Unit II. Symmetric Cryptography

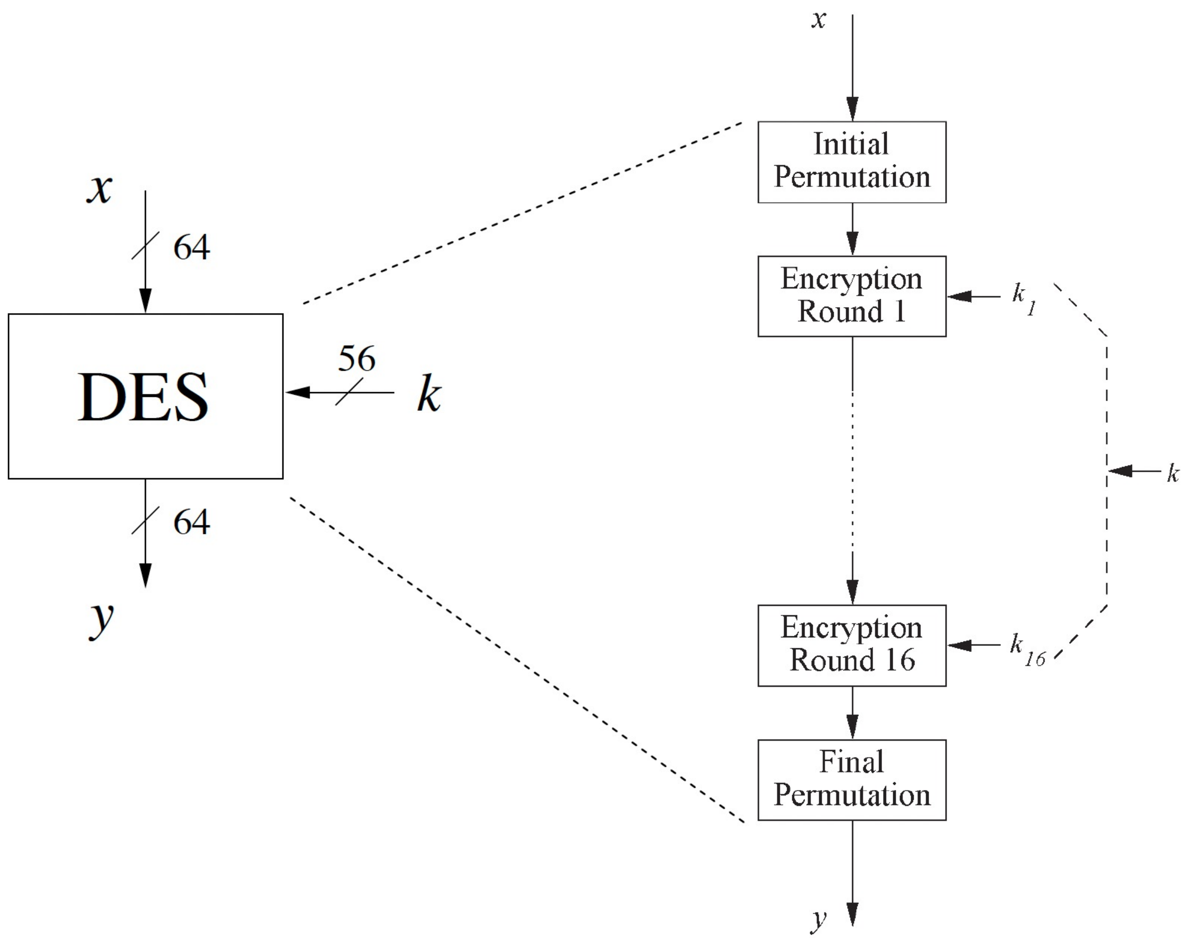
Data Encryption Standard (DES)

Data Encryption Standard (DES)

