MH4311 Cryptography

Lecture 5

Block Cipher (Part 2, DES)

Wu Hongjun

Lecture Outline

- Classical ciphers
- Symmetric key encryption
  - One-time pad & information theory
  - Block cipher
    - Introduction
    - DES, Double DES, Triple DES
    - AES
    - Mode of Operations
    - Attacks
  - Stream cipher
- Hash function and Message Authentication Code
- Public key encryption
- Digital signature
- Key establishment and management
- Introduction to other cryptographic topics

Lecture Outline (cont.)

- **DES design**
  - **Feistel network**
- **Double DES**
- **Triple DES**

Recommended Reading

- CTP Section 3.5
- HAC Section 7.4

- Wikipedia:
  - Feistel network
    http://en.wikipedia.org/wiki/Feistel_cipher
  - DES
    http://en.wikipedia.org/wiki/Data_Encryption_Standard
  - Triple DES
    http://en.wikipedia.org/wiki/Triple_DES

Data Encryption Standard (DES): History

- **1973: NBS (now named NIST) request for encryption algorithms**
  Only IBM submitted, but the design is disappointing
- **1974: NBS request for the second time**
  IBM submitted a new one with 64-bit key
  NSA involved in the design, suggested modification to reduce the key size
- **1976: NBS adopts the IBM design as Data Encryption Standard (FIPS 46)**
- **1983: DES was reaffirmed for the first time**
- **1988: DES was reaffirmed for the second time (FIPS 46-1)**
- **1993: DES was reaffirmed for the third time (FIPS 46-2)**
- **1998: DES cracked in 55 hrs (brute force)**
- **1999: DES cracked in 22 hrs 15 minutes (brute force)**
- **1999: DES was reaffirmed for the forth time (FIPS 46-3, with Triple-DES)**
- **2001: NIST adopts Rijndael (AES) as replacement to DES**
- **2005: NIST withdraws FIPS 46-3**
- **2006: DES cracked in 9 days ($10,000 hardware cost, brute force)**

Data Encryption Standard (DES)

- **Block cipher**
- **64-bit block size**
- **56-bit key**
  - **Plus 8 redundant bits for parity checking:**
    for every 7 key bits, there is one parity bit that indicates whether the number of '1' in those 7 bits is even or odd
    ➔ simple error detection
- **Feistel network**
- **16 rounds**
  + initial permutation
  + final permutation (the inverse of initial permutation)

DES: Feistel Network

- **Feistel network**
  - **Invented by Horst Feistel**
    - **A pioneer on block cipher**
    - **Co-designer of DES**

Horst Feistel
(1915—1990)

