2021年绿城杯网络安全大赛—Crypto(密码)部分

原创

base呗 于 2021-09-30 16:54:50 发布 245 收藏

分类专栏： CTF 密码学 RSA 文章标签： python 密码学

本文链接：https://blog.csdn.net/weixin_52805837/article/details/120568951


CTF 同时被 3 个专栏收录

9 篇文章 0 订阅

订阅专栏


密码学

5 篇文章 0 订阅

订阅专栏


RSA

5 篇文章 0 订阅

订阅专栏

2021年绿城杯网络安全大赛—Crypto(密码)部分

文章目录

RSA1

已知

```
M = 2021 * m * 1001 * p
c = m**e mod n = k*p
```

所以

```
M = 2021 * m * 1001 * p
c = m**e mod n = k*p
```

即可分解n
脚本为：

```
#!/usr/bin/python
from Crypto.Util.number import *
import gmpy2

e = 0x10001
n = 17365231154926348364478276872558492775911760603002394353723603461898405740234715001820111548600914907617003806652492391686710256274156677887101997175692277729648456087534987616743724646598234466094779540729413583826355145277980479040157075453694250572316638348121571218759769533738721506811175866990851972838466307594226293836934116659685215775643285465895317755892754473332034234495795936183610569571016400535362762699517686781602302045048532131426035260878979892169441059467623523060569285570577199236309888155833013721997933960457784653262076135561769838704166810384309655788983073376941843467117256002645962737847
c = 69449671088154377354289412867841194031383197134557321559250559286465369625976729418058313121306893380149134520812964002728627104472072650997504016578281658360131228486568391008547199651886800973754911932491277255996603837468270318030660264979892988564202162502060350681809637974547921511910714336459462459149167326370071170851994428944956674555445174834040065366071214806786880004204222813805393685198071621750997638919886481179377779510698999752601900189958349045414475627183074339065920212266668856388770203040056144507630813370828386084147561622538256974204935099145785469516341275023936470687229953637533219126 76
p = gmpy2.gcd(c, n)
q = n // p
d = gmpy2.invert(e, (q-1) * (p-1) )
m = gmpy2.powmod(c, d, n)
m = m // (2021*1001*p)
m = long_to_bytes(m)
print(m)
#flag{Math_1s_1nterest1ng_hah}
```

warmup

“纺射密码”
参数全部已知
题目脚本为：

```
from Crypto.Util.number import *
from flag import flag
assert flag[:5]=='flag{'

str1 = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'
def encode(plain_text, a, b, m):
 cipher_text = ''
 for i in plain_text:
  if i in str1:
   addr = str1.find(i)
   cipher_text += str1[(a*addr+b) % m]
  else:
   cipher_text += i
 print(cipher_text)


encode(flag,37,23,52)
# cipher_text = 'aoxL{XaaHKP_tHgwpc_hN_ToXnnht}'
```

真的解不出来，大佬脚本如下：

```
str1 = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'
cipher_text = 'aoxL{XaaHKP_tHgwpc_hN_ToXnnht}'
a=37
b=23
m=52

for i in cipher_text:
    for j in range(33,128):
        if(chr(j)in str1):
            addr= str1.find(chr(j))
            cipher_text = str1[(a * addr + b) % m]
        else:
            cipher_text = chr(j)
        if(cipher_text==i):
            print(chr(j),end='')
```

RSA2-PLUS

这题rsa是下午放出来的，当时在上课也没细看
两段脚本，看了大佬的wp后才发现原来是flag被分成两段了

第一段：

```
#!/usr/bin/python
from Crypto.Util.number import *
import gmpy2

n = 6348779979606280884589422188738902470575876294643492831465947360363568026280963989291591157710389629216109615274754718329987990551836115660879103234129921943824061416396264358110216047994331119920503431491509529604742468032906950984256964560405062345280120526771439940278606226153077959057882262745273394986607004406770035459301695806378598890589432538916219821477777021460189140081521779103226953544426441823244765828342973086422949017937701261348963541035128661464068769033772390320426795044617751909787914185985911277628404632533530390761257251552073493697518547350246993679844132297414094727147161169548160586911
c = 6201882078995455673376327652982610102807874783073703018551044780440620679217833227711395689114659144506630609087600915116940111002026241056808189658969089532597757995423694966667948250438579639890580690392400661711864264184444018345499567505424672090632235109624193289954785503512742400960515331371813467034511130432319427185134018830006918682733848618201088649690422818940385123599468595766345668931882249779415788129316594083269412221804774856038796248038700275509397599351533280014908894068141056694660319816046357462684688942519849441237878018480036145051967731081582598773076490918572392784684372694103015244826


e = 0x10001
p1 = q1 = 0

def isqrt(n):
    x=n
    y=(x+n//x)//2
    while(y<x):
        x=y
        y=(x+n//x)//2
    return x

def fermat(n):
    t0=isqrt(n)+1
    counter=0
    t=t0+counter
    temp=isqrt((t*t)-n)
```

```
    #while((temp*temp)!=((t*t)-n)):
    while True :
        counter+=1
        t=t0+counter
        temp=isqrt((t*t)-n)
        s=temp
        p=t+s
        q=t-s
        if (temp*temp)==((t*t)-n) and p != p1 and q != q1:
            return p,q


p1,q1=fermat(n)
print("found 1")
p2,q2=fermat(n)
print("found 2")

list = [p1,q1,p2,q2]
primes = []

assert p1*q1 == n
assert p2*q2 == n

for i in list:
    for j in list:
        if i != j:
            tmp = int(gmpy2.gcd(i,j))
            if tmp != 1 :
                if tmp not in primes:
                    primes.append(tmp)

phi = 1
n = 1

for i in primes:
    phi *= i-1
    n *= i

d = gmpy2.invert(e,phi)
m = gmpy2.powmod(c,d,n)
print(long_to_bytes(m))
# flag{Euler_funct1ons
```

