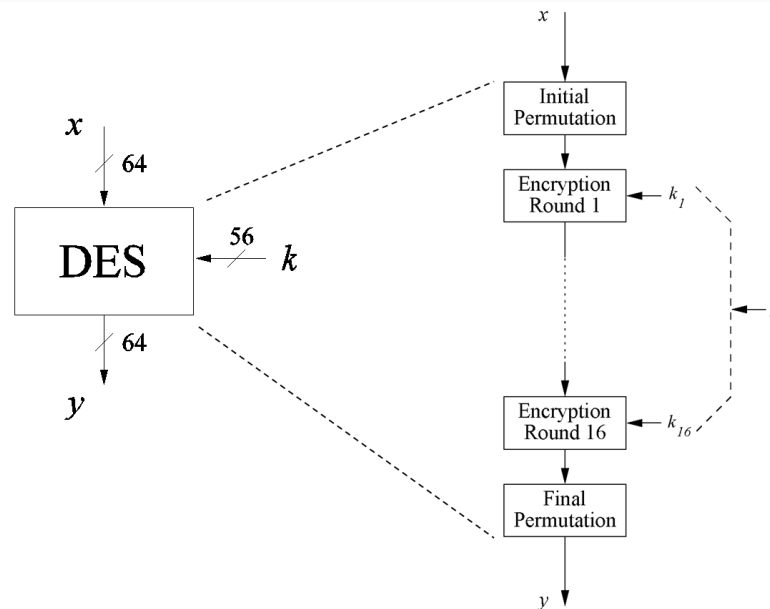


# Data Encryption Standard (DES)



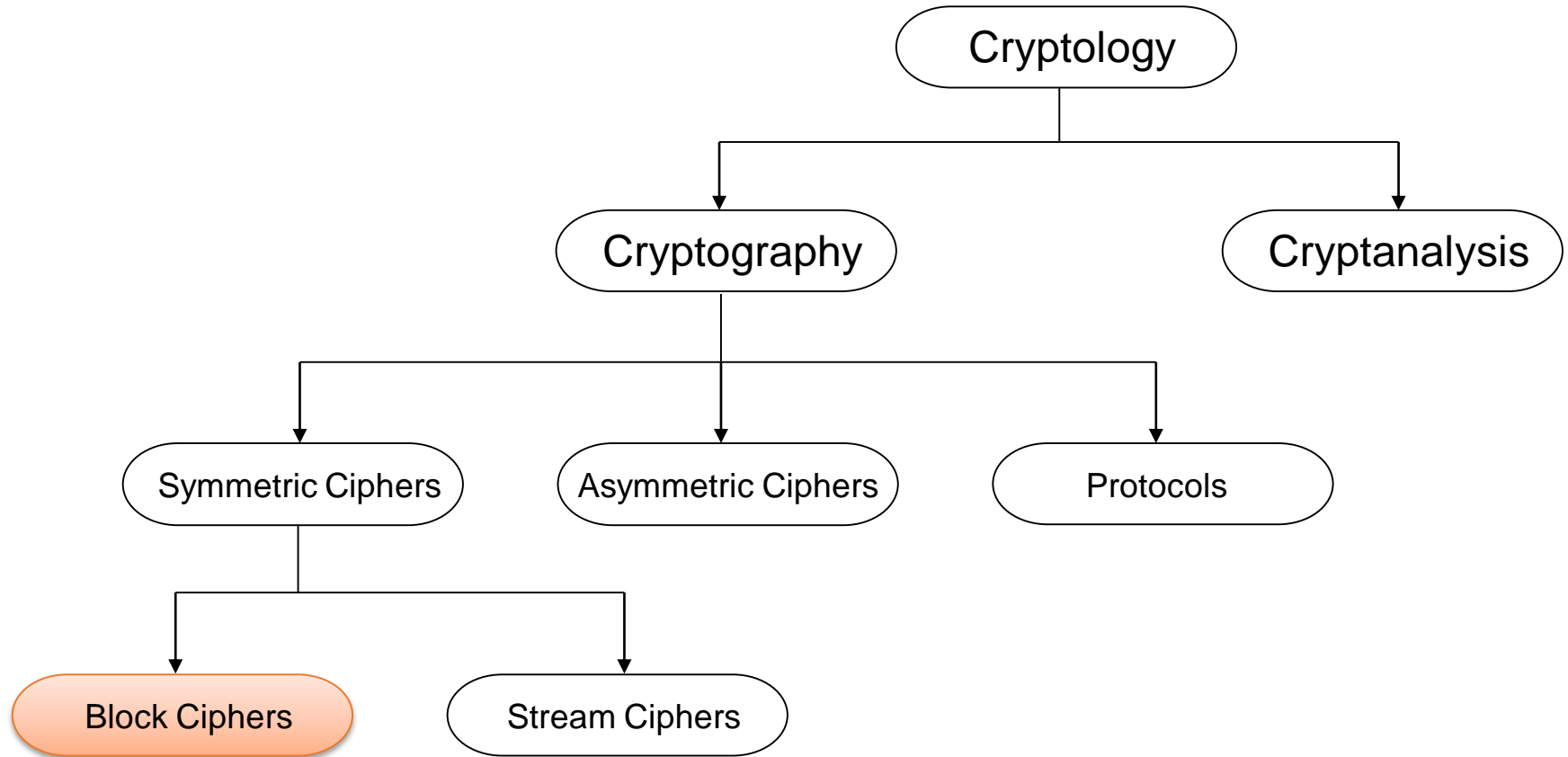
# Outline

1. Introduction to DES
2. Overview of DES Algorithm
3. Internal Structure of DES
4. Security of DES

# Introduction to DES

---

# Classification of DES in the Field of Cryptology



**You are here!**

# DES Facts

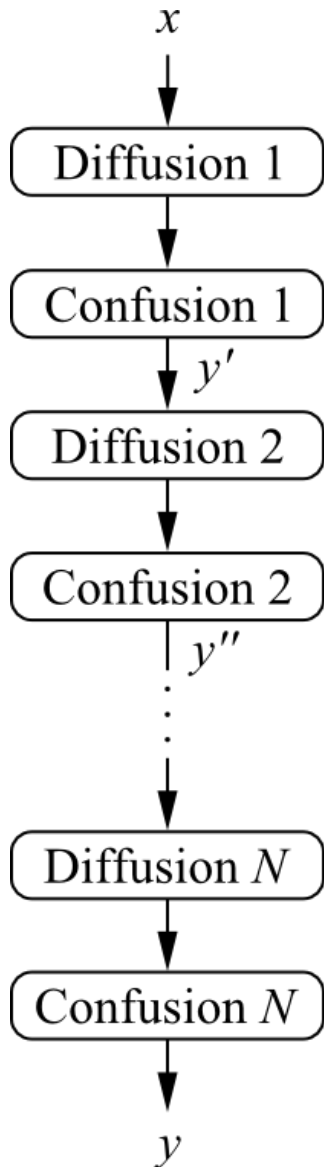
- Data Encryption Standard (DES) encrypts **blocks of size 64 bits**
- Developed by **IBM** based on the cipher *Lucifer* with input from the *National Security Agency* (NSA)
- **Standardized 1977** by the **National Bureau of Standards** (NBS) today called *National Institute of Standards and Technology* (NIST)
- Most popular **block cipher** until 2000
- By far best studied and cryptanalyzed 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

