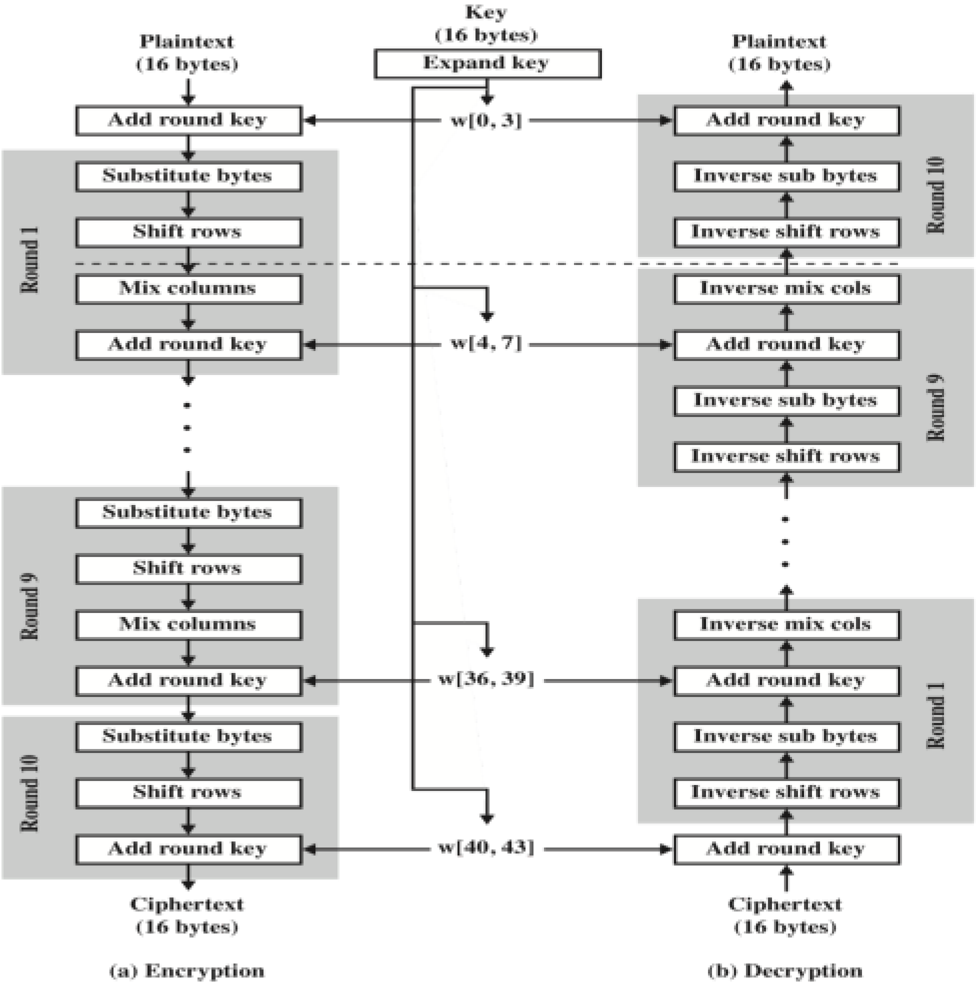
Project #2: Encryption and Decryption using AES algorithm



