Crypto hw8 yanliang

112889478 3/29/2018

1.

Since it is the hill encrypt so let A three rows, (a11,a12,a13), 2nd row (a21,a22,a23),3rd row is (a31,a31,a33)

Since (1,0,0)->(101)

After the multiplication we get a11=1,a21=0,a31=1

Similairly from (110)->(110)

We get a12=0 a22=1 a31=1

And from (111)->(001)

We get a13=1 a23=1 a33=0

So the three columns of A is (1,0,1),(0,1,1),(1,1,0)

2.

0x93=10010011

So in polynomial it will be a^7+a^4+a+1

And by sage following code get the inverse of It is a^6 + a^5 + a^3 + a^2 + 1

//sage code

sage: f2x.<x>=GF(**2**)[]

sage: aes.<a>=QuotientRing(f2x,x^**8**+x^**4**+x^**3**+x+**1**)

sage: (a^**7**+a^**4**+a+**1**)^(-**1**)

a^6 + a^5 + a^3 + a^2 + 1

//end of sage code

In binary it will b e(01101101)

And before multiply the mixColumns matrix, let reverse that

To be 10110110, and after the multiplication of the matrix, we get 11111101

Then use 10111111 xor 01100011 we get 11011100=0xdc

3.

The python code is attach here

```
def getBinary(my_hexdata):
    scale = 16 ## equals to hexadecimal
    num_of_bits = 8
    return bin(int(my_hexdata, scale))[2:].zfill(num_of_bits)
def getHex(my_binary):
    return hex(int(my_binary,2))
def fieldMul(number,bina,myascii):
    if(number==1):
        return bina;
    elif(number==2):
        tempbin=makeDecimalToBinShift((myascii))
        if(bina[0]=='1'):
            return XOR(tempbin,"00011011")
        else:
            #print("did u come here mam?")
            return tempbin
    else:
        first=fieldMul(1,bina,myascii)
        second=fieldMul(2,bina,myascii)
        return XOR(first,second)
def XOR(a,b):
    return bin(int(a,2)^int(b,2))[2:].zfill(len(a))
def makeDecimalToBinShift(number):
    a=bin(number)[2:].zfill(8)
    return a[1:]+'0'
def makeDecimalToBin(number):
    a=bin(number)[2:].zfill(8)
    return a;
```

```
inputdata=[]
col1=['O','H','I','N']
col2=['K','O','L','O']
col3=['L','M','L','I']
col4=['A','A','I','S']
m=[]
row1=[2,3,1,1]
row2=[1,2,3,1]
row3=[1,1,2,3]
row4=[3,1,1,2]
m.append(row1)
m.append(row2)
m.append(row3)
m.append(row4)
result1=[]
rcol1=[]
rcol2=[]
rcol3=[]
rcol4=[]
result1.append(rcol1)
result1.append(rcol2)
result1.append(rcol3)
result1.append(rcol4)
def getOneOutput(col,rcol,rowNo,m):
    count=0
    tempresult='00000000'
    minuse=m[rowNo]
    for ele in col:
        temp=fieldMul(minuse[count],makeDecimalToBin(ord(ele)),ord(ele))
```

```
        #print(temp)
        tempresult=XOR(tempresult,temp)
        #print("temp result "+str(rowNo)+": "+tempresult)
        count=count+1
    #print(temp)
    rcol.append(hex(int(str.encode(tempresult),2)))
for rowNo in range(0,4):
    getOneOutput(col1,rcol1,rowNo,m)
for rowNo in range(0,4):
    getOneOutput(col2,rcol2,rowNo,m)
for rowNo in range(0,4):
    getOneOutput(col3,rcol3,rowNo,m)
for rowNo in range(0,4):
    getOneOutput(col4,rcol4,rowNo,m)
```

Result1

Final result

[['0x41', '0x4a', '0x47', '0x4c'], ['0x44', '0x4e', '0x4d', '0x40'], ['0x4a', '0x4b', '0x42', '0x47'], ['0x5b', '0x4b', '0x67', '0x6d']]