# Block Cipher Primitives: Confusion and Diffusion

Claude Shannon established that two primitive operations are required to build strong encryption algorithms:

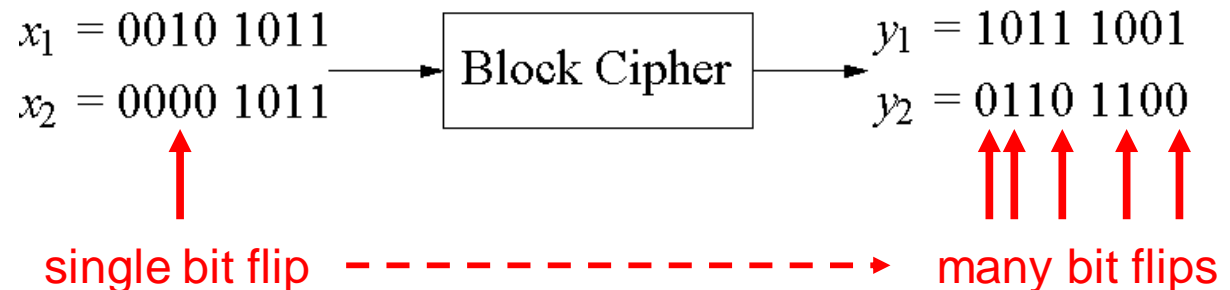
1. **Confusion**: An encryption operation where the **relationship between key and ciphertext is obscured**
    - Commonly achieved using **substitution**
    - Small change in key → large change in ciphertext
  2. **Diffusion**: An encryption operation where the **influence of one plaintext symbol is spread over many ciphertext symbols** with the goal of hiding statistical properties/patterns of the plaintext
    - Commonly achieved using **bit permutation**
    - Small change in plaintext → large change in ciphertext
- **Alternate** both confusion and diffusion functions 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:**

