

Week 2 . Lecture Notes

- Topics:
- Data Encryption Standard (DES)
 - Triple DES and Modes of operation.
 - Stream Cipher
 - Pseudorandom Sequence

■ Data Encryption Standard (DES)

- DES is a 16-round Feistel cipher having block length 64: it encrypts a plaintext bitstring x (of length 64) using a 56-bit key, K , obtaining a ciphertext bitstring (of length 64).
- DES is the most widely used encryption scheme, adopted by NIST.
- Data is encrypted in 64-bit blocks using a 56 bit key and delivering a 64 bit output (FIPS PUB 46).

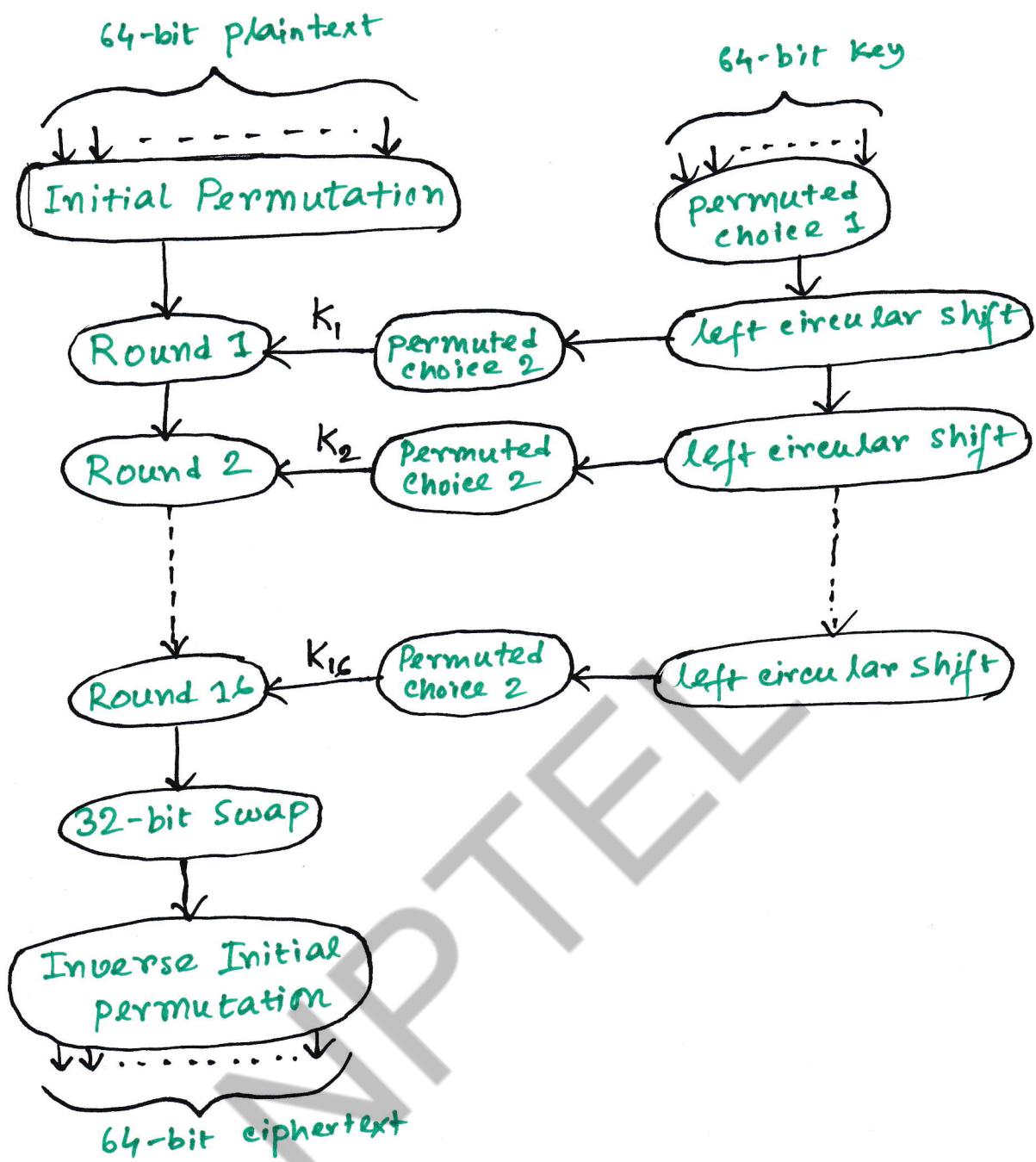


Figure 2: Block diagram of DES algorithm

- Overall scheme for DES is shown in figure 2. Two inputs are the plaintext (64 bits) and key (56 bits). The processing of the plaintext proceeds in three phases as can be seen from left hand side of the figure:

1. Initial permutation (IP) rearranging the bits to form the "permuted input".
 2. Followed by 16 iterations of the same function (substitution and permutation). The output of the last iteration consists of 64 bits which is a function of the plaintext and key. The left and right halves are swapped to produce the preoutput.
 3. Finally, the preoutput is passed through a permutation (IP^{-1}) which is simply the inverse of the initial permutation (IP). The output of IP^{-1} is the 64-bit ciphertext.
- The use of the key can be seen in the right hand portion of figure 2.
 - Initially the key is passed through a permutation function (PC_1 - shown in table 1).

