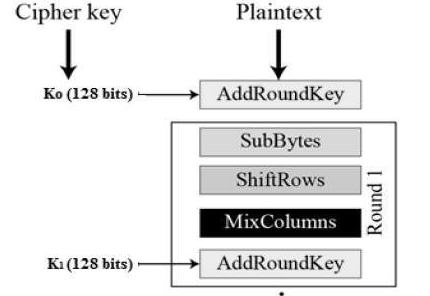
# Encryption steps



# Code

# -\*- coding: utf-8 -\*-

"""

Created on Tue Mar 24 20:06:12 2020

@author: waqar

"""

def rotate(list\_, num):

return list\_[num:] + list\_[:num]

import numpy as np

A=[0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00]

key0=[0x2b,0x28,0xab,0x09,0x7e,0xae,0xf7,0xcf,0x15,0xd2,0x15,0x4f,0x16,0xa6,0x88,0x3c]

key1=[0xa0,0x88,0x23,0x2a,0xfa,0x54,0xa3,0x6c,0xfe,0x2c,0x39,0x76,0x17,0xb1,0x39,0x05]

sbox = [0x63, 0x7c, 0x77, 0x7b, 0xf2, 0x6b, 0x6f, 0xc5, 0x30, 0x01, 0x67,

0x2b, 0xfe, 0xd7, 0xab, 0x76, 0xca, 0x82, 0xc9, 0x7d, 0xfa, 0x59,

0x47, 0xf0, 0xad, 0xd4, 0xa2, 0xaf, 0x9c, 0xa4, 0x72, 0xc0, 0xb7,

0xfd, 0x93, 0x26, 0x36, 0x3f, 0xf7, 0xcc, 0x34, 0xa5, 0xe5, 0xf1,

0x71, 0xd8, 0x31, 0x15, 0x04, 0xc7, 0x23, 0xc3, 0x18, 0x96, 0x05,

0x9a, 0x07, 0x12, 0x80, 0xe2, 0xeb, 0x27, 0xb2, 0x75, 0x09, 0x83,

0x2c, 0x1a, 0x1b, 0x6e, 0x5a, 0xa0, 0x52, 0x3b, 0xd6, 0xb3, 0x29,

0xe3, 0x2f, 0x84, 0x53, 0xd1, 0x00, 0xed, 0x20, 0xfc, 0xb1, 0x5b,

0x6a, 0xcb, 0xbe, 0x39, 0x4a, 0x4c, 0x58, 0xcf, 0xd0, 0xef, 0xaa,

0xfb, 0x43, 0x4d, 0x33, 0x85, 0x45, 0xf9, 0x02, 0x7f, 0x50, 0x3c,

0x9f, 0xa8, 0x51, 0xa3, 0x40, 0x8f, 0x92, 0x9d, 0x38, 0xf5, 0xbc,

0xb6, 0xda, 0x21, 0x10, 0xff, 0xf3, 0xd2, 0xcd, 0x0c, 0x13, 0xec,

0x5f, 0x97, 0x44, 0x17, 0xc4, 0xa7, 0x7e, 0x3d, 0x64, 0x5d, 0x19,

0x73, 0x60, 0x81, 0x4f, 0xdc, 0x22, 0x2a, 0x90, 0x88, 0x46, 0xee,

0xb8, 0x14, 0xde, 0x5e, 0x0b, 0xdb, 0xe0, 0x32, 0x3a, 0x0a, 0x49,

0x06, 0x24, 0x5c, 0xc2, 0xd3, 0xac, 0x62, 0x91, 0x95, 0xe4, 0x79,

0xe7, 0xc8, 0x37, 0x6d, 0x8d, 0xd5, 0x4e, 0xa9, 0x6c, 0x56, 0xf4,

0xea, 0x65, 0x7a, 0xae, 0x08, 0xba, 0x78, 0x25, 0x2e, 0x1c, 0xa6,

0xb4, 0xc6, 0xe8, 0xdd, 0x74, 0x1f, 0x4b, 0xbd, 0x8b, 0x8a, 0x70,

0x3e, 0xb5, 0x66, 0x48, 0x03, 0xf6, 0x0e, 0x61, 0x35, 0x57, 0xb9,

0x86, 0xc1, 0x1d, 0x9e, 0xe1, 0xf8, 0x98, 0x11, 0x69, 0xd9, 0x8e,

0x94, 0x9b, 0x1e, 0x87, 0xe9, 0xce, 0x55, 0x28, 0xdf, 0x8c, 0xa1,

0x89, 0x0d, 0xbf, 0xe6, 0x42, 0x68, 0x41, 0x99, 0x2d, 0x0f, 0xb0,