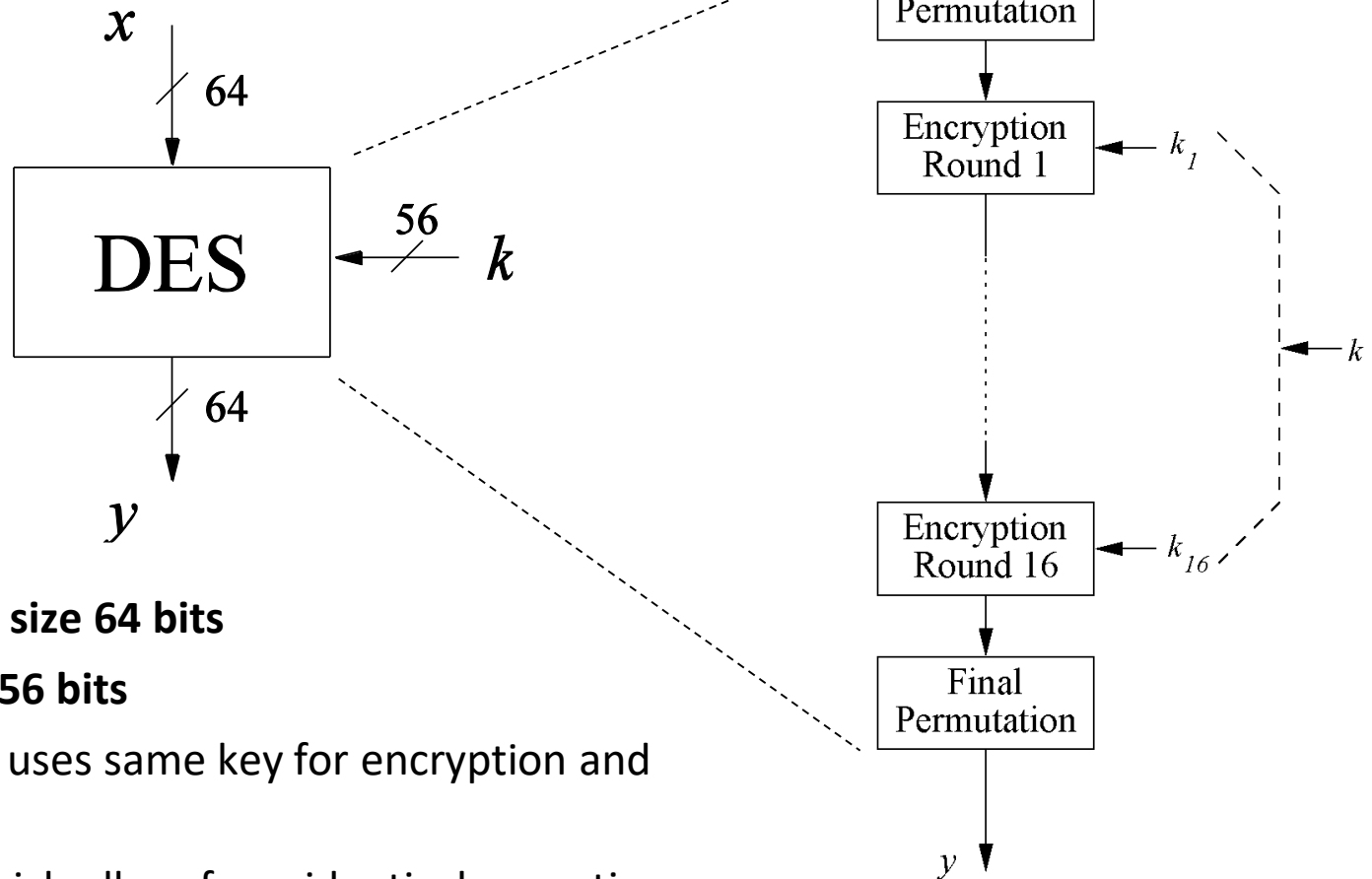


# Overview of DES Algorithm

---

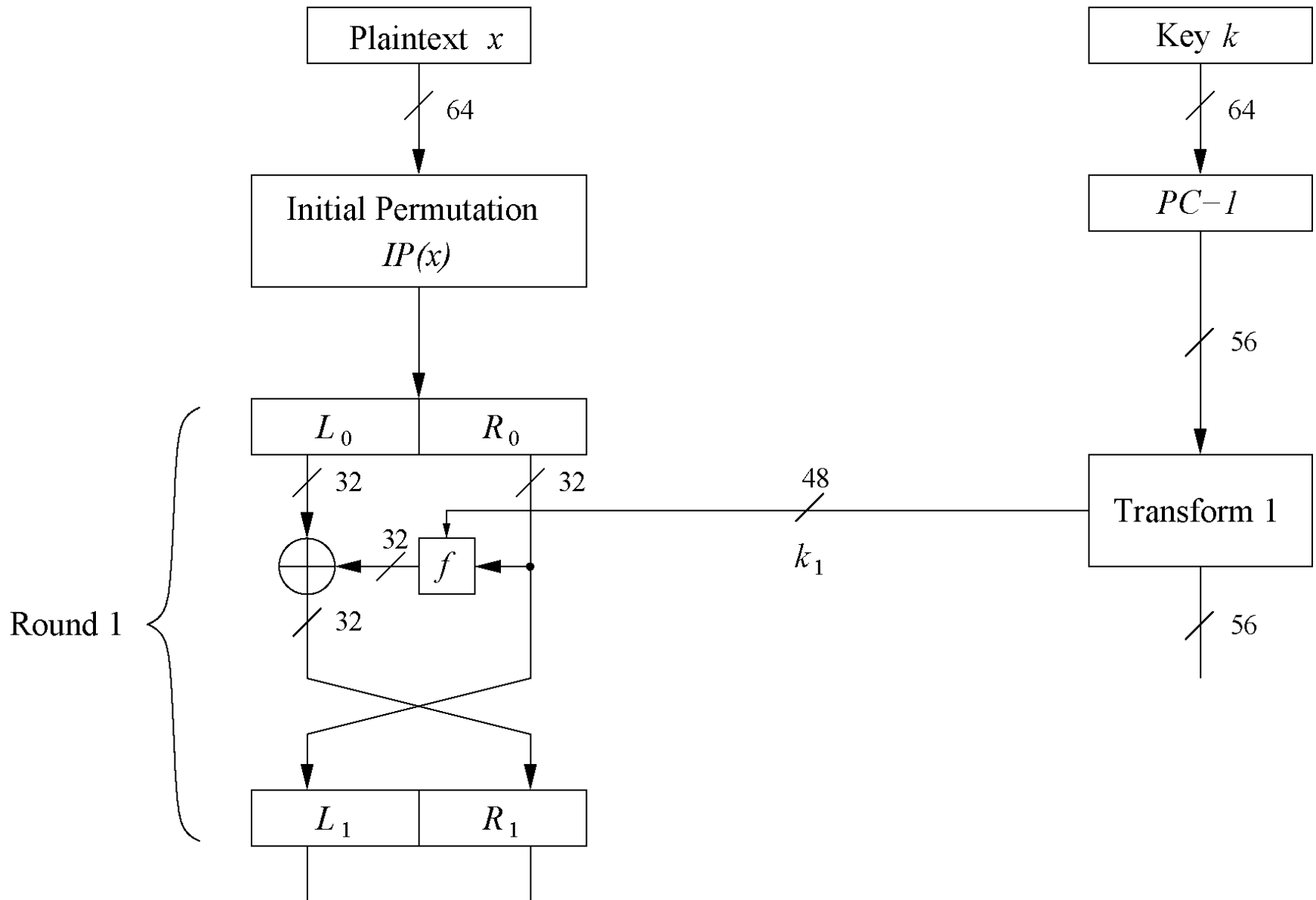


# Overview of DES Algorithm



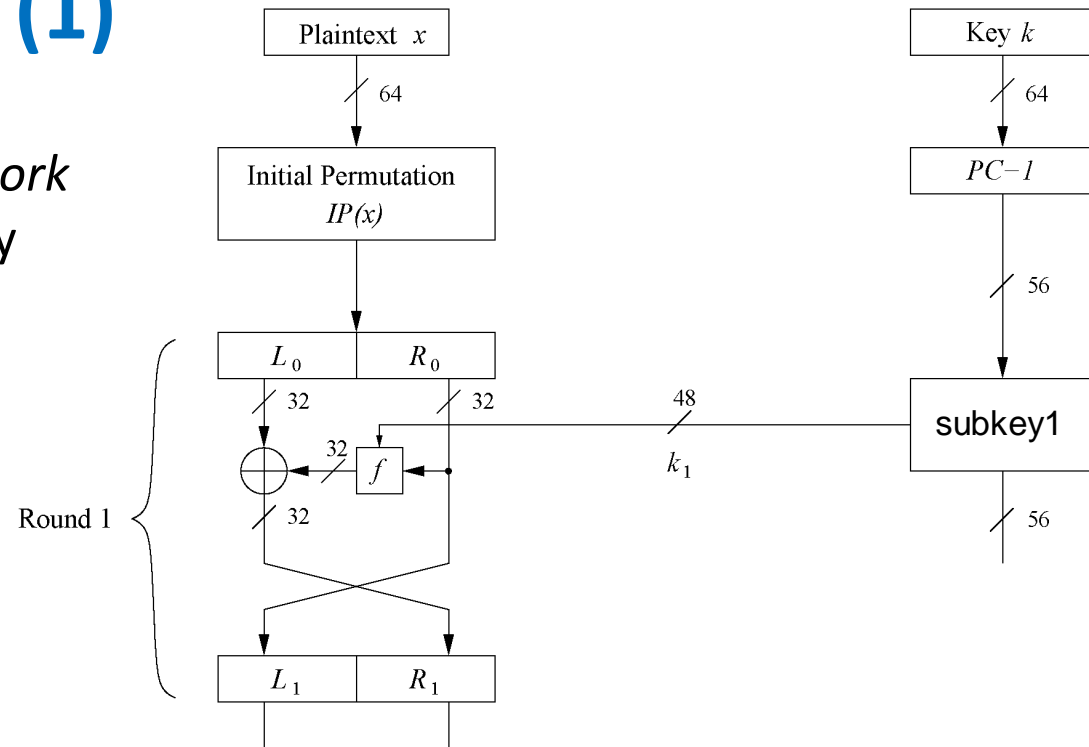
- **Encrypts blocks of size 64 bits**
- **Uses a key of size 56 bits**
- Symmetric cipher: uses same key for encryption and decryption
- Uses **16 rounds** which all perform identical operation
- Different subkey in each round derived from main key
- Each round uses a generated **subkey** and the **output**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

# Encryption Round



# DES Feistel Network (1)

- DES structure is a *Feistel network* which an approach adopted by many block ciphers
- Advantage: encryption and decryption differ only in keyschedule