• For each of the 16 iterations, a subkey (K_i) is produced by a combination of a left circular shift and a permutation (PC_2 - shown in table 1) which is the same for each iteration. However, the resulting subkey is different for each iteration because of repeated shifts.

• Detail of a single iteration: Following figure 3 and focusing on the LHS of the diagram, for the i th iteration of 16, the left and right halves are treated as separate 32 bit quantities L and R .

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

(a) Initial Permutation (IP)

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

(b) Inverse Initial Permutation (IP⁻¹)

Table I: Permutation tables used in DES

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

(c) Expansion Permutation (E)

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

(d) Permutation Function (P)

Table 1 (Cont...): Permutation tables used in DES

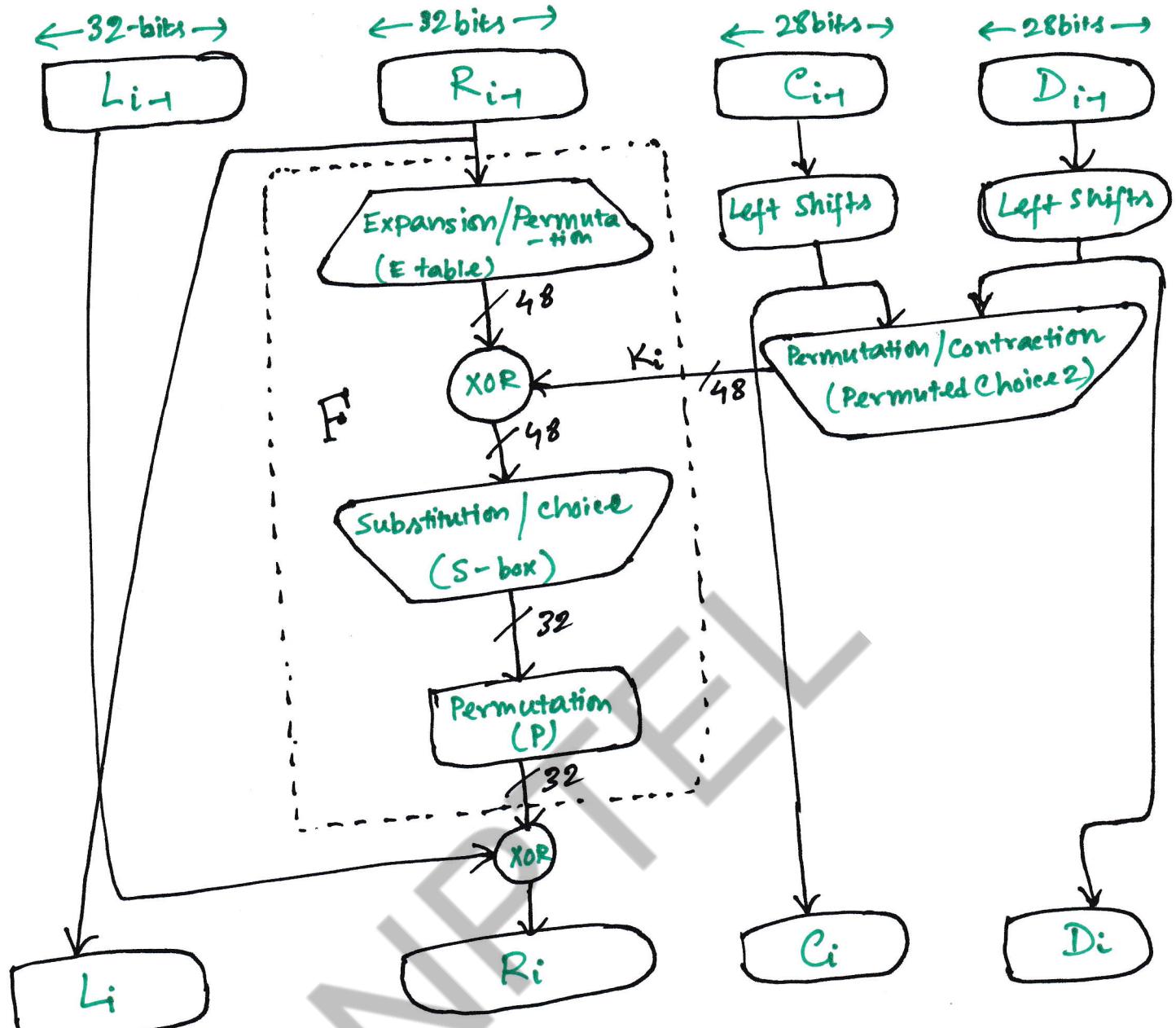


Figure 3: The details of one round of DES

- The overall processing at each iteration is:
 - $L_i = R_{i-1}$, i.e. the left hand output L_i of an iteration is the right hand input R_{i-1} .

