Student Manual/DR. Bashar Al-Esawi

Simplified DES=SDES

Block Cipher (depends on Bit Manipulation):

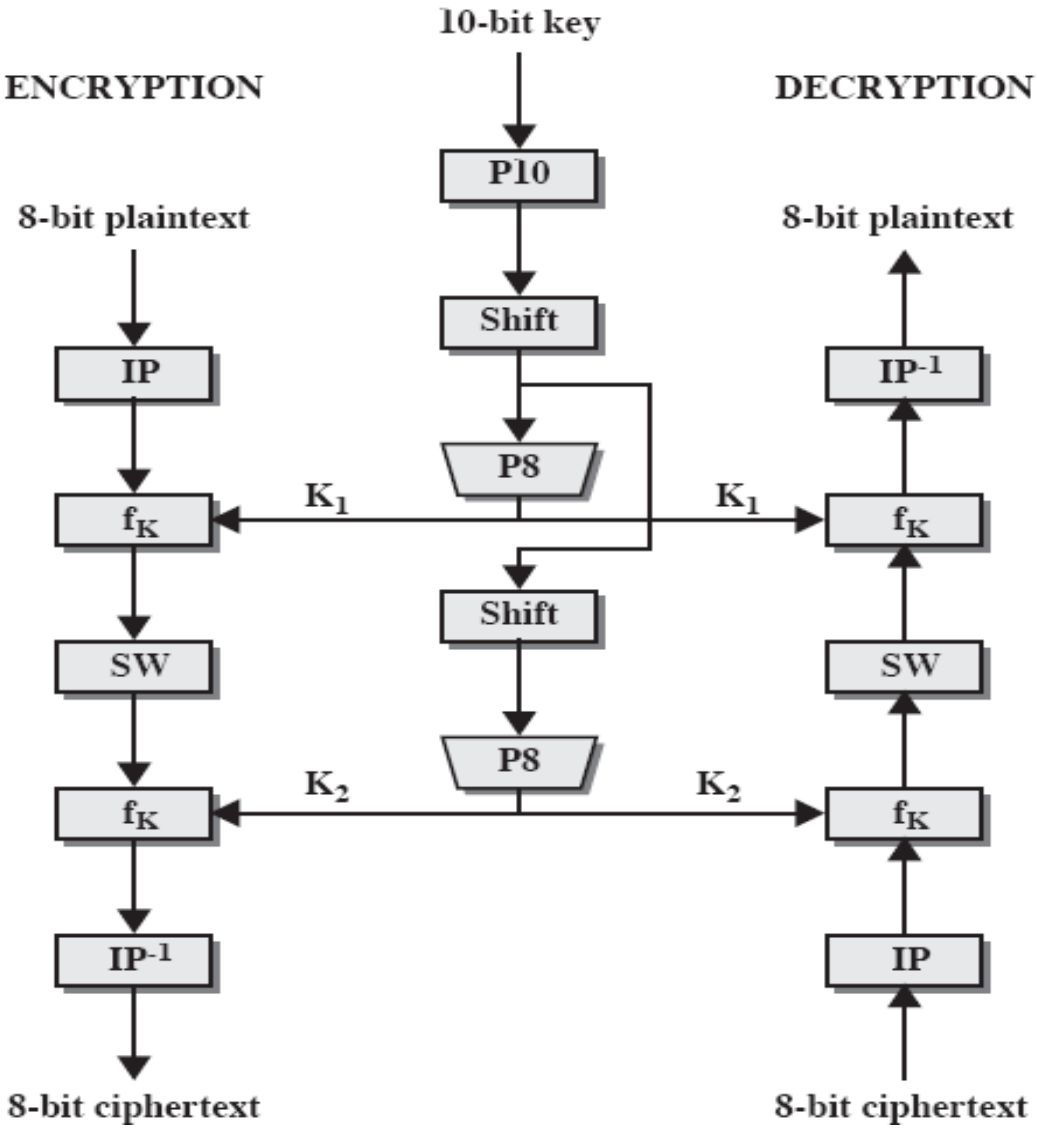
S-DES or Simplified Data Encryption Standard
The process of encrypting a plan text into an encrypted message with the use of S-DES has been divided into multi-steps which may help you to understand it as easily as possible.