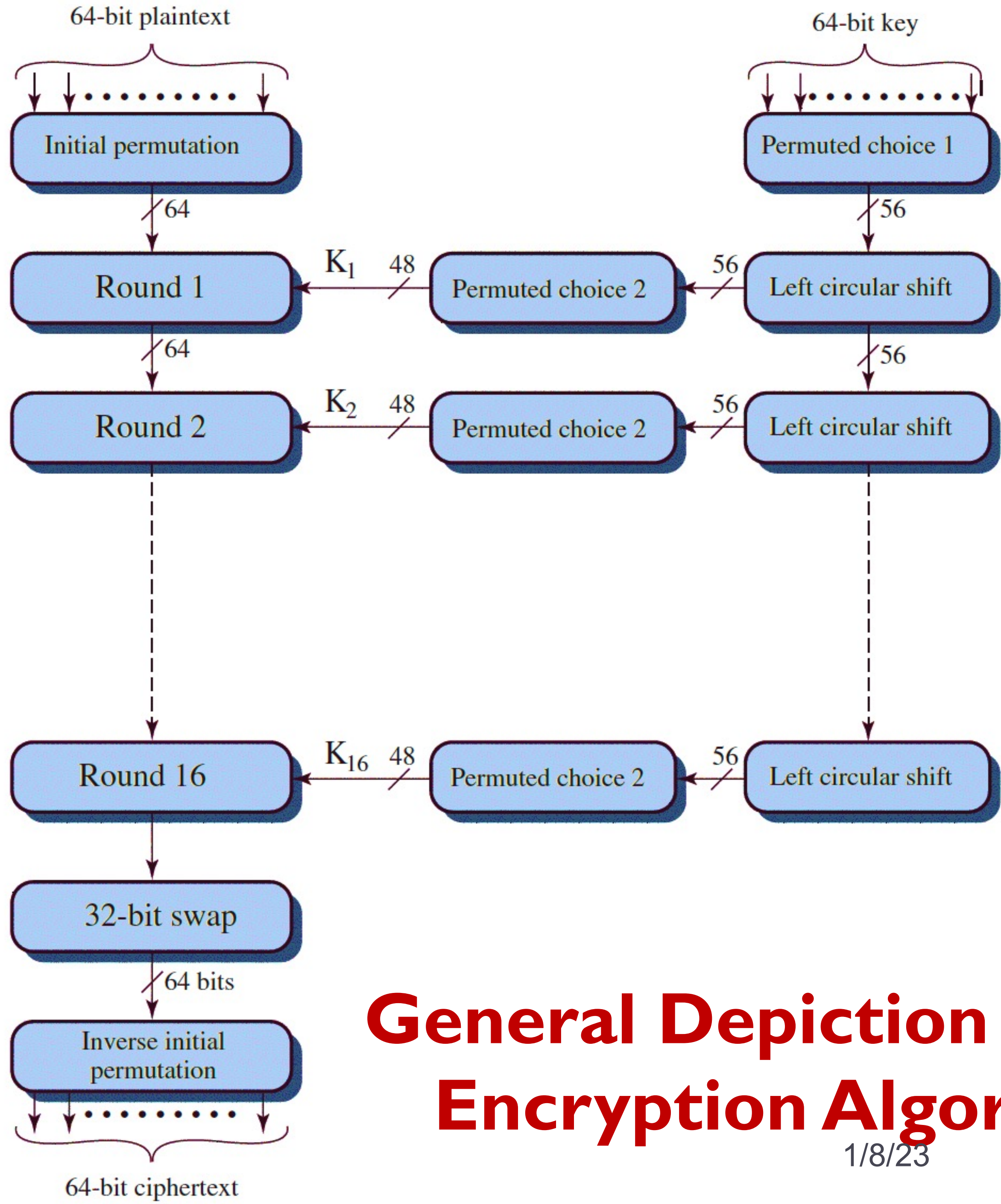
- First developed in 1974 by IBM and NSA (based on LUCIFER that was 64-bit encryption with 128 bit key size).
- Also known as Data Encryption Algorithm (DEA) that was adopted in 1977 by National Bureau of Standards (presently called NIST) as the Data Encryption Standard.
- Data are encrypted in 64-bit blocks using a 56-bit key.
- The same steps, with the same key in reverse order, are used to perform the decryption.
- DES has the same structure as Feistel cipher except the F function and additional initial and final permutations IP and IP^{-1}, respectively.
- IP and IP^{-1} have no cryptographic significance.

