de1ctf 部分writeup解析

原创

逃课的小学生 于 2020-05-09 00:21:19 发布 1114 收藏 2
分类专栏： ctf crypto

1.NLFSR:

在ctf-wiki中我们可以找到一种对NLFDSR的攻击方法——https://wiki.x10sec.org/crypto/streamcipher/fsr/nfsr/。由于对多个LFSR组合时不能实现均匀混合吗，我们可以发现有的值和最后结果相近，我们可以据此对NLFSR中的部分LFSR进行爆破。最后我们可以对剩下的LFSR进行爆破。首先我们可以根据多个LFSR组合判断哪些LFSR爆破。

```
ass=0
bs=0
cs=0
ds=0
print "%s,%s,%s,%s,%s"%("ao","bo","co","do","re")
for i in xrange(0b10000,0b100000):
 ao=int(bin(i)[-4],10)
 bo=int(bin(i)[-3],10)
 co=int(bin(i)[-2],10)
 do=int(bin(i)[-1],10)
 re=(ao*bo) ^ (bo*co) ^ (bo*do) ^ co ^ do

 print "%d,%d,%d,%d,%d"%(ao,bo,co,do,re)
 if ao==re:
  ass=ass+1
 if bo==re:
  bs=bs+1
 if co==re:
  cs=cs+1
 if do==re:
  ds=ds+1
print  "%d,%d,%d,%d"%(ass,bs,cs,ds)
```

我们发现结果中a和最后结果有75%相似，我们首先对a进行爆破。

```
def lfsr(r, m): return ((r << 1) & 0xffffff) ^ (bin(r & m).count('1') % 2)
def guess(beg, end, num, mask):
 f=open('data','r')
 data = f.read()[:1600]
 f.close()
 now = 0
 res = 0
 for i in range(beg, end):
  r = i
  cnt = 0
  jie=0
  for j in range(num * 8):
   r = lfsr(r, mask)
   lastbit=r & 1
   if lastbit == int(data[j],10):
    cnt=cnt+1
  if cnt > now:
   now=cnt
   res = i
   print cnt, res
 return res

guess(pow(2,18),pow(2,19),100,0x505a1)
```

我们发现a=363445时，a所对应的lfsr和总的nlfsr结果有接近75%类似，我们已将a猜出，接下来我们观察之前的结果发现当a为1时，b所对应的lfsr和总的nlfsr结果有接近75%类似，当a为0时，b所对应的lfsr的反和总的nlfsr结果有接近75%类似。由于a已经确定，我们借助a爆破b

```
def lfsr(r, m): return ((r << 1) & 0xffffff) ^ (bin(r & m).count('1') % 2)
def guess(beg, end, num, mask):
 f=open('data','r')
 data = f.read()[:800]
 f.close()
 now = 0
 res = 0
 for i in range(beg, end):
  r = i
  cnt = 0
  a=363445
  for j in range(num * 8):
   r = lfsr(r, mask)
   a = lfsr(a, 0x505a1)
   lastbit=r & 1
   alastbit = a & 1
   if alastbit==1 and lastbit == int(data[j],10):
    cnt=cnt+1
   if alastbit==0 and lastbit != int(data[j],10):
    cnt=cnt+1
  if cnt > now:
   now = cnt
   res = i
   print now, res
 return res
guess(pow(2,18),pow(2,19),100,0x40f3f)
```

我们得到b是494934，剩下的c和d可以直接爆破获得

```
def lfsr(r, m): return ((r << 1) & 0xffffff) ^ (bin(r & m).count('1') % 2)
a=363445
b=494934
ma, mb, mc, md = 0x505a1, 0x40f3f, 0x1f02, 0x31
f=open('data','r')
data = f.read()[:800]
f.close()
for ha in range(pow(2,12),pow(2,13)):
 for hah in range(pow(2,5),pow(2,6)):
  c=ha
  d=hah
  flag = True
  a=363445
  b=494934
  for j in  xrange(30 * 8):
   a = lfsr(a, ma)
   b = lfsr(b, mb)
   c = lfsr(c, mc)
   d = lfsr(d, md)
   [ao, bo, co, do] = [i & 1 for i in [a, b, c, d]]
   lastbit=(ao*bo) ^ (bo*co) ^ (bo*do) ^ co ^ do
   if  not  str(lastbit)== data[j]:
    #print j
    flag = False
    break
  if flag:
   print "c is %d,d is %d"%(ha,hah)
```