- $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$.
- \oplus is the exclusive OR function - therefore the right hand output R_i is the exclusive OR of L_{i-1} and a complex function "F" of R_{i-1} and K_i .
- The function F is shown in figure 4. The subkey K_i used is 48 bits and the R_{i-1} input is first expanded to 48 bits using a table that defines a permutation plus an expansion (duplication of 16R plus an expansion (shown in table I)). The resulting 48 bits are XORed with K_i .

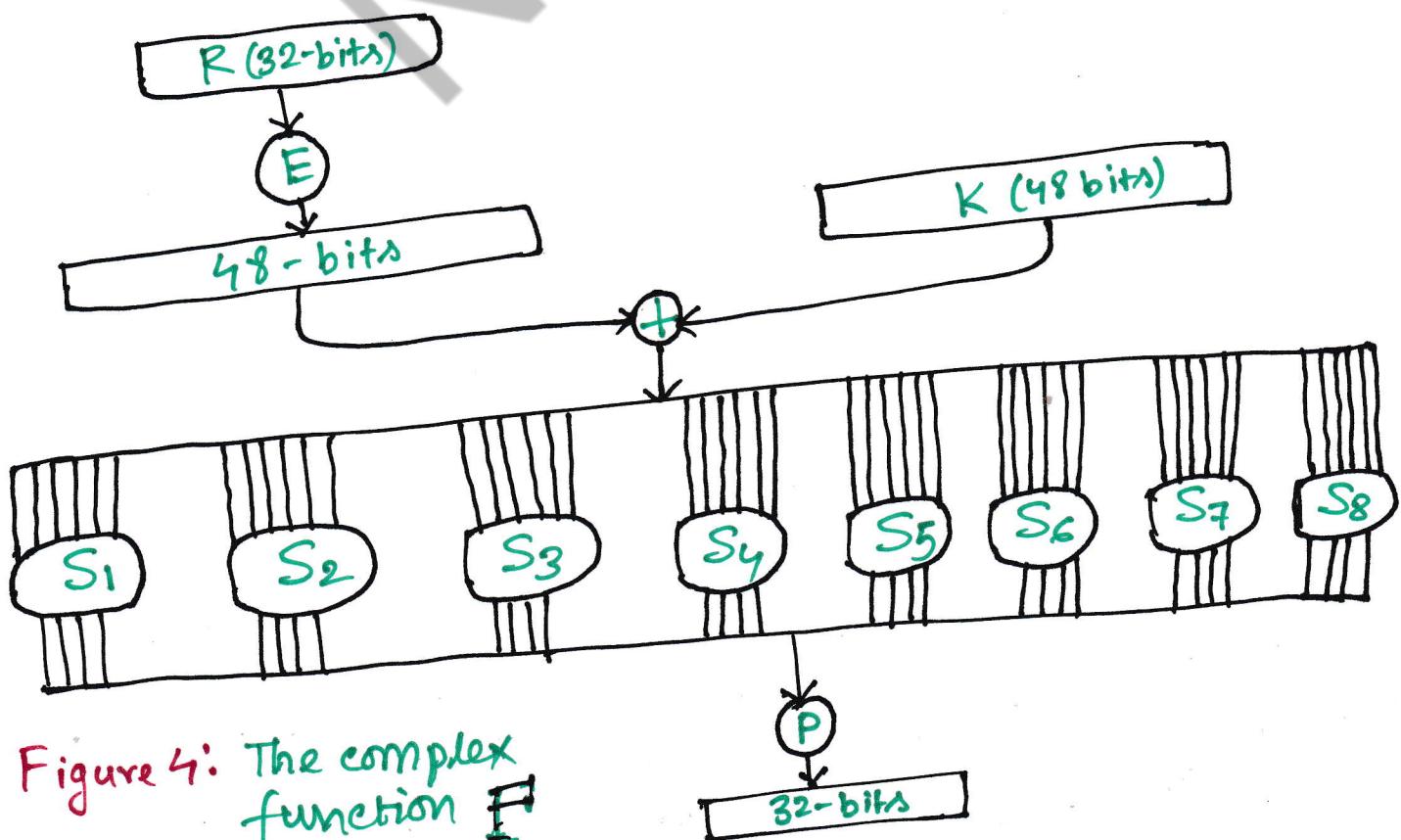
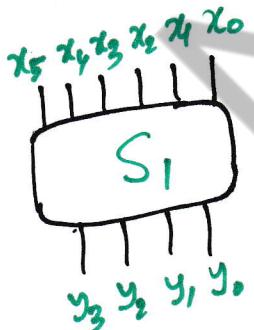


Figure 4: The complex function F

● Single iteration Continued :

- This 48-bit result passes through a substitution function that produces a 32 bit output which is then permuted as shown in table 1.

