Example of DES Single Round Calculation

1. Getting Text

a. Getting Text

For DES calculation, let the input text be "**Ansary**".
So, Input text = "Ansary"

b. Converting to Binary

The input text is converted to binary.

A	=	0	1	0	0	0	0	0	1
n	=	0	1	1	0	1	1	1	0
s	=	0	1	1	1	0	0	1	1
a	=	0	1	1	0	0	0	0	1
r	=	0	1	1	1	0	0	1	0
y	=	0	1	1	1	1	0	0	1

c. Breaking into 64 bit Blocks

The input text is turned to 64 bit block(s) and for padding, 0x80 (10000000) is used.

0	1	0	0	0	0	0	1
0	1	1	0	1	1	1	0
0	1	1	1	0	0	1	1
0	1	1	0	0	0	0	1
0	1	1	1	0	0	1	0
0	1	1	1	1	0	0	1
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0

2. Example Key

For DES Calculation, the following key is used.

0	0	1	1	0	1	0	0
0	0	1	0	1	1	0	1
1	0	1	1	0	1	0	1
1	0	1	0	1	0	0	0
0	0	0	1	1	1	0	1
1	1	0	1	1	0	1	1
1	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0

3. Initial Permutation (IP)

Initial permutation is done on the input.

input:							
0	1	0	0	0	0	0	1
0	1	1	0	1	1	1	0
0	1	1	1	0	0	1	1
0	1	1	0	0	0	0	1
0	1	1	1	0	0	1	0
0	1	1	1	1	0	0	1
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0

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

result:							
0	0	1	1	1	1	1	1
0	0	1	1	0	1	0	0
0	0	0	0	0	0	1	0
0	0	1	0	1	1	0	1
1	1	0	0	0	0	0	0
0	0	1	1	1	1	1	0
0	0	1	0	0	0	1	0
0	0	0	1	0	1	1	0

4. Permuted Choice – 1 (PC – 1)

The 64 bit key is permuted according to PC-1. After this, 56 bit key is achieved.

key:							
0	0	1	1	0	1	0	0
0	0	1	0	1	1	0	1
1	0	1	1	0	1	0	1
1	0	1	0	1	0	0	0
0	0	0	1	1	1	0	1
1	1	0	1	1	0	1	1
1	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0

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:	0	1	1	0	1	1	0
	0	0	0	1	0	0	0
	0	0	0	0	0	0	1
	1	1	1	0	1	1	1
D:	0	0	1	0	0	0	0
	0	1	0	0	1	0	1
	1	1	0	0	1	1	1
	0	1	0	0	1	0	1

5. Key Scheduler

After PC-1, 56 bit key is achieved from 64 bit key. Then, in each round of the 16 rounds, 56 bit key is split into left and right halves where each half has 28 bits and left circular shift is done on each half separately and permutation is done with PC-2 to achieve 48 bit key from 56 bit key. The number of shifts for all rounds is not the same.

a. Left Circular Shift

Round No	No of Shift	C	D
1	1	1101100001000000000111101110	0100000100101110011101001010
2	1	1011000010000000001111011101	1000001001011100111010010100
3	2	1100001000000000111101110110	0000100101110011101001010010
4	2	0000100000000011110111011011	0010010111001110100101001000
5	2	0010000000001111011101101100	1001011100111010010100100000
6	2	1000000000111101110110110000	0101110011101001010010000010
7	2	0000000011110111011011000010	0111001110100101001000001001
8	2	0000001111011101101100001000	1100111010010100100000100101
9	1	0000011110111011011000010000	1001110100101001000001001011
10	2	0001111011101101100001000000	0111010010100100000100101110
11	2	0111101110110110000100000000	1101001010010000010010111001
12	2	1110111011011000010000000001	0100101001000001001011100111
13	2	1011101101100001000000000111	0010100100000100101110011101
14	2	1110110110000100000000011110	1010010000010010111001110100
15	2	1011011000010000000001111011	1001000001001011100111010010
16	1	0110110000100000000011110111	0010000010010111001110100101

b. Permuted Choice -2 (PC – 2)

For each round, the input for PC – 2 comes from the shifted C and D sub-keys that are achieved after left circular shifts.

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

For Round 1,

56 bit key						
1	1	0	1	1	0	0
0	0	1	0	0	0	0
0	0	0	0	0	1	1
1	1	0	1	1	1	0
0	1	0	0	0	0	0
1	0	0	1	0	1	1
1	0	0	1	1	1	0
1	0	0	1	0	1	0

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

48 bit key					
0	0	0	0	1	1
0	0	0	0	1	1
1	0	0	1	1	0
0	0	1	1	0	1
1	0	0	0	1	1
1	0	0	0	0	1
0	0	1	0	0	1
1	1	1	1	0	0

For all other rounds, 48 bit keys can be generated from 56 bit keys using PC – 2 in similar approach.

