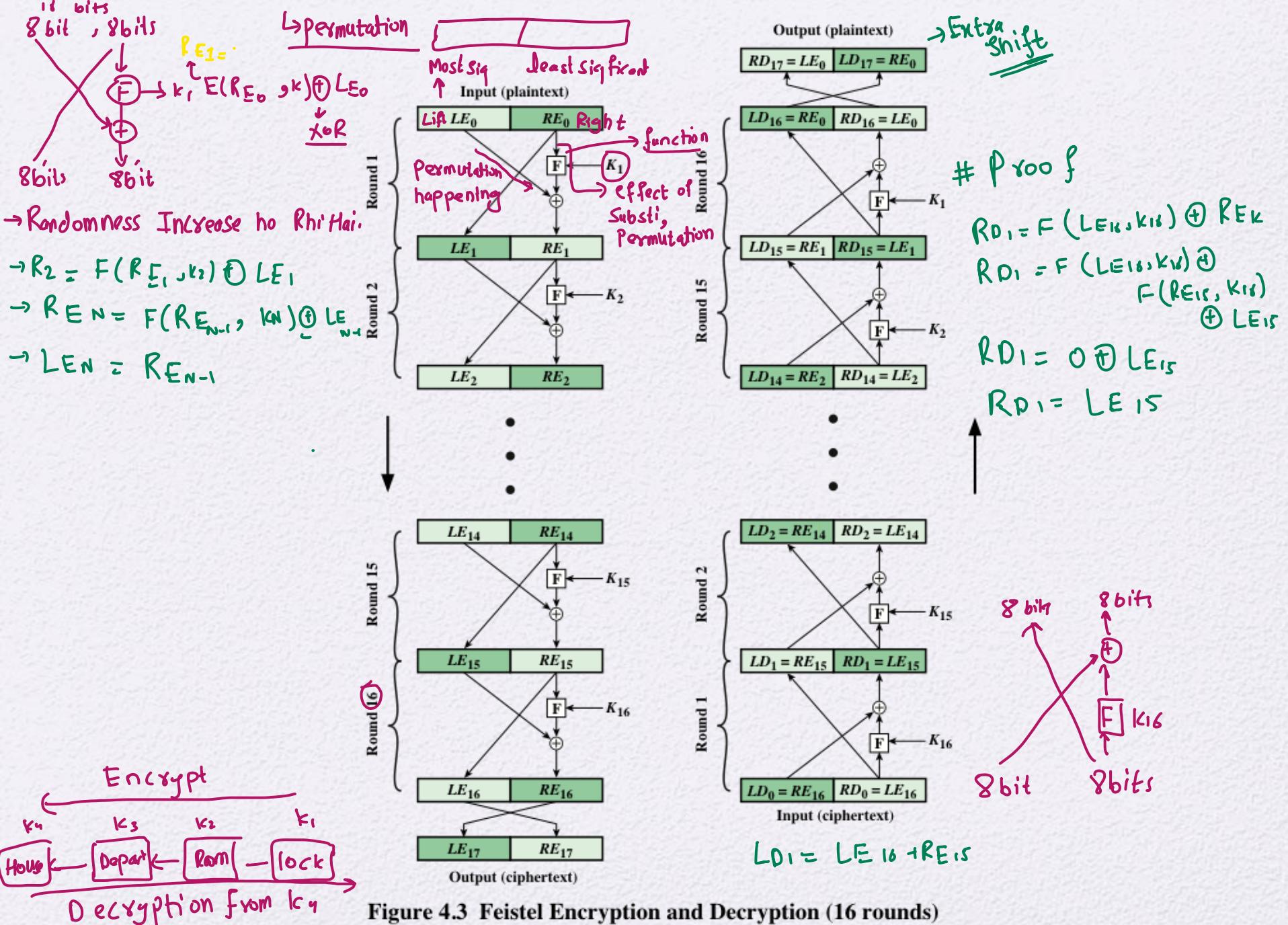


Figure 4.3 Feistel Encryption and Decryption (16 rounds)

# Feistel Cipher Design Features

- ✓ • Block size
  - Larger block sizes mean greater security but reduced encryption/decryption speed for a given algorithm
- ✓ • Key size
  - Larger key size means greater security but may decrease encryption/decryption speeds
- ✓ • Number of rounds
  - The essence of the Feistel cipher is that a single round offers inadequate security but that multiple rounds offer increasing security
- ✓ • Subkey generation algorithm
  - Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis
- Round function F ✓
  - Greater complexity generally means greater resistance to cryptanalysis
- Fast software encryption/decryption
  - In many cases, encrypting is embedded in applications or utility functions in such a way as to preclude a hardware implementation; accordingly, 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 and therefore develop a higher level of assurance as to its strength

