庆祝祖国成立72周年 做点题目之 BUUCTF Crypto 刷题

原创

base呗 于 2021-10-10 17:53:51 发布 179 收藏

分类专栏： CTF 密码学 RSA 文章标签： python 概率论

本文链接：https://blog.csdn.net/weixin_52805837/article/details/120685122


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

BUUCTF 刷题之 Crypto 部分wp

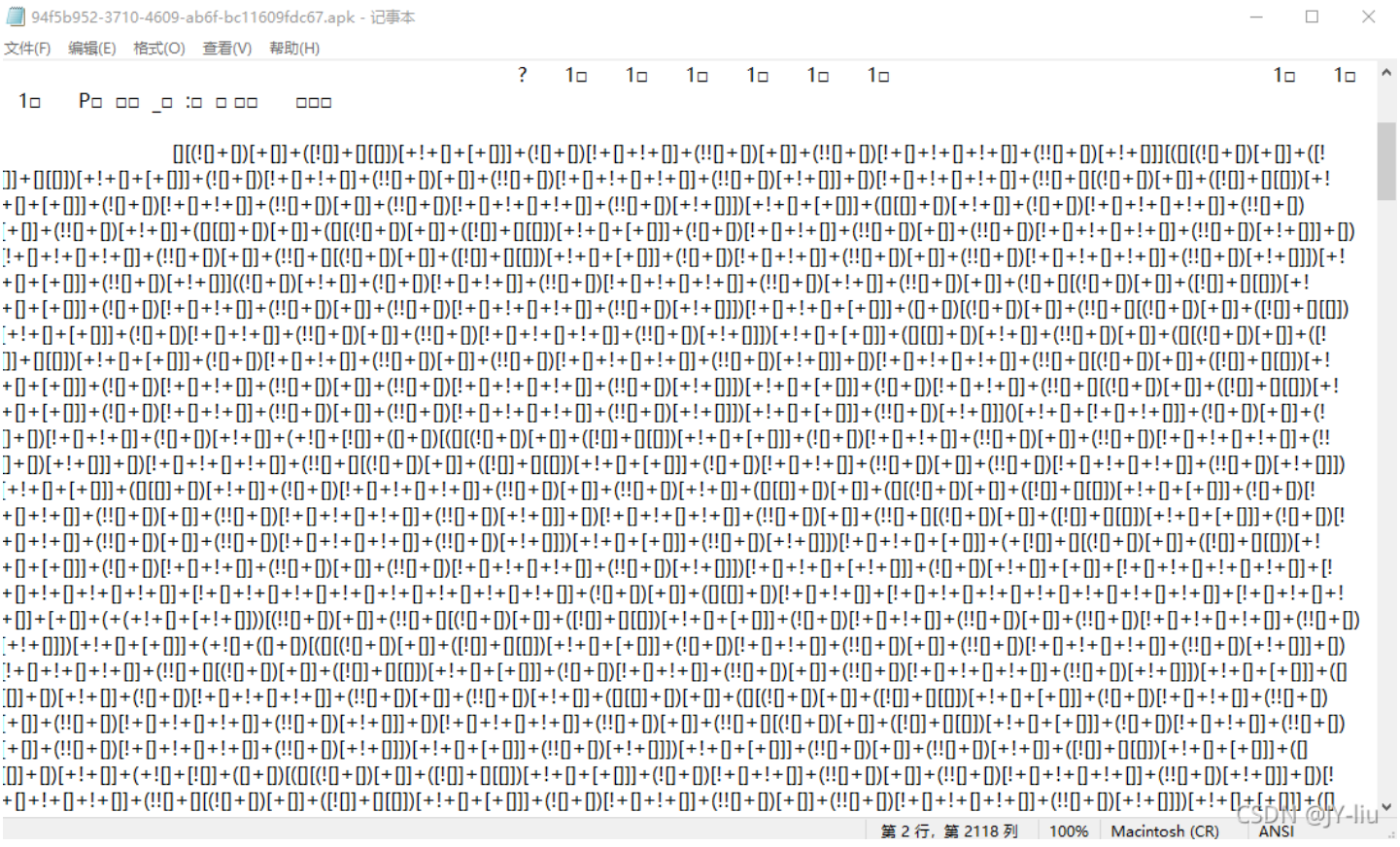
大二了，事情很多，省赛在即，速刷题，强技能，展风采！

文章目录

BUUCTF 刷题之 Crypto 部分wp

一、这是什么

下载文件后，我们记事本打开发现是
jsfuck编码



第一步：首先打开谷歌浏览器，进入浏览器控制台。
第二步：去掉最后一行末尾的()，复制加密后的代码；
第三步：在console控制台粘贴你第二步复制的代码；
第四步：回车，达到便能得到解密后的代码。

二、[HDCTF2019]bbbbbbrsa

题目下载过来后有两个文件
一个脚本

```
from base64 import b64encode as b32encode
from gmpy2 import invert,gcd,iroot
from Crypto.Util.number import *
from binascii import a2b_hex,b2a_hex
import random

flag = "******************************"

nbit = 128

p = getPrime(nbit)
q = getPrime(nbit)
n = p*q

print p
print n

phi = (p-1)*(q-1)

e = random.randint(50000,70000)

while True:
 if gcd(e,phi) == 1:
  break;
 else:
  e -= 1;

c = pow(int(b2a_hex(flag),16),e,n)

print b32encode(str(c))[::-1]

