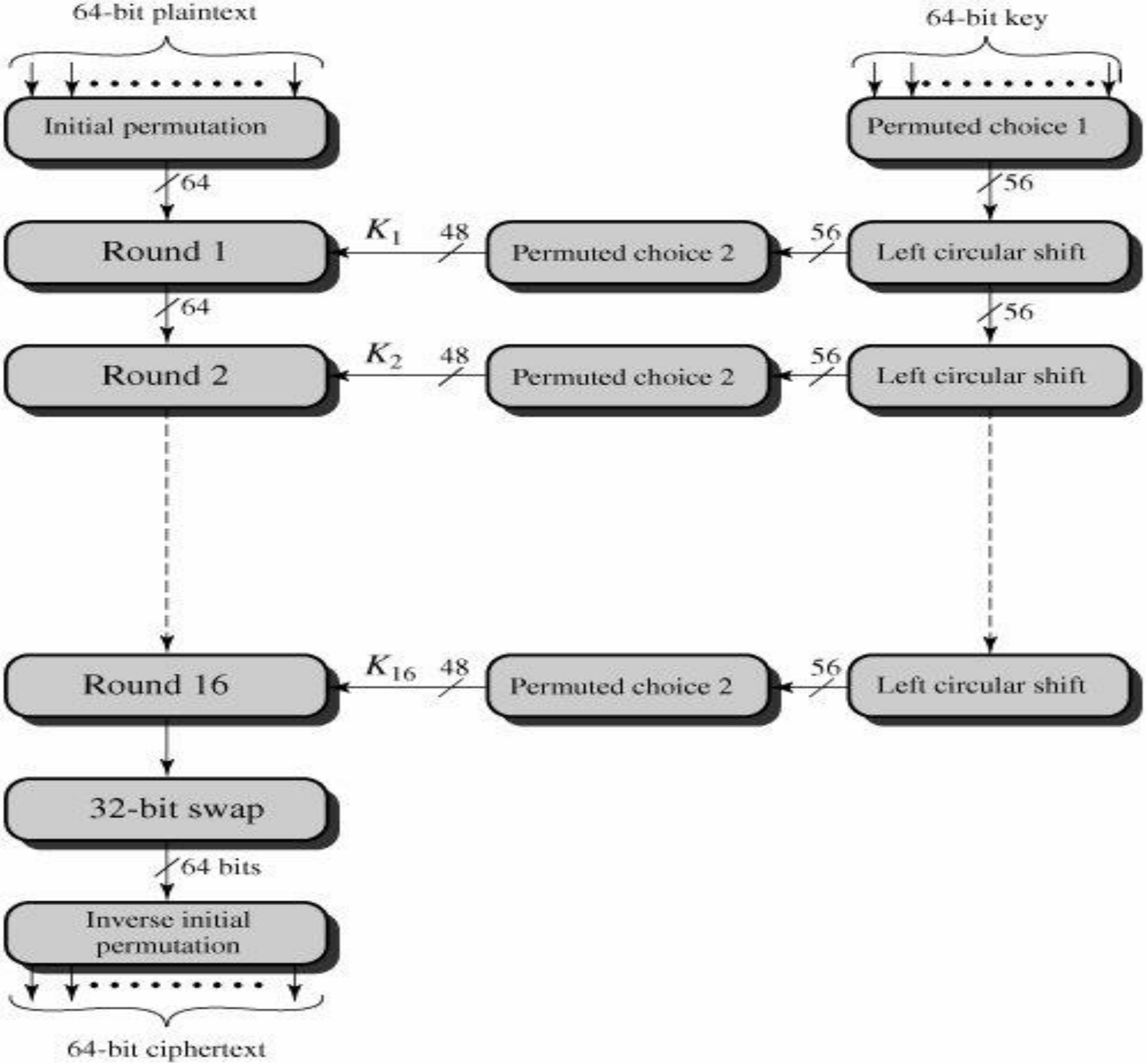
DES (Data Encryption Standard)

