**Program for PRESENT\_64\_128**

Input:-

Plain Text:- 64-bit- 0000000000000000

Key :- 128 – bit key :- k[0]=0000000000000000 (MSB 64) k[1]=0000000000000000 (LSB 64)

Output:-

Cipher Text at round 31 should be

Cipher Text :- 018839aa80a7d618

**Note : Use k[0] as subkey in the program.**

#include<stdio.h>

unsigned long long int sbox(unsigned long long int);

unsigned long long int player(unsigned long long int);

unsigned int p[64]={ 0, 16, 32, 48, 1, 17, 33, 49, 2, 18, 34, 50, 3, 19, 35, 51,4, 20, 36, 52, 5, 21, 37, 53, 6, 22, 38, 54, 7, 23, 39, 55,8, 24, 40, 56, 9, 25, 41, 57, 10, 26, 42, 58, 11, 27, 43, 59,12, 28, 44, 60, 13, 29, 45, 61, 14, 30, 46, 62, 15, 31, 47, 63};

unsigned int s[16]={0xc,5,6,0xb,9,0,0xa,0xd,3,0xe,0xf,8,4,7,1,2};

int main()

{

unsigned long long int a=0x0000000000000000;

unsigned long long int k[2]={0x0000000000000000,0x0000000000000000},temp,m,i,b1,g,e;

for(i=1;i<32;i++)

{

printf("\nRound :-%d",i); // Shows the value of Round

printf("\nSub Key :- %llx",k[0]); // Always Take K[0] as subkey for process because as per PRESENT Key scheduling standards we take left most 64 bit from 128 bit key

printf("\n Plain Text :- %llx",a); // Plain Text Input

a=a^k[0]; // a is input i.e. plain text and Ex-or with k[0] subkey

printf("\nKey Ex-or: -%llx",a);// Output after Ex-oring 64-bit subkey and plaintext

a=sbox(a);// apply S-box

printf("\n S-Box: %llx",a);// Output of S-box

a=player(a);// Apply P-Layer

printf("\n Cipher Text: %llx",a); // Output of P-layer i.e. this is the cipher text of respective round

//------------------------ Key Scheduling Algorithm Starts----------------------//

//----------------Shift The 128 bit data circularly left by 61-------------------//

temp=k[1];

k[1]= (temp<<61)|(k[0]>>(64-61));

k[0]= (k[0]<<61)|(temp>>(64-61));

//------------------------Apply S-box for bit K127 to K120 -----------------//

b1=0;

for(g=0;g<2;g++)

{

b1|= s[(k[0]>>((g+14)\*4))&0xf]<<g\*4;

}

k[0] &= 0x00ffffffffffffff;

k[0] |= ((((k[0]>>56)&0xff)|b1) << 56);

//-----------------Ex-or With round Counter to bits K66 K65 K64 K63 K62 ----------------//

e=((k[0]&0x7)<<2)|((k[1]>>62)&0x3);// e is a just temparory variable for storing round counter value

e=e^i;

k[0] &= 0xfffffffffffffff8;

k[0] |=((e>>2)&0x7);

k[1] &= 0x3fffffffffffffff;

k[1] |= (e&0x3)<<62;

e=0;

printf("\n-----------------------------------------------");

}

}

unsigned long long int sbox(unsigned long long int in)

{

unsigned long long int i,temp=0,temp1=0;

for(i=0;i<8;i++)

{

temp |= s[(in >> i\*4) &0xf] << i\*4;

temp1 |=s[(in >> (i+8)\*4) &0xf] << i\*4;

}

in = temp1<<32|(temp&0xffffffff);

return (in);

}

unsigned long long int player(unsigned long long int b)

{

unsigned long long int c=0,i;

for(i=0;i<64;i++)

{

c |=((b>>i) & 0x1) << p[i];

}

return(c);

}

Output:-