2.ECDH

这是在之前JWE上的一个漏洞，在题目中我们发现在重新输入公钥时没有检测公钥所对应点是否在原先的椭圆曲线上。而无论在点的加法还是乘法都不涉及b，所以我们可以选择一个不在椭圆曲线上的点改变椭圆曲线以降低密钥的阶。然而我们使用多组密钥构成多组同余方程。然后使用中国剩余定理求解密钥。首先我们来找低阶的椭圆曲线上的点以及对应的阶

```
q = 0xdd7860f2c4afe6d96059766ddd2b52f7bb1ab0fce779a36f723d50339ab25bbd
a = 0x4cee8d95bb3f64db7d53b078ba3a904557425e2a6d91c5dfbf4c564a3f3619fa
FF = GF(q)
su=1
while su<q:
 for b in xrange(1000,20000):
  E = EllipticCurve([FF(a), FF(b)])
  ss = str(E.order().factor()).split('*')
  for i in ss:
   if '^' not in i:
    i = int(i.strip())
    if 100 < i < 10000:
     su=su*i
     P = E.random_point() * Integer(E.order()/i)
     print i, P
```

注意将结果中阶相同的值删掉，且选取的阶之间互斥。并保证所选阶的乘积大于模P

```
from pwn import *
import gmpy2
zero = (0,0)
def crt(ll):
 N=160100368002089863336612588235555840059619800749562435161043158894093919344765004036542503281
 su=0
 for i in ll:
  Mi=N/i
  Miphi=gmpy2.invert(Mi,i)
  su=(su+Mi*Miphi*ll[i])%N
 return su
def add(p1,p2):
 if p1 == zero:
  return p2
 if p2 == zero:
  return p1
 (p1x,p1y),(p2x,p2y) = p1,p2
 if p1x == p2x and (p1y != p2y or p1y == 0):
     return zero
 if p1x == p2x:
     tmp = (3 * p1x * p1x + a) * gmpy2.invert(2 * p1y , q) % q
 else:
     tmp = (p2y - p1y) * gmpy2.invert(p2x - p1x , q) % q
 x = (tmp * tmp - p1x - p2x) % q
 y = (tmp * (p1x - x) - p1y) % q
 return (int(x),int(y))

def mul(n,p):
 r = zero
 tmp = p
 while 0 < n:
     if n & 1 == 1:
         r = add(r,tmp)
     n, tmp = n >> 1, add(tmp,tmp)
 return r
def proof(a,b):
 ss="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
 print b
 for i in ss:
  for j in ss:
   for k in ss:
    for o in ss:
     jie=i+j+k+o+a
     ll=hashlib.sha256(jie).hexdigest()
     if ll==b:
      return jie
dic=上文结果{阶:{点}}
shou=(30850180072163607425727221374286281042832069301111007754078234396927723013762 , 129917155716996486657
q = 0xdd7860f2c4afe6d96059766ddd2b52f7bb1ab0fce779a36f723d50339ab25bbd
a = 0x4cee8d95bb3f64db7d53b078ba3a904557425e2a6d91c5dfbf4c564a3f3619fa
io=remote("134.175.225.42",8848)
io.recvuntil("+")
tian=io.recvuntil(")")[:-1]
io.recvuntil("== ")
sh256=io.recvuntil("\n")[:-1]
print tian.encode("hex")
print sh256.encode("hex")
io.recvuntil(":")
haha=proof(tian,sh256)
```

```
io.sendline(haha[:4])
print haha
ls={}
msg="f"*128
msghex=0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
io.recvuntil("X:\n")
io.sendline(str(shou[0]))
io.recvuntil("Y:\n")
io.sendline(str(shou[1]))
io.recvuntil("Tell me your choice:\n")
io.sendline("Encrypt")
io.recvuntil("Give me your message(hex):\n")
io.sendline(msg)
io.recvuntil("The result is:\n")
miwen=int(io.recvuntil("\n")[:-1],16)