0x54, 0xbb, 0x16]

print("cipher text:")

original=np.reshape(A, (-1, 4))

print (original)

print("Key 0:")

print (np.reshape(key0, (-1, 4)))

print ("Key 1:")

print (np.reshape(key1, (-1, 4)))

print ("Sbox:")

print (np.reshape(sbox, (-1, 16)))

#add round key

for i in range (len(A)):

A[i]^=key0[i]

print ("Initial Key Addition :")

print (np.reshape(A, (-1, 4)))

#SubByte

for i in range (len(A)):

A[i]=sbox[A[i]]

print ("After SBox :")

print (np.reshape(A, (-1, 4)))

#shift row

A1=A[0:4]

A2=A[4:8]

A3=A[8:12]

A4=A[12:16]

a=rotate(A1,0)

b=rotate(A2,1)

c=rotate(A3,2)

d=rotate(A4,3)

A=a+b+c+d

print ("After Shift :")

print (np.reshape(A, (-1, 4)))

#Mix Columns

d=[0xd4,0xbf,0x5d,0x30]

print ("After Mix Columns:")

A=np.reshape(A, (-1, 4))

col1=A[:,0]

col2=A[:,1]

col3=A[:,2]

col4=A[:,3]

for i in range (4):

if i==0:

a[i]=col1[i]\*d[i]

elif i==1:

a[i]=col1[i-1]\*d[i]^col1[i]\*d[i-1]

elif i==2:

a[i]=col1[i]\*d[0]^col1[i-1]\*d[1] ^ col1[i-2]\*d[2]

elif i==3:

a[i]=col1[i]\*d[0]^col1[i-1]\*d[1] ^ col1[i-2]\*d[2] ^col1[i-3]\*d[3]

for i in range (4):

if i==0:

b[i]=col2[i]\*d[i]

elif i==1:

b[i]=col2[i-1]\*d[i]^col2[i]\*d[i-1]

elif i==2:

b[i]=col2[i]\*d[0]^col2[i-1]\*d[1] ^ col2[i-2]\*d[2]

elif i==3:

b[i]=col2[i]\*d[0]^col2[i-1]\*d[1] ^ col2[i-2]\*d[2] ^col2[i-3]\*d[3]

for i in range (4):

if i==0:

c[i]=col3[i]\*d[i]

elif i==1:

c[i]=col3[i-1]\*d[i]^col3[i]\*d[i-1]

elif i==2:

c[i]=col3[i]\*d[0]^col3[i-1]\*d[1] ^ col3[i-2]\*d[2]

elif i==3:

c[i]=col3[i]\*d[0]^col3[i-1]\*d[1] ^ col3[i-2]\*d[2] ^col3[i-3]\*d[3]

for i in range (4):

if i==0:

d[i]=col4[i]\*d[i]

elif i==1:

d[i]=col4[i-1]\*d[i]^col4[i]\*d[i-1]

elif i==2:

d[i]=col4[i]\*d[0]^col4[i-1]\*d[1] ^ col4[i-2]\*d[2]

elif i==3:

d[i]=col4[i]\*d[0]^col4[i-1]\*d[1] ^ col4[i-2]\*d[2] ^col4[i-3]\*d[3]

A=np.vstack((a, b,c,d)).T % 256

print (A)

for i in range (len(A)):

A[i]^=key1[i]

print ("Round Key Addition :")

print (np.reshape(A, (-1, 4)))

# Output