DES: Feistel Network

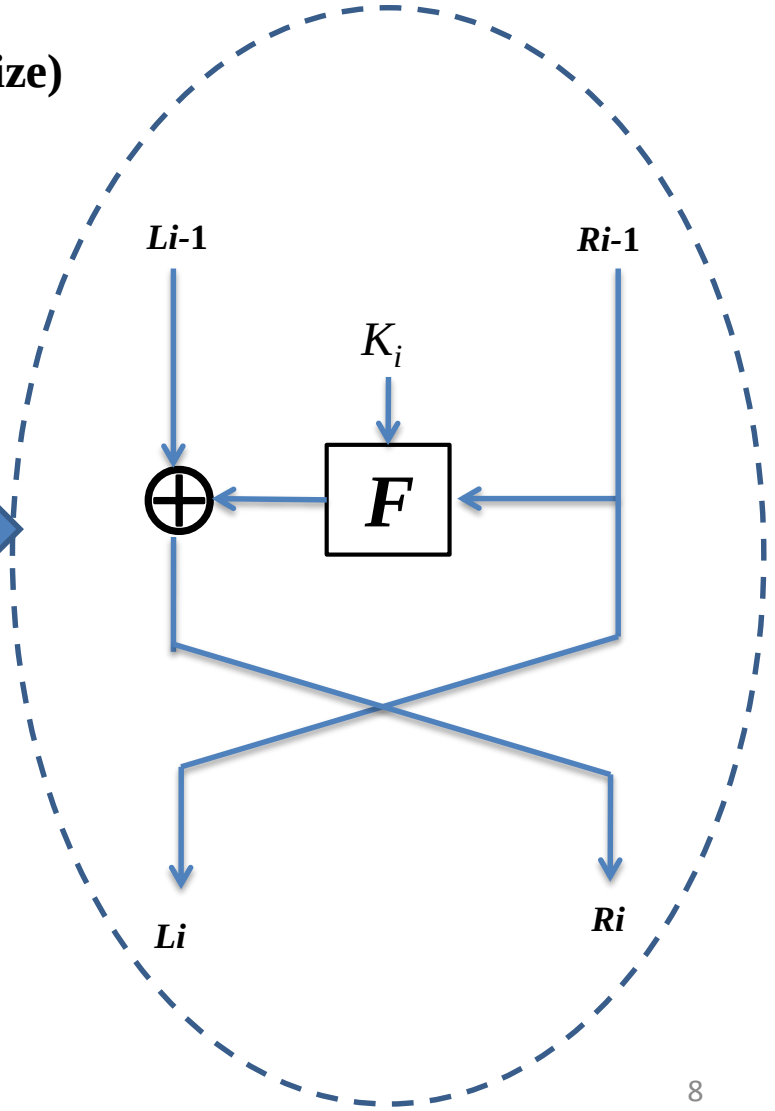
Encryption:

$(L_0, R_0) = P$ (**L_i & R_i are the same size**)

$$L_1 = R_0$$
$$R_1 = L_0 \oplus F(K_1, R_0)$$
$$\Lambda$$
$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus F(K_i, R_{i-1})$$
$$\Lambda$$
$$L_r = R_{r-1}$$
$$R_r = L_{r-1} \oplus F(K_r, R_{r-1})$$

$C = (R_r, L_r)$

One Round



DES: Feistel Network

Encryption:

$(L_0, R_0) = P$

$L_1 = R_0$

$R_1 = L_0 \oplus F(K_1, R_0)$

Λ

$L_i = R_{i-1}$

$R_i = L_{i-1} \oplus F(K_i, R_{i-1})$

Λ

$L_r = R_{r-1}$

$R_r = L_{r-1} \oplus F(K_r, R_{r-1})$

$C = (R_r, L_r)$

Decryption:

$(L_0, R_0) = C$

$L_1 = R_0$

$R_1 = L_0 \oplus F(K_r, R_0)$

Λ

$L_i = R_{i-1}$

$R_i = L_{i-1} \oplus F(K_{r-i+1}, R_{i-1})$

Λ

$L_r = R_{r-1}$

$R_r = L_{r-1} \oplus F(K_1, R_{r-1})$

$P = (R_r, L_r)$

Encryption & decryption :

