[CS304] Introduction to Cryptography and Network Security

Course Instructor: Dr. Dibyendu Roy — Winter 2022-2023
Scribed by: Rathva Kaushikkumar Sanjaybhai (202051156) — Week : 3 (5-6th lecture #)

1 OTP(One time padding):

OTP provides perfect secrecy under some condition.

1.1 Encryption :

$P \rightarrow$ plain text
$K \rightarrow$ Secret key

$$Enc(P,k) \ = \ p \oplus k = \ C$$

1.2 decryption :

$Dec(C,k) \ = \ C \oplus k = \ CP$

$$P_r[message|Ciphertext] = P_r[message]$$

1.3 Conditions for it to have perfect secrecy:

1. The secret key k can't be used to encrypt the message.
2. length(k) $\geq$ length(p)
3. k is uniformly selected from any space.

2 OTP on one bit of message :

message $\rightarrow$ $m \ \epsilon \ \{0,1\}$ where, key k $\epsilon \ \{0,1\}$

$$P_r[m=0] \ = \ P \qquad P_r[k=0] = 0.5$$

$$P_r[m=1] \ = \ 1-P \qquad P_r[k=1] = 0.5$$

2.1 Encryption :

$C \ = \ m \bigoplus \ k$

for cipher text to be 0 :
either m = k = 0 or m = k = 1 are 2 possibilities.

So ,

$$\begin{aligned} P[C=0] \ &= \ P_r[m=0, k=0] \ + \ P_r[m=1, k=1] \\ &= \ P_r[m=0] \cdot P_r[k=0] \ + \ P_r[m=1] \cdot P_r[k=1] \\ &= \ P \times (0.5) \ + \ (1-P) \times 0.5 \end{aligned}$$

similar can be proven for C = 1.

$P_r[M=m|C=c] = P_r[M=m]$

1. $P_r(A/B) \ = \ \frac{P_r(AB)}{P_r(B)}$
2. $P_r(AB) \ = \ P_r(B/A) \cdot P(A)$

2.2 Perfect secrecy of OTP:

$$P_r[M=0|C=0] = \frac{P_r[M=0, C=0]}{P_r[C=0]}$$

$P_r[M=0, C=0] \ = \$ Probability of M and C being 0.

$$= \frac{P_r[C=0|M=0] \times P_r[M=0]}{1/2}$$

Here,We are assuming that $P_r[C=0|M=0] \ = \ 0.5$.

$$= \frac{\cancel{1/2} \times P_r[M=0]}{\cancel{1/2}}$$

$$= P_r[M=0]$$

C depends on k and M given m=0 k can be 1 or 0.

$P_r[M=0|C=0] \ = \ P_r[M=0]$ So it provides perfect secrecy.

2.3 OTP with out Condition :

1. Reuse secret key.

   $$M_1 \bigoplus k = C_1$$
   $$M_2 \bigoplus k = C_2$$

   $$C_1 \bigoplus C_2 = (M_1 \bigoplus k) \bigoplus (M_2 \bigoplus k)$$
   $$= M_1 \bigoplus M_2$$

   So the xor of cipher texts will give diffrence between cipher text and message/plain text.

2. length of key $\geq$ length of plain text.
   let's suppose len(k) $<$ len(P)

   $$C \ = \ P \oplus \ k$$

   $$\begin{array}{ccl} & P \ = & p_1\ p_2\ .\ .\ .\ .\ p_l\ p_{l+1}\ .\ .\ .\ p_n \\ \bigoplus & k \ = & k_1\ k_2\ .\ .\ .\ .\ k_l\ k_1\ .\ .\ .\ k_t \end{array}$$

   $$C \ = \ (p_1 \bigoplus k_1)(p_2 \bigoplus k_2) \ldots (p_l \bigoplus k_l)(p_{l+1} \bigoplus k_1) \ldots (p_n \bigoplus k_t)$$

3. If we take k from a non-uniformly.The $P_r[C = 0|M = 0]$ will not be 0.5.
   So $P_r[M = 0|C = 0]$ will not be equal to $P_r[M = 0]$ and OTP will not have perfect secrecy.