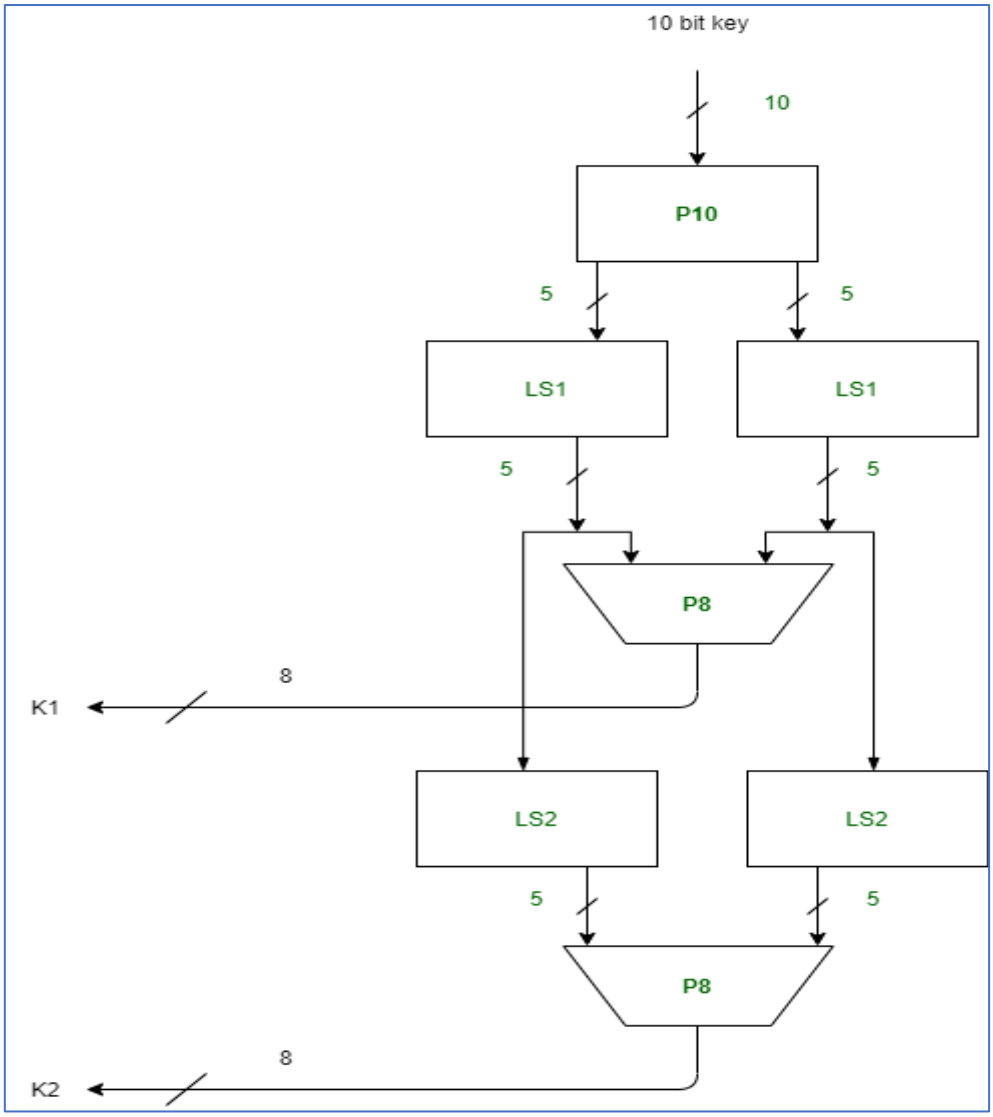
Points should be remembered.

1. It is a block cipher.
2. **It has 8-bits block size of plain text or cipher text.**
3. **It uses 10-bits key size for encryption.**
4. It is a ***symmetric*** cipher.
5. **It has Two Rounds.**

https://www.c-sharpcorner.com/article/s-des-or-simplified-data-encryption-standard/

http://homepages.neiu.edu/~jakwon1/sdes.html **step by step example**


10-bit key
ENCRYPTION
DECRYPTION
P10
8-bit plaintext
8-bit plaintext
Shift
IP
IP-1
P8
K1
K1
fK
fK
Shift
SW
SW
P8
K2
K2
fK
fK
IP-1
IP
8-bit ciphertext
8-bit ciphertext


10 bit key
10
P10
5
5
LS1
LS1
5
5
P8
8
K1
LS2
LS2
5
5
P8
8
K2

https://www.youtube.com/watch?v=3jGMCyOXOV8

https://www.youtube.com/watch?v=fdJokZ_gkl4

Simplified DES

- Input (plaintext) block: 8-bits
- Output (ciphertext) block: 8-bits
- Key: 10-bits
- Rounds: 2
- Round keys generated using permutations and left shifts
- Encryption: initial permutation, round function, switch halves
- Decryption: Same as encryption, except round keys used in opposite order

S-DES Operations

- P10 (permutate)

```
Input : 1 2 3 4 5 6 7 8 9 10
Output: 3 5 2 7 4 10 1 9 8 6
```

- P8 (select and permutate)

```
Input : 1 2 3 4 5 6 7 8 9 10
Output: 6 3 7 4 8 5 10 9
```

- P4 (permutate)

```
Input : 1 2 3 4
Output: 2 4 3 1
```

Steps of S-DES can be summarized as follow:

Step1: Plain Text must be 8-Bits as a block.

Step2: Key that agree between S/R must be 10-bits

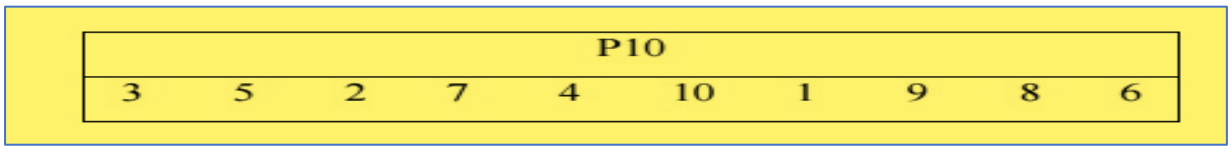
Step3: Cipher text must be 8-Bits.