# Feistel Example

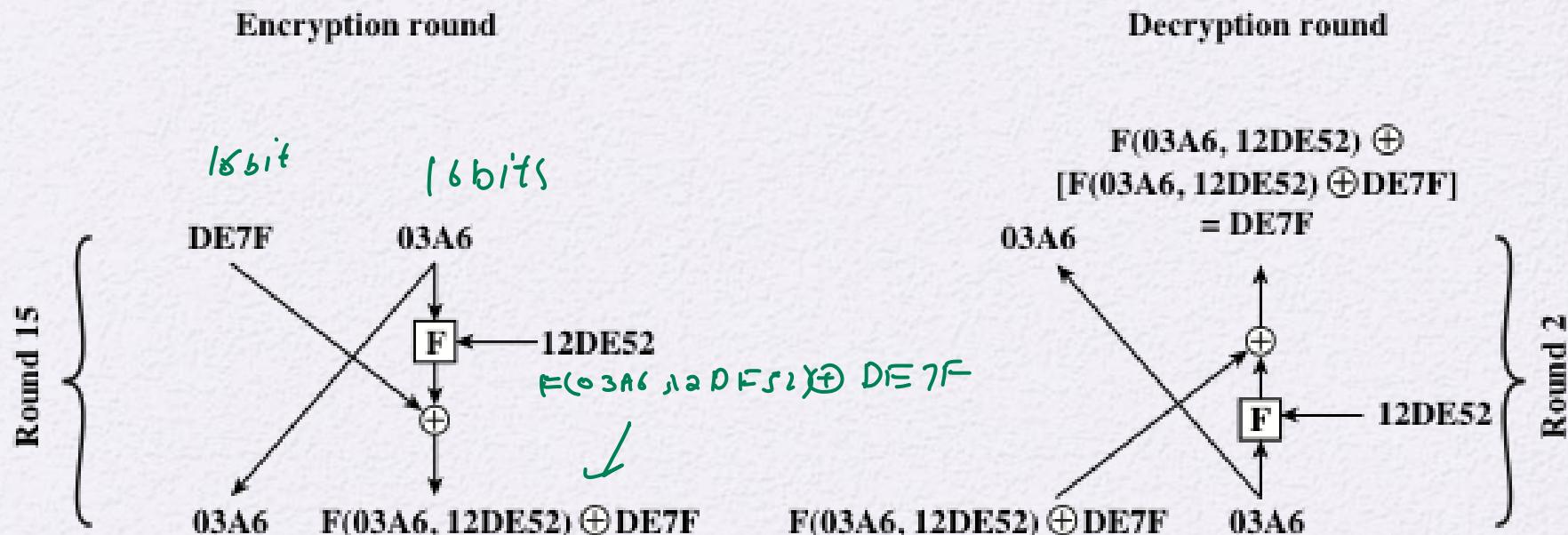


Figure 4.4 Feistel Example

# Data Encryption Standard (DES)

- Issued in 1977 by the National Bureau of Standards (now NIST) as Federal Information Processing Standard 46
- Was the most widely used encryption scheme until the introduction of the Advanced Encryption Standard (AES) in 2001
- Algorithm itself is referred to as the Data Encryption Algorithm (DEA)
  - Data are encrypted in 64-bit blocks using a 56-bit key
  - The algorithm transforms 64-bit input in a series of steps into a 64-bit output
  - The same steps, with the same key, are used to reverse the encryption

8 bits for parity → Check for Data Integrity  
↳ may be  
↳ odd parity  
↳ even parity

