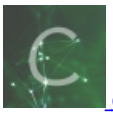
BUUCTF 打卡1

原创

路由(_)生 于 2021-08-19 17:27:03 发布 38 收藏

分类专栏： crypto 文章标签： python

本文链接：https://blog.csdn.net/qq_52193383/article/details/119784153



由于后面的知识盲点太多了，就从头开始把之前不会的再做一遍。

1.RSA5

一大串数据，20次加密过程中的m、e相同。我以为是广播攻击，想着用剩余定理来解，自己编的脚本差劲，就用了书上的脚本。解出个'\x01'。回头再看一下广播攻击的前提，加密指数e较低！！！e = 65537应该算大吧。于是只能将这20个n进行分解，在n5的时候解出了p、q。

```
import libnum,gmpy2
n = 22822039733049388110936778173014765663663303811791283234361230649775805923902173438553927805407463106104699773994158375704033093471761387799852168337898526980521753614307899669015931387819927421875316304591521901592823814417756447695701045846773508629371397013053684553042185725059996791532391626429712416994990889693732805181947970071429309599614973772736556299404246424791660679253884940021728846906344198854779191951739719342908761330661910477119933428550774242910420952496929605686154799487839923424336353747442153571678064520763149793294360787821751703543288696726923909670396821551053048035619499706391118145067
c = 15406498580761780108625891878008526815145372096234083936681442225155097299264808624358826686906535594853622687379268969468433072388149786607395396424104318820879443743112358706546753935215756078345959375299650718555759698887852318017597503074317356745122514481807843745626429797861463012940172797612589031686718185390345389295851075279278516147076602270178540690147808314172798987497259330037810328523464851895621851859027823681655934104713689539848047163088666896473665500158179046196538210778897730209572708430067658411755959866033531700460551556380993982706171848970460224304996455600503982223448904878212849412357
p = 13258580638379860030542695730761256760422356262676419021133313624664372381104614933785296682872905247672555236113243737052154870766497712316527930505297186801275550916040864110054874404662151687798186418007649752409320140455803630182021627496863882524515075577255925957554410191859031106846660161847246483249 9
q = 17213033849932627874808865964211853990326330664462548981326985404970451412059813493478631677191226024836907594886403622960556395007049199264312583859414938163136212054261554515869692536091608647010798777124664545943384132075904866124601687518063545835779913180673477712914184572810281637881560766366013182743 3
e = 65537
phi = (p-1)*(q-1)
d = int(gmpy2.invert(e,phi))
m = pow(c,d,n)
print(libnum.n2s(m))
```

2.rsa2

提示hash前要encode，于是
flag = “flag{” + hashlib.md5(hex(d).encode()).hexdigest() + “}”
print(flag)#flag{8159e6c4abdd3b94ce461ed9a1a24017}
可还是错的。
后面看了一下大佬的，才知道这个程序在python2中才能运行成功。我之前还以为是我不会md5和hash（确实不会）。后面按照大佬的指示，用网站的python2运行才算出结果。
菜鸟工具 python

```
import gmpy2,libnum
import hashlib

N = 10199180977755325347027675139926474013115768232925267350179215450700615843443200914199536724196252570595004625340018888465826249653470643879151507188586089755273665689956691573129722581725063987364337631010399217064690655724283289391490205358108750251278730332274778042021088485216658671763655905815254497947 1
q = 90468539152235033517870318889776271069345640432047835931186781819915963165828770575564631525796216990106105695265730319547795207814485506777675652074071 83
p = 11273732364123571293429600400343309403733952146912318879993851141423284675797325272321856863528776914709992821287788339848962916204774010644058033316303937
e = 46731919563265721307105180410302518676676135509737992912625092976849075262192092549323082367518264378630543338219025744820916471913696072050291990620486581719410354385121760761374229374847695148230596005409978383369740305816082770283909611956355972181848077519920922059268376958811713365106925235218265173085
#e很大
phi = (p-1)*(q-1)
d = int(gmpy2.invert(e,phi))
#flag = "flag{" + hashlib.md5(hex(d)).hexdigest() + "}"
#
print(flag)
```

3.[HDCTF2019]bbbbbbrsa

代码中写的是对c进行base32编码，但是用base32解码不对，使用base64解码。根据代码推测e是要进行爆破的（50000<e<70000）。用try except来处理e模phi没有逆元的情况。

```
import base64,gmpy2,libnum,sympy
from Crypto.Util.number import long_to_bytes
from binascii import a2b_hex,b2a_hex
p = 177077389675257695042507998165006460849
n = 37421829509887796274897162249367329400988647145613325367337968063341372726061
c = '==gMzYDNzIjMxUTNyIzNzIjMyYTM4MDM0gTMwEjNzgTM2UTN4cjNwIjN2QzM5ADMwIDNyMTO4UzM2cTM5kDN2MTOyUTO5YDM0czM3MjM'
q = n//p
Cipher = c[::-1]
c = int(base64.b64decode(Cipher))
#print(c)
phi = (p-1)*(q-1)
e = 50001
'''
while e<70000:
        if gmpy2.gcd(e,phi) == 1:
                try:
                        d = int(gmpy2.invert(e,phi))
                        m = pow(c,d,n)
                        m = str(long_to_bytes(m))
                        if 'CTF' in m or 'flag' in m:
                                print(m)
                except:
                        pass
        e = sympy.nextprime(e)

'''
for e in range (50000,70000):
    if  gmpy2.gcd (e,phi) == 1:
        try:
                d = gmpy2.invert (e,phi)
                m = pow (c,d,n)
                m = str(long_to_bytes(m))
                if 'CTF' in m or 'flag' in m:
                    print(m)
        except:
                pass
```