Step4: Key generation

Step5: Ciphertext = $\mathbf{IP}^{-1}$ ($\mathbf{f}_{k2}$ (**SW** ($\mathbf{f}_{k1}$ (**IP** (plaintext)))))

1	2	3	4	5	6	7	8	9	10
1	1	0	0	1	1	0	0	1	0
3	5	2	7	4	10	1	9	8	6
0	1	1	0	0	0	1	1	0	1

Inx=[3,5,2,7,4,10,1,9,8,6]



Key(10-Bits)=1100110010

$K_1 = P_8(shift(P_{10}(key)))$

=P8(shift(0110001101))

=P8(1100011010)=10100001

This step (Circular-Shift Left)<<0110001101<<1100011010

1	2	3	4	5	6	7	8	9	10
1	1	0	0	0	1	1	0	1	0
6	3	7	4	8	5	10	9	P8	
1	0	1	0	0	0	0	1		

Key(10-Bits)=1100110010

$K_2 = P_8(\text{shift}(\text{shift}(P_{10}(\text{key}))))$

P10(key)= 0110001101<<1100011010<<1000110101

K2=P8(1000110101)=10001110

1	2	3	4	5	6	7	8	9	10
1	0	0	0	1	1	0	1	0	1
6	3	7	4	8	5	10	9		
1	0	0	0	1	1	1	0		

P8							
6	3	7	4	8	5	10	9

This can be called S-DES

One of the modern Methods

DES-3DES- RC4-AES-etc…

Encryption:

$$\text{Ciphertext} = IP^{-1}(f_{K2}(SW(f_{k1}(IP(\text{plaintext})))))$$

Encryption involves the sequential application of five functions.

Initial and Final Permutations

The input to the algorithm is an 8-bit block of plaintext, which we first permute using the IP function:

IP							
2	6	3	1	4	8	5	7

This retains all 8 bits of the plaintext but mixes them up.

Consider the plaintext to be **11110011**.

IP=**11010111**

1	1	1	1	0	0	1	1
8	7	6	5	4	3	2	1
2	**6**	**3**	**1**	**4**	**8**	**5**	**7**
1	1	0	1	0	1	1	1

Permuted output = 10111101

At the end of the algorithm, the inverse permutation is used:

IP –1							
4	1	3	5	7	2	8	6

The Function f_k

The most complex component of S-DES is the function f_k, which consists of a combination of permutation and substitution functions. The functions can be expressed as follows. Let L and R be the leftmost 4 bits and rightmost 4 bits of the 8-bit input to f K, and let F be a mapping (not necessarily one to one) from 4-bit strings to 4-bit strings. Then we let

Fk# =(L XOR F(R XOR SK),R)

Example: L and R (Right and Left of BITS)

Let (IP (plaintext)=10111101

Let K1=(**10111101**) and

Suppose **F(1101,SK)** and (**SK=4132**)=**1110** **(this 4-bits).**

Therefore:

Fk1 =(1011 XOR 1110,1101)

=(010111101)

3 The Switch Function

The function f K only alters the leftmost 4 bits of the input. The switch function (SW) interchanges the left and right 4 bits

Example:

=SW (01011101)= (11010101)

Fk2=(11010101) and SK=4132

Fk2 =(L XOR F(R,SK),R)

Fk2 =(1101 XOR F(0101,4132),0101)

Fk2 =(1101 XOR 0110,0101)

Fk2 =(10110101) 8-bits

IP-1=01110011

Therefore the ciphertext=”01110011”

IP –1							
4	1	3	5	7	2	8	6

Decryption:

Plaintext = IP-1 (FK1(SW(FK2 (IP(Ciphertext))))

https://www.brainkart.com/article/Simplified-Data-Encryption-Standard-(S-DES)_8343/

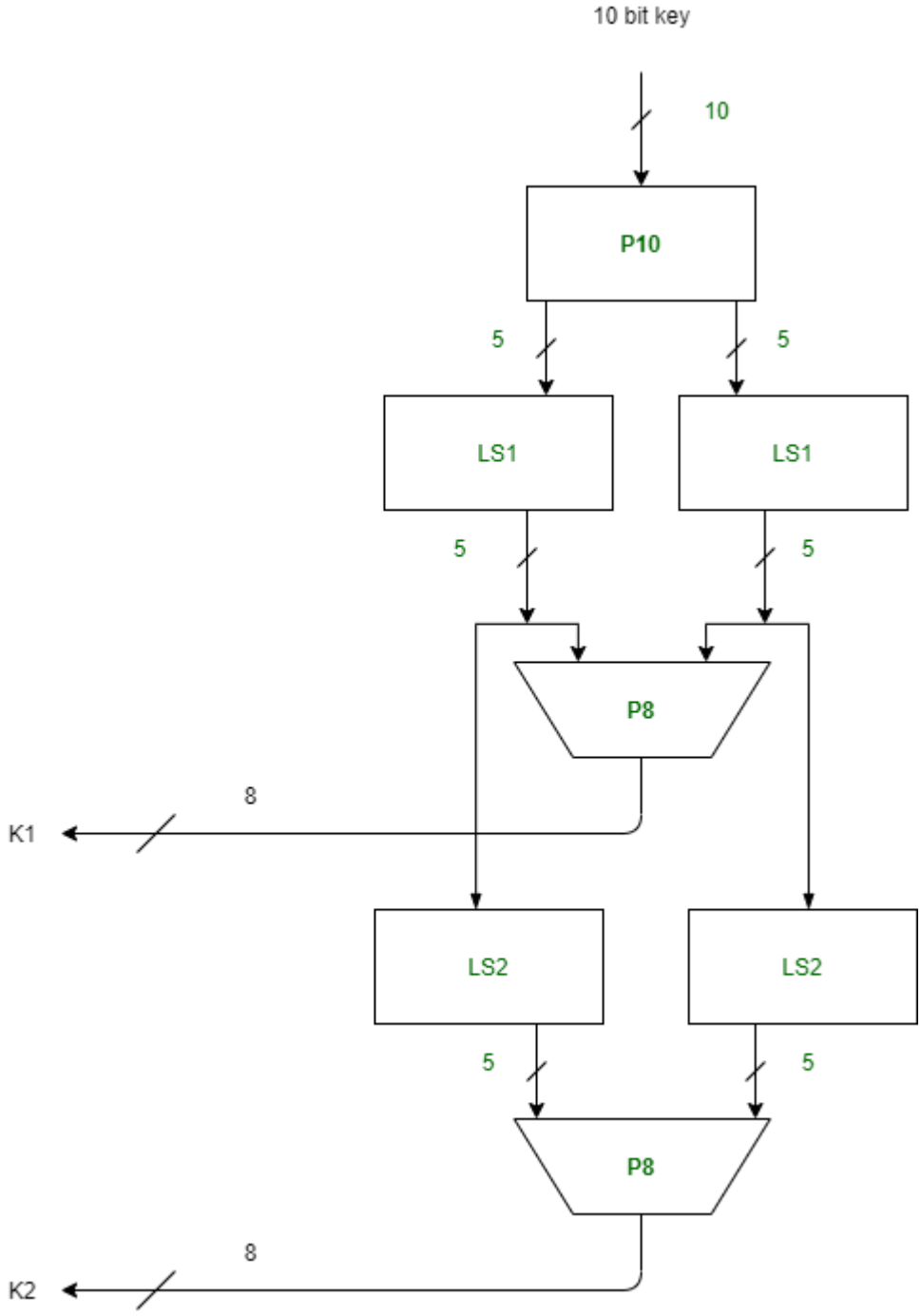
https://codebeautify.org/string-binary-converter