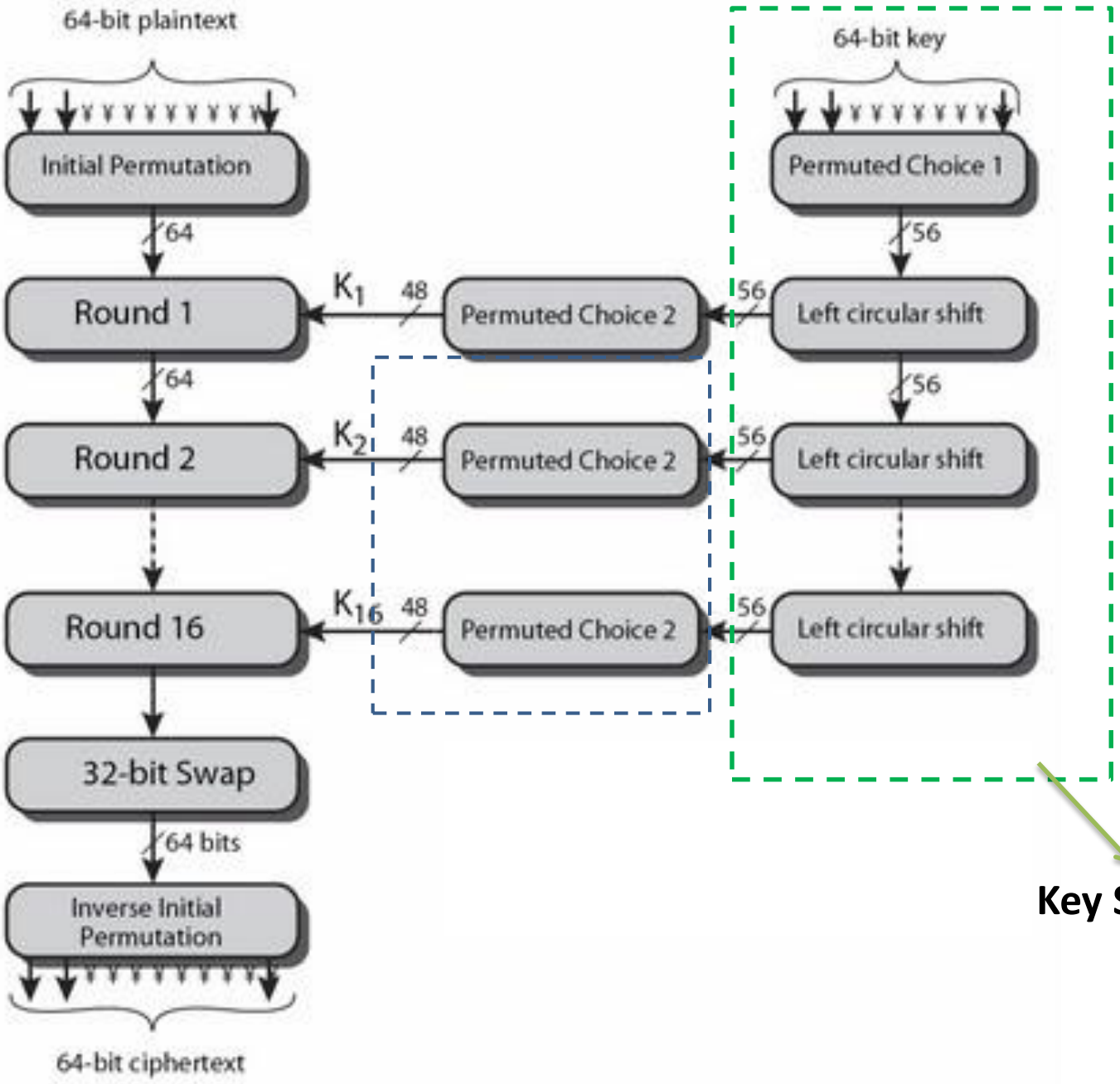
1) **The same Feistel Network**
2) **But the order of the round keys is reversed**

DES: Feistel Network

- **Properties of Feistel Network**
  - **Always invertible**

$$L_i = R_{i-1} \qquad R_{i-1} = L_i$$
$$R_i = L_{i-1} \oplus F(K_i, R_{i-1}) \iff L_{i-1} = R_i \oplus F(K_i, L_i)$$

