XCTF—RSA-gcd

原创

loadding... 于 2020-10-13 20:43:28 发布 549 收藏 2

分类专栏： CTF

本文链接：https://blog.csdn.net/qq_41137110/article/details/109060391

题目地址

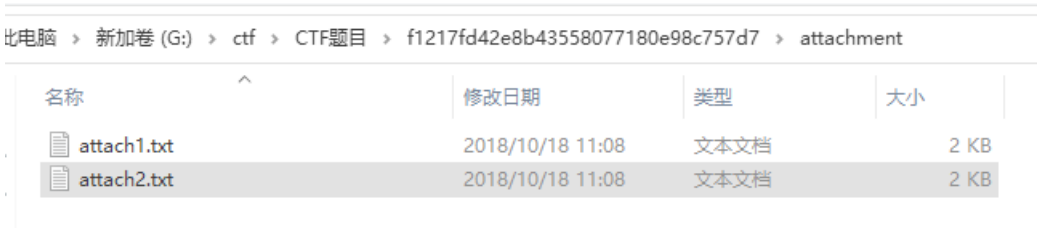
题目名称：RSA_gcd

题目描述：暂无

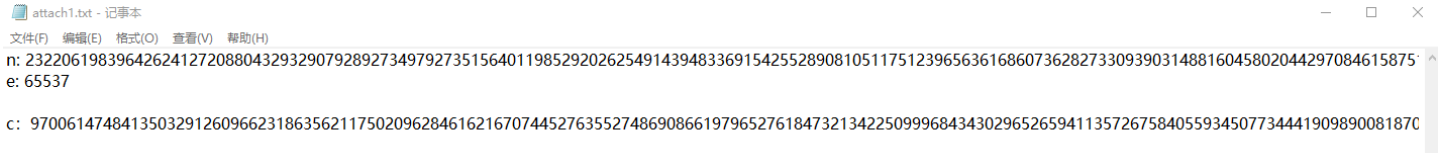
题目附件：附件1

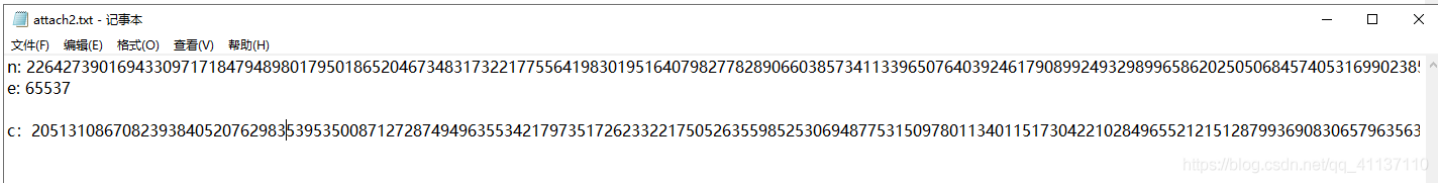
解题思路：

下载附件，解压得到两个文件



文件内容如下：





给出了模数n、公钥指数e、密文c，就是常规的RSA解密，属于常规破解模数，直接用脚本解密就好了。

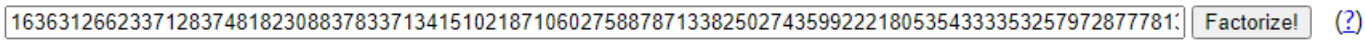
解题过程：

登录网站http://factordb.com/,分别解出p和q，如下

attach1.txt：

Result:
number
2264273901...11<617> = 1383766045...39<309> · 1636312662...49<309>

attach2.txt：



Result:
number
1636312662...49<309> = 1636312662...49<309>

python3脚本解密：

```
# -*- coding: cp936 -*-
import base64
from Crypto.PublicKey import RSA
def egcd(a,b):
    if a==0:
        return (b,0,1)
    else:
        g,y,x=egcd(b%a,a)
        return (g,x-(b//a)*y,y)
def modinv(a,m):
    g,x,y=egcd(a,m)
    if g!=1:
        raise Exception('modular inverse does not exist')
    else:
        return x%m
p1 = 13837660453353041240023955834042470031241270269902248111935779905471587782929163529083271983503314058069005
38656770793162419191691663751236919176752359794627721036813987252858085510419246248828409015838928581742707144 71
36653175897524186847093813823830700578265129617957996186980184139568278264591684852377 1439
q1 = 16780741164967646254666111964411308191554237875577832705715619128445315088766234341490891695315489718361354
80835589194103596424500013436448140211598287248447308811119550509923985350634098281694621809706295377926769986 47
88011046352755503404087148596436141808048111305995941061644677221803814115705100709 1689351
n1 = p1*q1

p2=  13837660453353041240023955834042470031241270269902248111935779905471587782929163529083271983503314058069005
38656770793162419191691663751236919176752359794627721036813987252858085510419246248828409015838928581742707144 71
36653175897524186847093813823830700578265129617957996186980184139568278264591684852377 1439
q2=  16363126623371283748182308837833713415102187106027588787133825027435992221805354333353257972877781350995626
16626154931791606697155038339494203083115817366743322681315346025816268170392373935992226882716073251315297906 40
37576569783274661470048368165891175393652048238359247874309323326137145171884427576209 4649
n2=p2*q2
e = 65537
d1=modinv(e,(p1-1)*(q1-1))#RSA私钥
d2=modinv(e,(p2-1)*(q2-1))#RSA私钥

c1=97006147484135032912609662318635621175020962846162167074452763552748690866197965276184732134225099968434302 96
52659411357267584055934507734441909890081870957764232490040558249968360478698114409987802178456754065404083391 20
63141709913653416394888766281465200682852378794478801329251224801006820925858507273130504236563822120838520746 27
02807311214428394122583971919630360405535396978465350388415412090505030610010709097258065742300902460418914865 06
98093929424553725261094479957392084423522109695639109571611162999859407576250734543094552349277591579082807800 04
53705320783486744734994213028476446922815870053311973844961
c2=20513108670823938405207629835395350087127287494963553421797351726233221750526355985253069487753150978011340 11
51730422102849655212151287993690830657963563952859051542607092631971958287653971892678663489461886527520764721 72
15575594028261521222837036704243520358415932607823892150215108376890874248075678127735835773454569491759192115 01
27540286087770229112383605858821811640935475859936319249757754722093551370392083736485637225052738864742947137 89
03631357097964100088455769852976969226810436499166505993493208189015128350070504254608726758579740699278466209 05
98137486916620289690560034322364029613842389870313723646350 8
#解出明文
m1=pow(c1,d1,n1)#得到的是10进制数据
m2=pow(c2,d2,n2)#得到的是10进制数据

hex1=hex(m1)#得到16进制数据，最后转字符串就行了
hex2=hex(m2)#得到16进制数据，最后转字符串就行了
hex=hex1+hex2[2:]
print(hex)#输出16进制数据
flag=base64.b16decode(hex[2:].upper())
print(flag)#输出解码后的字符串
```

解密结果：

```
C:\Users\zww>
C:\Users\zww>
C:\Users\zww>python C:\Users\zww\Desktop\rsa.py
0x666c61677b3333364242353137324144453232374645363842414134344644413733463342 7d
b'flag{336BB5172ADE227FE68BAA44FDA73F3B}'

C:\Users\zww>_
```

得到flag{336BB5172ADE227FE68BAA44FDA73F3B}