CTF-Crypto题目分析__3

原创

建议密码为123456 于 2021-04-26 19:58:45 发布 145 收藏

分类专栏： ctf-crypto题目分析 文章标签： python 信息安全 编程语言 ssl anaconda

本文链接：https://blog.csdn.net/a6b6c6d5488/article/details/116166817

CTF-Crypto题目分析__3

题目描述：

ezrsa

实训描述： Related Message Attack，最后以SeBaFi{}的形式提交得到的{}中的内容。

实训附件： 下载

下载后，发现有个python文件

ezrsa.py	2020/8/24 16:50	PY 文件	5 KB

解题思路：打开ezrsa.py

ezrsa.py - E:\ctf-穿山甲题目\crypto\ezrsa\ezrsa.py (3.7.7)

File Edit Format Run Options Window Help

```
from secret import flag
from Crypto.Util.number import getPrime, inverse, bytes_to_long, long_to_bytes
from sympy import isprime

m = bytes_to_long(flag)

i=0
p=getPrime(1024)
r=getPrime(1024)
while True:
    i+=1
    q = 5*p+i
    if isprime(q) :
        print(i)
        break

n=p*q*r
e = 65537
c = pow(m, e, n)
p3 = pow(p, 3, n)
q3 = pow(q, 3, n)

print c
print e
print n
print p3
print q3

#c=12183662430097407569702141030761787779939870463641299704388507008195928098942
#e=65537
#n=20361372240024088786698455948788052559208001789410016096382703853157107986024
#p3=36398477312664730121119969097654652596845401345841803683723385709488921968l6
#q3=70307771277791732066335828473460011579914774560021919261228365991551489094 65
```

SyntaxError

Missing parentheses in call to 'print'. Did you mean print(c)?

确定

Ln: 19 Col: 0

发现：

1 需要通过对密文c进行解密得到明文flag，*c=m^e mod n*

2 q = 5*p+i q是p的五倍多一点

3 n=pqr,

因为p<q,所以p的三次<pqr=n,即p的三次 mod n =p的三次，

p=对p的三次开三次方，顺推出q,r

#tips：n=pqr时，phi=(p-1)(q-1)(r-1)

- 解题过程：

  解题脚本如下：

  from Crypto.Util.number import *
  from sympy import isprime
  from gmpy2 import *
  c=1218366243009740756970214103076178777993987046364129970438850700819592809894297201215059392716
  188015198455646772944872890852610718644895309389367569752667967246025721356145547903837404176049
  471223254213813221555022225325708998451718582152444119458892355099751213325036728386931902713973
  346624951699406493497256072128672701144456182011715822238682141719427539392824041334601104897253
  400751798188565870441724607035197511892844987686408988375257738231272598073373838705352328804718
  694651888826674015400521517951730035684243695758667803548521583047482992849004687688989661437295
  625891100277891657740685003609703352618011312648019967034465271567094915976392623051313095922177
  119561818660545890857730858224822479660307631623824931974735553302013438356840659919631794403356
  688177272804295818642244977550048251702638031743909858689978621179833344058155432719697169100409
  27833496696049703621334172902517666284662473059140662717708823
  q3=703077712777917320663358284734600115799147745600219192612283659915514890946505406780080761536
  110844256094205886540318867262929703970306592780177164633481787133513488913989464872952745254109
  844984220283898398250855175066966254061553432715082986996442900613089173147209991293771740612044
  338028354857127031742172204283563973296697581276408401522125511594050845644227990225067766513638
  098890268237087560214583313593721074079052875630105198199435155324785201835552664101243467066473
  292449179094923551960089928951549504635355947580693520002932156354955316723541903992427640605985
  865947632971880965707299738594726265474318124288570955820924958948203667342872303530072228022919
  272719248777221751867383820964630054827595745099482822132929966621645796174618988535692969867429
  494424372973985092711123123506000511978165224523453758318123271596419167524120656288810725256956
  6488402724441835466680342239244581162530424964324562530832713397
  p3=363984773126647301211199690976546525968454013458418036837233857094889219681609583878142302099
  640745740818822523852092748380909107999315155507678137288251881017468715006790387044843629950155
  738050879323825447183327550763473294796490746161918211278791113305427587212024355855669790052842
  767935218196131295866088180073167813448166407471107667229017838999640335707680980542259185114530
  642595172562784335220723369381047461888239414069133474208600896726011774048695564006819044060998
  409565769542353601647546822941918748935956380073726121297592166380372911242022203900547883047745
  559216709252007450924189482930420940671378108295929962367429492724955608348622303685867407717310
  451801360162844750450060644782154068746536161644763157997657975499602165363080407353535212931541
  311876483627075125040564968378648725182324782894720233668053884957149878035335727210369751091057
  6879383751704763858882439578045020243015928994208017750848637513
  n=2036137224002408878669845594878805255920800178941001609638270385315710798602486026272168500041
  771926061193573163407785212743214036179276720258163181654454697275003449406127677987840954477970
  791426167963376477257504030471236163431808628978395155584202102843879964925265204121134182545150
  075176087257240225074798249538426367766952657582518373335380069416142536029952114372668138748509
  728183221900968276852330473725276390793964221254295984663046462813502520348907569869998071598668
  934106996438777952325420302142486535505421512231616020107360410531776811228191433406534942094671
  711656363488336831624749504221633040837217671449901277841016047838450333561032110826370624332974
  578563259970774053438698894525957889761431758254675165848091718846417899702628433686102729928907
  304567775434274638640850569524380068532328385202032504464960454857508992754193588480032712187519
  1739922436199496098842684301207745090701158839031935190703347091
  e=65537

```
p=iroot(p3,3)[0]
i=0
while True:
i+=1
q = 5p+i
if isprime(q):
break
r=n//(pq)
phi=(p-1)(q-1)(r-1)
d=invert(e,phi)
m=pow(c,d,n)
flag=long_to_bytes(m)
print(flag)
```

- flag{d3cada43775097d86a7daec938af3c73}

如果写的不好或者写的有问题，请大佬多多指正，也可以私信~~