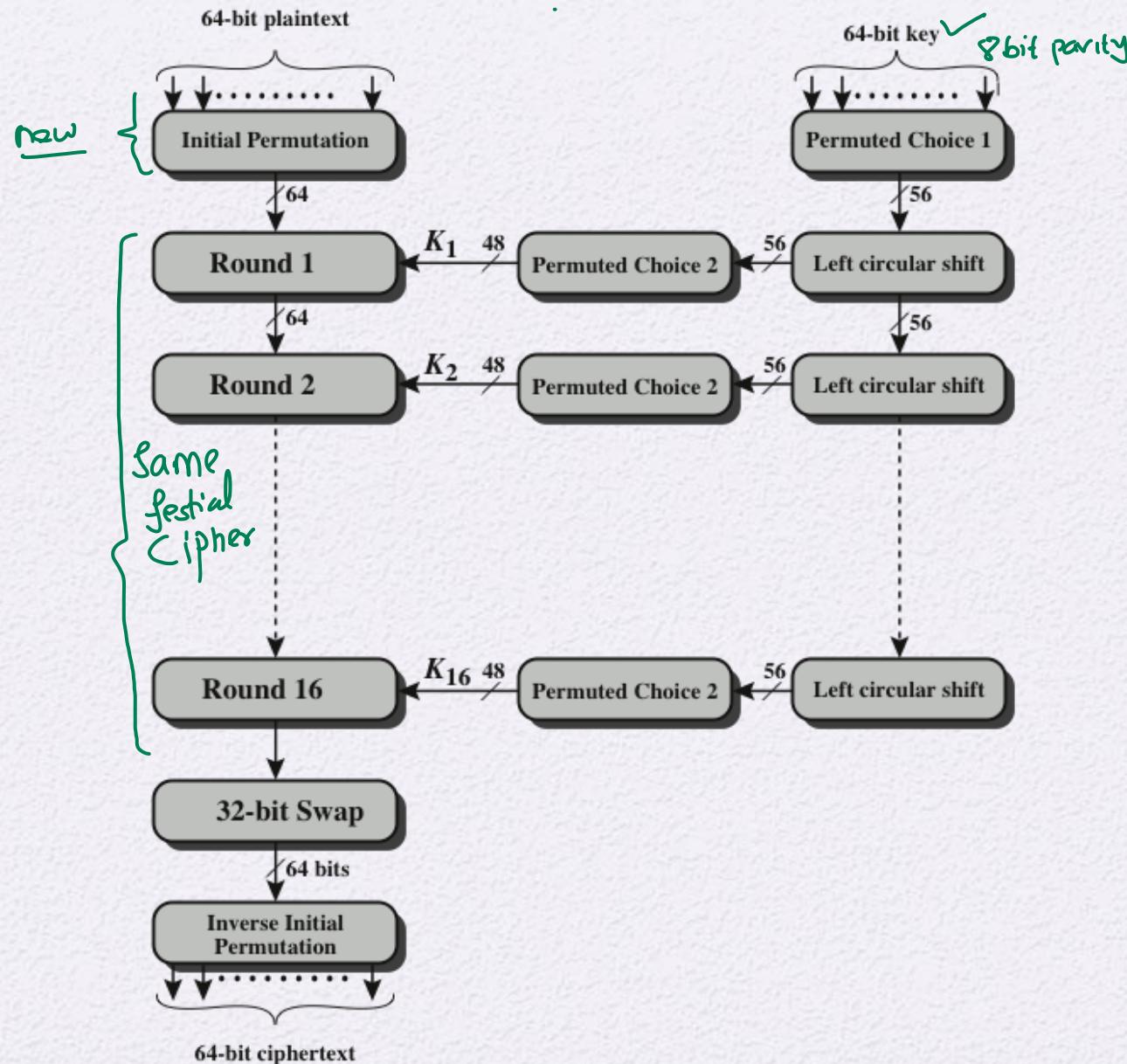
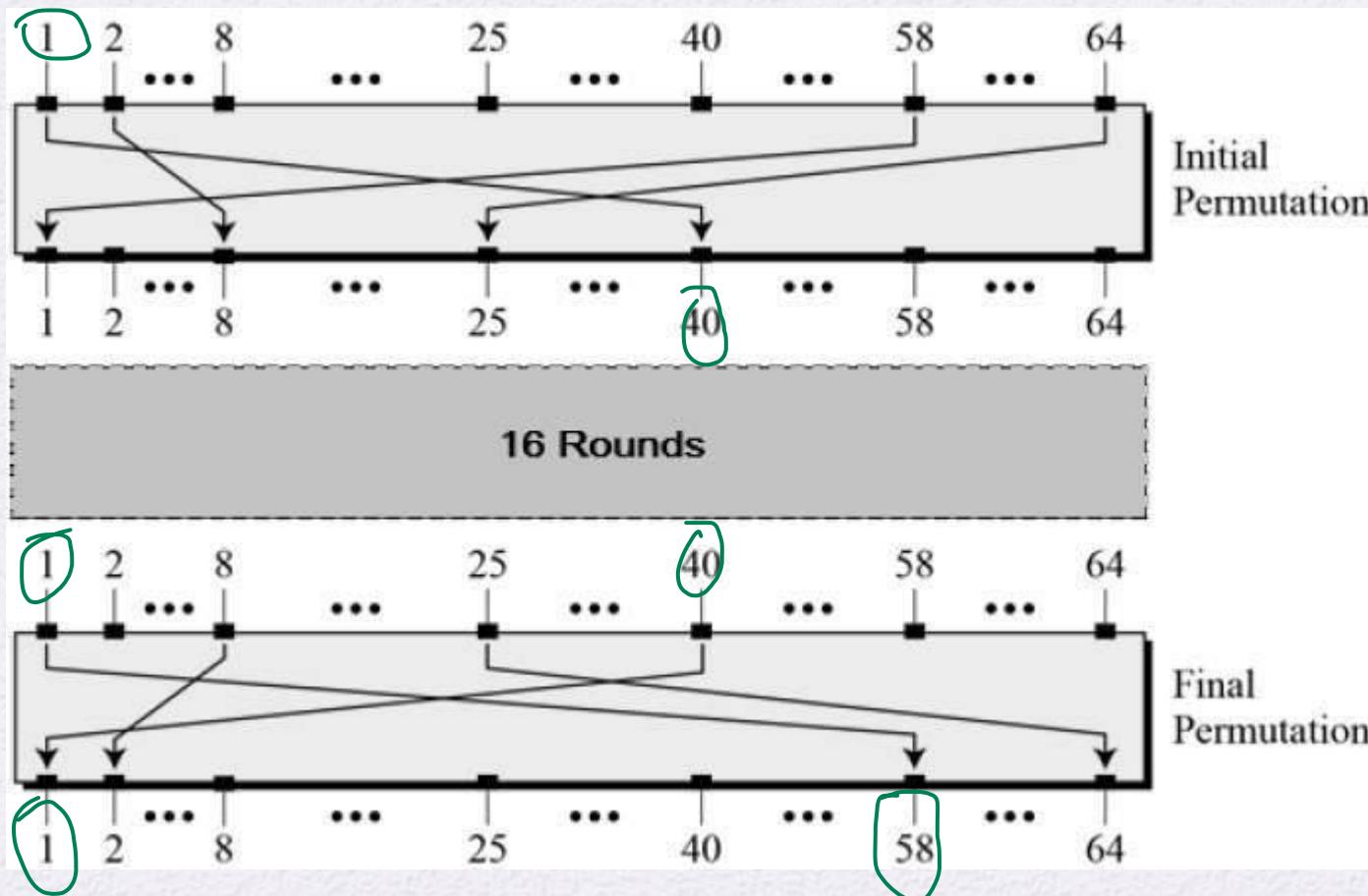