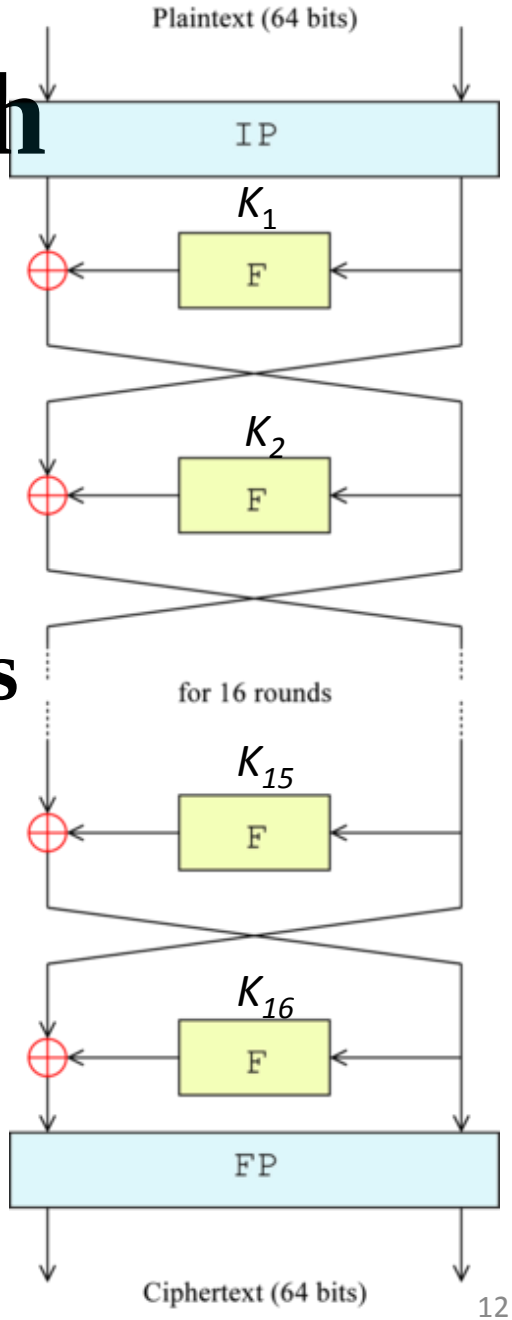
  - **The same Feistel network is used for both encryption & decryption**
    - **Except that the order of round keys is reversed**
      - $K_1, K_2, K_3, \ldots, K_r$ **for encryption**
      - $K_r, \ldots, K_3, K_2, K_1$ **for decryption**


64-bit plaintext
Initial Permutation
64
Round 1
64
Round 2
Round 16
32-bit Swap
64 bits
Inverse Initial Permutation
64-bit ciphertext
64-bit key
Permuted Choice 1
56
Left circular shift
56
Left circular shift
Left circular shift
K_1
48
Permuted Choice 2
56
K_2
48
Permuted Choice 2
56
K_{16}
48
Permuted Choice 2
56
Key Schedule

DES

DES: Computation Path

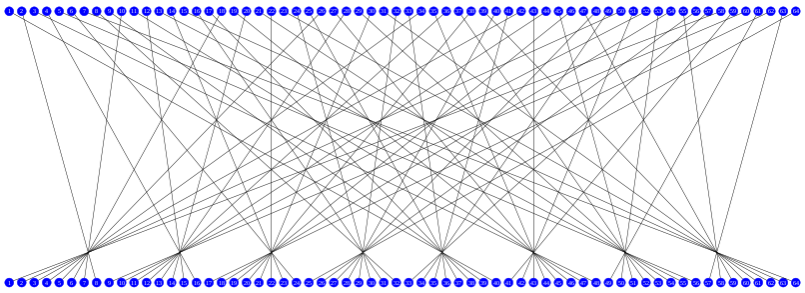
- **Initial permutation (*IP*)**
  - **To increase the security**
  - **But not that useful**
- **Feistel network with 16 rounds**
  - **Notice the last round**
- **Final permutation (IP^{-1})**
  - **The inverse of the initial permutation**

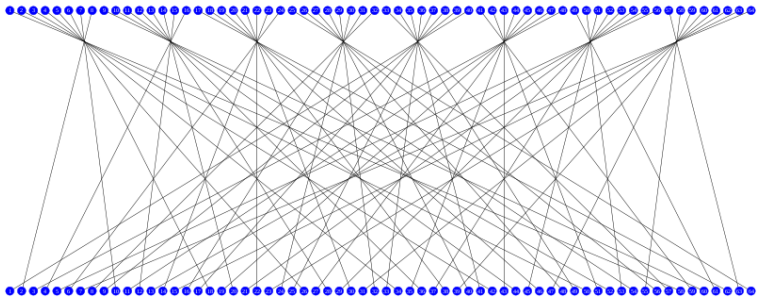


DES:Initial Permutation(IP) and Final Permutation(IP^{-1})