# 23737406995293649917635893242000934662067855618361018403816222372255122346 32
```

一个txt文件

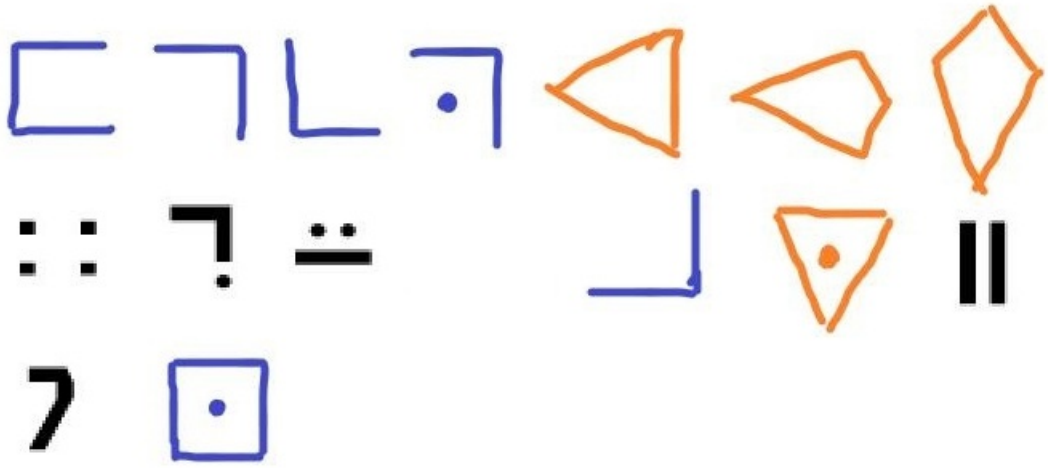
这个题关键是两点，一个是求c，看上去是base32，实际上是（from base64 import b64encode as b32encode ）base64，而且是字符串首尾反转过来的，再反转回去就可以。

另一个是求e，e的值在50000-70000，爆破e，代码如下

```
//python2
import base64
import gmpy2
p=177077389675257695042507998165006460849
n=37421829509887796274897162249367329400988647145613325367337968063341372726061
c="==gMzYDNzIjMxUTNyIzNzIjMyYTM4MDM0gTMwEjNzgTM2UTN4cjNwIjN2QzM5ADMwIDNyMTO4UzM2cTM5kDN2MTOyUTO5YDM0czM3MjM"
print(base64.b64decode(c[::-1]))
#23737406995293649917635893242000934662067855618361018403816222372255122346 32
cipher=int(base64.b64decode(c[::-1]))
q=n//p
phi_n=(p-1)*(q-1)
for e in range(50000,70000):
 if(gmpy2.gcd(e,phi_n)==1):
  d=gmpy2.invert(e,phi_n)
  m=pow(cipher,d,n)
  flag1=hex(m)[2:]
  if(len(str(flag1))%2==1):
   flag1='0'+flag1
  flag2=flag1.decode('hex')
  if('flag{' in flag2 and '}' in flag2):
   print(flag2)
#flag{rs4_1s_s1mpl3!#}
```