Figure 4.5 General Depiction of DES Encryption Algorithm

# Initial and final permutation

for Randomness

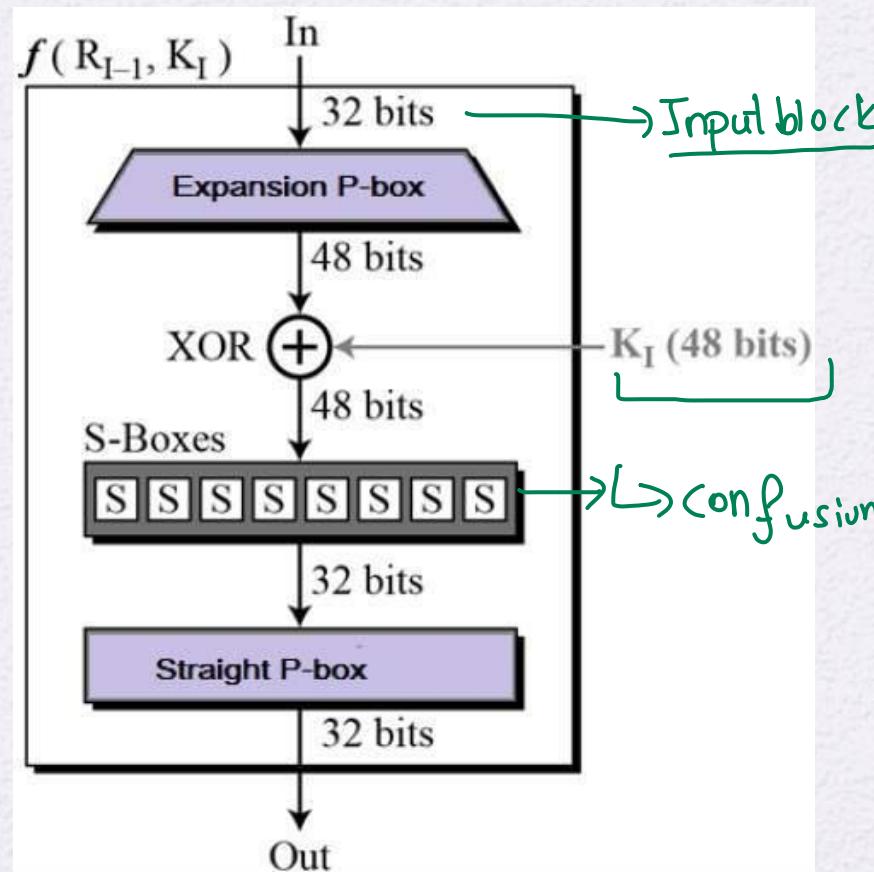
↳ Diffusion & randomness:



# Round Function

F

- Applied to the right most 32-bits.