Simplified Data Encryption Standard Key Generation

- Difficulty Level : Medium
- Last Updated : 27 Sep, 2021

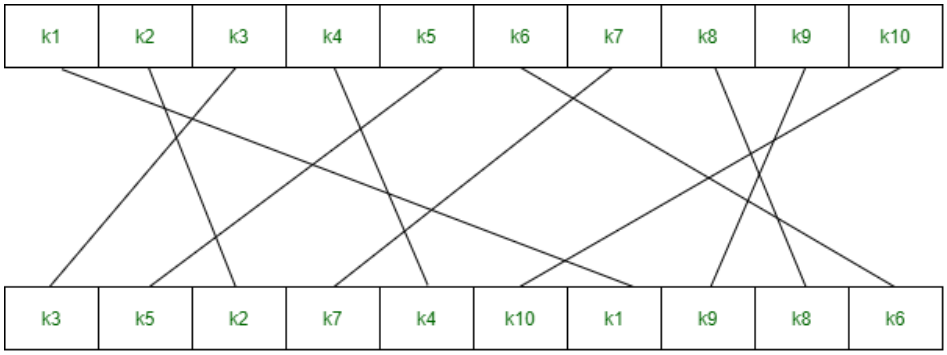
Simplified Data Encryption Standard (S-DES) is a simple version of the DES Algorithm. It is similar to the DES algorithm but is a smaller algorithm and has fewer parameters than DES. It was made for educational purposes so that understanding DES would become simpler. It is a block cipher that takes a block of plain text and converts it into ciphertext. It takes a block of 8 bit.
It is a symmetric key cipher i.e. they use the same key for both encryption and decryption. In this article, we are going to demonstrate key generation for s-des encryption and decryption algorithm. We take a random 10-bit key and produce two 8-bit keys which will be used for encryption and decryption.

Key Generation Concept: In the key generation algorithm, we accept the 10-bit key and convert it into two 8 bit keys. This key is shared between both sender and receiver.

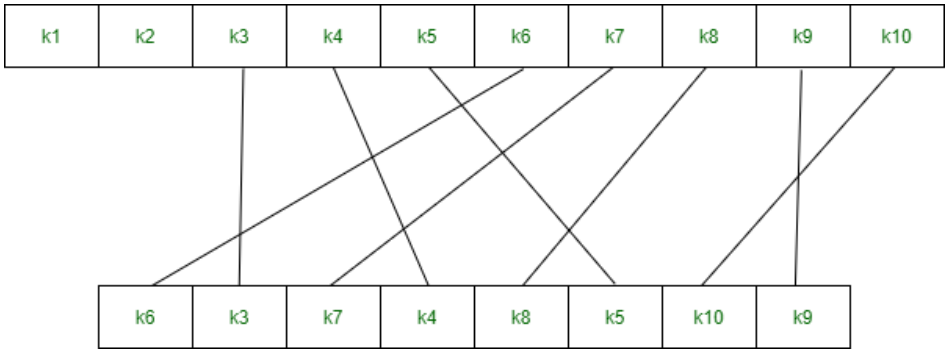


In the key generation, we use three functions:

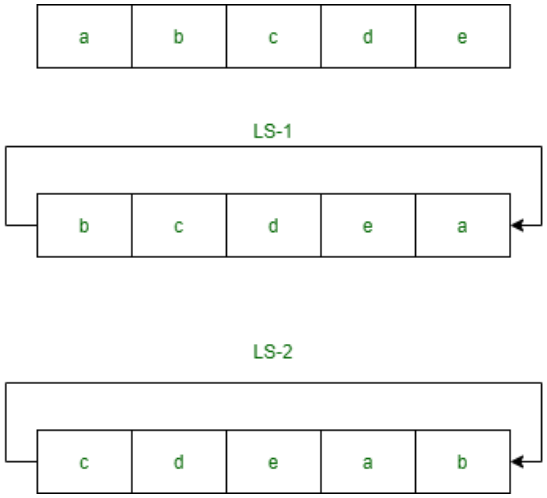
1. Permutation P10



2. Permutation P8



3. Left Shift



Step 1: We accepted a 10-bit key and permuted the bits by putting them in the P10 table.

```
Key = 1 0 1 0 0 0 0 0 1 0
```

```
(k1, k2, k3, k4, k5, k6, k7, k8, k9, k10) = (1, 0, 1, 0, 0, 0, 0, 0, 1, 0)
```

```
P10 Permutation is: P10(k1, k2, k3, k4, k5, k6, k7, k8, k9, k10) = (k3, k5, k2, k7, k4, k10, k1, k9, k8, k6)
```

```
After P10, we get 1 0 0 0 0 0 1 1 0 0
```

Step 2: We divide the key into 2 halves of 5-bit each.

```
l=1 0 0 0 0, r=0 1 1 0 0
```

Step 3: Now we apply one bit left-shift on each key.

```
l = 0 0 0 0 1, r = 1 1 0 0 0
```

Step 4: Combine both keys after step 3 and permute the bits by putting them in the P8 table. The output of the given table is the first key K1.

```
After LS-1 combined, we get 0 0 0 0 1 1 1 0 0 0
```

```
P8 permutation is: P8(k1, k2, k3, k4, k5, k6, k7, k8, k9, k10) = (k6, k3, k7, k4, k8, k5, k10, k9)
```

```
After P8, we get Key-1 : 1 0 1 0 0 1 0 0
```

Step 5: The output obtained from step 3 i.e. 2 halves after one bit left shift should again undergo the process of two-bit left shift.

```
Step 3 output - l = 0 0 0 0 1, r = 1 1 0 0 0
```

```
After two bit shift - l = 0 0 1 0 0, r = 0 0 0 1 1
```

Step 6: Combine the 2 halves obtained from step 5 and permute them by putting them in the P8 table. The output of the given table is the second key K2.

```
After LS-2 combined = 0 0 1 0 0 0 0 0 1 1

P8 permutation is: P8(k1, k2, k3, k4, k5, k6, k7, k8, k9, k10) =
(k6, k3, k7, k4, k8, k5, k10, k9)

After P8, we get Key-2 : 0 1 0 0 0 0 1 1
```

Final Output:

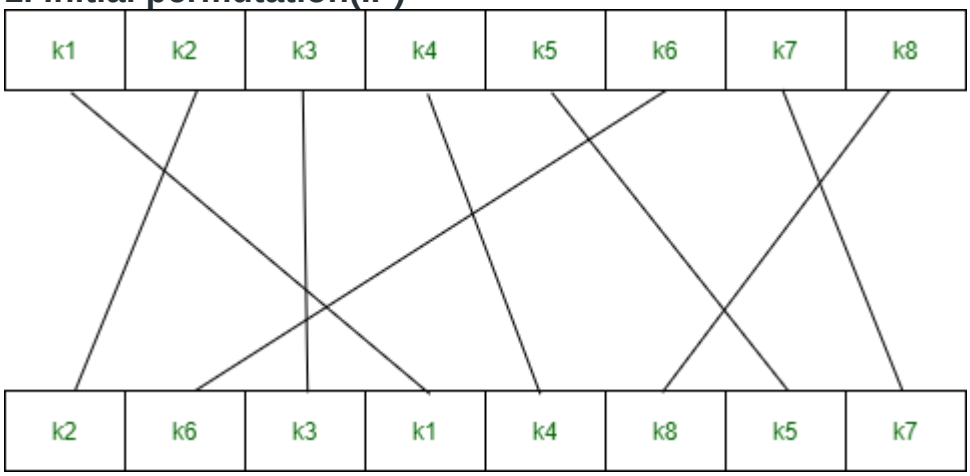
```
Key-1 is: 1 0 1 0 0 1 0 0

Key-2 is: 0 1 0 0 0 0 1 1
```

SDES Ciphering:

Components :

