小塔WEEK4努力加班的wp

crypto

ECC

先用sagemath求出m值以及对应的数据值

以下是放进sagemath的代码:

```
p =
74997021559434065975272431626618720725838473091721936616560359000648651891507
a =
61739043730332859978236469007948666997510544212362386629062032094925353519657
b =
87821782818477817609882526316479721490919815013668096771992360002467657827319
k =
93653874272176107584459982058527081604083871182797816204772644509623271061231
E = EllipticCurve(GF(p),[a,b])
c1
=E(144556136662118995760188351651324381020119882646071465119382497448719649460 84
,25506582570581289714612640493258299813803157561796247330693768146763035791942)
c2
=E(375548711626194567091835091226739296364576222518801992350547345237824838699 31
,71392055540616736539267960989304287083629288530398474590782366384873814477806)
m = c1-k*c2
cipher_left =
68208062402162616009217039034331142786282678107650228761709584478779998734710
cipher_right =
27453988545002384546706933590432585006240439443312571008791835203660152890619
left = cipher_left / m[0]
right = cipher_right / m[1]
print(left)
print(right)
```

再放进python里使用n2s后得到flag

PRNG

伪随机数

已知624个即可预测之后小范围内的生成数

```
from random import Random
from libnum import n2s

def inverse_right(res, shift):
    tmp = res
    bits = len(bin(res)[2:])
```

```
    for i in range(bits // shift):
        tmp = res ^ tmp >> shift
    return tmp

def inverse_right_mask(res, shift, mask):
  tmp = res
  bits = len(bin(res)[2:])
  for i in range(bits // shift):
     tmp = res ^ tmp >> shift & mask
  return tmp

def inverse_left(res, shift):
   tmp = res
   bits = len(bin(res)[2:])
   for i in range(bits // shift):
     tmp = res ^ tmp << shift
   return tmp

 def inverse_left_mask(res, shift, mask):
  tmp = res
  bits = len(bin(res)[2:])
  for i in range(bits // shift):
   tmp = res ^ tmp << shift & mask
  return tmp

def recover(y):
   y = inverse_right(y, 18)
   y = inverse_left_mask(y, 15, 4022730752)
   y = inverse_left_mask(y, 7, 2636928640)
   y = inverse_right(y, 11)
   return y

def clone(record):
   state = [recover(i) for i in record]
   gen = Random()
   gen.setstate((3, tuple(state + [0]), None))
   return gen

data =[888058162, 3094055443...]
ans =[3437104340, 508103176..]

g = clone(data)
for i in range(624):
   g.getrandbits(32)

for i in range(21):
 key = g.getrandbits(32)
 flag=n2s(ans[i]^key)
 print(flag)
```

运行后得到flag