Description of system

DESCRIPTION

Plain text : 128 bit

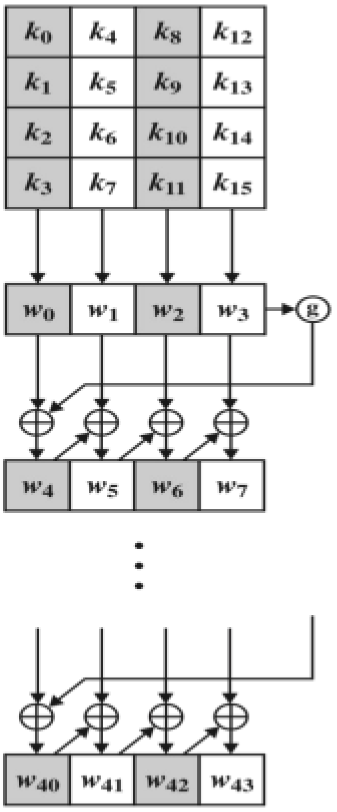
# rounds : 10

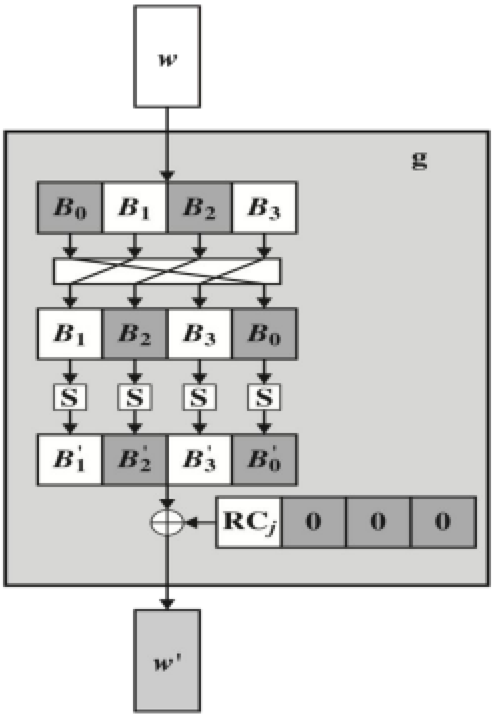
Key size : 128 bit

All computations are in

Irreducible polynomial : xA8 + xA4 + xA3 + X + 1

MODULES


k_0 k_4 k_8 k_{12}
k_1 k_5 k_9 k_{13}
k_2 k_6 k_{10} k_{14}
k_3 k_7 k_{11} k_{15}
w_0 w_1 w_2 w_3
g
w_4 w_5 w_6 w_7
w_{40} w_{41} w_{42} w_{43}



(b) Function g

1. GET SUBKEYS

   a. It takes the initial key (seed) and number of rounds as input

   b. Retums the key list consisting of keys for each of the rounds.

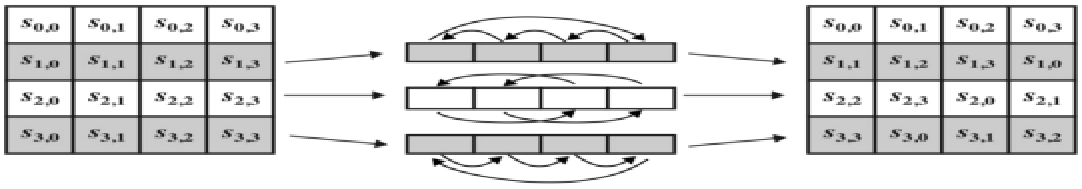
2.SUBSTITUTE BYTES

   a. Performs the transformation of the 4x4 input state matrix.

   b. For each element, calls "SUBSTITUTE" which replaces the byte using

   S-Boxes/lnverse-S-Boxes implemented as a lookup-table.

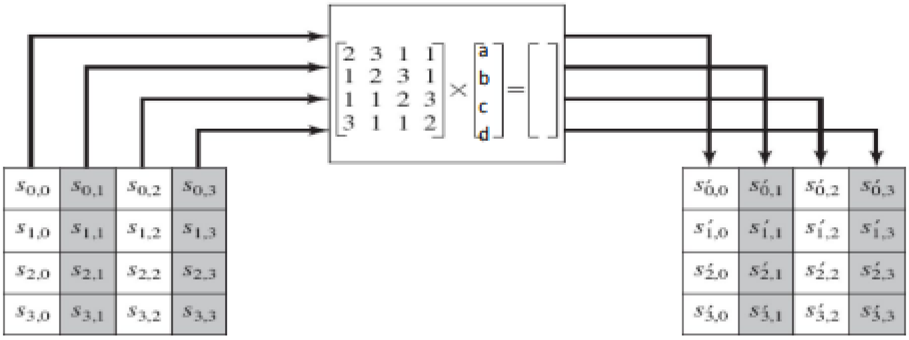
3.SHIFT_ROWS


$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,1}$ $s_{1,2}$ $s_{1,3}$ $s_{1,0}$
$s_{2,2}$ $s_{2,3}$ $s_{2,0}$ $s_{2,1}$
$s_{3,3}$ $s_{3,0}$ $s_{3,1}$ $s_{3,2}$

a. Performs the left shift row transformation on the input 4x4 state matrix.

b. IJsing numpy.roll for fast and easy implementation.