gongyao=miwen^msghex
gyzuobiao=(int(hex(gongyao)[2:66],16),int(hex(gongyao)[66:130],16))
sign=True
for j in xrange(1693+1):
 if mul(j,shou)==gyzuobiao:
  ls[1693]=j
  sign=False
  break
if sign:
 print "1693"
 io.interactive()
for i in dic:
 sign=True
 io.recvuntil("Tell me your choice:\n")
 io.sendline("Exchange")
 io.recvuntil("X:\n")
 io.sendline(str(dic[i][0]))
 io.recvuntil("Y:\n")
 io.sendline(str(dic[i][1]))
 io.recvuntil("Tell me your choice:\n")
 io.sendline("Encrypt")
 io.recvuntil("Give me your message(hex):\n")
 io.sendline(msg)
 io.recvuntil("The result is:\n")
 miwen=int(io.recvuntil("\n")[:-1],16)
 gongyao=miwen^msghex
 gyzuobiao=(int(hex(gongyao)[2:66],16),int(hex(gongyao)[66:130],16))
 for j in xrange(i+1):
  if mul(j,dic[i])==gyzuobiao:
   sign=False
   ls[i]=j
   break
 if sign:
  print i
  io.interactive()
se=crt(ls)
print "yes"
io.recvuntil("Tell me your choice:\n")
io.sendline("Backdoor")
io.recvuntil("Give me the secret:\n")
io.sendline(str(se))
io.interactive()
#(脚本存在一些问题，需要多跑十几次，原因未知)
```

```
[+] Opening connection to 134.175.225.42 on port 8848: Done
636e69786b493454796538414c457467
65373136646135333138376463346436303363383439373165326630616165646 33
306265393539623933326334663434353331623930336138
e716da53187dc4d603c84971e2f0aaedc58316f10be959b932c4f44531b903a8
hqTwcnixkI4Tye8ALEtg
yes
[*] Switching to interactive mode
Wow! How smart you are! Here is your flag:
De1CTF{c47b5984-1a7c-49f5-a2e3-525d83b50ecf}Something error!
[*] Got EOF while reading in interactive
$
```


3.easyrsa

参考cryptanalysis of rsa and its variants第7章，唯一的问题时我们不知道δ是多少，我们需要爆破验证

```
N= 24402191928494981635640497435944050736451453218629774561432653700273120014058697415669445779441226800209
e1= 40463163242918669105715145616579959622951583647029334603894688270728728652939208142665152287104389702 57
e2= 10895986718189312854870245261598411076951978229292592994243405039714982648044851876570648614223964976 30
cipher= 50892498886184599475480747595245896064785788153360599491767181570240226780248417588568132413351913 1
i=1.356
s=""
while i>1.00:
 B=[]
 B.append([1,-N,0,pow(N,2)])
 B.append([0,e1,-e1,-e1*N])
 B.append([0,0,e2,-e2*N])
 B.append([0,0,0,e1*e2])
 B=Matrix(ZZ,B)
 D=[]
 D.append([N,0,0,0])
 D.append([0,floor(N**(0.5)),0,0])
 D.append([0,0,floor(N**i),0])
 D.append([0,0,0,1])
 D=Matrix(ZZ,D)
 M=B*D
 K = M.LLL()
 v2=K[0]
 X=M.solve_left(v2)
 phi=floor((e1*X[1])/X[0])
 if phi<N:
  b=phi-N-1
  if b**2>=4*N:
   s=s+str(phi)+"\n"
 i=i-0.001

f=open("changshi.txt","w")
f.write(s)
f.close()
print "wancheng"
```