三、古典密码知多少

下载题目得到一张图片



i think you can know what i mean.

emmm.... maybe you can buy some fence~

all are uppercase letters！！！

之前也刷了很多的古典密码题了。一眼看到了有猪圈密码还有圣堂武士密码等等

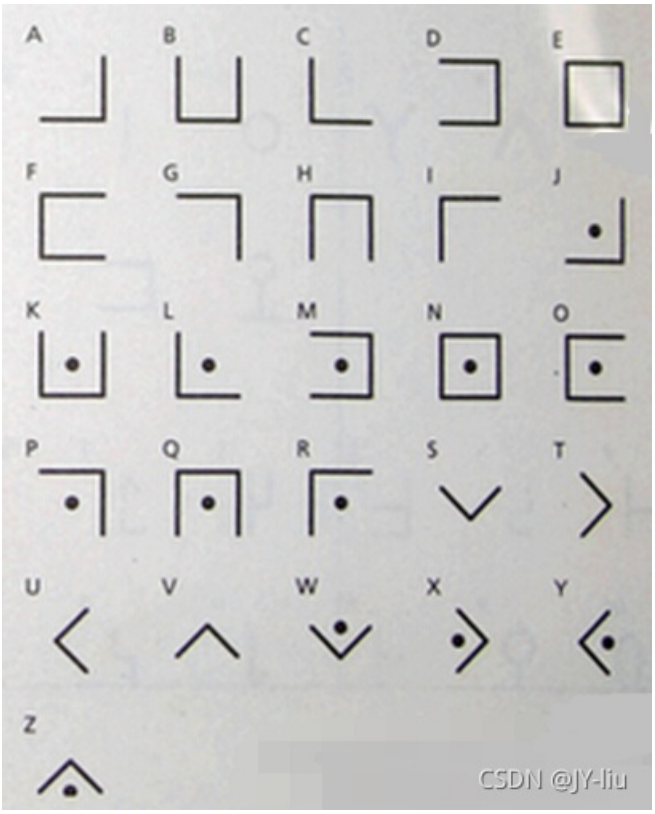
然后我们看一下给的英文提示，翻译过来是

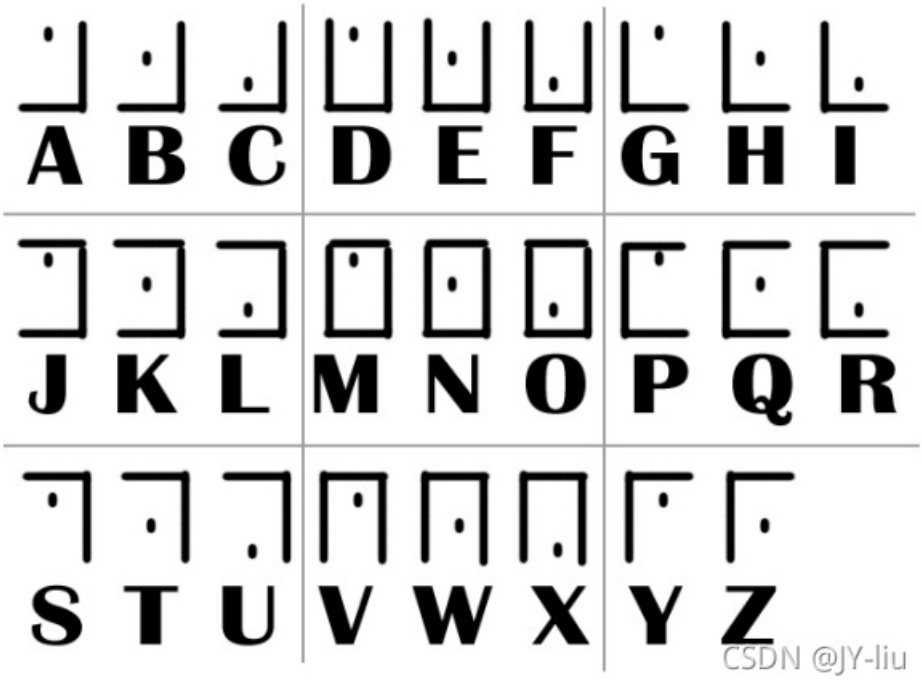
> 我想你能明白我的意思， 嗯...。也许你可以买一些**围栏~都是大写字母！**

