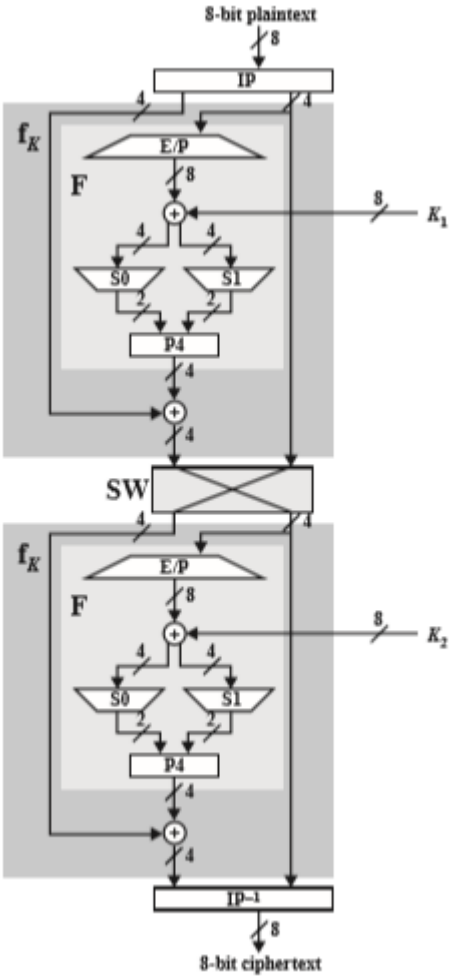
S-DES

PART-2

The plain text is of size 8-bits



Key-1 = 1 0 1 0 0 1 0 0

Key-2 = 0 1 0 0 0 0 1 1

How To Encrypt the Plain Text into Cipher Text in S-DES After Generating Keys?

Now, let's start Encryption of plain text into cipher text.

Encryption of Plain text into Cipher text in S-DES:

Come on do it, **step by step**.
Note: the size of input text is 8 bit and output also will be 8-bit. Or the block size is 8-bit/one byte always.

Step 1:

Suppose this is our plain text in binary which is 8-bit.

Plain text: 01110010

Step 2:

Put the plain text into IP-8(initial permutation) table and permute the bits.

IP-8

I/P	1	2	3	4	5	6	7	8
O/P	2	6	3	1	4	8	5	7

	1	2	3	4	5	6	7	8
I/P	0	1	1	1	0	0	1	0
O/P	1	0	1	0	1	0	0	1

Output = 1 0 1 0 1 0 0 1

Step 3:

Now break the bits into two halves, each half will consist of 4 bits.
The halves will be right and left.
Two Halves of the bits:

Left half {1 0 1 0}
right half {1 0 0 1}

Step 4:

Take the right 4 bits and put them into E.P (expand and per-mutate) Table.
Bits of right half: 1001

EP Table:

I/P	1	2	3	4				
O/P	4	1	2	3	2	3	4	1

	1	2	3	4				
I/P	1	0	0	1				
O/P	1	1	0	0	0	0	1	1

O/P = 11000011

Step 5: Now, just take the output and XOR it with First key Or K 1 (which we created in previous topic that is how to generate key.).

O/P = 1 1 0 0 0 0 1 1

KEY1 = 1 0 1 0 0 1 0 0

0 1 1 0 0 1 1 1

Step 6:

Once again split the output of XOR's bit into two halves and each half will consist of 4 bits.

Splitting them into two halves:

Left half : 0 1 1 0

Right half : 0 1 1 1

Now put each half into the **s-boxes**, there is only two s-boxes. **S-0 and S-1.**

$$S0 = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{bmatrix} \end{array} \qquad S1 = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{bmatrix} \end{array}$$

But how to put them in into S-Boxes?

The most first and most last bit will be consider the row and other remaining, which are, 2 and 3, will be considered the columns.

i.e., here I'm taking the left half: which is 0 1 1 0

Now I will take First and last bit which are: 0 and 0. These will be row.

And I also will take 2nd and 3rd bits which are: 1 1. These will be column number.

00 means=0th **row**
11 means = 3rd **col**

so ,the left half need to keep in S-0 table so the value at 0th row 3rd column is **1 0**

now for the right half = 0 1 1 1
Now I will take First and last bit which are: 0 and 1. These will be row.

And I also will take 2nd and 3rd bits which are: 1 1. These will be column number.
01 means = 1st row
11 means = 3rd col

So the right half I need to keep In s-1 table/ matrix so the value of 1st row and 3rd col is 1 1

Step 7:

Now combine these two halves together.

Left half: 1 0
right half: 1 1

It will be: 1 0 1 1

Step 8: Now take these 4 bits and put them in **P-4** (permutation 4) table and get the result.

P-4

I/P	1	2	3	4
O/P	2	4	3	1

	1	2	3	4
I/P	1	0	1	1
O/P	0	1	1	1

Output = 0 1 1 1

Step-9

Now get XOR the output with left 4 bits of Initial Permutation. The left bits of initial per-mutation are in **step 3**, which are **1 0 1 0**.(please, in step 3).
Let them to be XOR.

P-4 = 0 1 1 1
IPL = 1 0 1 0

1 1 0 1

Step 10:

Now get the right half of the initial permutation, which is **step 3,** and combine that with this out- put.

The out-put of XOR in **step 9:** 1 1 0 1
Right half of IP (initial permutation): 1 0 0 1
Let's combine both. 1 1 0 0 – 1 0 0 1= 1 1 0 0 1 0 0 1

Now the output is 8 bits.: **1 1 0 0 1 0 0 1**

Step 11: Now once again break the out-put into two halves, left and right;

Left: {1 1 0 0} right: {1 0 0 1}

Step 12:

Now swap both halves, which means put the left half in place of right and vice versa.

Result:
Left half: {1 0 0 1} right half: {1 1 0 0}

Step 13:

Now let's take these halves and once again start the same procedure from **step 2** or initial Permutation BUT be careful on using key in this stage we use second key or K2 (not K1). And put that into IP^{-1} (IP inverse) Table. What you get will be your final cipher text

After Second Round

Output= 1 1 1 0 1 1 1 0

The final step is we need to put the output in IP^{-1} table

	1	2	3	4	5	6	7	8
IP	2	6	3	1	4	8	5	7
IP^{-1}	4	1	3	5	7	2	8	6

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
I/P	**1**	**1**	**1**	**0**	**1**	**1**	**1**	**0**
O/P	**0**	**1**	**1**	**1**	**1**	**1**	**0**	**1**

Hence the Cipher-Text = 0 1 1 1 1 1 0 1

THANK YOU