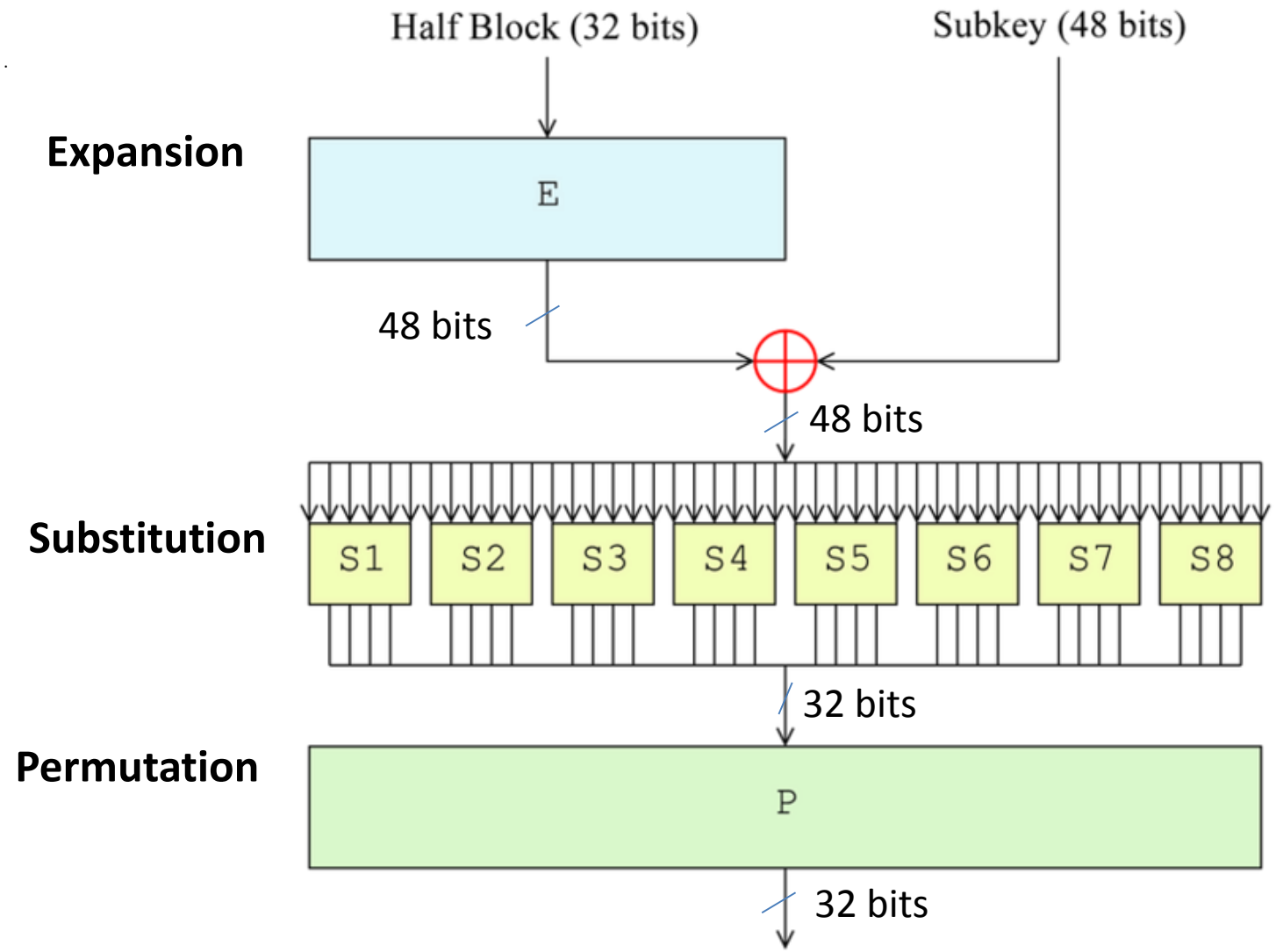
IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