DES Basic Overview

- 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 the identical operation
- Different subkey in each round derived from the main key



General Depiction of DES Encryption Algorithm



DES Encryption

Encryption Phases:

1. Initial permutation (IP):
   - the 64-bit plaintext passes through and rearranges the
   - bits to produce the permuted input
2. Sixteen rounds of substitutions and permutations.
   - 16 rounds which all perform the identical operation
   - The output of the last round consists of 64 bits that is a function of the input plaintext and the key.
   - The left and right halves of the output are swapped to produce the pre-output.
3. Final permutation (IP-1)
   - Inverse of the initial permutation function.

DES Encryption

Key scheduling: -

- Initially, the 64-bit key is passed through a permutation function.
- Then, for each of the sixteen rounds, a subkey (K_i) is produced by the combination of a left circular shift and a permutation.
- The permutation function is the same for each round, but a different subkey is produced because of the repeated shifts of the key bits.

DES: IP and IP^{-1}

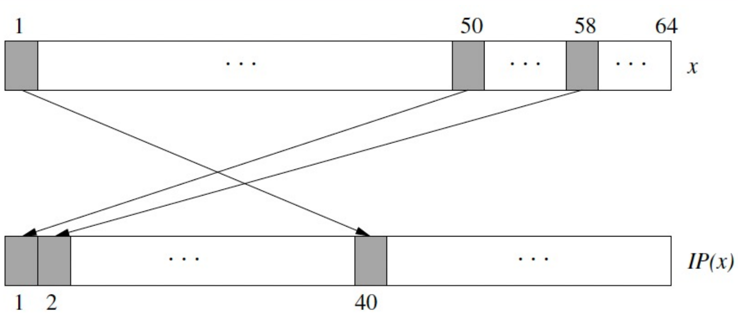
- Initial and final permutations are defined by tables.
- Both tables consist of 64 bits each numbered from 1 to 64.
- Each entry in the permutation table indicates the position of a numbered input bit in the output.

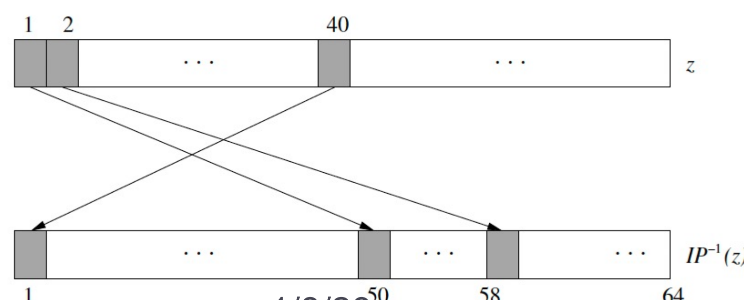
(a) Initial Permutation (IP)

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

(b) Inverse Initial Permutation (IP^{-1})