0000

0001

1011

1100

1101

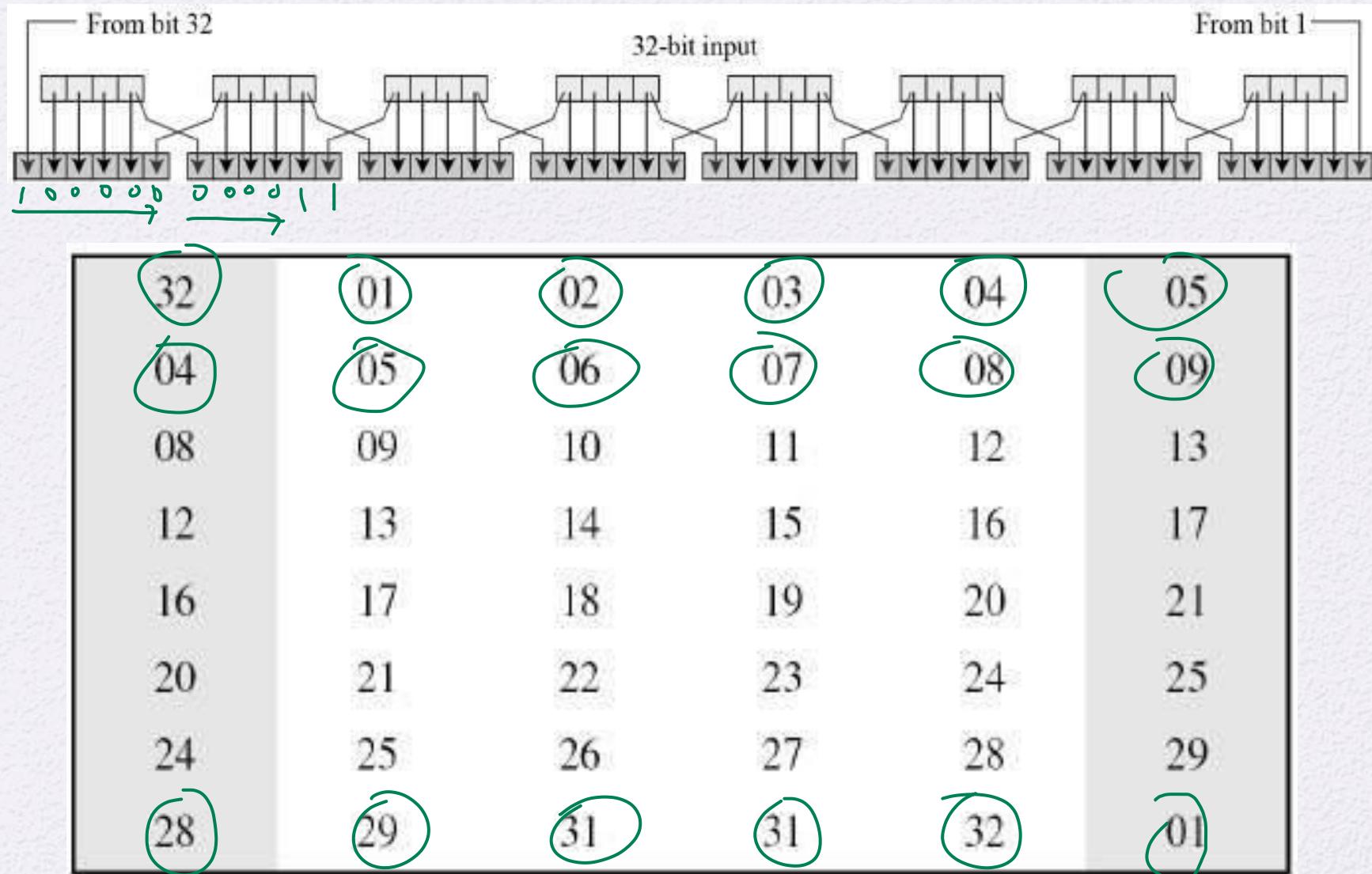
1111

0000

10 11

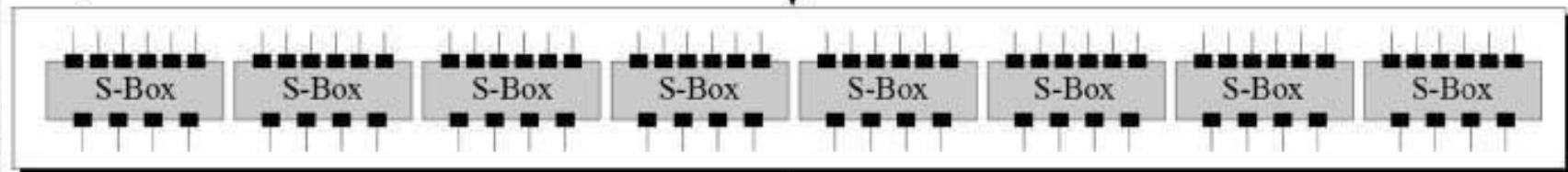


## Expansion Permutation Box

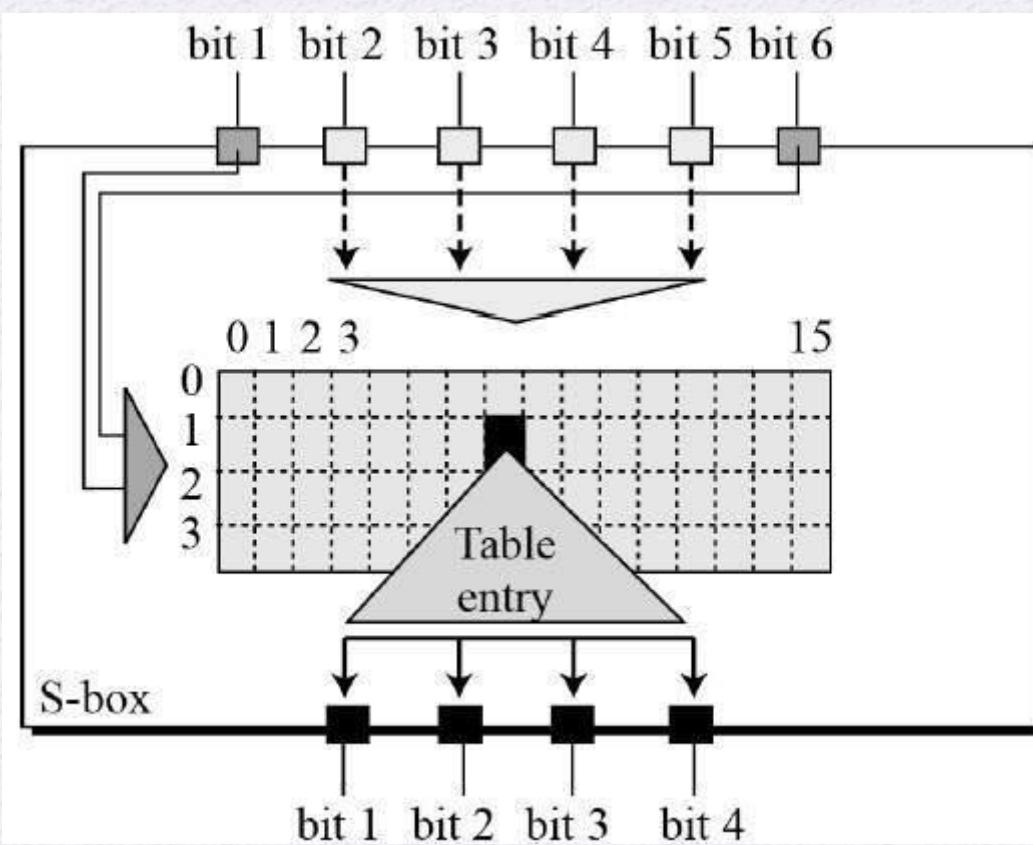


Array of S-Boxes

48-bit input



32-bit output



# Example (Sbox)

Row #	S <sub>1</sub>	1	2	3	...	7	15	Column #
0	14	4	13	1	2	15	11	8
1	0	15	7	4	14	2	13	1
2	4	1	14	8	13	6	2	11
3	15	12	8	2	4	9	1	7