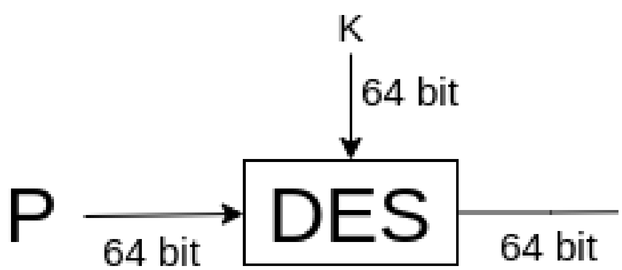
3 Data Encryption Standard (DES):

- It's a block cipher Designed by IBM in 1970s and was proprietary until 1977.

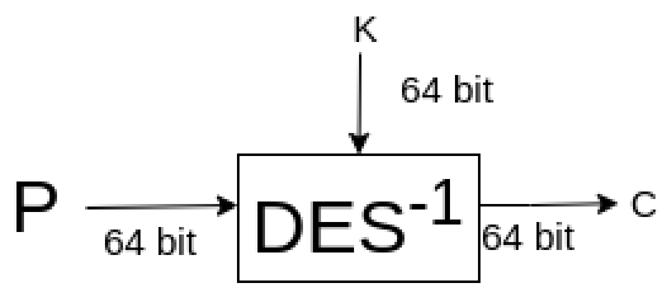
Characteristics of DES :

1. Block size = 64 bit.
2. Number of rounds = 16
3. Secret key size = 64 bit
   Out of 64 key consists of 56 bit actual key and 8 bit parity bits.
4. It's based on Feistel Network.

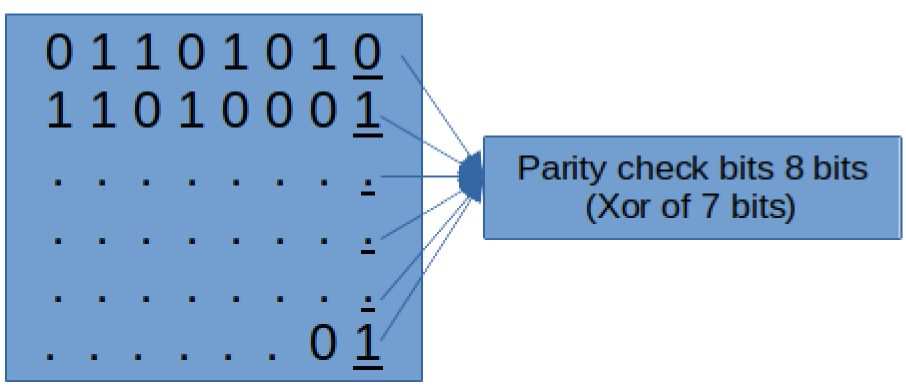
3.1 Encryption :



3.2 decryption :