4.MIX_COLUMNS


2 3 1 1
1 2 3 1
1 1 2 3
3 1 1 2
×
a
b
c
d
=
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s'_{0,0}$ $s'_{0,1}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{1,1}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,1}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,1}$ $s'_{3,2}$ $s'_{3,3}$

a. Performs the mix column transformation on the input 4x4 state matrix.

b. Using "galois" python library which is an extension to the numpy library, helps in faster matrix multiplication in Galois Field.

c. GF = galois.GF(2**8, (1,0,0,0,1,1,0,1,1))

5.ADD_ROUND KEY

a. Performs the add round key transformation on the input state matrix.

b. Nothing more than just element-wise, bitwise xor operation between the state matrix and the subkey.

6.ENCRYPT

a. ENCRYPT_ROUND implements a single round of encryption allowing to omit the MixColumns transformation (as needed for round 10).

b. ENCRYPT module performs the initial add round key, followed by the 10 calls to the ENCRYPT ROUND module above, the tenth one specifying to omit the mix

7.DECRYPT

a.DECRYPT_ROUND implements a single round of decryption allowing to omit the InverseMixColumns transformation (as needed for round 1 All the transformations are inverses w.r.t the ENCRYPT ROUND.

b.DECRYPT module performs 10 calls to the DECRYPT_ROIJND module above, the first one omitting the inverse mix columns, followed by the final add round key

8.Operation.

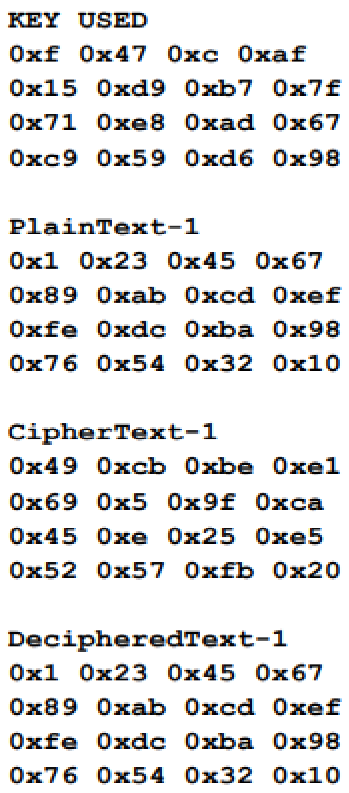
Other Helper Functions

a. Cal_decimal - converts binary to decimal

b. Cal_subKey - calculates the g function of last 32-bit of previous round sub-key

c. Print hex - Prints a state as hex codes.

Sample Input and Output



```
KEY USED
0xf 0x47 0xc 0xaf
0x15 0xd9 0xb7 0x7f
0x71 0xe8 0xad 0x67
0xc9 0x59 0xd6 0x98

PlainText-1
0x1 0x23 0x45 0x67
0x89 0xab 0xcd 0xef
0xfe 0xdc 0xba 0x98
0x76 0x54 0x32 0x10

CipherText-1
0x49 0xcb 0xbe 0xe1
0x69 0x5 0x9f 0xca
0x45 0xe 0x25 0xe5
0x52 0x57 0xfb 0x20

DecipheredText-1
0x1 0x23 0x45 0x67
0x89 0xab 0xcd 0xef
0xfe 0xdc 0xba 0x98
0x76 0x54 0x32 0x10
```

```
PlainText-2
0x1 0x89 0xfe 0x76
0x23 0xab 0xdc 0x54
0x45 0xcd 0xba 0x32
0x67 0xef 0x98 0x10

CipherText-2
0xff 0x8 0x69 0x64
0xb 0x53 0x34 0x14
0x84 0xbf 0xab 0x8f
0x4a 0x7c 0x43 0xb9

DecipheredText-2
0x1 0x89 0xfe 0x76
0x23 0xab 0xdc 0x54
0x45 0xcd 0xba 0x32
0x67 0xef 0x98 0x10
```