S(i, j) < 16, can be represented with 4 bits

Example: B = 101111 → middle 4

$$b_1 b_6 = 11 = 3 \text{ (row)} \quad \checkmark$$

$$b_2 b_3 b_4 b_5 = 0111 = 7 \text{ (column)} \quad \checkmark$$

Let output of sbox be C

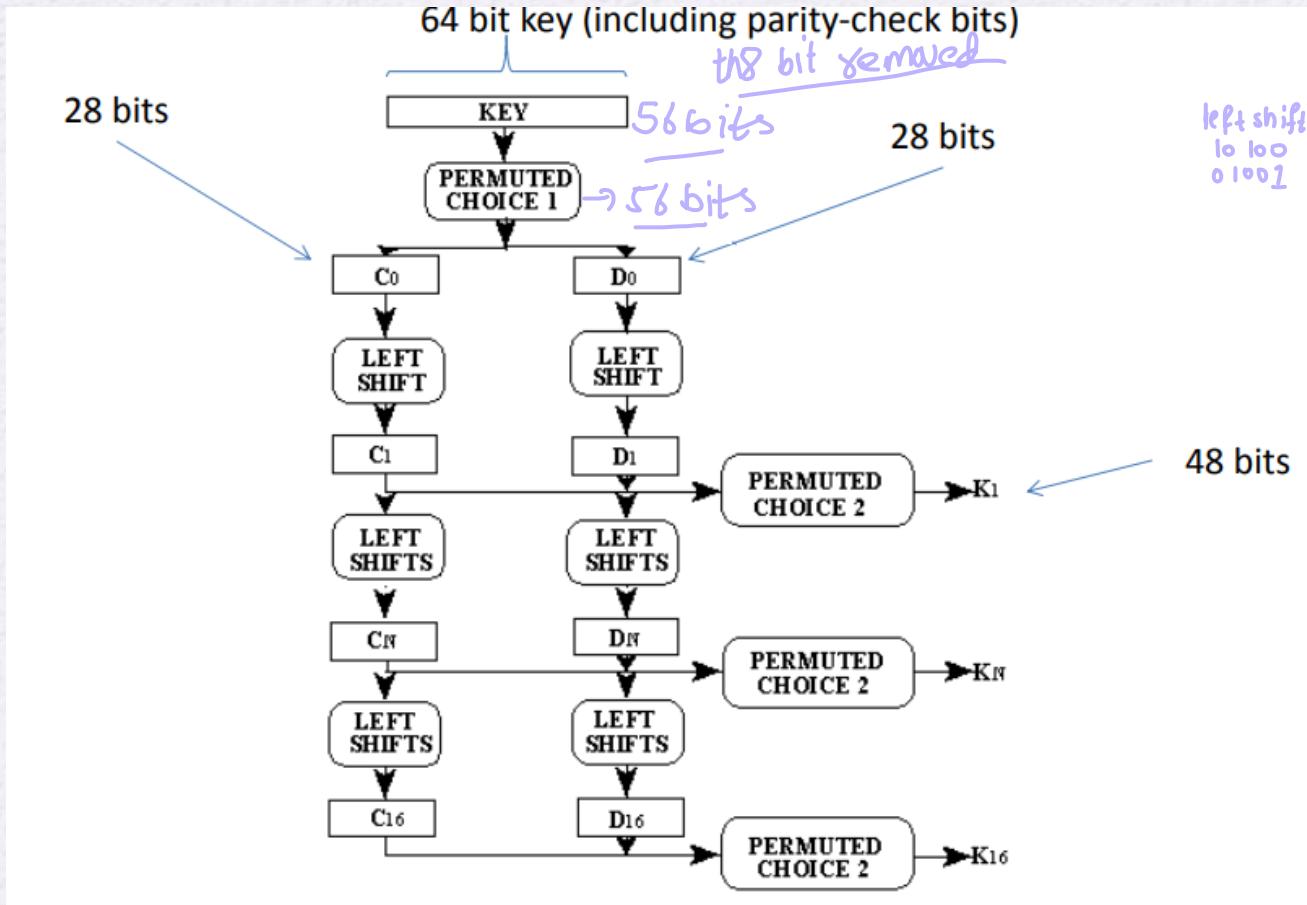
$$C = 7 = 0111$$

# Straight Permutation

I don't know

0	1	3	4	5	6	7	8	
0	16	07	20	21	29	12	28	17
1	01	15	23	26	05	18	31	10
2	02	08	24	14	32	27	03	09
3	19	13	30	06	22	11	04	25