6. Round 1

L_0:							
0	0	1	1	1	1	1	1
0	0	1	1	0	1	0	0
0	0	0	0	0	0	1	0
0	0	1	0	1	1	0	1

R_0:							
1	1	0	0	0	0	0	0
0	0	1	1	1	1	1	0
0	0	1	0	0	0	1	0
0	0	0	1	0	1	1	0

K:					
0	0	0	0	1	1
0	0	0	0	1	1
1	0	0	1	1	0
0	0	1	1	0	1
1	0	0	0	1	1
1	0	0	0	0	1
0	0	1	0	0	1
1	1	1	1	0	0

- **E-bit Selection Table**

1	1	0	0	0	0	0	0
0	0	1	1	1	1	1	0
0	0	1	0	0	0	1	0
0	0	0	1	0	1	1	0

➡

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

=

0	1	1	0	0	0
0	0	0	0	0	0
0	0	0	1	1	1
1	1	1	1	0	0
0	0	0	1	0	0
0	0	0	1	0	0
0	0	0	0	1	0
1	0	1	1	0	1

- **XOR with Sub-key**

$$\begin{array}{|cccccc|}\hline 0&1&1&0&0&0\\ 0&0&0&0&0&0\\ 0&0&0&1&1&1\\ 1&1&1&1&0&0\\ 0&0&0&1&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&0\\ 1&0&1&1&0&1\\ \hline\end{array} \text{ XOR } \begin{array}{|cccccc|}\hline 0&0&0&0&1&1\\ 0&0&0&0&1&1\\ 1&0&0&1&1&0\\ 0&0&1&1&0&1\\ 1&0&0&0&1&1\\ 1&0&0&0&0&1\\ 0&0&1&0&0&1\\ 1&1&1&1&0&0\\ \hline\end{array} = \begin{array}{|cccccc|}\hline 0&1&1&0&1&1\\ 0&0&0&0&1&1\\ 1&0&0&0&0&1\\ 1&1&0&0&0&1\\ 1&0&0&1&1&1\\ 1&0&0&1&0&1\\ 0&0&1&0&1&1\\ 0&1&0&0&0&1\\ \hline\end{array}$$

- **S-boxes**

-*S_1:*	Row: 01= 1	Column: 1101 = 13	Value = 5	= 0101
-*S_2:*	Row: 01= 1	Column: 0001 = 1	Value = 13	= 1101
-*S_3:*	Row: 11= 3	Column: 0000 = 0	Value = 1	= 0001
-*S_4:*	Row: 11= 3	Column: 1000 = 8	Value = 9	= 1001
-*S_5:*	Row: 11= 3	Column: 0011 = 3	Value = 7	= 0111
-*S_6:*	Row: 11= 3	Column: 0010 = 2	Value = 2	= 0010
-*S_7:*	Row: 01= 1	Column: 0101 = 5	Value = 9	= 1001
-*S_8:*	Row: 01= 1	Column: 1000 = 8	Value = 12	= 1100

- **Permutation**

$$\begin{array}{|cccc|}\hline 0&1&0&1\\ 1&1&0&1\\ 0&0&0&1\\ 1&0&0&1\\ 0&1&1&1\\ 0&0&1&0\\ 1&0&0&1\\ 1&1&0&0\\ \hline\end{array} \quad ➡ \quad \begin{array}{|cccc|}\hline 16&7&20&21\\ 29&12&28&17\\ 1&15&23&26\\ 5&18&31&10\\ 2&8&24&14\\ 32&27&3&9\\ 19&13&30&6\\ 22&11&4&25\\ \hline\end{array} = \begin{array}{|cccc|}\hline 1&0&1&0\\ 1&1&1&0\\ 0&0&1&0\\ 1&1&0&0\\ 1&1&0&0\\ 0&0&0&0\\ 1&1&1&1\\ 0&0&1&1\\ \hline\end{array}$$

- **XOR Left and Right**

$$\begin{array}{|cccccccc|}\hline 0&0&1&1&1&1&1&1\\ 0&0&1&1&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&1&0&1&1&0&1\\ \hline\end{array} \text{ XOR } \begin{array}{|cccccccc|}\hline 1&0&1&0&1&1&1&0\\ 0&0&1&0&1&1&0&0\\ 1&1&0&0&0&0&0&0\\ 1&1&1&1&0&0&1&1\\ \hline\end{array} = \begin{array}{|cccccccc|}\hline 1&0&0&1&0&0&0&1\\ 0&0&0&1&1&0&0&0\\ 1&1&0&0&0&0&1&0\\ 1&1&0&1&1&1&1&0\\ \hline\end{array}$$

L_1:							
1	1	0	0	0	0	0	0
0	0	1	1	1	1	1	0
0	0	1	0	0	0	1	0
0	0	0	1	0	1	1	0

R_1:							
1	0	0	1	0	0	0	1
0	0	0	1	1	0	0	0
1	1	0	0	0	0	1	0
1	1	0	1	1	1	1	0

7. Inverse Initial Permutation on Round 1 Result

IP^{-1} simply reverses what was done by IP.

input							
1	1	0	0	0	0	0	0
0	0	1	1	1	1	1	0
0	0	1	0	0	0	1	0
0	0	0	1	0	1	1	0
1	0	0	1	0	0	0	1
0	0	0	1	1	0	0	0
1	1	0	0	0	0	1	0
1	1	0	1	1	1	1	0

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

result							
1	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1
0	0	0	1	0	0	1	1
0	0	1	1	0	0	1	0
1	0	1	1	0	0	1	1
0	0	0	1	0	1	0	0
0	1	0	0	1	0	1	0
1	1	0	0	1	0	1	0

8. Encrypted Message

The result of IP^{-1} can be converted back to ASCII as the following

1	0	0	0	0	0	0	0	=	€
0	0	0	1	1	1	1	1	=	US
0	0	0	1	0	0	1	1	=	DC3
0	0	1	1	0	0	1	0	=	2
1	0	1	1	0	0	1	1	=	³
0	0	0	1	0	1	0	0	=	DC4
0	1	0	0	1	0	1	0	=	J
1	1	0	0	1	0	1	0	=	Ê