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





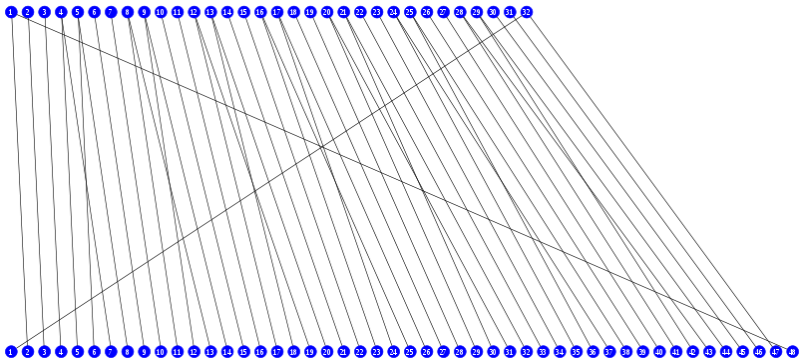
DES: Function F (DES round function)

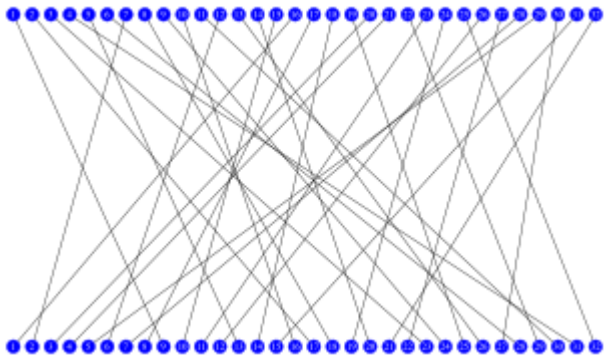


DES: Expansion & Permutation in Function F

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

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





DES: Substitution in Function F

- **Sbox: substitution table**
- **8 different Sboxes used in DES**
  - **Non-invertible**
  - **6-bit input, 4-bit output**
    - **6×4-bit Sbox**
  - **Example: the fifth Sbox of DES**

$S_5(011011) = 1001$

S_5		Middle 4 bits of input														
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Outer bits	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

DES: SBoxes

- **Why design these Sboxes in this way?**

S_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_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_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_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_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
4	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_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_7

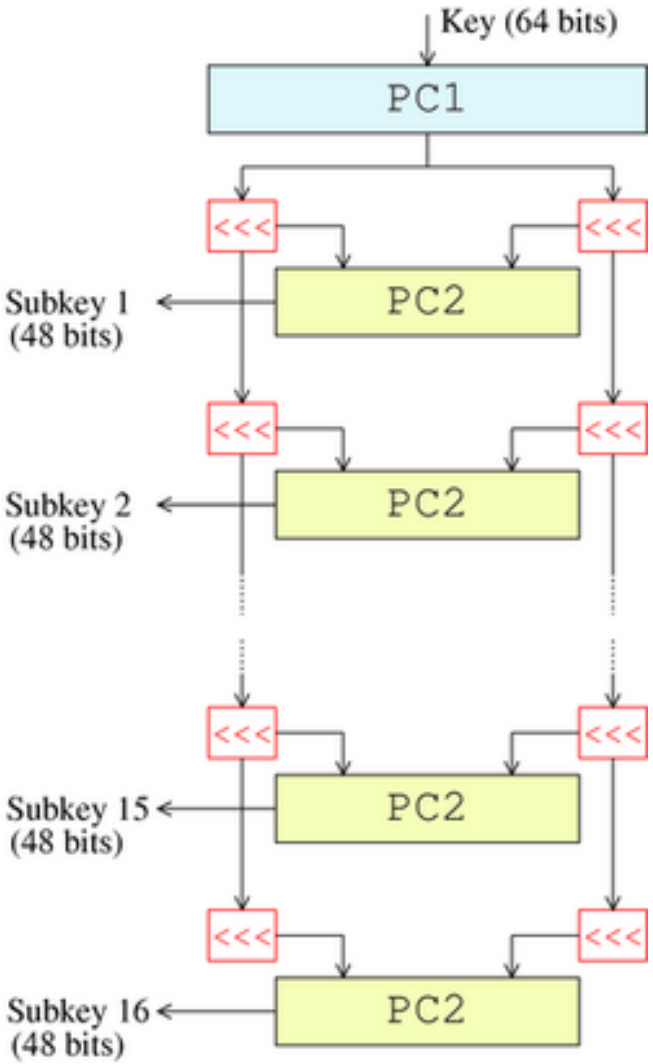
4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_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