# Key Generation



# DES Permuted Choice 1 and 2 (PC-1, PC-2)

Parity-check bits (namely, bits 8,16, 4,32,40,48,56,64) are not chosen, they do not appear in **PC-1**



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

Left							
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	
Right							
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	



**PC-2** selects the 48-bit subkey for each round from the 56-bit key-schedule state

# Rotation

**Example:** From the original 64-bit key

$K = 00010011\ 00110100\ 01010111\ 01111001\ 10011011\ 10111100\ 11011111\ 11110001$

we get the 56-bit permutation

$K' = 1111000\ 0110011\ 0010101\ 0101111\ 0101010\ 1011001\ 1001111\ 0001111$

Next, split this key into left and right halves,  $C_\theta$  and  $D_\theta$ , where each half has 28 bits.

**Example:** From the permuted key  $K'$ , we get

$C_\theta = 1111000\ 0110011\ 0010101\ 0101111$   
 $D_\theta = 0101010\ 1011001\ 1001111\ 0001111$

**Example:** From original pair  $C_\theta$  and  $D_\theta$  we obtain:

$C_1 = 1110000|11001100101010|1011111$  }  $PC_2 \rightarrow 48\text{ bit} \rightarrow K_1$   
 $D_1 = 10101010110011001111000111101$

$C_2 = 1100001100110010101010111111$   
 $D_2 = 0101010110011001111000111101$

$C_3 = 0000110011001010101011111111$   
 $D_3 = 0101011001100111100011110101$

$C_4 = 0011001100101010101111111100$   
 $D_4 = 0101100110011110001111010101$

Iteration Number	Number of Left Shifts
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

# DES Weak Keys

- Weak keys: keys make the same sub-key to be generated in more than one round.
- Result: reduce cipher complexity
- Weak keys can be avoided at key generation.
- DES has 4 weak keys
  - 01010101 01010101
  - FEEFEFEF FEEFEFEF
  - E0E0E0E0 F1F1F1F1
  - 1F1F1F1F 0E0E0E0E



# DES Decryption

- Decryption uses the same algorithm as encryption, except that the subkeys  $K_1, K_2, \dots, K_{16}$  are applied in reversed order

# Table 4.2

## DES Example

(Table can be found on page 132 in textbook)

Round	<i>Ki</i>	<i>Li</i>	<i>Ri</i>
IP		5a005a00	3cf03c0f
1	1e030f03080d2930	3cf03c0f	bad22845
2	0a31293432242318	bad22845	99e9b723
3	23072318201d0c1d	99e9b723	0bae3b9e
4	05261d3824311a20	0bae3b9e	42415649
5	3325340136002c25	42415649	18b3fa41
6	123a2d0d04262a1c	18b3fa41	9616fe23
7	021f120b1c130611	9616fe23	67117cf2
8	1c10372a2832002b	67117cf2	c11bfc09
9	04292a380c341f03	c11bfc09	887fbc6c
10	2703212607280403	887fbc6c	600f7e8b
11	2826390c31261504	600f7e8b	f596506e
12	12071c241a0a0f08	f596506e	738538b8
13	300935393c0d100b	738538b8	c6a62c4e
14	311e09231321182a	c6a62c4e	56b0bd75
15	283d3e0227072528	56b0bd75	75e8fd8f
16	2921080b13143025	75e8fd8f	25896490
IP-1		da02ce3a	89ecac3b

Note: DES subkeys are shown as eight 6-bit values in hex format

Round		$\delta$	Round		$\delta$
	02468aceeca86420 12468aceeca86420	1	9	c11bf09887fb0c6c 99f911532eed7d94	32
1	3cf03c0fbad22845 3cf03c0fbad32845	1	10	887fb0c6c600f7e8b 2eed7d94d0f23094	34
2	bad2284599e9b723 bad3284539a9b7a3	5	11	600f7e8bf596506e d0f23094455da9c4	37
3	99e9b7230bae3b9e 39a9b7a3171cb8b3	18	12	f596506e738538b8 455da9c47f6e3cf3	31
4	0bae3b9e42415649 171cb8b3ccaca55e	34	13	738538b8c6a62c4e 7f6e3cf34bc1a8d9	29
5	4241564918b3fa41 ccaca55ed16c3653	37	14	c6a62c4e56b0bd75 4bc1a8d91e07d409	33
6	18b3fa419616fe23 d16c3653cf402c68	33	15	56b0bd7575e8fd8f 1e07d4091ce2e6dc	31
7	9616fe2367117cf2 cf402c682b2cefbc	32	16	75e8fd8f25896490 1ce2e6dc365e5f59	32
8	67117cf2c11bf09 2b2cefbc99f91153	33	IP-1	da02ce3a89ecac3b 057cde97d7683f2a	32