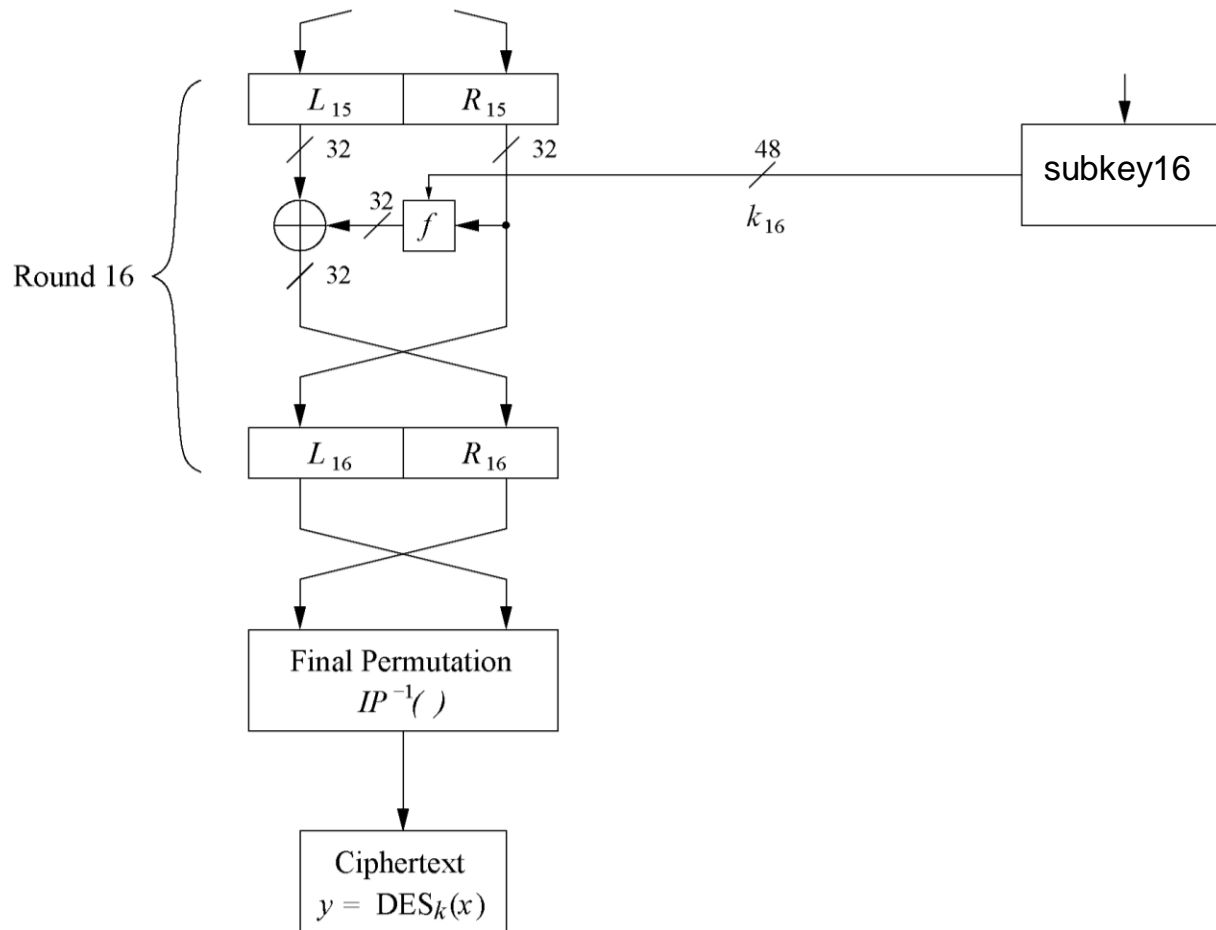


- Bitwise initial permutation, then 16 rounds
  1. Plaintext is **split** into 32-bit halves  $L_i$  and  $R_i$
  2.  $R_i$  is fed into the **function**  $f$ , the output of which is then XORed with  $L_i$
  3. Left and right half are swapped
- Rounds can be expressed as:

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

# The DES Feistel Network (2)

- L and R swapped again at the end of the cipher, i.e., after round 16 followed by a **final permutation**



# Internal Structure of DES

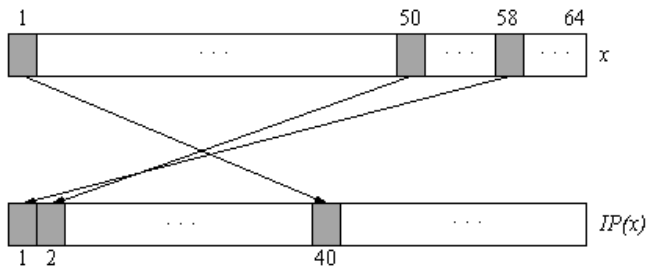
---

# Initial and Final Permutation

- Bitwise Permutations
- Described by tables  $IP$  and  $IP^{-1}$

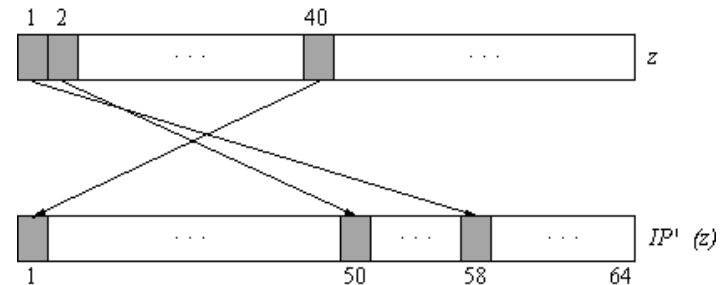
Initial Permutation

$IP$							
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



[https://en.wikipedia.org/wiki/DES\\_supplementary\\_material](https://en.wikipedia.org/wiki/DES_supplementary_material)

# The f-Function

- **Main operation of DES**

- *f*-Function inputs:

$R_{i-1}$  and round key  $k_i$

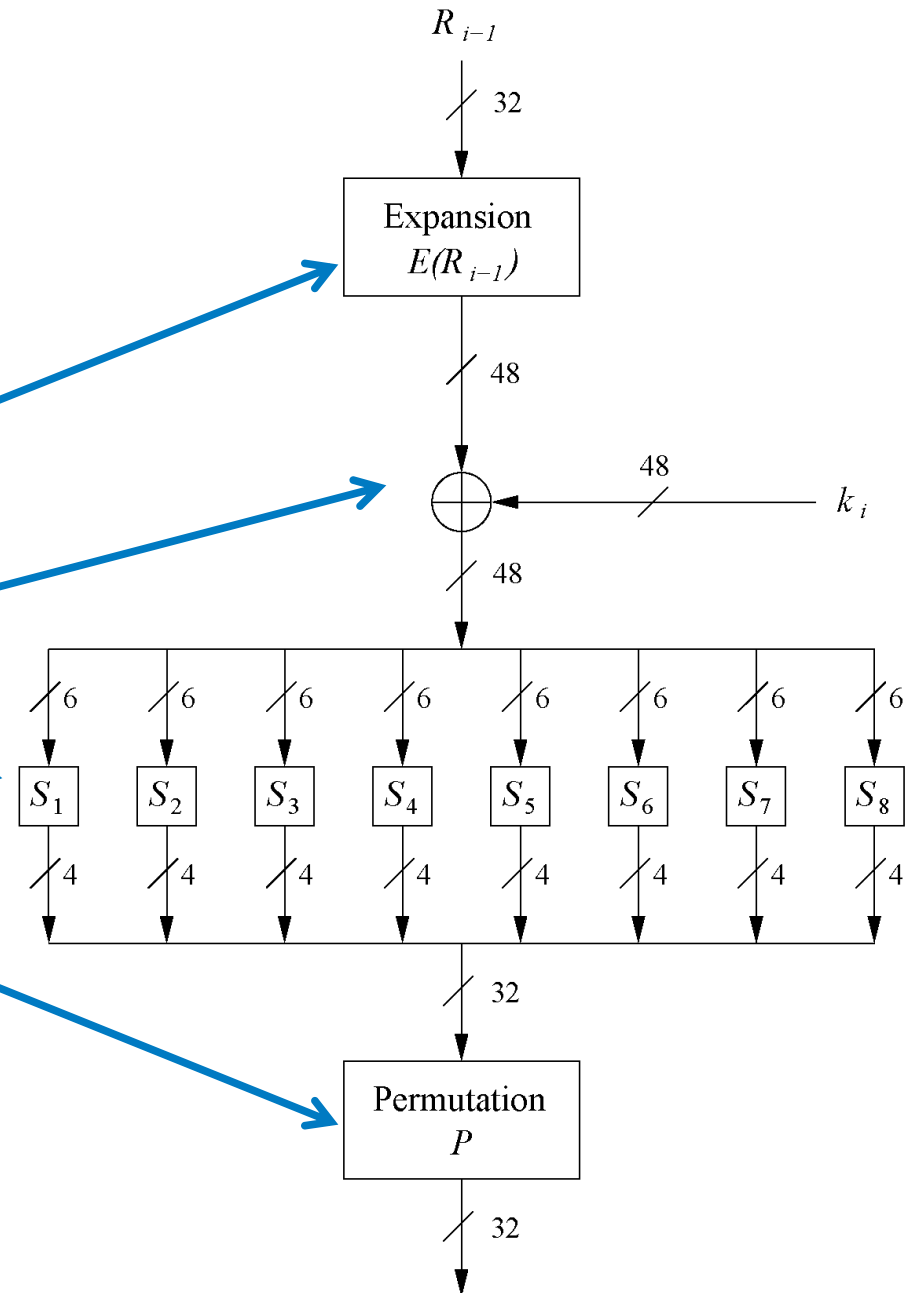
- **4 Steps:**

1. Expansion  $E$

2. XOR with round key

3. S-box substitution

4. Permutation

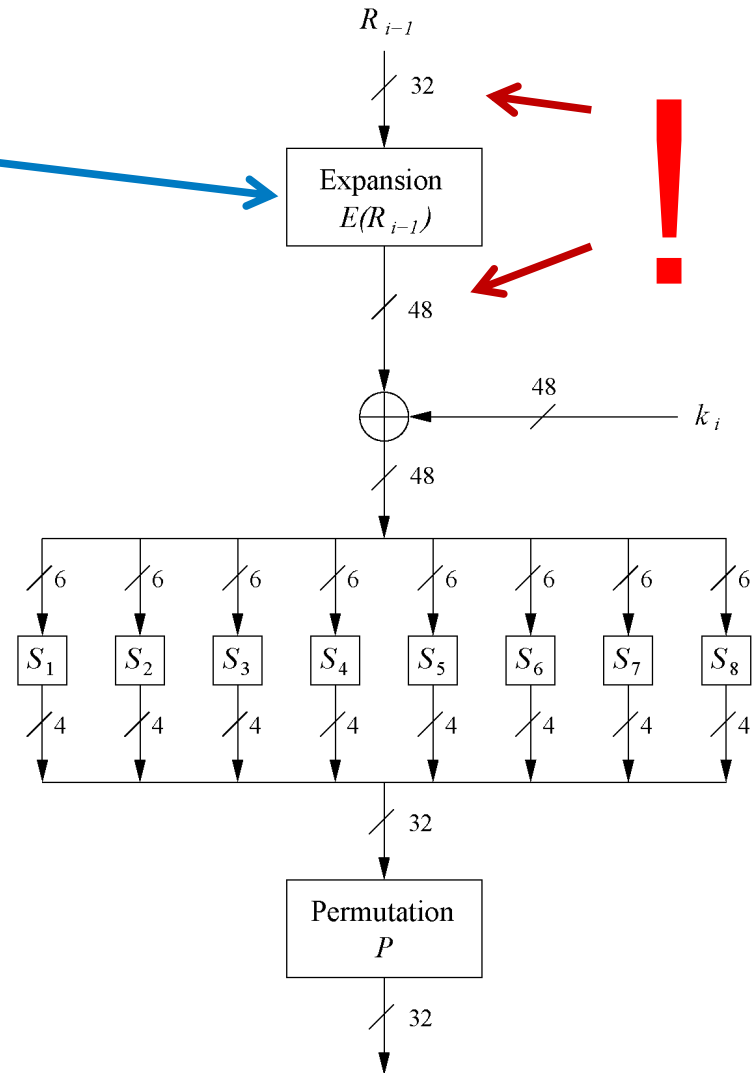
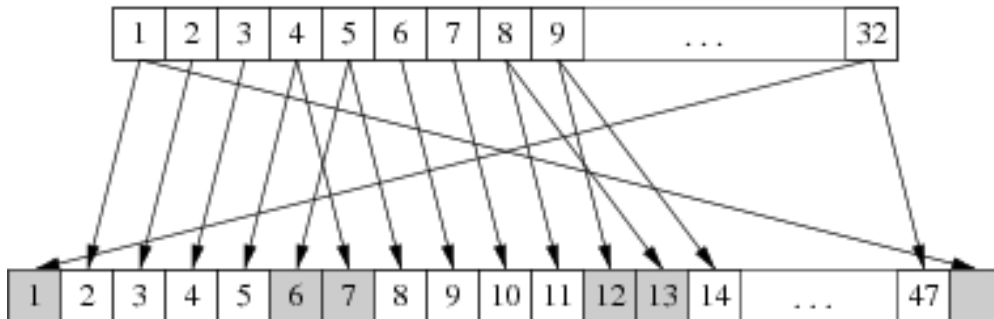