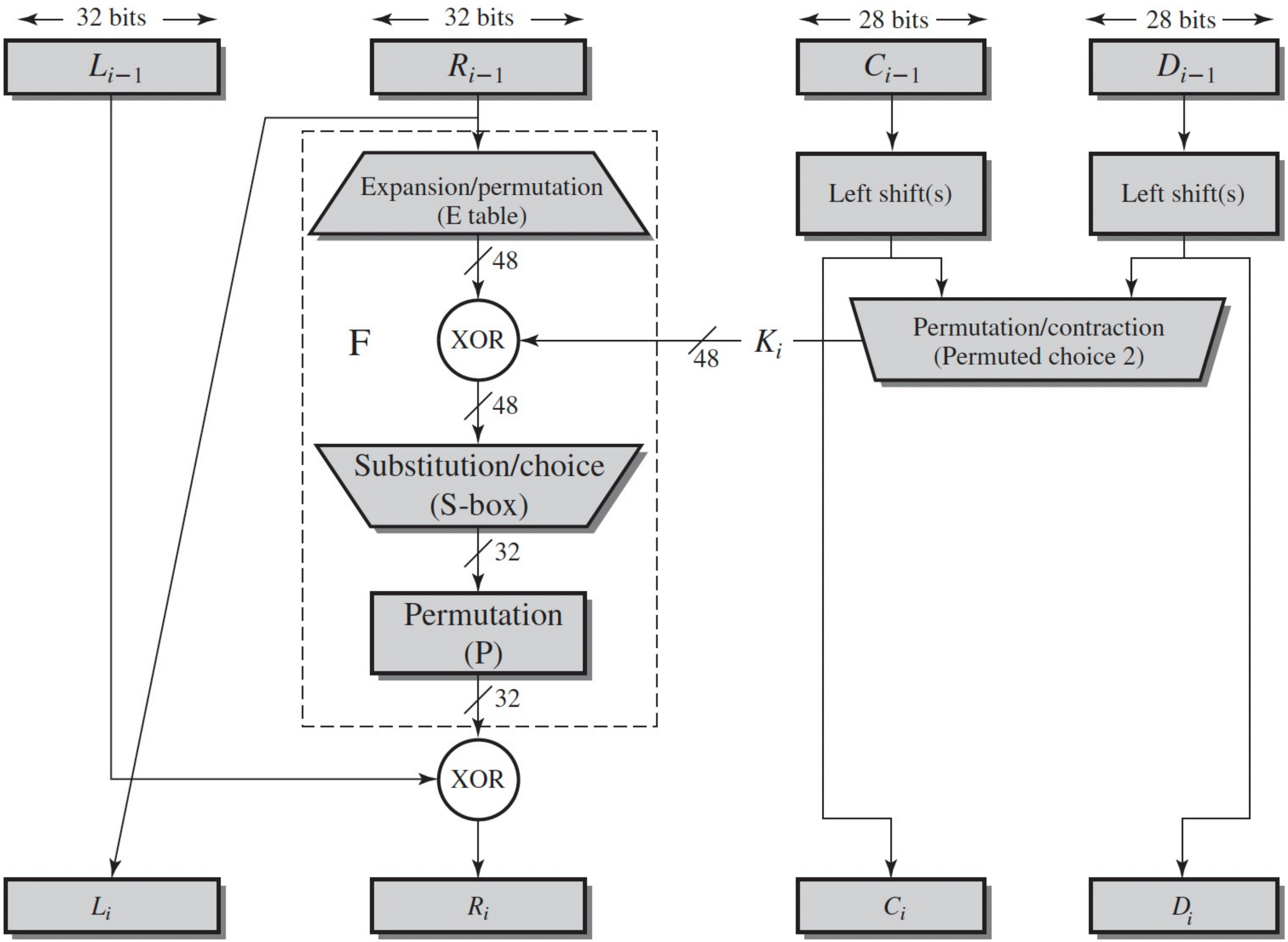
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25





DES: Internals of a Single Round

- Plaintext block of 64 bits is divided into two halves, L and R of 32 bits each.
- Input R is expanded to 48 bits (using Expansion Permutation table), where 16 of the R bits are duplicated.
- Expanded 48 bits are X-ORed with round key K_i.
- X-ORed 48-bits pass through a substitution function (by using S-box) that produces a 32-bit output.
- Those 32-bits are permuted as defined by the permutation table.
- Single round of DES algorithm is depicted in the following figure:



Single Round of DES Algorithm

DES: Internals of a Single Round

(c) Expansion Permutation (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

(d) Permutation Function (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: S-box Substitution

- The substitution consists of a set of eight S-boxes, each of which accepts 6 bits as input and produces 4 bits as output.
- The first and last bits of the input to box S_i form a 2-bit binary number to select one of four substitutions defined by the four rows in the table for Si .
- The middle four bits select one of the sixteen columns.