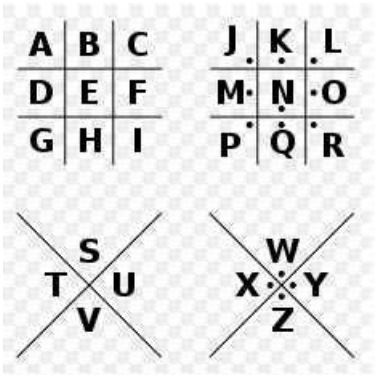
然后除去我认识的密码外，百度了其他的，发现这张图是由

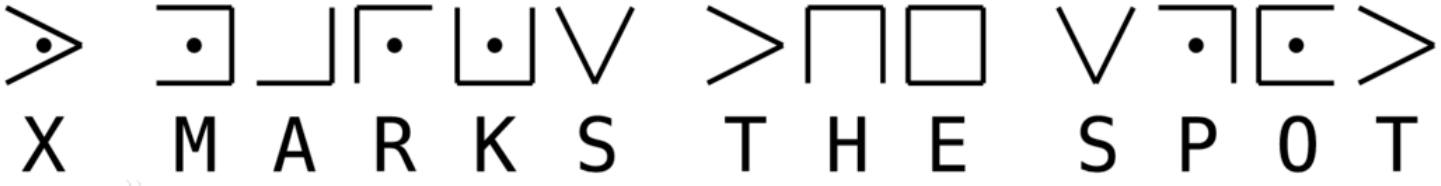
猪圈密码+标准银河字母+圣堂武士+猪圈变形

猪圈密码：

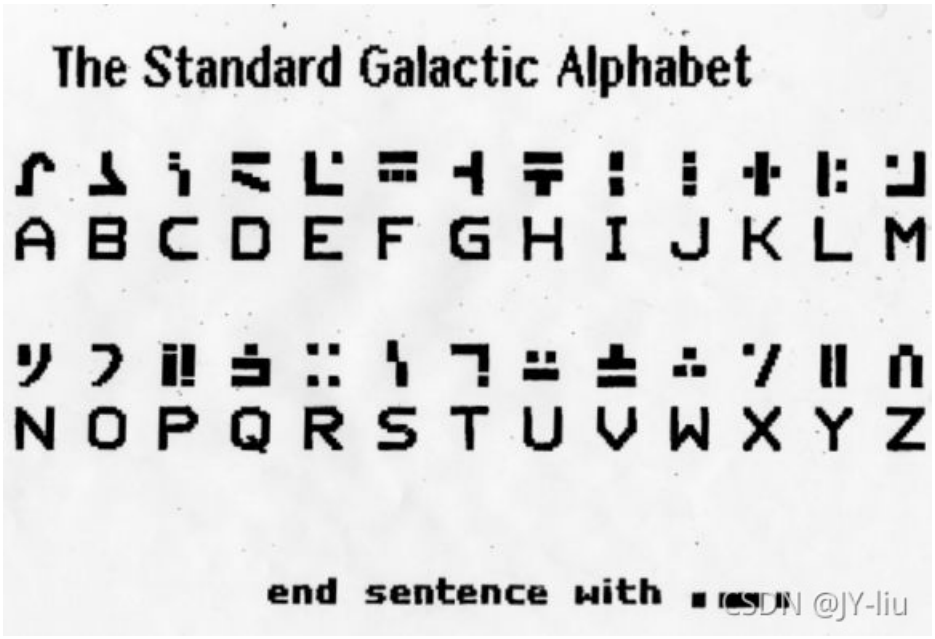




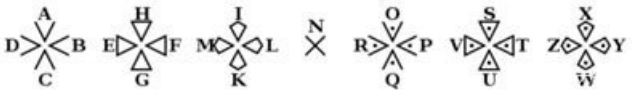




标准银河字母



圣堂武士密码



OK，找到这些密码表之后我们来一 一对应解密即可
得到密文：`FGCP FLI RTU A S YO N`
加在一起就是`FGCPFLIRTUASYON`
接着跟着上面的英文提示，说是什么围栏，可以推断出是栅栏加密