# The Expansion Function E

## 1. Expansion E

- main purpose: increases diffusion

$E$												
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							

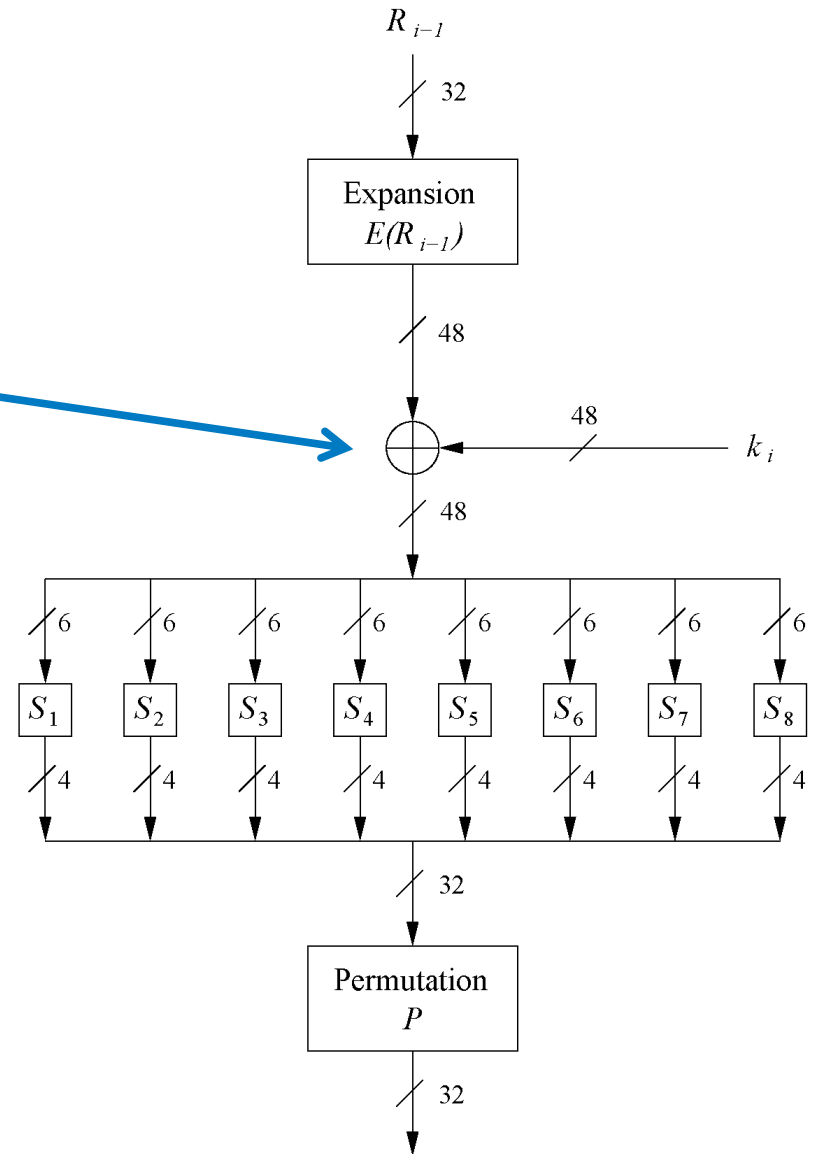




# XOR Round Key

## 2. XOR Round Key

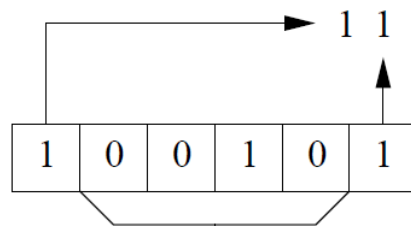
- Bitwise XOR of the round key and the output of the expansion function  $E$
- Round keys are derived from the main key in the DES keyschedule (in a few slides)



# The DES S-Boxes

## 3. S-Box substitution

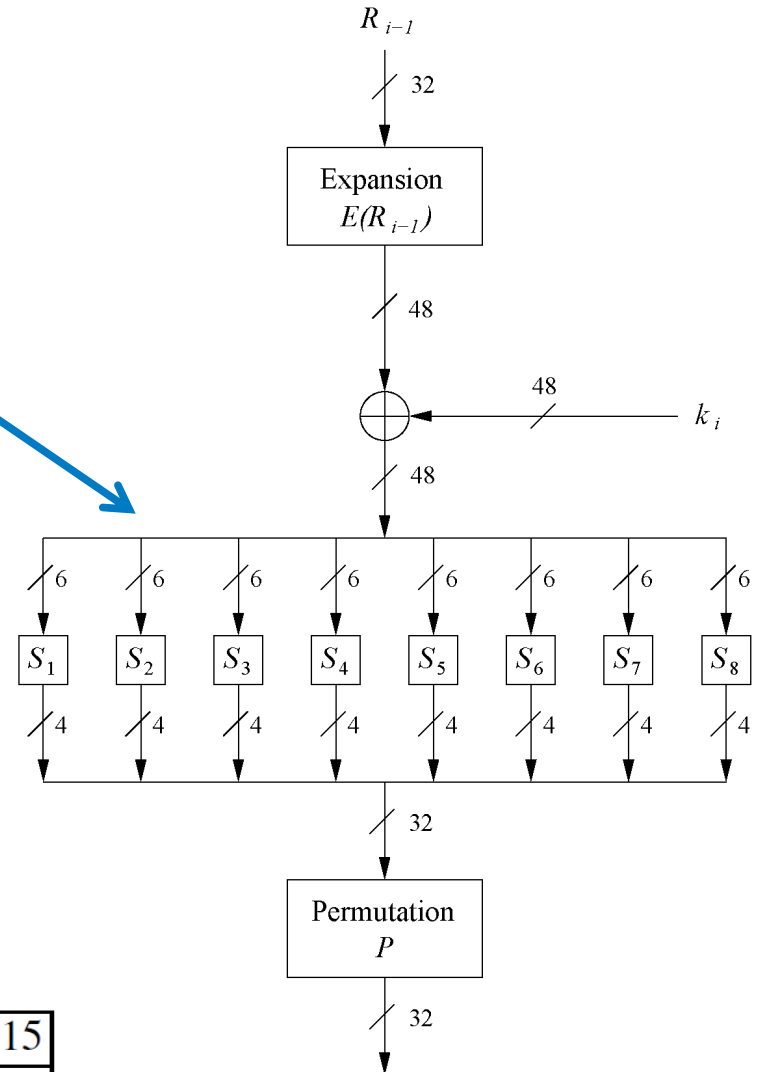
- **Eight** substitution tables
- Each S-box maps 6 bits of input to 4 bits of output
- Non-linear and resistant to differential cryptanalysis
- **Crucial element for DES security!**



Row Index **3**

Column Index **2**

$S_1$	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	<b>08</b>	02	04	09	01	07	05	11	03	14	10	00	06	13