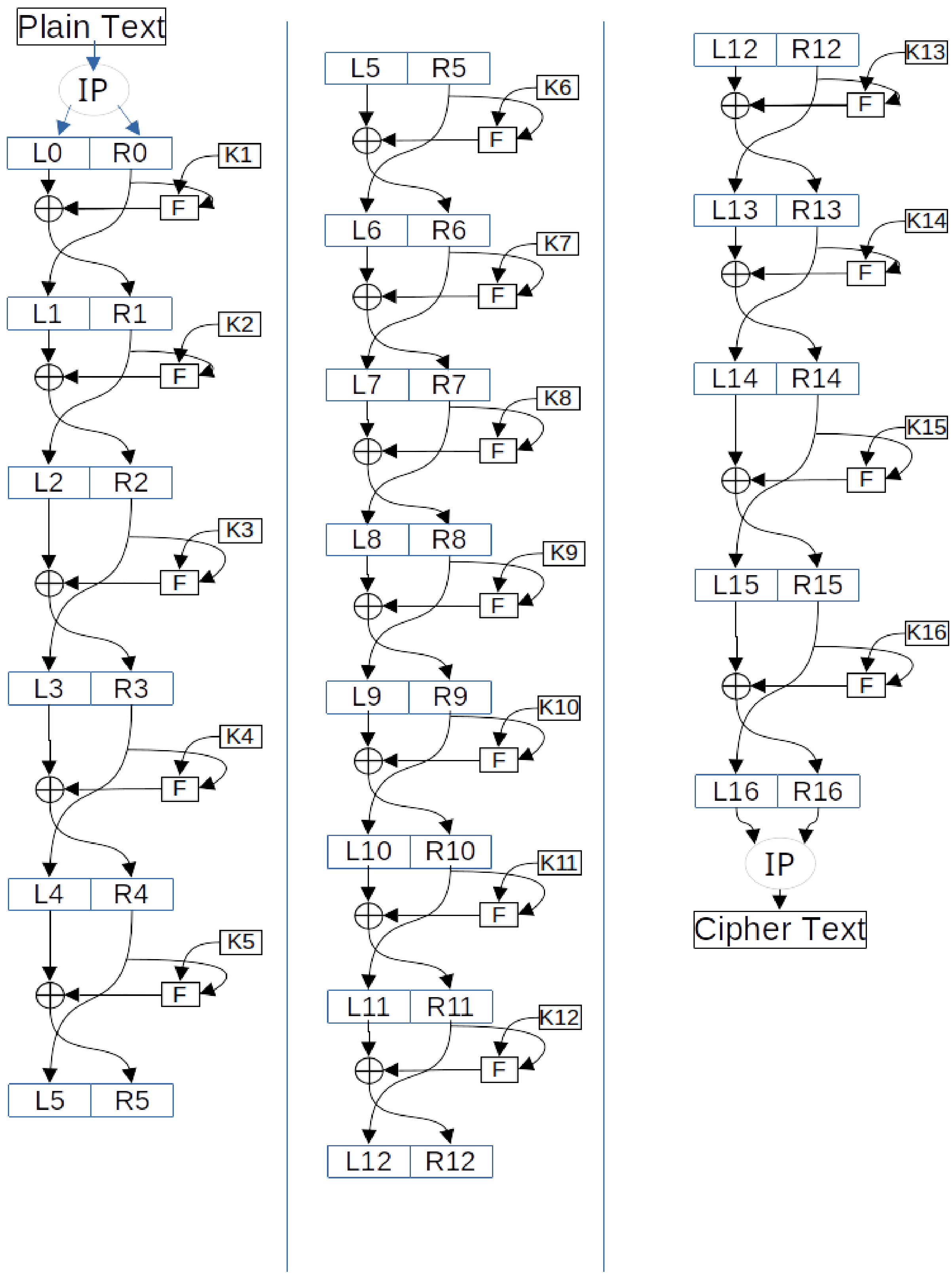
3.3 Parity Check :