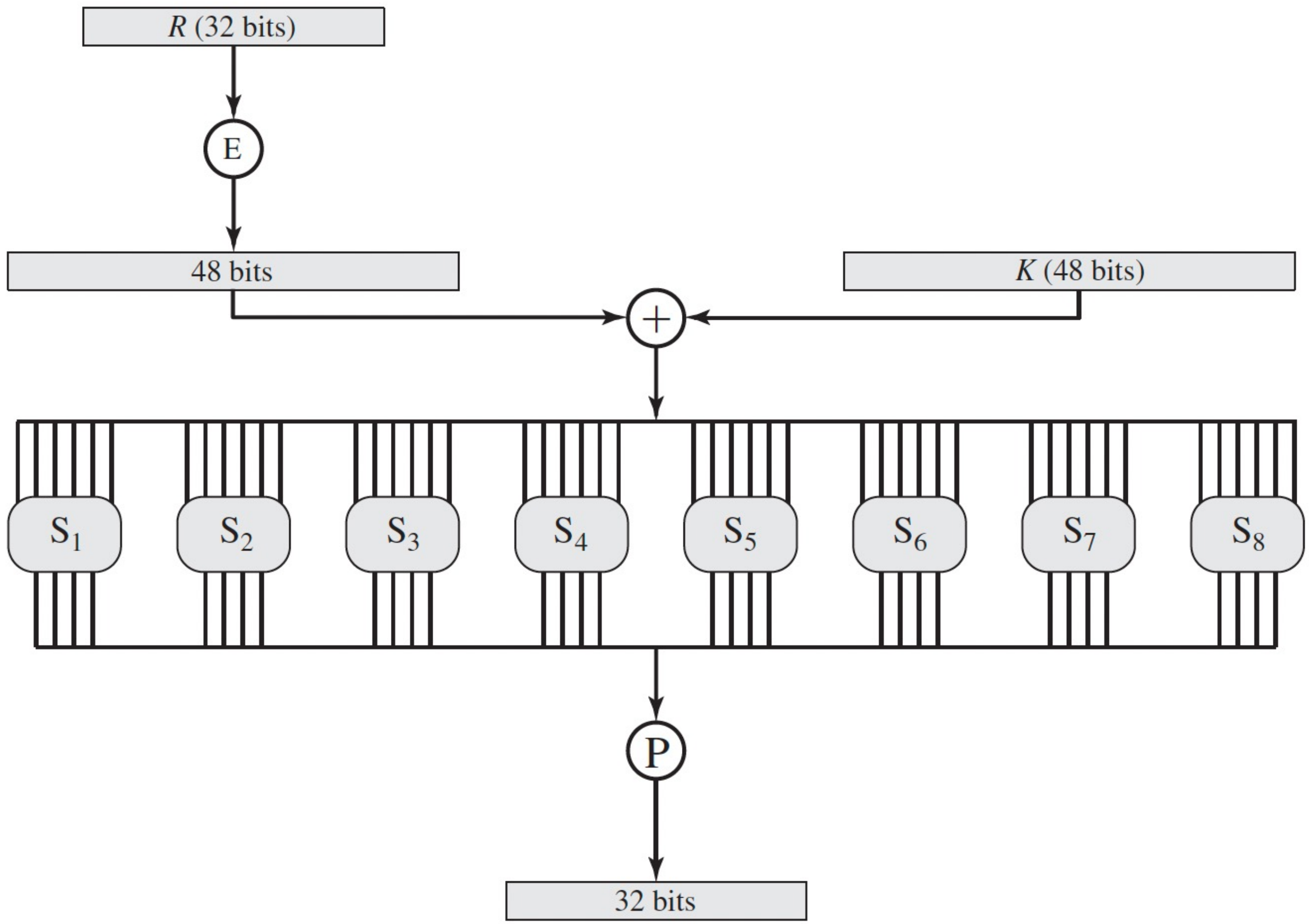
Middle bits

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
01	0	15	7	4	14	2	13	1	10	6	12	11	6	5	3	8
10	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
11	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

1st and last bits

For example, S_1(101010) = 6 = 0110.

DES: S-box Substitution



DES: S-box Substitution

	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
S_1	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

	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
S_2	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

	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
S_8	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 Generation

- A 64-bit key (numbered 1 – 64) is used as input to the algorithm.
- Every eighth bit (8, 16,... … , 56, 64) is ignored.
- The key is first subjected to a permutation governed by the table Permuted Choice One (PC-1).
- The resulting 56-bit key is then treated as two 28-bit quantities, labeled C_0 and D_0.
- At each round, C_{i-1} and D_{i-1} are separately subjected to a circular left shift or (rotation) of 1 or 2 bits, as governed by Shift tables.
- These shifted values serve as input to the table Permuted Choice Two (PC-2), which produces a 48-bit output that serves as input to the function F.
- They also serve as input to the next round.