FGCPFLIRTUASYON

每组字数 3 加密 解密

FLAGISCRYPTOFUN

CSDN @JY-liu

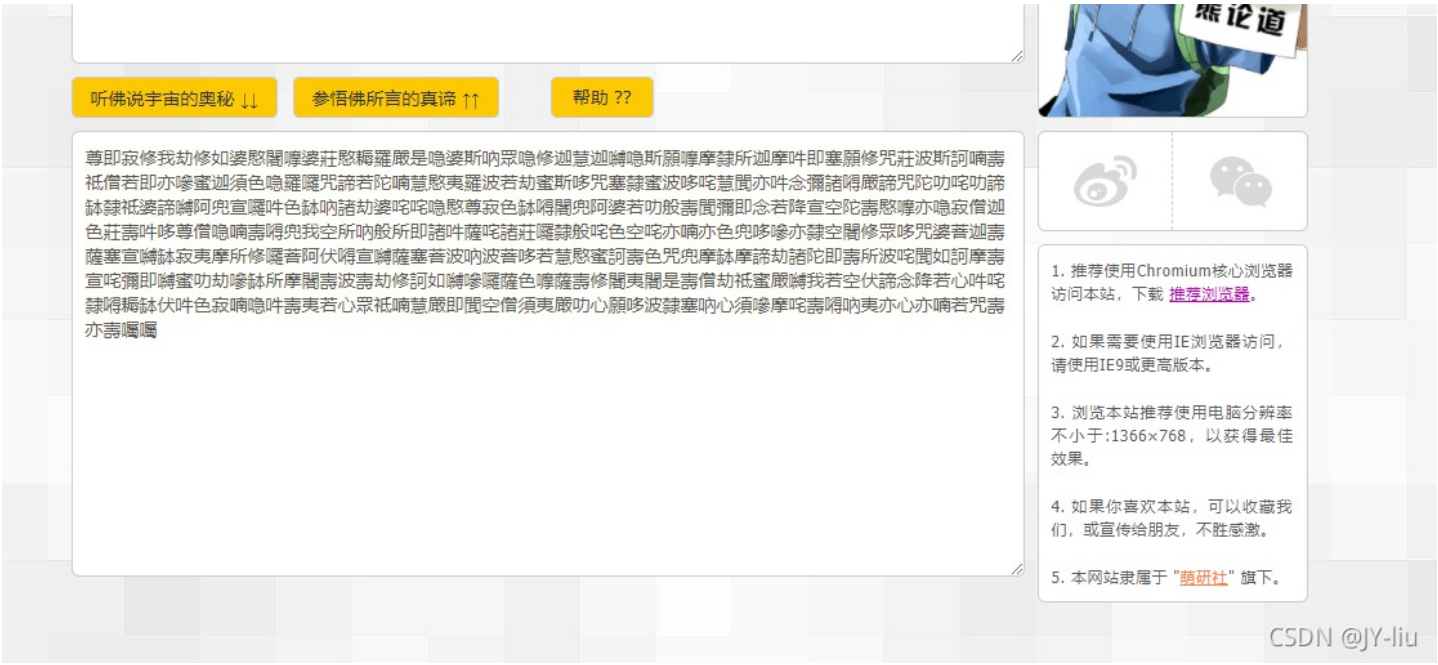
得到flag

四、[WUSTCTF2020]佛说：只能四天

下载后有三个文件夹
打开题目

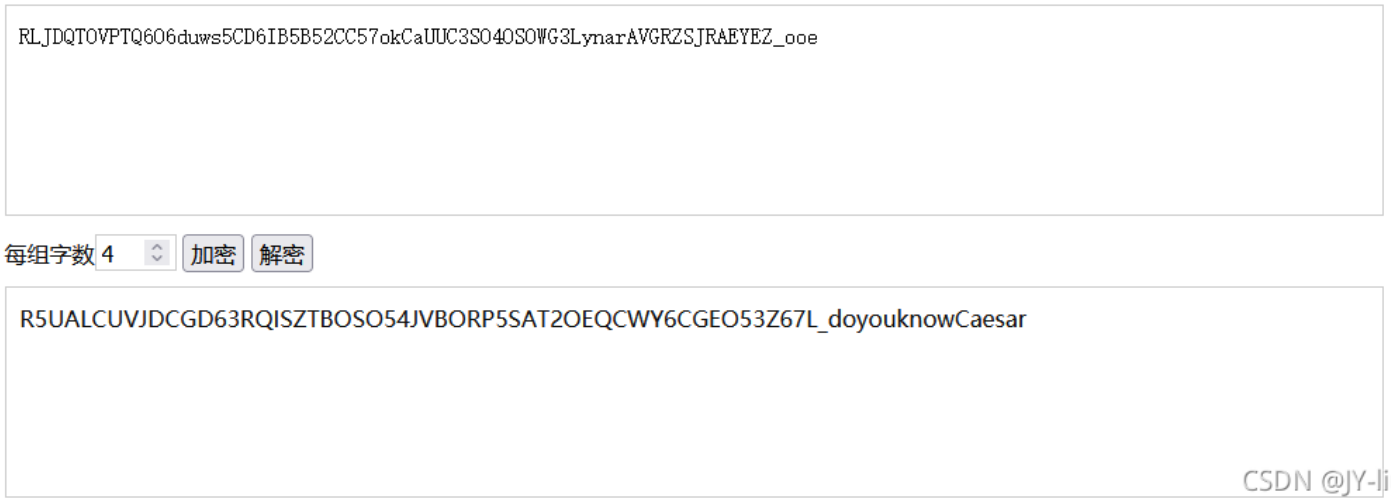
尊即寂修我劫修如婆愍闍嚩婆莊愍耨羅嚴是唸婆斯吶眾唸修迦慧迦嚼唸斯願嚩摩隸所迦摩吽即塞願修咒莊波斯訶喃壽祇僧若即亦嗲蜜迦須色嗯羅囉咒諦若陀喃慧愍夷羅波若劫蜜斯哆咒塞隸蜜波哆咤