- The substitution function consists of 8 "S-boxes" each of which accepts 6 bits input and produces 4 bits of output (table 2). The first and last input bits of S_i select one of 4 rows and the four middle bits select the column. The contents of the selected cell constitute the 4 bit output.



$x_0x_5 \rightarrow$ Row number i
 $x_4x_2x_3x_4 \rightarrow$ column number j] in S_i

$$y_3y_2y_1y_0 = ((S_1)_{ij})_2$$

Example :-

If $x_5x_4x_3x_2x_1x_0 = 110101$ then $x_0x_5 = (3)_2$
 and $x_4x_2x_3x_4 = 0101$ ~~.....~~ $= (5)_2$. Thus i=3,
 $j=5 \Rightarrow (S_1)_{ij} = 09 \Rightarrow y_3y_2y_1y_0 = 1001$

Key generation is shown in figure 3 where the 56 bit key used as input to the algorithm is first permuted (PC_1 (table 3) and is then treated as two 28 bit quantities C_0 and D_0).

S_1	$\begin{bmatrix} 14 & 4 & 13 & 1 & 2 & 15 & 11 & 8 & 3 & 10 & 6 & 12 & 5 & 9 & 0 & 7 \\ 0 & 15 & 7 & 4 & 14 & 2 & 13 & 1 & 10 & 6 & 12 & 11 & 9 & 5 & 3 & 8 \\ 4 & 1 & 14 & 8 & 13 & 6 & 2 & 11 & 15 & 12 & 9 & 7 & 3 & 10 & 5 & 0 \\ 15 & 12 & 8 & 2 & 4 & 9 & 1 & 7 & 5 & 11 & 3 & 14 & 10 & 0 & 6 & 13 \end{bmatrix}$
-------	--

S_2	$\begin{bmatrix} 15 & 1 & 8 & 14 & 6 & 11 & 3 & 4 & 9 & 7 & 2 & 13 & 12 & 0 & 5 & 10 \\ 3 & 13 & 4 & 7 & 15 & 2 & 8 & 14 & 12 & 0 & 1 & 10 & 6 & 9 & 11 & 5 \\ 0 & 14 & 7 & 11 & 10 & 4 & 13 & 1 & 5 & 8 & 12 & 6 & 9 & 3 & 2 & 15 \\ 13 & 8 & 10 & 1 & 3 & 15 & 4 & 2 & 11 & 6 & 7 & 12 & 0 & 5 & 14 & 9 \end{bmatrix}$
-------	--

S_3	$\begin{bmatrix} 10 & 0 & 9 & 14 & 6 & 3 & 15 & 5 & 1 & 13 & 12 & 7 & 11 & 4 & 2 & 8 \\ 13 & 7 & 0 & 9 & 3 & 4 & 6 & 10 & 2 & 8 & 5 & 14 & 12 & 11 & 15 & 1 \\ 13 & 6 & 4 & 9 & 8 & 15 & 3 & 0 & 11 & 1 & 2 & 12 & 5 & 10 & 14 & 7 \\ 1 & 10 & 13 & 0 & 6 & 9 & 8 & 7 & 4 & 15 & 14 & 3 & 11 & 5 & 2 & 12 \end{bmatrix}$
-------	--

S_4	$\begin{bmatrix} 7 & 13 & 14 & 3 & 0 & 6 & 9 & 10 & 1 & 2 & 8 & 5 & 11 & 12 & 4 & 15 \\ 13 & 8 & 11 & 5 & 6 & 15 & 0 & 3 & 4 & 7 & 2 & 12 & 1 & 10 & 14 & 9 \\ 10 & 6 & 9 & 0 & 12 & 11 & 7 & 13 & 15 & 1 & 3 & 14 & 5 & 2 & 8 & 4 \\ 3 & 15 & 0 & 6 & 10 & 1 & 13 & 8 & 9 & 4 & 5 & 11 & 12 & 7 & 2 & 14 \end{bmatrix}$
-------	--

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	19	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	3	2	12

S_8	13	2	8	4	6	15	31	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

Table 2 : S-box details

1	2	3	4	5	6	7	8
9	10	12	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	45	45	47	48	
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

(a) Input Key

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

(b) Permutated choice one (PC1)

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

(c) Permutated choice Two (PC2)

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

Table 3: DES Key Schedule.

④ Single iteration cont:

- At each iteration C_i and D_i are subjected to a circular left shift or rotation of 1 or 2 bits (table 3). They serve as the input to the next iteration and also are input into Permutated choice 2 (PC_2) whose output serves as K_i the input to E' .

• Attacks on Cryptosystems :

- Ciphertext-only attack: In this attack the attacker knows only the ciphertext to be decoded. The attacker will try to find the key or decrypt one or more pieces of ciphertext.
- Known plaintext attack: The attacker has a collection of plaintext-ciphertext pairs and is ~~try~~ trying to find the key or to decrypt some other ciphertext that has been encrypted with the same key.
- Chosen plaintext attack: Here the attacker can choose the plaintext to be encrypted and read the corresponding ciphertext.
- Chosen ciphertext attack: The attacker has the ability to select any ciphertext and study the plaintext produced by decrypting them.
- Chosen text attack: The attacker has abilities required in previous two attacks.