DES: Key Generation

(a) Input Key

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

(b) Permuted Choice One (PC-1)

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

(c) Permuted Choice Two (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

(d) Schedule of Left Shifts

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

DES Decryption

- As with any Feistel cipher, decryption uses the same algorithm as encryption, except that the application of the subkeys is reversed.
- Compared to encryption, only the key schedule is reversed, i.e., in decryption round 1, subkey 16 is needed; in round 2, subkey 15; ……;etc.
- Thus, when in decryption mode, the key schedule algorithm has to generate the round keys as the sequence K_{16}, K_{15}, . . . ,K_1.
- Additionally, the initial and final permutations are reversed.

Avalanche Effect in DES

- Avalanche effect:
  - A small change in the plaintext or in the key results in a significant change in the ciphertext.
  - an evidence of high degree of diffusion and confusion
  - a desirable property of any encryption algorithm

- DES exhibits a strong avalanche effect
  - Changing 1 bit in the plaintext affects 34 bits in the ciphertext on average.
  - 1-bit change in the key affects 35 bits in the ciphertext on average.

DES: Security Issues

Key-length

- There are 2^{56} possible keys ($= 7.2 \times 10^{16}$ keys approx) - a brute-force attack is impractical.
- In July 1998, a special-purpose machine "DES cracker" that was made by Electronic Frontier Foundation (EFF) with $250,000 broke a DES encryption in less than three days.

S-boxes

- The design criteria for S-boxes, and for the entire algorithm, were not made public, so, there is a suspicion that the boxes were constructed in such a way that cryptanalysis is possible for an opponent who knows the weaknesses in the S-boxes.

DES: Security Issues

Other successful attacks on DES

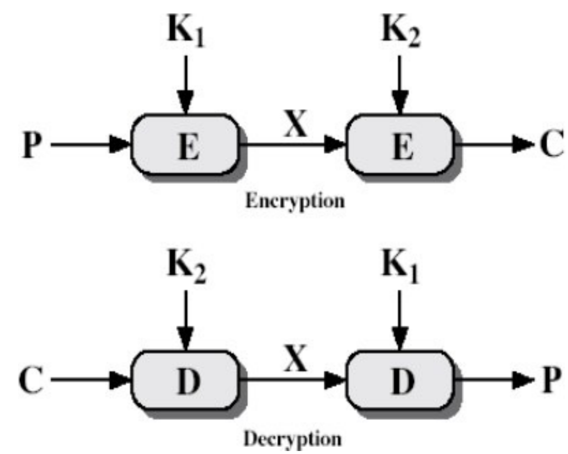
- Differential cryptanalysis
  - Possible to find a key with 2^{47} plaintext-ciphertext samples
  - Known-plaintext attack
- Linear cryptanalysis:
  - Possible to find a key with 2^{43} plaintext-ciphertext samples
  - Known-plaintext attack

Multiple Encryption with DES

- In 2001, NIST published the Advanced Encryption Standard (AES) to replace DES.
- But users in commerce and finance are not ready to give up on DES.
- As a temporary solution to DES's security problem, one may encrypt a message (with DES) multiple times using multiple keys:
  - 2DES is not much securer than the regular DES
  - So, 3DES with either 2 or 3 keys is used

2DES/Double DES

- DES uses a 56-bit key, this raised concerns about brute force attacks.
- One proposed solution: double DES (2DES)
- In double DES, DES is applied twice using two keys, K_1 and K_2.
- Encryption: $C = E_{K2}(E_{K1}(P))$
- Decryption: $P = D_{K1}(D_{K2}(C))$
- Key length: 56 x 2 = 112 bits



- This should have thwarted brute-force attacks?

Wrong!