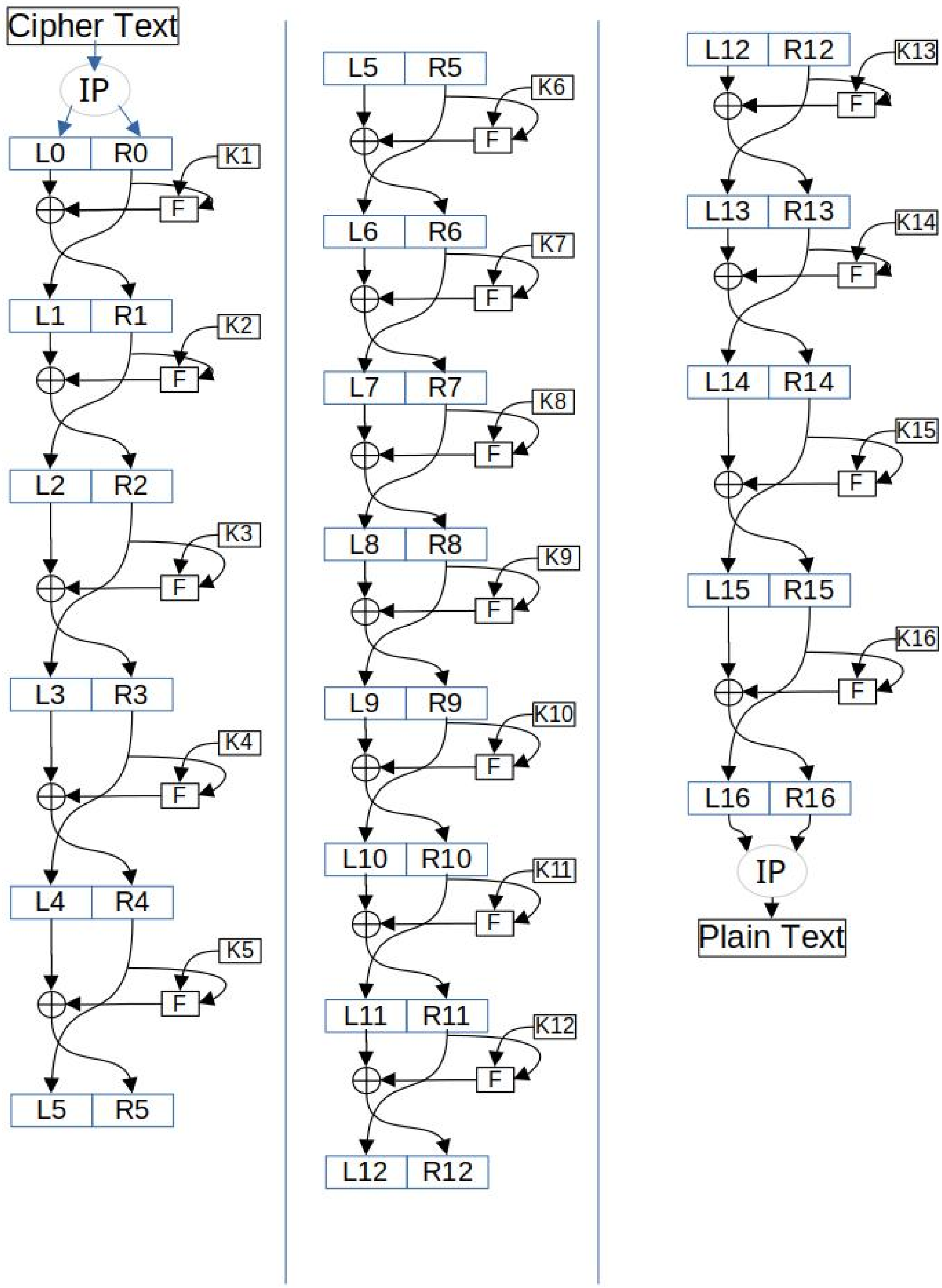
After discarding 8 parity bits we have Secret key of length 58 bits.
In DES 16 round keys $k_1 - k_{16}$are generated by key scheduling function G(n) which takes secret key as input.

4 Structure of DES :

4.1 Encryption :



4.2 Decryption :



4.3 IP (Initial Permutation):

IP : $\{0,1\}^{64} \rightarrow \{0,1\}^{64}$

IP lookup table

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

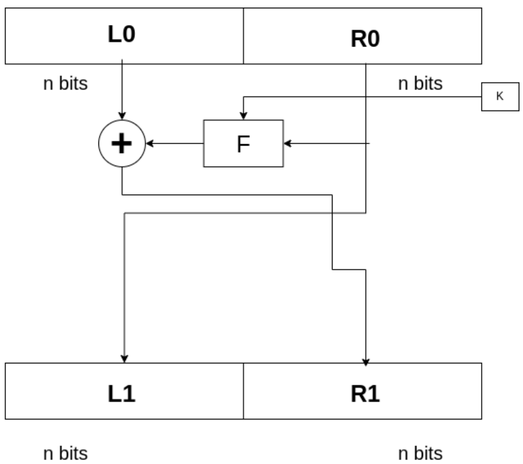
IP$(m_1m_2 \ . \ . \ . \ m_{64}) \ = \ (m_58m_50 \ . \ . \ . \ m_15m_7)$