- Basics
  - DES is a Fiestel type of Substitution – Permutation Network (SPN) cipher.
  - It was approved by federal standard in November 1976, It was published in 15th January 1977, adopted in 1977 by national bureau of standard and now NIST(National Institute of Standard and Technology) .
  - Data encrypted in 64-bit blocks using 56 bit key.
  - The algorithm transforms 64-bit input in a series of steps into a 64-bit output. The same steps, with the same key, are used to reverse the encryption

- DES Encryption Process
  - There are two input to the encryption function
    - Plaintext to be encrypted (64 bit length)
    - Key (56 bits)
  - The processing of the plaintext proceeds in three phases.
    - First, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the *permuted input* .
    - This is followed by a phase consisting of 16 rounds of the same function, which involves both permutation and substitution functions.
    - The output of the last (sixteenth) round consists of 64 bits that are a function of the input plaintext and the key

- The left and right halves of the output are swapped to produce the **preoutput** .
- Finally, the preoutput is passed through a permutation (IP^{-1}) that is the inverse of the initial permutation function, to produce the 64-bit cipher-text.

- **Right halve**
  - The 56-bit key is used. Initially, the key is passed through a permutation function.
  - Then, for each of the 16 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



64-bit plaintext
64-bit key
Initial permutation
Permuted choice 1
64
56
Round 1
K_1
48
Permuted choice 2
56
Left circular shift
64
56
Round 2
K_2
48
Permuted choice 2
56
Left circular shift
Round 16
K_{16}
48
Permuted choice 2
56
Left circular shift
32-bit swap
64 bits
Inverse initial permutation
64-bit ciphertext

(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

- Representation
  - The input to a table consists of 64 bits numbered from 1 to 64.
  - The 64 entries in the permutation table contain a permutation of the numbers from 1 to 64.
  - Each entry in the permutation table indicates the position of a numbered input bit in the output, which also consists of 64 bits

- To see that these two permutation functions are indeed the inverse of each other, consider the following 64-bit input M :

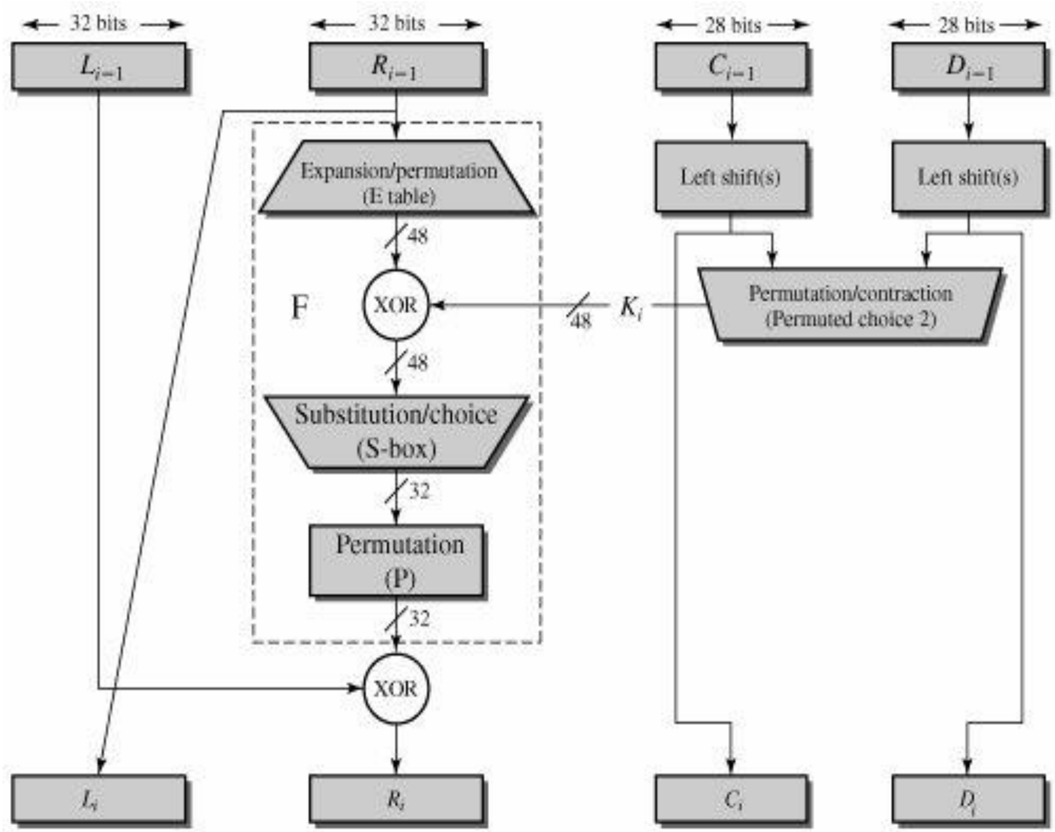
M_{1}	M_{2}	M_{3}	M_{4}	M_{5}	M_{6}	M_{7}	M_{8}
M_{9}	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}
M_{17}	M_{18}	M_{19}	M_{20}	M_{21}	M_{22}	M_{23}	M_{24}
M_{25}	M_{26}	M_{27}	M_{28}	M_{29}	M_{30}	M_{31}	M_{32}
M_{33}	M_{34}	M_{35}	M_{36}	M_{37}	M_{38}	M_{39}	M_{40}
M_{41}	M_{42}	M_{43}	M_{44}	M_{45}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