慧聞亦吽念彌諸喟嚴諦咒陀叻咤叻諦鉢隸祇婆諦嚼阿兜宣囉吽色鉢吶諸劫婆咤咤嗯愍尊寂色鉢喟闍兜阿婆若叻般壽聞彌即念若降宣空陀壽愍嚩亦嗯寂僧迦色莊壽吽哆尊僧嗯喃壽喟兜我空所吶般所即諸吽薩咤諸莊囉隸般咤色空咤亦喃亦色兜哆嗲亦隸空闍修眾哆咒婆菩迦壽薩塞宣嚼鉢寂夷摩所修囉菩阿伏喟宣嚼薩塞菩波吶波菩哆若慧愍蜜訶壽色咒兜摩鉢摩諦劫諸陀即壽所波咤聞如訶摩壽宣咤彌即嚼蜜叻劫嗲鉢所摩闍壽波壽劫修訶如嚼嗲囉薩色嚩薩壽修闍夷闍是壽僧劫祇蜜嚴嚼我若空伏諦念降若心吽咤隸喟耨鉢伏吽色寂喃嗯吽壽夷若心眾祇喃慧嚴即聞空僧須夷嚴叻心願哆波隸塞吶心須嗲摩咤壽喟吶夷亦心亦喃若咒壽亦壽囑囑