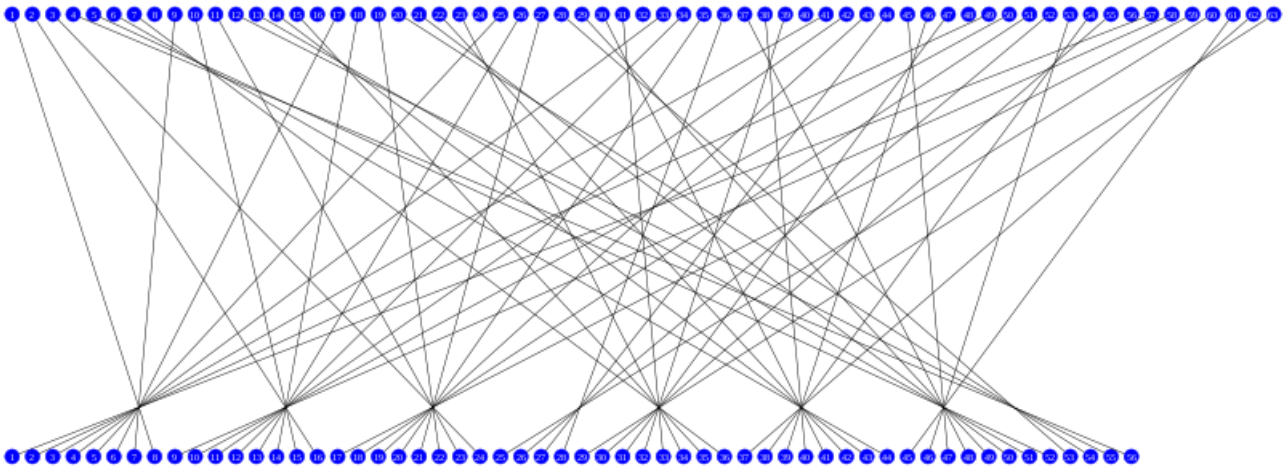
DES: Key Schedule

- **Permuted Choice 1 (PC1)**
  - **Remove 8 parity bits**
  - **Permutation**
- **Rotation positions for each round:**
  - **1,1,2,2,2,2,2,2,**
    1,2,2,2,2,2,2,1
- **Permuted Choice 2 (PC2)**
  - **Permute 56 key bits**
  - **Extract 48 bits as a subkey (round key)**

