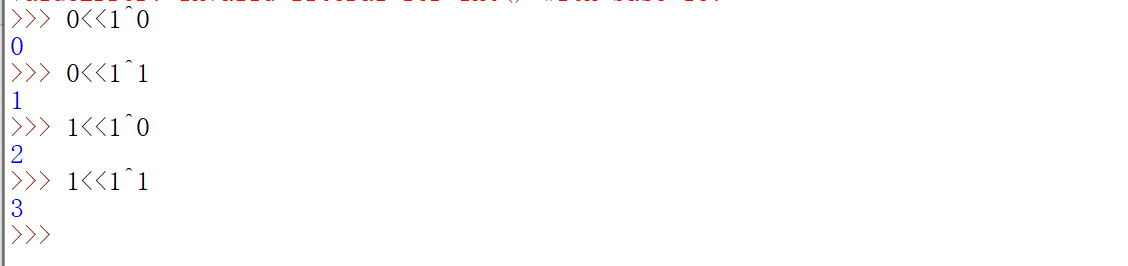
常规的lfsr，只是两个lfsr产生的输出流out1,out2,再计算out=ou1<<1^out2,可以发现out和out1,out2的组合是一一对应的，也就是说通过out就可以反求out1，out2,对应如下：



输出的tmp就是out的值，分别还原out1，out2流，然后就是n级lfsr已知掩码mask的话，已知n位输出就可以求到初始状态，就很简单了。最后求到flag。

Exp：

with open(r"key.txt","rb") as f:

c=f.read()

f.close

from Crypto.Util.number import \*

c=bin(bytes\_to\_long(c))[2:]

C=[]

for i in range(0,len(c),2):

C.append(int(c[i:i+2],2))

out1=""

out2=""

for i in C:

if i==3:

out1+="1"

out2+="1"

if i==2:

out1+="1"

out2+="0"

if i==1:

out1+="0"

out2+="1"

if i==0:

out1+="0"

out2+="0"

KEY=[]

mask="1010110101101101001011010111101100010110011110001010110101101101"

for i in range(len(mask)):

if mask[i]=="1":

KEY.append(i)

flag1=""

out1=out1[:64]

for i in range(64):

im="1"+flag1+out1[:63-i]

K=int(im[KEY[0]])

for j in range(1,len(KEY)):

K^=int(im[KEY[j]])

if K==int(out1[63-i]):

flag1="1"+flag1

else:

flag1="0"+flag1

out2=out2[:64]

flag2=""

for i in range(64):

im="1"+flag2+out2[:63-i]

K=int(im[KEY[0]])

for j in range(1,len(KEY)):

K^=int(im[KEY[j]])

if K==int(out2[63-i]):

flag2="1"+flag2

else:

flag2="0"+flag2

flag1=hex(int(flag1,2))[2:]

flag2=hex(int(flag2,2))[2:]

import hashlib

print("flag{"+hashlib.md5((flag1+flag2).encode("utf8")).hexdigest()+"}")

#flag{729c6024d4bed977c361eec011cee114}

写的有点粗糙前面还原out1，out2，后面就是逆lfsr。

Rsa：三素数rsa，已知一组n,e,d，其中d=invert(e,lcm(p-1,q-1,r-1))再用n,7去加密flag

首先根据e，d，n可以爆破求出phi,lam=gcd(p-1,q-1,r-1)

e\*d-1=kphi/lam

爆破公约数lam，lam\*(e\*d-1)=kphi

k=kphi//n +1=lam(e\*d-1)//n+1

exp: from Crypto.PublicKey import RSA

d=6848345389131232097250291554774004483864247462767351912899751705063009304102012225840379809975695209148626539132005837425458215616241927473070938221119168859672496841054510245128799501775634965114887272776082740599056090759932017

#解析公钥文件

with open(r"npubkey.pem", 'r') as f:

key=RSA.importKey(f.read())

n=key.n

e=key.e

#读取密文

cipher=int(open(r"nflag.enc","rb").read().decode(),16)

print(n,e)

#爆破lam,求phi

for lam in range(1,200):

kphi=lam\*(e\*d-1)

k=kphi//n+1

if kphi%k==0:

phi=kphi//k

print(phi,lam)

break

e=7

#e和phi不互素，只是其中一个素数因子有7,把它剔除再求解。

from gmpy2 import \*

m=pow(2,phi//7,n)

1==pow(2,phi,n)

qr=gcd(m-1,n)

p=n//qr

d=invert(e,phi//(p-1))

from Crypto.Util.number import \*

print(long\_to\_bytes(pow(cipher,d,qr)))

#flag{618b6744a7e82e58b4a510dbb43174fac42c2cdc}