IP^{-1} lookup table

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

$IP^{-1}(m_1m_2 \ . \ . \ . \ m_{64}) \ = \ (m_40m_8 \ . \ . \ . \ m_57m_25)$

4.4 Round function of DES :



$f : \{0,1\}^{32} \times \{0,1\}^{48} \rightarrow \{0,1\}^{32}$
$f : (R_i, k_i) \ = \ X_i$
where, R_i is 32 bit.
k_i is 48 bit.
X_i is 32 bit.

$$f(R_i, k_i) = P(S(E(R_i) \bigoplus k_i))$$

where, $E : Expenionfunction : \{0,1\}^{32} \rightarrow \{0,1\}^{48}$
$S : Sbox : \{0,1\}^{48} \rightarrow \{0,1\}^{32}$
$P : Permutation : \{0,1\}^{32} \rightarrow \{0,1\}^{32}$

4.5 Expansion function :

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

$E(x_1 x_2 . . . x_{32})$
$= \{x_{32}\ x_1\ . . .\ x_4\ x_5\ . . .\ x_{32}\ x_1\ \}$

4.6 Sbox :

S(x) = y, where x is 48 bit and y is 32 bit.
$X = B_1\ B_2\ B_3\ B_4\ B_5\ B_6\ B_7\ B_8$
Where length of B_i is 64 bit.
$S_1\ S_2\ S_3\ S_4\ S_5\ S_6\ S_7\ S_8$
$S_i(B_i) = C_i$
$S_i : \{0,1\}^6 \rightarrow \{0,1\}^4\ where, i = 1,2,3,...,8$
$S(x) = (S_1(B_1),\ . . .\ , S_8(B_8))$
$B_i = b_1\ b_2\ b_3\ b_4\ b_5\ b_6 \quad b_i \epsilon \{0,1\}$
$r = (2 \times b_1 + b_6)$ it's just interger representation of $b_1 b_6$
c is integer representation of $b_2 b_3 b_4 b_5$.
here, $0 \leq r \leq 3\ and\ 0 \leq c \leq 15$

Now using following table compute the $S_i\ from B_i$.

row	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]
									S_1							
[0]	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
[1]	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
[2]	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
[3]	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
									S_2							
[0]	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
[1]	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
[2]	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
[3]	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
									S_3							
[0]	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
[1]	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
[2]	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
[3]	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
									S_4							
[0]	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
[1]	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
[2]	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
[3]	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
									S_5							
[0]	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
[1]	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
[2]	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
[3]	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
									S_6							
[0]	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
[1]	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
[2]	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
[3]	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
									S_7							
[0]	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
[1]	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
[2]	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
[3]	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
									S_8							
[0]	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
[1]	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
[2]	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
[3]	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

4.7 Permutation :

$P : \{0,1\}^{32} \rightarrow \{0,1\}^{32}$

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

$P(x_1\ x_2\ x_3\ .\ .\ .\ .\ x_32) \ = \ (x_{16}\ x_7.\ .\ .\ .\ x_{22}\ x_{11}\ x_4\ x_{25})$

4.8 Des summary:

1. 16 rounds
2. 64 bit block size
3. key size of 64 bits
4. IP and IP^{-1}
5. Round function
6. Key scheduling algorithm

5 Key scheduling algorithm DES:

Input : 64 bit key k.
Output : 16 round keys.

1. Define $V_i,\ 1 \leq i \leq 16$
   if i ϵ 1,2,9,16
   $V_i = 1$
   else $V_i = 2$
2. Discard 8 parity bits from k.
3. $T = PC_1(\tilde{k}) \quad PC_1 : \{0,1\}^{56} \rightarrow \{0,1\}^{56}$
4. $(C_0, D_0) = T \quad Where$ C_0 is of 28 bit and D_0 is of 28 bit.
5. for i = 1 to 16
   $C_i = (C_{i-1} \hookleftarrow v_i)$
   $D_i = (D_{i-1} \hookleftarrow v_i)$

   $K_i = PC_2(C_i, D_i)$
   $PC_2 : \{0,1\}^{56} \rightarrow \{0,1\}^{48}$
6. Round key = $k_1\ k_2\ k_3\ k_4\ .\ .\ .\ .\ k_{16}$

PC1						
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
above for C_i; below for D_i						
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

PC2					
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