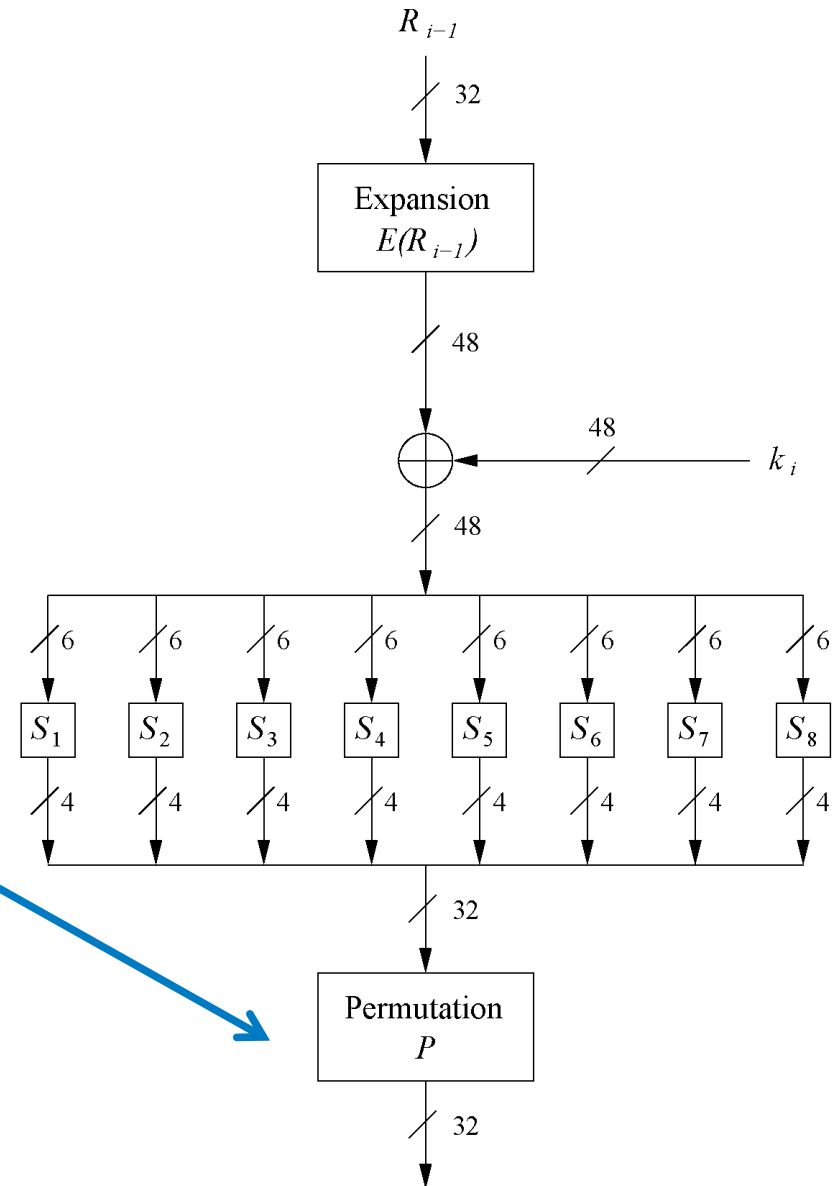


# The Permutation P

## 4. Permutation P

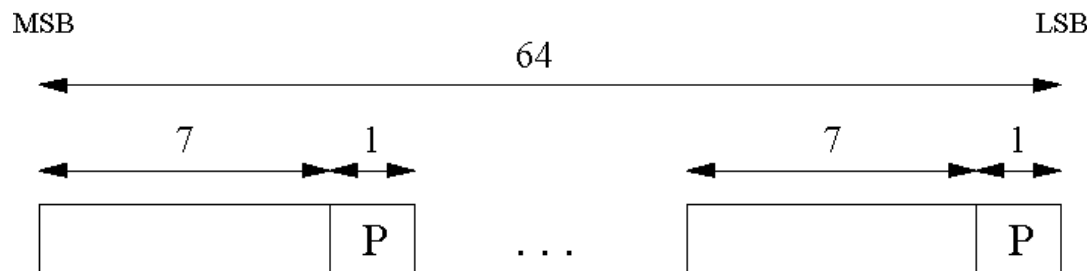
- Bitwise permutation
- Introduces diffusion
- Output bits of one S-Box affect several S-Boxes in next round

$P$								
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	



# Key Schedule (1)

- 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 not used:



**Last bit of every byte is not used**

- Permuted Choice (PC-1)** is used to permute the key is split it into left-side and right-side (28 bits each). Not that the **last bit of every byte is ignored**. (the bits 8, 16, 24, 32, 40, 48, 56 and 64 are not used at all)

**PC-1**

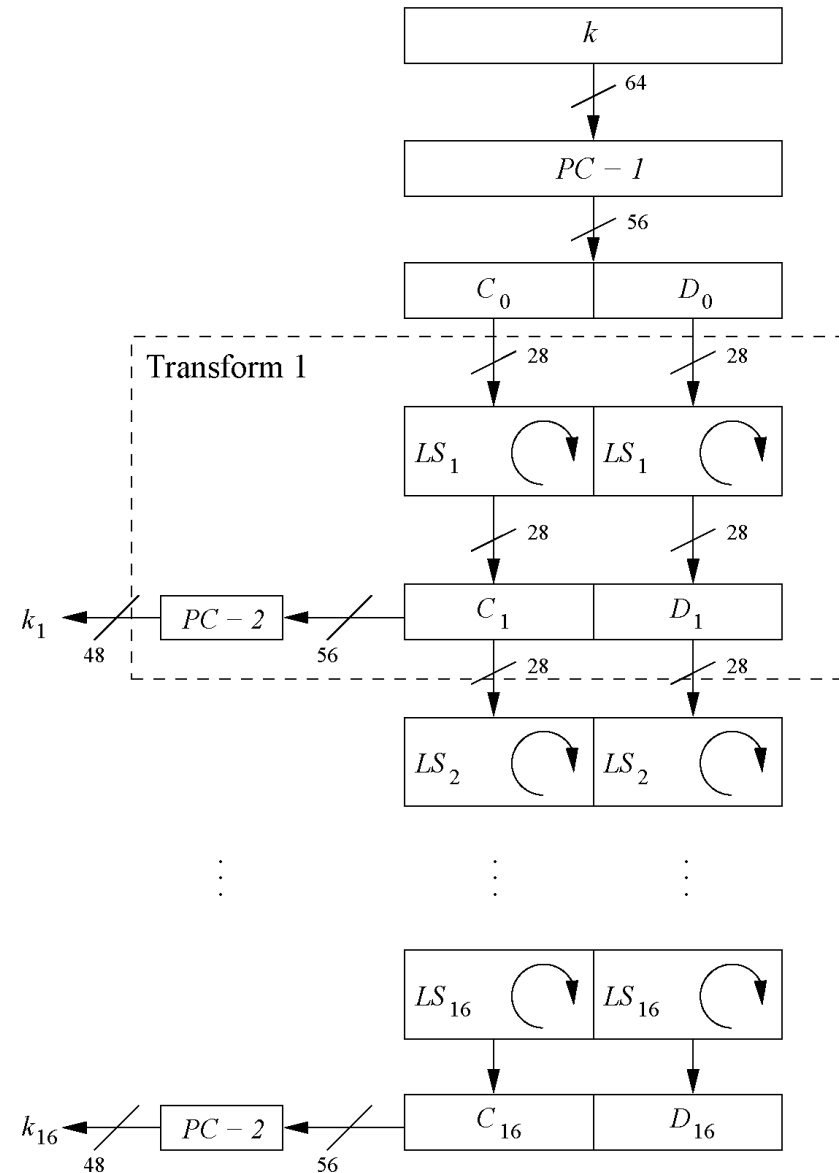
Left							Right						
57	49	41	33	25	17	9	63	55	47	39	31	23	15
1	58	50	42	34	26	18	7	62	54	46	38	30	22
10	2	59	51	43	35	27	14	6	61	53	45	37	29
19	11	3	60	52	44	36	21	13	5	28	20	12	4

# Key Schedule (2)

- 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** (i.e, Left Circular Shift)
- In **all other rounds** where the two halves are each **rotated left** by **two bits**
- In each round  $i$  **Permuted Choice  $PC-2$**  selects a permuted subset of 48 bits of  $C_i$  and  $D_i$  as round key  $k_i$   
i.e. **each  $k_i$  is a permutation of  $k$ !**

$PC - 2$							
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

- Note:** The total number of rotations:  
 $4 \times 1 + 12 \times 2 = 28 \Rightarrow D_0 = D_{16}$  and  $C_0 = C_{16}$ !