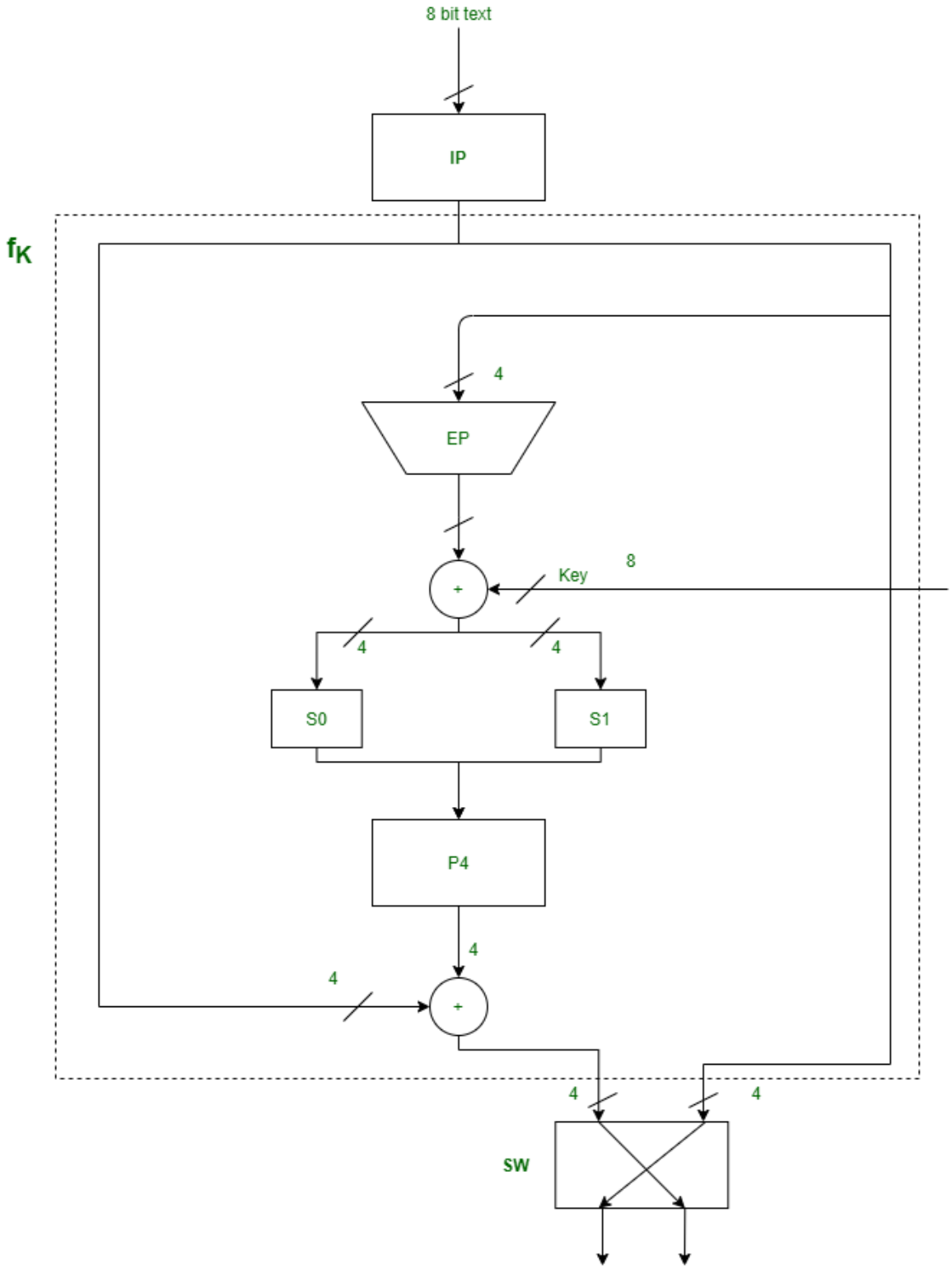
S-DES encryption involves four functions –

1. Initial permutation(IP) –



2. Complex function (f_k) –

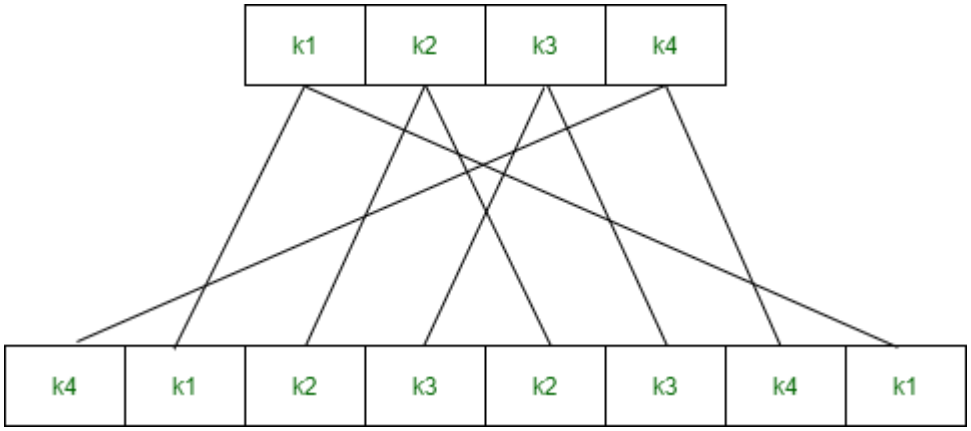
It is the combination of permutation and substitution functions. The below image represents a round of encryption and decryption. This round is repeated twice in each encryption and decryption.



Components in f_k are –

a. Expanded Permutation (EP) –

It takes a 4-bit input and converts it into an 8-bit output.



b. S-boxes (S0 and S1) –
It is a basic component of a symmetric key algorithm that performs substitution.