第二段：

```
#!/usr/bin/python
import gmpy2
from Crypto.Util.number import long_to_bytes
from z3 import *

tmp1 = 27477314676113846270813758230909738643779389179369138303385652430301081129410193345482448501052146891484615181987604350854187963754444425652074141849547939377713283098585652200856108841086281591329228868376165791912193001695691686584926115372109767131588346934897292575707808971510203224181852692598864557877 8
tmp2 = 18514724270030962172566965941723224386374076294232652258701085781018776172843355920566035157331579524980108190739141959926523082142273672741849552475156278397131571360099018592018959785627785130126477982765210498547680367230723634424036009539347854344573537848628061468892166199866227984167843139793429682559241317072979374002912607549039431398267184818771503468116379618249319324788996321340764624593443106354104274472601170229835219638093242557547840060892527576940077162990069687019966946826210112318408269749294366586682732614372434218768720577917368726530200897558912687470088583774711767599580037663378929000217
```

```
s = Solver()
p,q = Ints('p q')
s.add(p+q == tmp1)
s.add((p) * (q) == tmp2)
assert s.check() == sat
p = s.model()[p].as_long()
q = s.model()[q].as_long()
n = q*p*q*p*q
e = 65537
assert gmpy2.is_prime(q)
assert gmpy2.is_prime(p)
#a = (q-1) * q**2
#b = (p-1)
phi = (q**3 - q**2 ) * (p**2 - p)
d = gmpy2.invert(e, phi)

c = 25591090168544821761746024178724660839590948190451329227481168576490717242294520739865602061082558759751196452117720647426598261568572440942370039702932821941366792140173428488344932203576334292648255551171274828821657097667106792872200082579319963310503721435500623146012954474613150848083425126987554594651797477741828655238243550266972216752593788734836373144363217639612492397228808215205862281278774096317615918854403992620720969173788151215489908812749179861803144937169587452008097008940710091361183942268245271154461872102813602754439939747566507116519362821255724179093051041994730856401493996771276172343313045755916751082693149885922105491818225012844519264933137622929024918619477538521533548551789739698933067212305578480416163609137189891797209277557411169643568540392303036719952140554435338851671440952865151077383220305295001632816442144022437763089133141886924265774247290306669825085862351732336395617276100374237159580759999593028756939354840677333467281632435767033150052439262501059299035212928041546259933118564251119588970009016873855478556588250138969938599988198494567241172399453741709840486953189764289118312870580993115636710724139809708256360212728127786394411676427828431569046279687481368215137561500777480380501551616577832499521295655237360184159889151837766353116185320317774645294201044772828099074917077896631909654671612557207653830344897644115936322128351494551004652981550758791285434809816872381900401440743578104582305215488888563166054568802145921399726673752722820646807494657299104190123945675647n2 = 40588227045595304080360385041082238507044292731344465815296032905633525556943787610712651675460810768762763493579129831271018141591546207557410817432455139315527674932933085299277599173971912445226532235814580879585317211349524406424200622675880992390782025158621241499693400288031658194434641718026910652327933253877313106112861283314274635124734817398465059373562194694957841264834312640926278890386089611103714990646541470577351599526904458342660444968591197606820361364761648205241041444681145820799054413179462285509661124362074093583494932706249461954240408827087015525507173082129412234486228092002841868365895837463699200959915782767657258729794037776401995309244941171415842403617486719492483671490834562579225506831496881542530519595438932482796867853234159664409420977526102480385193101883785161080269573707156626838551506024455486506224305894501968583442346807126920740779780593650871645915149689424292912611578291912721896864772950410266629045542480009266574096080138709683466489568290569363478444349563498507530805502511051165160827192795520182720802422213364247355775222858214648603034743679187470844212529134374975737510982287957316878179964602394749601431823167982157434890459245394370728942790117156485268116758052636794417268680901420193002289035538753620555488506926366624641291881353268617130968991258983002165300186971963661666476600998389048880565199317280428349802824448329898502788492233381873026217202981921654673840142095839603360666049476100561268336225902504932800605464136192275593886736746497955270280541423593
assert n == n2
m = gmpy2.powmod(c,d,n)
print(long_to_bytes(m))
# _1s_very_interst1ng}
```

组合得flag：`flag{Euler_funct1ons_1s_very_interst1ng}`

前半部分,两次费马分解即可, 四舍五入, 去年“祥云杯”原题
后半部分,z3直接解了, 然后用欧拉函数就可以了

总结

2021年绿城杯网络安全大赛—Crypto(密码)部分总结：

这是第一届绿城杯，很遗憾没有达到进入决赛条件，再接再厉吧

对于rsa部分的学习还是冰山一角，希望能进步的快一点吧，密码学还是蛮有意思的。