Exhaustive search attack on DES:

- A brute force method which will try for all possible keys.
- The resistance against exhaustive search depends on:
 - size of the key space.
 - implementation in software or special purpose hardware.
 - the number of parallel processors available.
 - the speed at which each key can be processed.
 - the cost of each processor and the overall cost of implementing the attack.

So, key size of DES = 2^{56}

If 2^{32} parallel processors available
then time to mount exhaustive search attack = $2^{56-32} = 2^{24}$ sec
(assuming encryption oracle runs in 1 sec)

- DES decryption : The decryption process with DES is essentially the same as the encryption process and is as follows:

Use the ciphertext as the input to the DES algorithm but use the keys K_i in reverse order. That is use K_{16} on the first iteration, K_1 on the second until K_1 which is used on the 16th and the last iteration.

- Triple DES : Due to exhaustive search attack on DES, Triple DES was realised.

- Triple DES is simply applying the DES algorithm three times using two different keys (see figure 9) :

$$C = E_{K_1} [D_{K_2} [E_{K_1} [P]]]$$

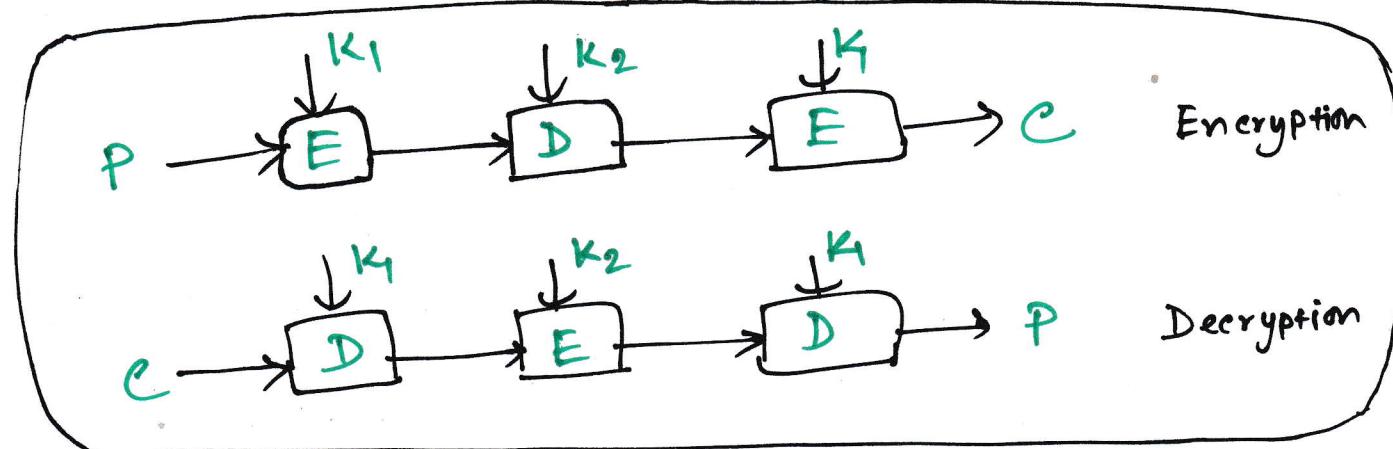


Figure 9: Triple DES

- The order of encryptions and decryptions determines name EDE or Encryption-Decryption-Decryption.

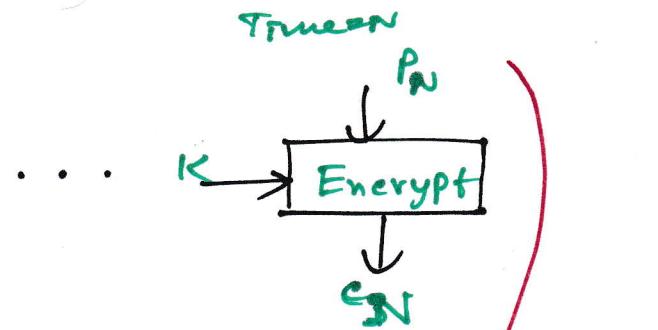
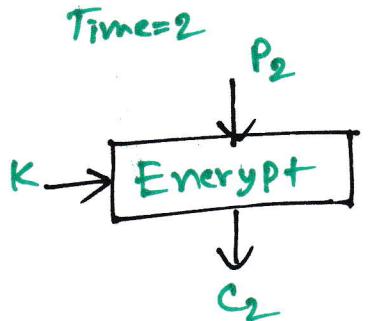
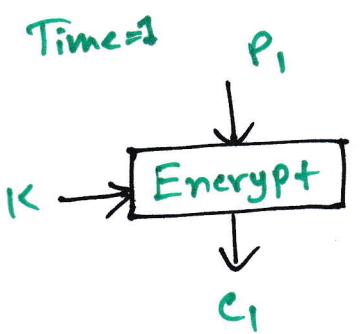
- Decryption is performed on the second iteration is simply for backward compatibility and offers no extra security to the algorithm.