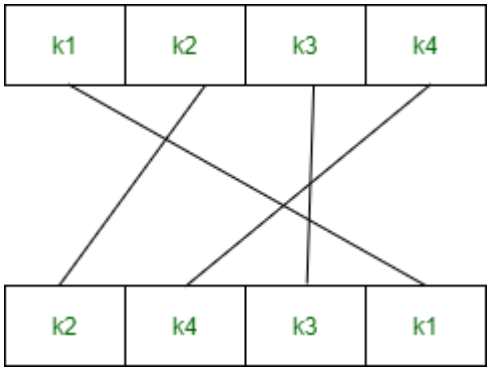
S0

1	0	3	2
3	2	1	0
0	2	1	3
3	1	3	2

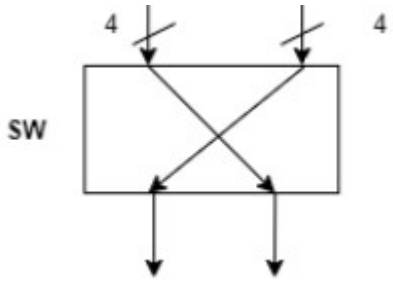
S1

0	1	2	3
2	0	1	3
3	0	1	0
2	1	0	3

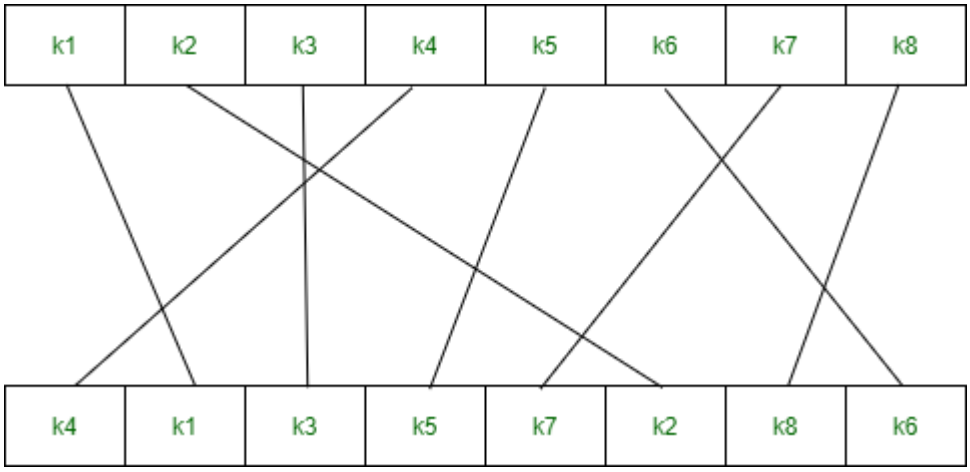
c. Permutation P4 –



3. Switch (SW) –



4. Inverse of Initial Permutation (IP^{-1}) –



First, **we need to generate 2 keys before encryption.**

```
Consider, the entered 10-bit key is - 1 0 1 0 0 0 0 0 1 0
```

Therefore,

```
Key-1 is - 1 0 1 0 0 1 0 0
```

```
Key-2 is - 0 1 0 0 0 0 1 1
```

Encryption –

```
Entered 8-bit plaintext is - 1 0 0 1 0 1 1 1
```

Step-1:

We perform initial permutation on our 8-bit plain text using the IP table. The initial permutation is defined as –

```
IP(k1, k2, k3, k4, k5, k6, k7, k8) = (k2, k6, k3, k1, k4, k8, k5, k7)
```

```
After ip = 0 1 0 1 1 1 0 1
```

Step-2:

After the initial permutation, we get an 8-bit block of text which we divide into 2 halves of 4 bit each.

```
l = 0 1 0 1  and r = 1 1 0 1
```

On the right half, we perform expanded permutation using EP table which converts 4 bits into 8 bits. Expand permutation is defined as –

```
EP(k1, k2, k3, k4) = (k4, k1, k2, k3, k2, k3, k4, k1)
```

```
After ep = 1 1 1 0 1 0 1 1
```

We perform XOR operation using the first key K1 with the output of expanded permutation.

```
Key-1 is - 1 0 1 0 0 1 0 0
```

```
(1 0 1 0 0 1 0 0) XOR (1 1 1 0 1 0 1 1) =  0 1 0 0 1 1 1 1
```

```
After XOR operation with 1st Key = 0 1 0 0 1 1 1 1
```

Again we divide the output of XOR into 2 halves of 4 bit each.

```
l = 0 1 0 0  and r = 1 1 1 1
```

We take the first and fourth bit as row and the second and third bit as a column for our S boxes.

```
S0 = [1,0,3,2
      3,2,1,0
      0,2,1,3
      3,1,3,2]

S1=  [0,1,2,3
      2,0,1,3
      3,0,1,0
      2,1,0,3]

For l = 0 1 0 0
row = 00 = 0, column = 10 = 2
S0 = 3 = 11

For r = 1 1 1 1
row = 11 = 3, column = 11 = 3
S1 = 3 = 11

After first S-Boxes combining S0 and S1 = 1 1 1 1
```

S boxes gives a 2-bit output which we combine to get 4 bits and then perform permutation using the P4 table. P4 is defined as –

```
P4(k1, k2, k3, k4) = (k2, k4, k3, k1)
After P4 = 1 1 1 1
```

We XOR the output of the P4 table with the left half of the initial permutation table i.e. IP table.

```
(0 1 0 1) XOR (1 1 1 1) = 1 0 1 0
After XOR operation with left nibble of after ip = 1 0 1 0
```

We combine both halves i.e. right half of initial permutation and output of ip.

```
Combine 1 1 0 1 and 1 0 1 0
After combine = 1 0 1 0 1 1 0 1
```

Step-3:
Now, divide the output into two halves of 4 bit each. Combine them again, but now the left part should become right and the right part should become left.

```
After step 3 = 1 1 0 1 1 0 1 0
```