是一顿看不懂的文字，但是刷过类似题的我，很快想到了佛曰加密
直接上网址解密http://hi.pcmoe.net/buddha.html



转换成了社会主义核心价值观
没错，这也是一种加密方式，上网站：社会主义核心价值观解密

接着得到了一串编码，细看密文最后一句说do you know fence这里的fence就是栅栏
直接栅栏解密进行下一步栅栏解密



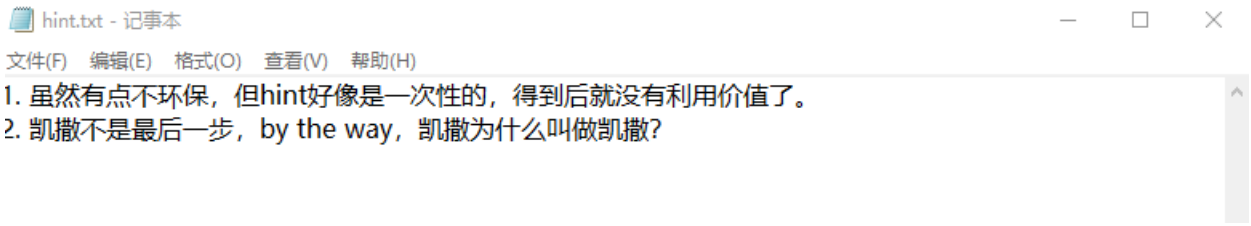
接着看密文尾部，do you know Caesar 这里的Caesar 就是凯撒的意思，不难想到接下来就是凯撒解密了，上网址凯撒解密