- This can be seen from :

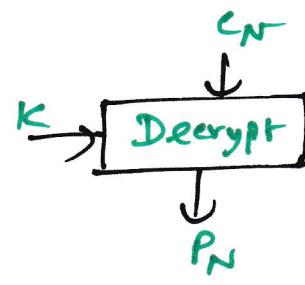
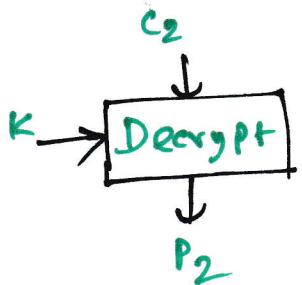
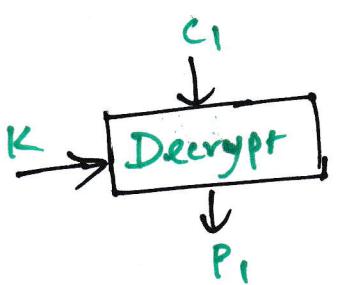
$$C = E_{K_3}[D_{K_1}[E_{K_2}[P]]] = E_{K_3}[P].$$

Modes of Operation:

<u>Mode</u>	<u>Description</u>
Electronic Codebook (ECB)	Each block of 64 plaintext bits is encoded independently using the same key
Cipher Block Chaining (CBC)	The input to the encryption algorithm is the XOR of the next 64-bits of plaintext and the preceding 64 bits of ciphertext.
Cipher Feedback (CFB)	Input is processed J bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output. which is XORed with plaintext to produce next unit of ciphertext
Output Feedback (OFB)	Similar to CFB, except that the input to the encryption algorithm is the preceding DES output.

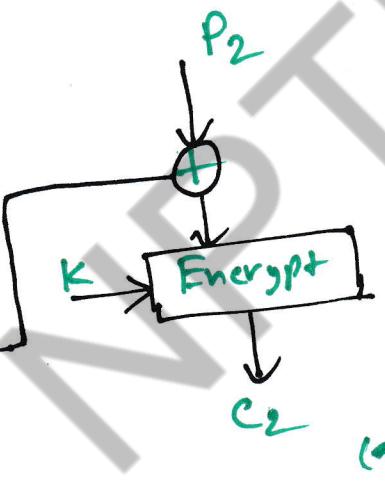
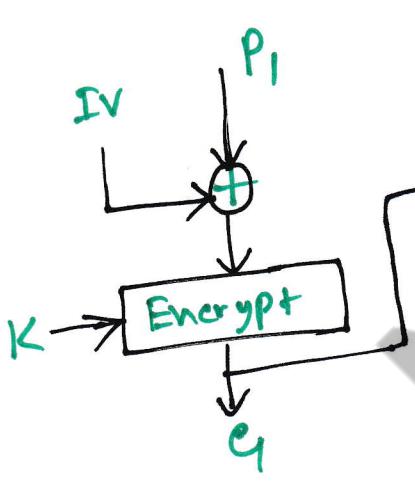


(a) Encryption

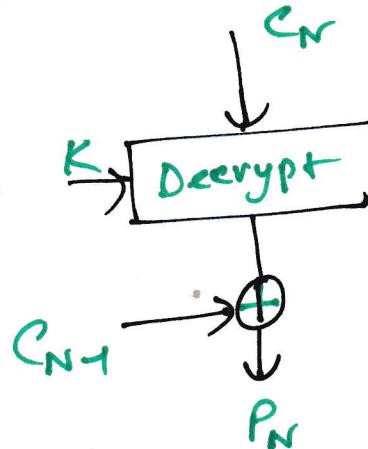
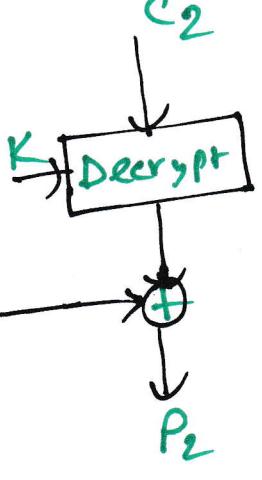
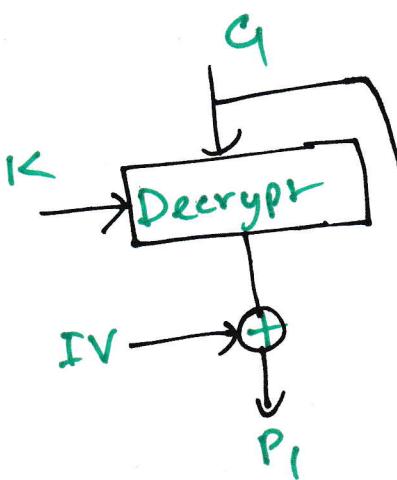
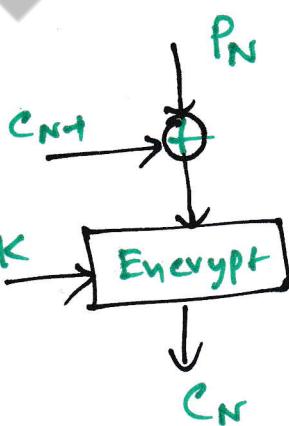