Step-4:
Again perform step 2, but this time while doing XOR operation after expanded permutation use key 2 instead of key 1.

```
Expand permutation is defined as - 4 1 2 3 2 3 4 1

After second ep = 0 1 0 1 0 1 0 1

After XOR operation with 2nd Key = 0 0 0 1 0 1 1 0

After second S-Boxes = 1 1 1 1


P4 is defined as - 2 4 3 1

After P4 = 1 1 1 1


After XOR operation with left nibble of after first part = 0 0 1 0

After second part = 0 0 1 0 1 0 1 0


l = 1 1 0 1  and r = 1 0 1 0
```

On the right half, we perform expanded permutation using EP table which converts 4 bits into 8 bits. Expand permutation is defined as –

```
EP(k1, k2, k3, k4) = (k4, k1, k2, k3, k2, k3, k4, k1)

After second ep = 0 1 0 1 0 1 0 1
```

We perform XOR operation using second key K2 with the output of expanded permutation.

```
Key-2 is - 0 1 0 0 0 0 1 1

(0 1 0 0 0 0 1 1) XOR (0 1 0 1 0 1 0 1) =  0 0 0 1 0 1 1 0

After XOR operation with 2nd Key = 0 0 0 1 0 1 1 0
```

Again we divide the output of XOR into 2 halves of 4 bit each.

```
l = 0 0 0 1  and r = 0 1 1 0
```

We take the first and fourth bit as row and the second and third bit as a column for our S boxes.

```
S0 = [1,0,3,2
      3,2,1,0
      0,2,1,3
      3,1,3,2]


S1 = [0,1,2,3
      2,0,1,3
      3,0,1,0
      2,1,0,3]
```

For l = 0 0 0 1

row = 01 = 1 , column = 00 = 0

S0 = 3 = 11

For r = 0 1 1 0

row = 00 = 0 , column = 11 = 3

S1 = 3 = 11

After first S-Boxes combining S0 and S1 = 1 1 1 1

S boxes gives a 2-bit output which we combine to get 4 bits and then perform permutation using the P4 table. P4 is defined as –

P4(k1, k2, k3, k4) = (k2, k4, k3, k1)

After P4 = 1 1 1 1

We XOR the output of the P4 table with the left half of the initial permutation table i.e. IP table.

(1 1 0 1) XOR (1 1 1 1) = 0 0 1 0

After XOR operation with left nibble of after first part = 0 0 1 0

We combine both halves i.e. right half of initial permutation and output of ip.

Combine 1 0 1 0 and 0 0 1 0

After combine = 0 0 1 0 1 0 1 0

After second part = 0 0 1 0 1 0 1 0

Step-5:

Perform inverse initial permutation. The output of this table is the cipher text of 8 bit.

Output of step 4 : 0 0 1 0 1 0 1 0

Inverse Initial permutation is defined as –

IP-1(k1, k2, k3, k4, k5, k6, k7, k8) = (k4, k1, k3, k5, k7, k2, k8, k6)

8-bit Cipher Text will be = 0 0 1 1 1 0 0 0

Public Key Cryptography (RSA)

Key generation:

1. Choose P,Q.
2. Compute N=P X Q.
3. Compute Euler(N)= (P-1)X(Q-1).
4. Choose (e) where:
    a. 1<e<Euler(N).
    b. GCD(e,Euler(N)=1.
5. Calculate (d):
    a. **d**Xe=??mod Euler(N)=**1**
6. the generated KEYS:
    a. KU (e,N).
    b. KR (d,N).

Encryption: $C = M^e$ MOD N.

Decryption: $M = C^d$ MOD N.

Example:

Let P=19 and Q=23.

N=P X Q= 437.

Euler (N)= (P-1) X (Q-1)= 18X22=396

Choose e 1<e<396 and GCD (e,Euler)=1

e=13, accepted

Calc. d where e X d=?? Mod Euler(N)=1

Note: قوة خوارزمية ال RSA هي في عدم القدرة على توقع الرقم المختار ضمن مدى الأرقام، وكذلك سرية واحتمالية الرقم الذي يحقق معكوس القسمة.

تبدا بسلسلة مكونة من 10 ارقام

انت تختار ال e بضوء ارقام محددة وفق شروط (4 فقط) على سبيل المثال.

انت يجب ان تحسب ال d بضوء شروط معكوس القسمة وهنا يؤثر اختيار ال e بالتأكيد على حساب ال d.

```
Public static int computeD (int e; int eulern)
{
        Int d=1;
        Repeat
            Modv= ((eXd)mod eulern;
            If (Modv==1) return (d) else d=d+1;
        Until (Modv==1)
}
```

http://umaranis.com/rsa_calculator_demo.html

https://www.cs.drexel.edu/~jpopyack/Courses/CSP/Fa17/notes/10.1_Cryptography/RSAWorksheetv4e.html

https://encryption-calc.herokuapp.com/

http://www.steyrerbrains.at/math/rsa.html

3,7,9,11,13,17,19,21,23,27,29,31,33,37,39,41,43

,

47,49,51,53,57,59,61,63,67,69,71,73,77,79,81,83,87,89,91,93,97,99,101,103,107,109,111,113,117,119,121,123,127,129,131,133,137,139,141,143,147,149,151,153,157,159

E=7 >>>>>> d (23,183)

e.d mod 160=1

e=9>>>>>>>>>>>d (89)

e.d mod 160=1

e=59>>>>>>>>>d=19

e.d mod 160=1???