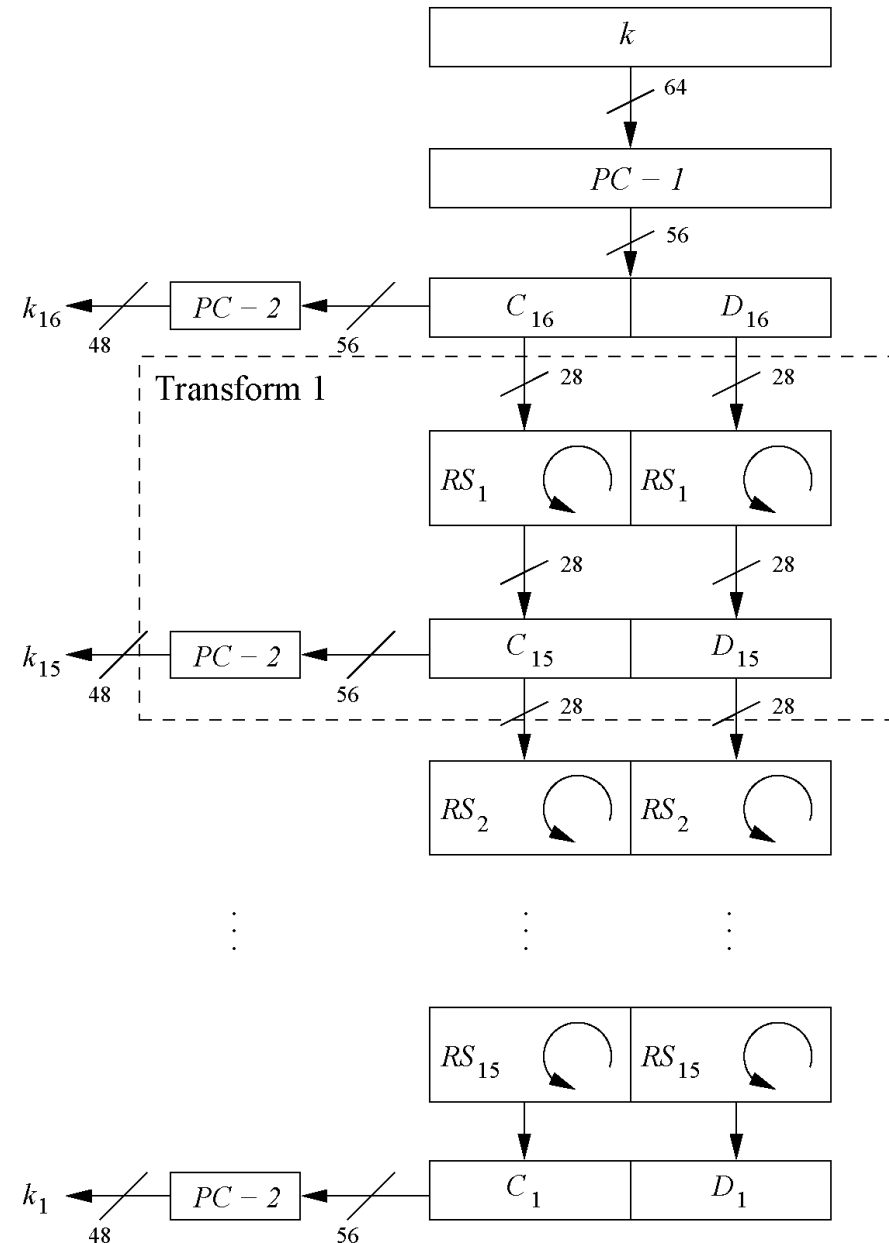


# Decryption

- In **Feistel ciphers** only the keyschedule has to be modified for decryption
- Generate the same 16 round keys in reverse order

## Reversed key schedule:

- As  $D_0 = D_{16}$  and  $C_0 = C_{16}$  the first round key can be generated by applying  $PC-2$  right after  $PC-1$  (no rotation here!)
- All other rotations of  $C$  and  $D$  can be reversed to reproduce the other round keys resulting in:
  - No rotation in round 1.
  - One bit rotation **to the right** in rounds 2, 9 and 16.
  - Two bit rotations **to the right** in all other rounds.



# Security of DES

---

# Security of DES

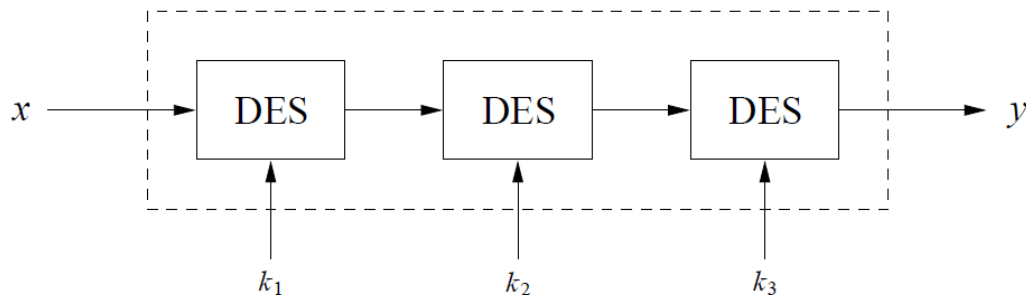
- **After the proposal of DES two major criticisms arose:**
  1. Key space is too small ( $2^{56}$  keys)
  2. S-box design criteria have been kept secret: Are there any hidden analytical attacks (*backdoors*), only known to the NSA?
- **Analytical Attacks:** DES is highly resistant to advanced *cryptanalysis*. So far there is no known analytical attack.
- **Exhaustive key search:** For a given pair of plaintext-ciphertext  $(x, y)$  test all  $2^{56}$  keys until the condition  $\text{DES}_k^{-1}(x)=y$  is fulfilled.  
⇒ Relatively easy given today's computer technology!



# Triple DES – 3DES

- Triple encryption applies the DES cipher algorithm three times to each data block
- Protect against brute-force attacks without the need to design a completely new block cipher (just by effective key length up to 168 bits)

$$y = DES_{k_3}(DES_{k_2}(DES_{k_1}(x)))$$



- No practical attack known today
- Used in some legacy applications such as in banking systems but slow

# Alternatives to DES

Algorithm	I/O Bit	Key lengths	Remarks
AES	128	128/192/256	DES replacement, worldwide used standard
Triple DES	64	up to 168 bits	Used for legacy applications
Mars	128	128/192/256	AES finalist
RC6	128	128/192/256	AES finalist
Serpent	128	128/192/256	AES finalist
Twofish	128	128/192/256	AES finalist

# Summary

- DES was the dominant symmetric encryption algorithm from the mid-1970s to the mid-1990s. Since 56-bit keys are no longer secure, the Advanced Encryption Standard (AES) was created
- Standard DES with 56-bit key length can be broken relatively easily nowadays through an exhaustive key search
- DES is robust against analytical attacks. Its security depends heavily on S-boxes
- By encrypting with DES three times in a row, triple DES (3DES) is created, against which no practical attack is currently known
- The default symmetric cipher is nowadays often **AES**

# Resources

- DES Animation

<http://kathrynneugent.com/des.html>

- DES Wikipedia page

[https://en.wikipedia.org/wiki/Data Encryption Standard](https://en.wikipedia.org/wiki/Data_Encryption_Standard)

- DES Tool

<http://des.online-domain-tools.com/>