Table 4.3 Avalanche Effect in DES: Change in Plaintext

Round		$\delta$	Round		$\delta$
	02468aceeca86420 02468aceeca86420	0	9	c11bf09887fbc6c 548f1de471f64dfd	34
1	3cf03c0fbad22845 3cf03c0f9ad628c5	3	10	887fb6c6c600f7e8b 71f64df4279876c	36
2	bad2284599e9b723 9ad628c59939136b	11	11	600f7e8bf596506e 4279876c399fdc0d	32
3	99e9b7230bae3b9e 9939136b768067b7	25	12	f596506e738538b8 399fdc0d6d208dbb	28
4	0bae3b9e42415649 768067b75a8807c5	29	13	738538b8c6a62c4e 6d208dbbb9bdeeeaa	33
5	4241564918b3fa41 5a8807c5488dbe94	26	14	c6a62c4e56b0bd75 b9bdeeaad2c3a56f	30
6	18b3fa419616fe23 488dbe94aba7fe53	26	15	56b0bd7575e8fd8f d2c3a56f2765c1fb	33
7	9616fe2367117cf2 aba7fe53177d21e4	27	16	75e8fd8f25896490 2765c1fb01263dc4	30
8	67117cf2c11bf09 177d21e4548f1de4	32	IP-1	da02ce3a89ecac3b ee92b50606b62b0b	30

Table 4.4 Avalanche Effect in DES: Change in Key

**Table 4.5** Average Time Required for Exhaustive Key Search

<b>Key Size (bits)</b>	<b>Cipher</b>	<b>Number of Alternative Keys</b>	<b>Time Required at <math>10^9</math> Decryptions/s</b>	<b>Time Required at <math>10^{13}</math> Decryptions/s</b>
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	$2^{55} \text{ ns} = 1.125 \text{ years}$	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	$2^{127} \text{ ns} = 5.3 \times 10^{21} \text{ years}$	$5.3 \times 10^{17} \text{ years}$
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	$2^{167} \text{ ns} = 5.8 \times 10^{33} \text{ years}$	$5.8 \times 10^{29} \text{ years}$
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	$2^{191} \text{ ns} = 9.8 \times 10^{40} \text{ years}$	$9.8 \times 10^{36} \text{ years}$
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	$2^{255} \text{ ns} = 1.8 \times 10^{60} \text{ years}$	$1.8 \times 10^{56} \text{ years}$
26 characters (permutation)	Monoalphabetic	$2! = 4 \times 10^{26}$	$2 \times 10^{26} \text{ ns} = 6.3 \times 10^9 \text{ years}$	$6.3 \times 10^6 \text{ years}$



# Strength of DES

- Timing attacks
  - One in which information about the key or the plaintext is obtained by observing how long it takes a given implementation to perform decryptions on various ciphertexts
  - Exploits the fact that an encryption or decryption algorithm often takes slightly different amounts of time on different inputs
  - So far it appears unlikely that this technique will ever be successful against DES or more powerful symmetric ciphers such as triple DES and AES



# Block Cipher Design Principles: Number of Rounds

The greater the number of rounds, the more difficult it is to perform cryptanalysis

In general, the criterion should be that the number of rounds is chosen so that known cryptanalytic efforts require greater effort than a simple brute-force key search attack

If DES had 15 or fewer rounds, differential cryptanalysis would require less effort than a brute-force key search

# Block Cipher Design Principles: Design of Function F

- The heart of a Feistel block cipher is the function F
- The more nonlinear F, the more difficult any type of cryptanalysis will be
- The SAC and BIC criteria appear to strengthen the effectiveness of the confusion function

The algorithm should have good avalanche properties

Strict avalanche criterion (SAC)

States that any output bit  $j$  of an S-box should change with probability  $1/2$  when any single input bit  $i$  is inverted for all  $i, j$

Bit independence criterion (BIC)

States that output bits  $j$  and  $k$  should change independently when any single input bit  $i$  is inverted for all  $i, j$ , and  $k$

↳ S-box  
↳ change one 1 bit  
↳ out put change 50% prob

↳ 1011 → Should be Indep-  
endent  
↓  
1100 to chan other.

# Block Cipher Design Principles: Key Schedule Algorithm

- With any Feistel block cipher, the key is used to generate one subkey for each round
- In general, we would like to select subkeys to maximize the difficulty of deducing individual subkeys and the difficulty of working back to the main key
- It is suggested that, at a minimum, the key schedule should guarantee key/ciphertext Strict Avalanche Criterion and Bit Independence Criterion

# Summary

- Traditional Block Cipher Structure
  - Stream ciphers
  - Block ciphers
  - Motivation for the Feistel cipher structure
  - Feistel cipher
- The Data Encryption Standard (DES)
  - Encryption
  - Decryption
  - Avalanche effect
- The strength of DES
  - Use of 56-bit keys
  - Nature of the DES algorithm
  - Timing attacks
- Block cipher design principles
  - Number of rounds
  - Design of function F
  - Key schedule algorithm