- where M_i is a binary digit. Then the permutation $X = \mathrm{IP}(M)$ is as follows:

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

- If we then take the inverse permutation $\boldsymbol{Y = \mathrm{IP}^{-1}(X) = \mathrm{IP}^{-1}(\mathrm{IP}(M))}$, it can be seen that the original ordering of the bits is restored.

- **Details of Single round**
  - The left and right halves of each 64-bit intermediate value are treated as separate 32-bit quantities , labeled L (left) and R (right)



- The round key K_i is 48 bits.
- The R input is 32 bits.
- This R input is first expanded to 48 bits by using a table that defines a permutation plus an expansion that involves duplication of 16 of the R bit

(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

- The resulting 48 bits are **XORed** with K_i. This 48-bit result passes through a substitution function that produces a 32-bit output

(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

- **Role of S-Boxes**
  - The substitution consists of a set of eight S-boxes, each of which accepts 6 bits as input and produces 4 bits as output.

- These transformations are defined in

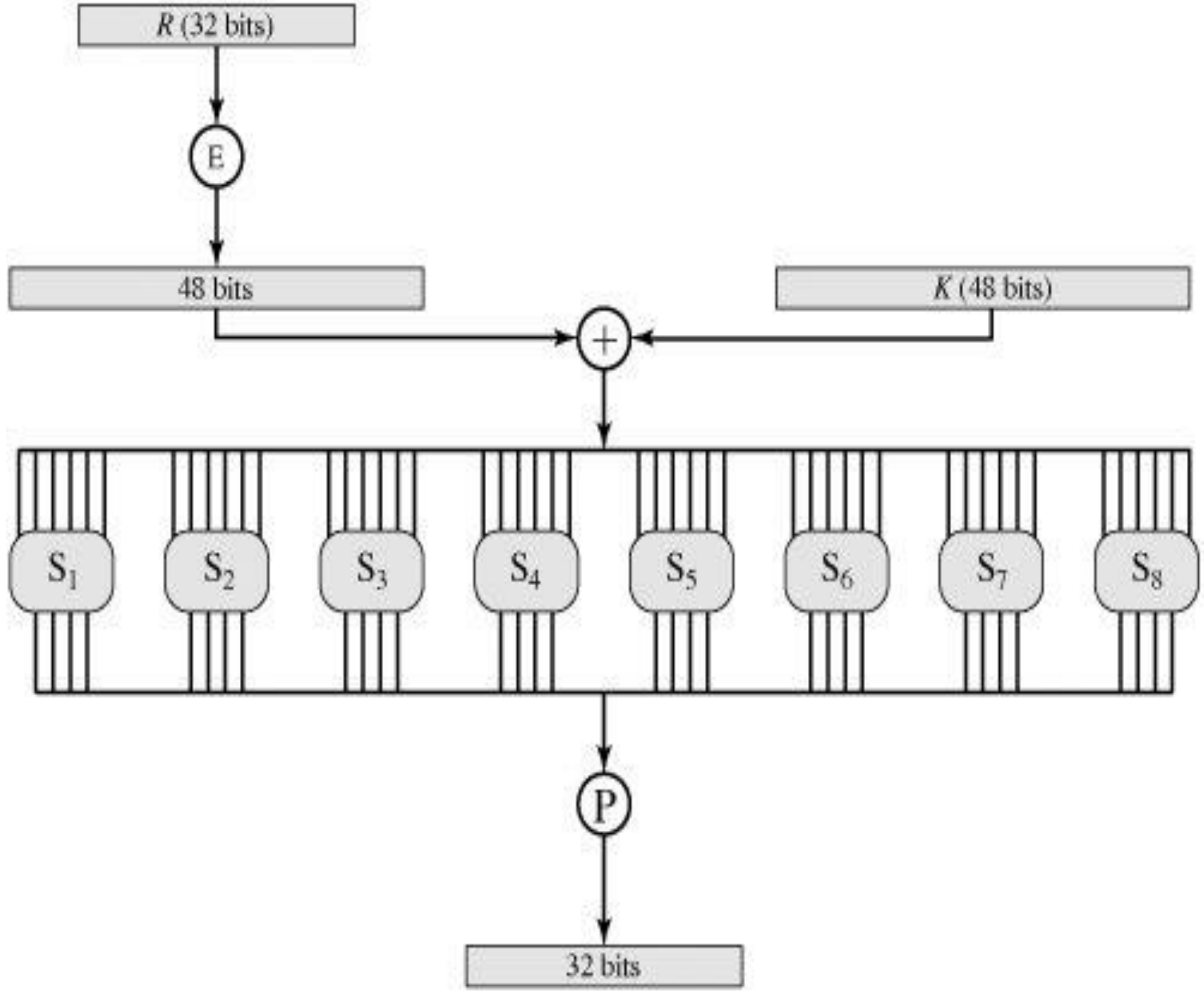
S_1	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	13	8	10	1	3	15	4	2	11	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

	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
S_5	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

	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
S_6	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

	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