Meet-in-the-Middle Attack on 2DES

- 2-DES: $C = E_{K2}(E_{K1}(P))$
  $C = D_{K2}(D_{K1}(P))$

- Given a known pair of plaintext/ciphertext (P, C), attack as follows:
  - Encrypt P with all 2^{56} possible keys for K_1.
  - Decrypt C with all 2^{56} possible keys for K_2.
  - If $E_{K1'}(P) = D_{K2'}(C)$, try the keys on another (P', C').
  - If works, (K1', K2') = (K1, K2) with high probability.
  - Takes $O(2^{56})$ steps; not much more than attacking 1-DES.

Triple DES with 2 Keys

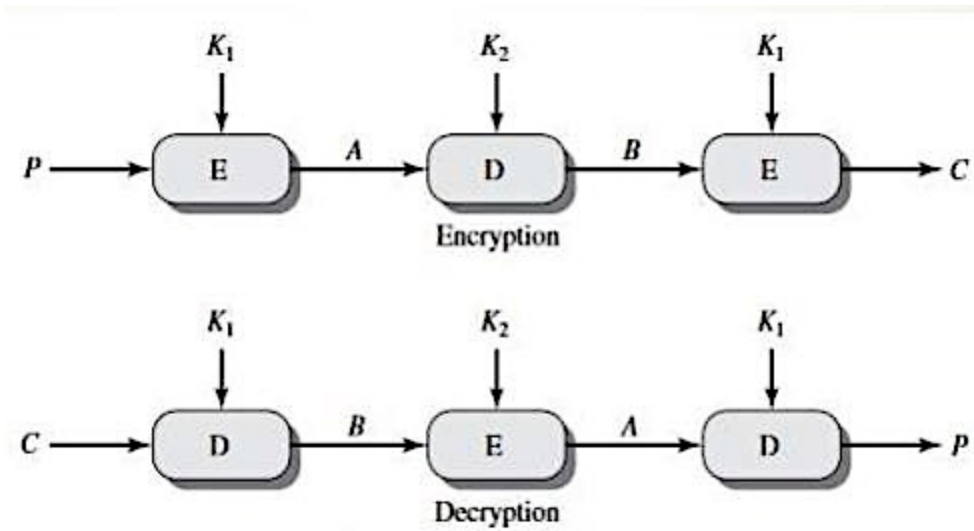
- Use three stages of DES for encryption and decryption.
- The 1st & 3rd stage use K_1 key and the 2nd stage uses K_2 key.
- To make triple DES compatible with single DES, the middle stage uses decryption in the encryption side and encryption in the decryption side.
- The function follows an encrypt-decrypt-encrypt (EDE) sequence

$$\mathbf{C = E_{K1}(D_{K2}(E_{K1}(P)))}$$
$$\mathbf{D = D_{K1}(E_{K2}(D_{K1}(C)))}$$

- It's much stronger than double DES with no practical attacks known.

Triple DES with 2 Keys

- Using triple DES with 2-key encryption, it raises the cost of meet-in-the-middle attack to 2^{112} attempts.

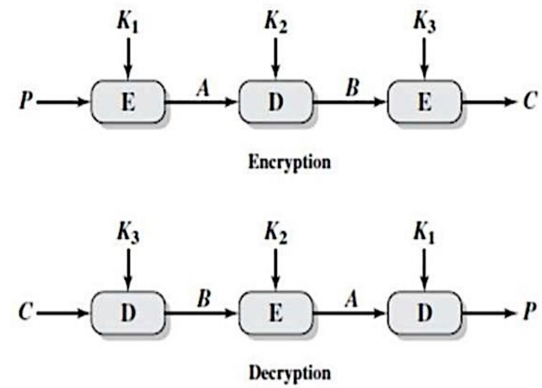


Triple DES with 3 Keys

- Uses three stages of DES for encryption and decryption with three different keys.
- Encryption and decryption in 3DES with 3 keys are as:

$C = E_{K3}(D_{K2}(E_{K1}(P)))$

$D = D_{K3}(E_{K2}(D_{K1}(C)))$



- If $K_1=K_2$, it becomes 3DES with 2 keys.
- If $K_1=K_2=K_3$, it becomes regular DES.
- So, it is backward compatible with both 3DES with 2 keys and the regular DES.
- Some internet application use 3DES with 3 keys. E.g., PGP and S/MIME