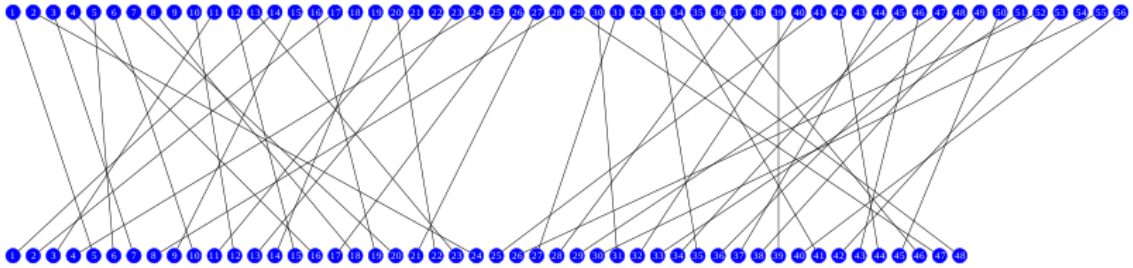


DES: Key Schedule

- **Permuted Choice 1**



- **Permuted Choice 2**



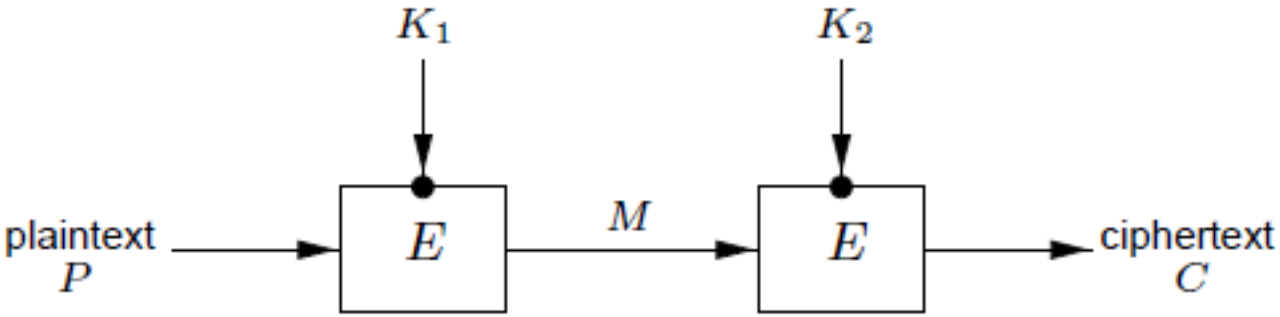
Multiple Encryption

- **DES: 56-bit key size**
  - **Quite small w.r.t. today's computing technology**
  - **How to increase the key size ?**
- **Multiple encryption with more than one key**
  - **Double DES**
  - **Triple DES**

Multiple Encryption

- **Double DES**
  - **Key size: 112 bits**
  - **Encryption:** $C = E_{K_2}(E_{K_1}(P))$
  - **Vulnerable to meet-in-the-middle attack (to learn later)**
    - **Recovering the 112-bit key with**

      2^{57} DES computations, 2^{56} memory



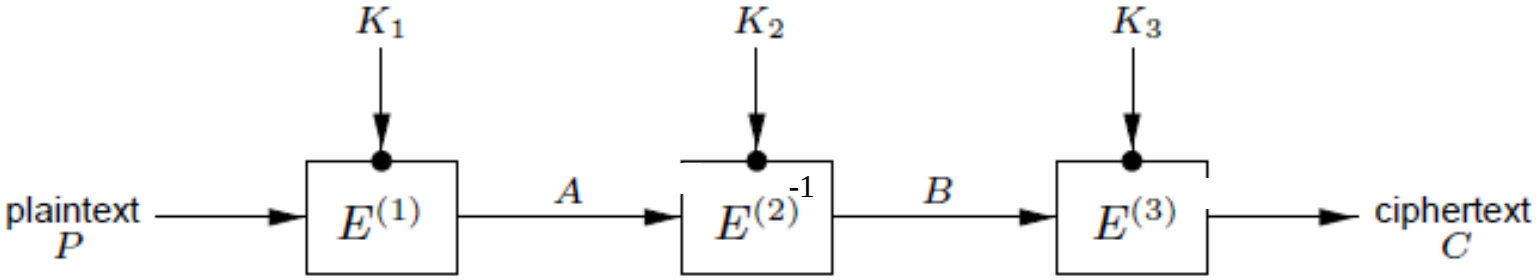
Multiple Encryption

- **Triple DES**
  - **Encryption:** $C = E_{k_3}(E_{K_2}^{-1}(E_{K_1}(P)))$

    Decryption: $P = E_{K_1}^{-1}(E_{K_2}(E_{K_3}^{-1}(C)))$
  - **Keying options**

    Option 1. K_1, K_2 & K_3 are independent ✓

    Option 2. K_1, K_2 are independent, $K_1 = K_3$



Summary

- **DES**
  - **Feistel Network**
    - **Always invertible**
    - **The same network for encryption and decryption**
      - **The order of the round keys are reversed**
  - **Key schedule**
    - **Linear**
- **Double DES, Triple DES**
  - **Their security ?**