S_7	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

	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


R (32 bits)
E
48 bits
K (48 bits)
+
S1
S2
S3
S4
S5
S6
S7
S8
P
32 bits

- 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 S $_i$.
- The middle four bits select one of the sixteen columns .
- The decimal value in the cell selected by the row and column is then converted to its 4-bit representation to produce the output.
- For example,
  - In S $_1$ for input 011001, the row is 01 (row 1) and the column is 1100 (column 12).
  - If The value in row 1, column 12 is 9, so the output is 1001.

Key Generation

- The bits of the key are numbered from 1 through 64; every eighth bit is ignored.
- The key is first subjected to a permutation governed by a table labeled Permuted Choice One.
- 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.
- These shifted values serve as input to the next round.
- They also serve as input to Permuted Choice Two, which produces a 48-bit output that serves as input to the function $F(R_{i-1}, K_i)$.

(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

(a) Change in Plaintext		(b) Change in Key	
Round	Number of bits that differ	Round	Number of bits that differ
0	1	0	0
1	6	1	2
2	21	2	14
3	35	3	28
4	39	4	32
5	34	5	30
6	32	6	32
7	31	7	35
8	29	8	34
9	42	9	40
10	44	10	38
11	32	11	31
12	30	12	33
13	30	13	28
14	26	14	26
15	29	15	34
16	34	16	35

- Strength of DES
  - Use of 56 bit keys
    - Attack on DES is impractical that is DES encryption per microsecond would take more than a thousand years to break the cipher .
    - There are 2^{56} possible keys, approximate 7.2×10^{16}