ECB mode

(b) Decryption

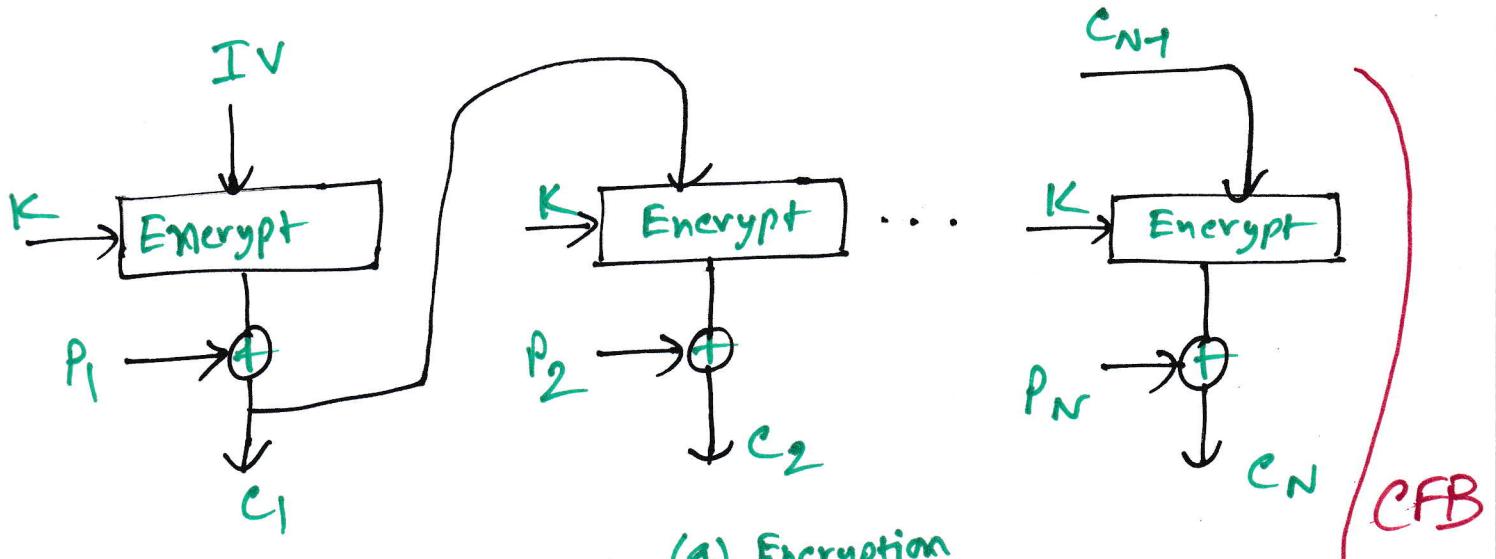


(a) Encryption



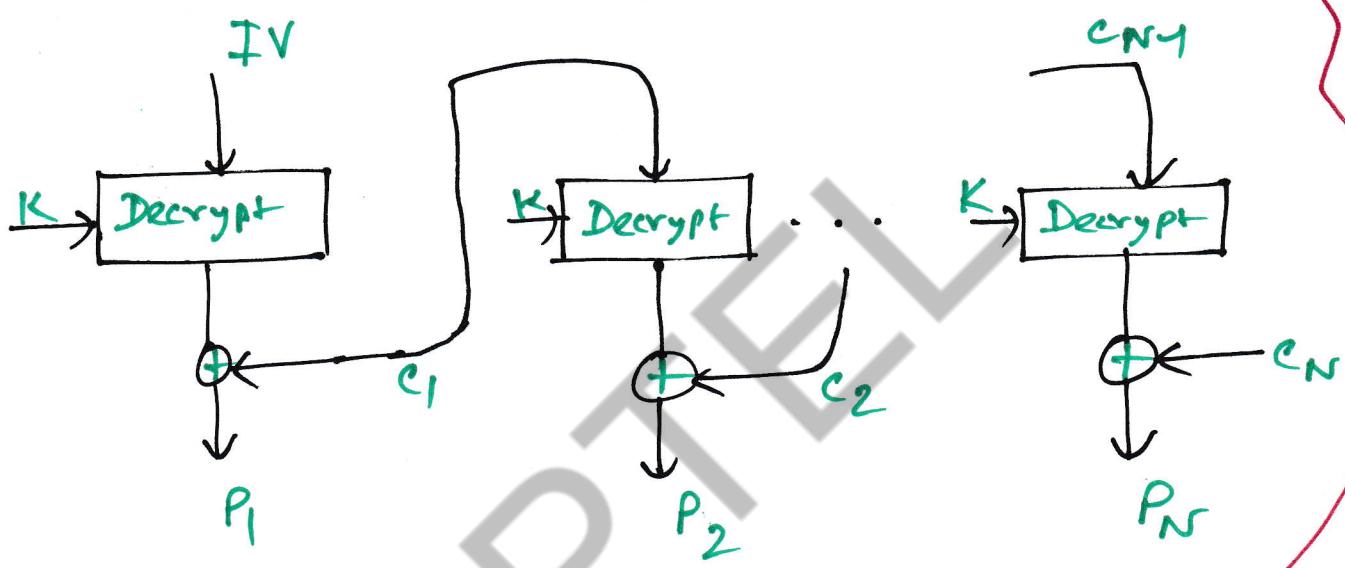
CBC mode

(b) Decryption



(a) Encryption

CFB
mode



(b) Decryption

• Stream Ciphers :

- Plaintext : binary string.
- Key stream: a pseudorandom bit string.
- Ciphertext: bit-wise XOR (addition modulo 2) of plaintext and key stream.
- Decryption: bit-wise XOR of ciphertext and key stream.

• Example :

P : 100011010101111011011

K : 010010101101001101101

C : 11000111000110110110

where $c = p \oplus k$ (encryption)

$p = c \oplus k$ (decryption)

Shannon's Notion of Perfect Secrecy :

- $\Pr(x|y) = \Pr(x)$ for all x and for all y where x is a plaintext and y is a ciphertext.
- The basic strength of stream-cipher lies in how "random" the key-stream is.
- If k_i is a true random sequence, then the cipher is called an one-time pad.
- One-time pad possesses perfect secrecy. However one-time pad is impractical.
- Main objective of a stream cipher construction is to get k_i as random as possible.

Illustration :

One bit encryption $C = P \oplus K$

$$\Pr(K=0) = \Pr(K=1) = \frac{1}{2}$$

$$\text{Let } \Pr(P=0) = 0.6, \quad \Pr(P=1) = 0.4$$

$$\begin{aligned}
 \Pr(P=0 | C=1) &= \frac{\Pr(P=0, C=1)}{\Pr(C=1)} \\
 &= \frac{\Pr(P=0, C=1)}{\Pr(P=0, C=1) + \Pr(P=1, C=1)} \\
 &= \frac{\Pr(P=0) \cdot \Pr(C=1 | P=0)}{\Pr(P=0) \cdot \Pr(C=1 | P=0) + \Pr(P=1) \Pr(C=1 | P=1)} \\
 &= \frac{\Pr(K=1) \cdot \Pr(P=0)}{\Pr(P=0) \Pr(K=1) + \Pr(P=1) \cdot \Pr(K=0)} \\
 &= \frac{\frac{1}{2} \times 0.6}{\frac{1}{2} \times 0.6 + \frac{1}{2} \times 0.4} = 0.6
 \end{aligned}$$

Randomness Measurements :

- Randomness of sequence : unpredictable property of sequence.
- Aim is to measure randomness of the sequence generated by a deterministic method called a generator.