根据验证的结果，我们一一尝试。

```
import gmpy2
phizonghe=上个程序文件内容
N= 24402191928494981635640497435944050736451453218629774561432653700273120014058697415669445779441226800209
e1= 40463163242918669105715145616579959622951583647029334603894688270728728652939208142665152287104389702257
e2= 10895986718189312854870245261598411076951978229292592994243405039714982648044851876570648614223964976307
cipher= 50892498886184599475480747595245896064785788153360599491767181570240226780248417588568132413351913l
phi=0
for i in phizonghe:
    a=1
    b=gmpy2.mpz(phi-N-1)
    c=gmpy2.mpz(N)
    delat=pow(b,2)-4*a*c
    if gmpy2.iroot(delat,2)[1]:
        phi=i
        break

print phi
d1=gmpy2.invert(e1,phi)
d2=gmpy2.invert(e2,phi)
m1=pow(cipher,d1,N)
m2=pow(cipher,d2,N)
print hex(m1)
print hex(m2)
```

4.Homomorphic

参考论文https://arxiv.org/pdf/1906.07127.pdf，我们对明文一个个进行碰撞验证

```
from pwn import *
import copy
import gmpy2
def proof(a,b):
 ss="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
 print b
 for i in ss:
  for j in ss:
   for k in ss:
    for o in ss:
     jie=i+j+k+o+a
     ll=hashlib.sha256(jie).hexdigest()
     if ll==b:
      return jie
def Roundq(A):
 q=pow(2,54)
 for i in range(len(A)):
  A[i]=A[i]%q
  if A[i]>(q/2):
   A[i]=A[i]-q
 return A
io=remote("106.52.135.150",8848)
#io=remote("106.52.180.168",8848)
io.recvuntil("+")
tian=io.recvuntil(")")[:-1]
io.recvuntil("== ")
sh256=io.recvuntil("\n")[:-1]
print tian.encode("hex")
print sh256.encode("hex")
```

```
io.recvuntil(":")
haha=proof(tian,sh256)
io.sendline(haha[:4])
print haha
io.recvuntil("The enc flag is: \n")
flags=io.recvuntil("Tel")[1:-5]
flagli=flags.split("]\n[")
jishu=0
flagc0=[]
flagc1=[]
while jishu<len(flagli):
 flaglj=flagli[jishu].split(",")
 flagcx=[]
 for i in flaglj:
  flagcx.append(int(i))
 flaglj=flagli[jishu+1].split(",")
 flagcy=[]
 for i in flaglj:
  flagcy.append(int(i))
 flagc0.append(flagcx)
 flagc1.append(flagcy)
 jishu=jishu+2
for hh in xrange(7,20):
 for kk in  xrange(32,128):
  c0=copy.deepcopy(flagc0[hh])
  c1=copy.deepcopy(flagc1[hh])
  c0[0]=c0[0]-kk*70368744177664
  cab=(Roundq(c0),Roundq(c1))
  for i in range(len(cab[0])):
   cab[0][i]=10*cab[0][i]
   cab[1][i]=10*cab[1][i]
  cab=(Roundq(cab[0]),Roundq(cab[1]))
  s0=str(cab[0][0])
  for i in range(1,len(cab[0])):
   s0=s0+","+str(cab[0][i])
  s1=str(cab[1][0])
  for i in range(1,len(cab[1])):
   s1=s1+","+str(cab[1][i])
  io.recvuntil("l me your choice:\n")
  io.sendline("Decrypt")
  io.recvuntil("Please input c0(Separated by commas):\n")
  io.sendline(s0)
  io.recvuntil("Please input c1(Separated by commas):\n")
  io.sendline(s1)
  io.recvuntil("The index:\n")
  io.sendline(str(0))
  io.recvuntil("The result is: \n")
  jie=io.recvuntil("\n")[:-1]
  if int(jie)==0:
   print jie,kk,hh
   break
io.interactive()
```