4.[ACTF新生赛2020]crypto-rsa0

```
import gmpy2,sympy
from Crypto.Util.number import long_to_bytes

p = 90185880664342063772402771624767392713862401730886765262953151639909683470229228412991282745514829264909083992371538834949647434361938539784599470602104 11
q = 75470056738777382578357297600377652133400366963507663242291436131799321451221306857785040624101370436359582088056986981698472935201495726050264927517402 23
c = 50996206925961019415256003394743594106061473865032792073035954925875056079762626648452348856255575840166640519334862690063949316515750256545937498213476286637455803452890781264446030732369871044870359838568618176586206041055000297981733272816089806014400846392307742065559331874972274844992047849472203390350
n = p*q
phi = (p-1)*(q-1)

for e in range(2,100000):
    if sympy.isprime(e) == True:
        if gmpy2.gcd(e,phi) == 1:
            try:
                d = int(gmpy2.invert(e,phi))
                m = pow(c,d,n)
                flag = str(long_to_bytes(m))
                if 'CTF' in flag or 'flag' in flag or 'ctf' in flag:
                    print('e = ',e)
                    print(flag)
            except:
                pass
```

5.[GWCTF 2019]BabyRSA

原来是我想的不仔细。根据原代码，我们可以知道p<q,c2<N，又因为c1 = F1 + F2，c2 = pow(F1, 3) + pow(F2, 3)，可以知道c1<c2<N，而且c2 =(F1+F2) * ($F1^2$ - F1 * F2 + $F2^2$) = (F1+F2)(($F1+F2)^2$ - 3 * F1 * F2)，从而推出F1 * F2。知道了F1 + F2和F1 * F2，我们就可以构造方程 x^2 -(F1+F2)*x+F1 * F2 = 0，解出F1 ，F2。

```python
import gmpy2
from Crypto.Util.number import long_to_bytes

N = 6365851495945747469090301601826908662229092564648472917830006518372279213372378996512879435977732709443840348589252957448807271016068414136400065276148731106514101558937765487378231529437978847291301497582791274300447392540004266109228345730949570825895394456108282794288145243134912620619305128290744662326331305991044908935720939438327403018096308475415925489212002882224327892086509499376383034294564688891001926138590737529238124542122399089489301783553313909335367710657918176439787630450308337123261628838106381200293783370929386621741197476878994846036283440794935566014224984053607319581627192961605840426710571602412848525229136762645962019061631m1 = 900099743414522432169869380283712575286049432089411765187174635547749678781526945864693777652961131656594987260127122886704588843739714198427509292876586402662196866469569298721157821730939797429587451216719285687094685260987159271898296004972831180516411073051288526970320533681151812160696266061655034651257252048755787012377892929662118240027614818152766662368690051291388627824768591030867260918604976148832829499550232224143332431932685647816216998704125578224043812138040266858312214307282907555978192593396166501586747132488416543385151994055320031737325204578139011702647130851070770014780833413390020698705853782570511502175117557614910215532391m2 = 48744398575740517342662818837565711760423550793696752299325797210887228369830523845446572321422687141427678891205818619703982124291273674282408062768097180251120691439467215924020691073585065199931610001469106729570813863936320359624469399556278028663711639473825077412975902108019732372480541466804231880601065281440507876973854891367546618155100552706530951536495061013720639325714835765966668709166274984856022545382636227170429269284759633953322908803882053208610942115857584107760126871317509787408353624900601894878941323878392284563349402360886525607196285658122989004389693902561360056428339132933145219906285893037456599163419149513793957453954631e = int('0x10001',16)
p = 7978628639024219849512313504303122605177732696849584563428609832361841296023909190260484961197571877020764995513107941779179201376468358888627061269240884115709971412571595639527258822141811855312091869723514699462695085113128637791232053223784521942612170165525277545132155203294999671081969688331633297246202510960803777476991m2 = 7978628639024219849512313504303122605177732696849584563428609832361841296023909190260484961197571877020764995513107941779179201376468358888627061269240884115709971412571595639527258822141811855312091869723514699462695085113128637791232053223784521942612170165525277545132155203294999671081969688331633297246202510960803777487371e = 65537phi = (p-1)*(q-1)
d = int(gmpy2.invert(e,phi))
c1 = pow(m1,d,N)# 这里的c1 = (F1 + F2)%N = F1 + F2
c2 = pow(m2,d,N)# c2 = pow(F1, 3) + pow(F2, 3) < N
#c2 = F1**3+F2**3 = (F1+F2)*(F1**2-F1*F2+F2**2) = c1*((F1+F2)**2-3*F1*F2) = c1*(c1**2-3*F1*F2)
#==>F1*F2 = (c1**2-c2//c1)//3
F1F2 = (c1**2-c2//c1)//3
#构造方程x**2-(F1+F2)*x+F1*F2 = 0
pd = int(gmpy2.iroot(c1**2-4*F1F2,2)[0])
F1 = (c1+pd)//2
F2 = (c1-pd)//2
flag1 = long_to_bytes(F1)
flag2 = long_to_bytes(F2)
print(flag1,'\n',flag2)
```