- The test is performed by taking a sample output sequence and subjecting it to various statistical tests to determine whether the

sequence possess certain kinds of attributes, a truly random sequence would be likely to exhibit.

- This is the reason the sequence is called pseudo-random sequence instead of random sequence and the generator is called pseudo-random sequence generator (PSG).
- The sequence $s = s_0, s_1, s_2, \dots$ is said to be periodic if there is some positive integer N such that $s_{i+N} = s_i$ and smallest N is called the period of sequence.

● Golomb's Randomness Postulates:

R-1: In every period the number of 1's differs from the number of 0's by at most 1. Thus $\left| \sum_{i=0}^{N-1} (-1)^{s_i} \right| \leq 1$.

R-2: In every period, half the runs have length 1, $\frac{1}{4}$ th have length 2, $\frac{1}{8}$ th have length 3, etc., as long as

the number of runs so indicated exceeds 1. Moreover for each of these lengths, there are almost equally many runs of 0's and of 1's.

R-3: The auto-correlation function

$$C(\tau) = \sum_{i=0}^{N-1} (-1)^{s_i + s_{i+\tau}}$$

is two-valued.

Explicitly

$$C(\tau) = \begin{cases} N, & \text{if } \tau \equiv 0 \pmod{N} \\ T, & \text{if } \tau \not\equiv 0 \pmod{N} \end{cases}$$

where T is constant.

- Example: Consider the periodic sequence s of period 15 with cycle

$$s^{15} = 011001000111101$$

R-1: There are seven 0's and eight 1's

R-2: Total runs is 8. 4 runs of length 1 (2 for each 0's and 1's), 2 runs of length 2 (1 for each 0's and 1's), 1 run of 0's of length 3 and 1 run of 1's of length 4.

R-3: The function $C(\tau)$ takes only two values:

$$C(0) = 15 \text{ and } C(\tau) = -1 \text{ for } 1 \leq \tau \leq 14.$$