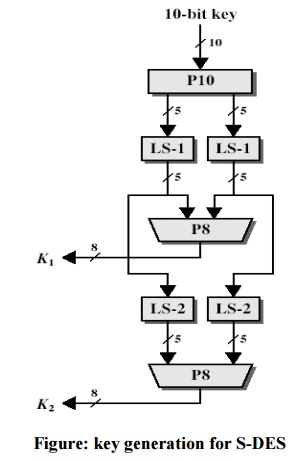
**Simplified DES**

**Key Generation**

****

Given K= 10100 00010

P10

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| I/P | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| O/P | 3 | 5 | 2 | 7 | 4 | 10 | 1 | 9 | 8 | 6 |

P8

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i/p | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| o/p | 6 | 3 | 7 | 4 | 8 | 5 | 10 | 9 |  |  |

K=1 0 1 0 0 0 0 0 1 0

1 2 3 4 5 6 7 8 9 10

Step 1: **Calculate P10 from the Key**

# P10

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| I/P | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| O/P | 3 | 5 | 2 | 7 | 4 | 10 | 1 | 9 | 8 | 6 |

Key value after P10 🡺10000 01100

Step 2: Split the Key as two 5-bit key

10000 🡪 Left shift by 1 🡺LS-1=00001

01100 🡪 Left shift by 1 🡺LS-1=11000

Step 3: **Calculate P8 from the Key**

Key as 10-bit key 0 0 0 0 1 1 1 0 0 0

1 2 3 4 5 6 7 8 9 10

P8

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i/p | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| o/p | 6 | 3 | 7 | 4 | 8 | 5 | 10 | 9 |  |  |

K1:10100100 - 10 bit key converted to 8-bit key

Step 4: Take the 10-bit key before P8

Key as 10-bit key 0 0 0 0 1 1 1 0 0 0

1 2 3 4 5 6 7 8 9 10

Step 5: Left shift :

LS-1= 00010 10001

LS-2=00100 00011

P8

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i/p | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| o/p | 6 | 3 | 7 | 4 | 8 | 5 | 10 | 9 |  |  |

Step 6: P8=01000 011

**K2=01000 011 ……………**

**DES:**

**EXPAND AND PERMUTATE**

**EP**

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

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

**IP**

**P4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **I/P** | **1** | **2** | **3** | **4** |
| **O/P** | **2** | **4** | **3** | **1** |

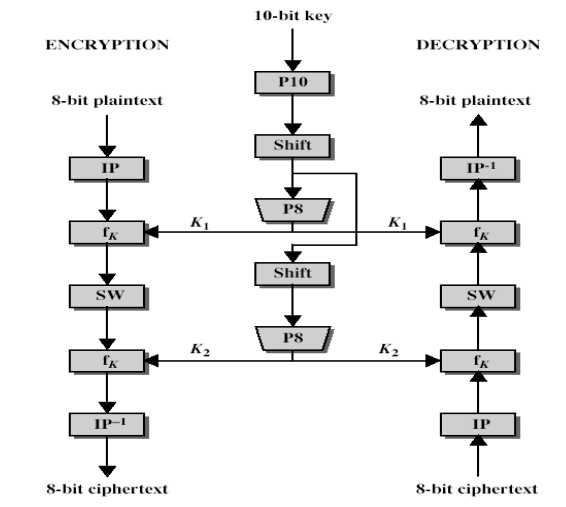
S0=[01 00 11 10



11 10 01 00



1. 10 01 11

11 01 11 10



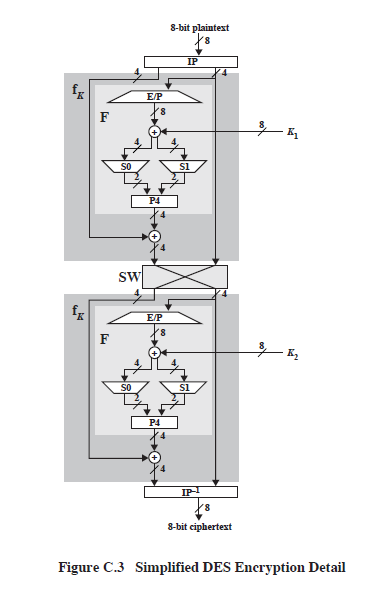
S1= 00 01 10 11

10 00 01 11



11 00 01 00

10 01 00 11



Step 1: Input as 0101 0110

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

IP= 1100 1001

Step 2:Take only right side 4-bit 1001 and do the expansion

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

EP= 1100 0011

K1= 1010 0100

Step 3: calculate the XOR operation

XOR=0110 0111

Step 4: S0=0110

S1=0111

Convert the 4 bit substitution to 2-bit data

S0=[01 00 **11** 10



11 10 01 00



1. 10 01 11

11 01 11 10



S-boxes Rule:

Bit1 and bit4 specifies as row 00

Bit2 and bit3 act as column 11

S0 row=00 =3

S0 col=11=3

S1= 00 01 10 11

10 00 01 11



11 00 01 00

10 01 00 11

S1 row=01=1

S1 col=11=3

S0S1=1011

Step 5:

P4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **I/P** | **1** | **2** | **3** | **4** |
| **O/P** | **2** | **4** | **3** | **1** |

P4=0111

Step 6: XOR ed

XOR(Left side 4-bit input =0101 and P4)

Step 6: Swapping

XOR=1101

Right side IP: 1001

11O1 1001



1001 1101

f1=1101 1001

f2….1110 1101

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

IP-1

# Cipher Text=0111 0111