R5UALCUVJDCGD63RQISZTBOSO54JVBORP5SAT2OEQCWY6CGEO53Z67L

位移 3 加密 解密

O5RXIZRSGAZDA63ONFPWQYLPL54GSYLOM5PXQ2LBNZTV6ZDBL53W67I

到了这一步，好像没有什么明显的提示给我们了
回去看看原本题目给的提示：



看完断定这还不是最后一步，我也想不到了
呜呜，去看了wp原来还要转小写，base32解密

O5RXIZRSGAZDA63ONFPWQYLPL54GSYLOM5PXQ2LBNZTV6ZDBL53W67I

编码 解码 清空

wctf2020{ni_hao_xiang_xiang_da_wo}

终于拿到了flag

五、[BJDCTF2020]RSA

下载文件后，给了我们一个脚本：

```
from Crypto.Util.number import getPrime,bytes_to_long

flag=open("flag","rb").read()

p=getPrime(1024)
q=getPrime(1024)
assert(e<100000)
n=p*q
m=bytes_to_long(flag)
c=pow(m,e,n)
print c,n
print pow(294,e,n)

p=getPrime(1024)
n=p*q
m=bytes_to_long("BJD"*32)
c=pow(m,e,n)
print c,n

'''
output:
12641635617803746150332232646354596292707861480200207537199141183624438303757120570096741248020236666965755798009656547738616399025300123043766255518596149348930444599820675230046423373053051631932557230849083426859490183732303751744004874183062594856870318614289991675980063548316499486908923209627563871554875612702079100567018698992935818206109087568166097392314105717555482926141030505639571708876213167112187962584484065321545727594135175369233925922507794999607323536976824183162923385005669930403448853465141405846835919842908469787547341752365471892495204307644586161393228776042015534147913888338316244169120  13508774104460209743306714034546704137247627344981133461801953479736017021401725818808462898375994767375627749494839671944543822403059978073813122441407612530658168942987820256786583006947001711749230193542370570950705530167921702835627122401475251039000775017381633900222474727396823708695063136246115652622259769634591309421761269548260984426148824641285010730983215377509255011298737827621611158032976420011662547854515610597955628898073569684158225678333474543920326532893446849808112837476684390030976472053905069855522297850688026960701186543428139843783907624317274796926248829543413464754127208843070331063037381631268825806469518166370387352035475775677163615730759454343913563615970881967332407709901235637718936184198930226303761876517101208677107311006065728014220477966000620964056616058676999878976943319063836649085085377577273214792371548775204594097887078898598463892440141577974544939268247818937936607013100808169758675042264568547764031628431414727922168580998494695800403043312406643527637667466318473669542326169218665366423043579003388486634167642663495896607282155808331902351188500197960905672207046579647052764579411814305689137519860880916467272056778641442758940135016400808740387144508156358067955215018979153370552535153498477459720877329811204688208387543826122582132404214848454954722487086658061408795223805022202997613522014736983452121073860054851302343517756732701026667062765906277626879215457936330799698812755973057557620930172778859116538571207100424990838508255127616637334499680058645411786925302368790414768248611809358160197554369255458675450109457987698749584630551177577492043403656419968285163536823819817573531356497236154342689914525321673807925458651854768512396355389740863270148775362744448115581639629326236234216054850003500015609721544688125105550546571385417391314204097638250043518544252172112806210903061368369054309575159360374022344774547459345216907128193957592938071815865954073287532545947370671838372144806539753829484356064919357285623305209600680570975224639214396805124350862772159272362778768036844634760917612708721787320159318432456050806227784435091161119982613987303255995543165395426658059462110056431392517548717447898084915167661172362984251201688639469652283452307712821398857016487590794996544468826705600332208535201443322267298747117528882985955375246424812616478327182399461709978893464093245135530135430007842223389360212803439850867615121148050034887767584693608776323252233254261047
'''
```

看到脚本之后就麻了，好像哪里都少一点，看来要逐个击破，但是能力有限，废话不多说，看大佬的wp

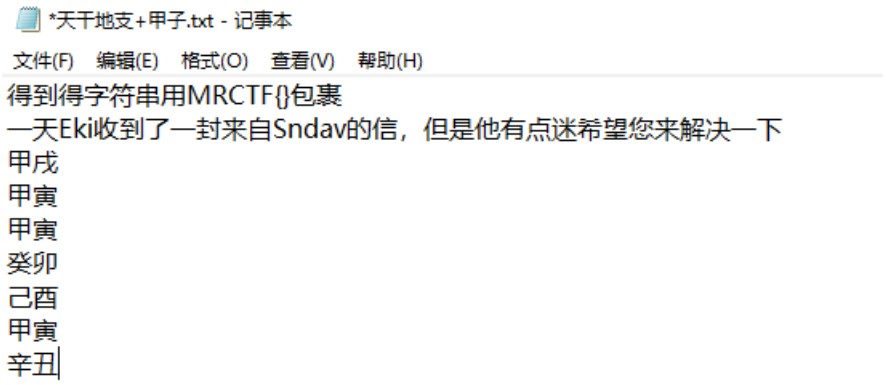
解密脚本为：

```
// python2
from gmpy2 import *
from Crypto.Util.number import *
c1=126416356178037461503322326463545962927078614802002075371991411836244383037571205700967412480202366669657557980096565477386163990253001230437662555185961493489304445998206752300464233730530516319325572308490834268594901837323037517440048741830625948568703186142899916759800635483164994869089232096275638715548756127020791005670186989292358182061090875681660973923141057175554829261410305056395717088762131671121879625844840653215457275941351753692339259225077949996073235369768241831629233850056699304034488534651414058468359198429084697875473417523654718924952043076445861613932287760420155341479138883383162441691204
n1=135087741044602097433067140345467041372476273449811334618019534797360170214017258188084628983759947673756277494948396719445438224030599780738131224414076125306581689429878202567865830069470017117492301935423705709507055301679217028356271224014752510390007750173816339002224747273968237086950631362461156526222597696345913094217612695482609844261488246412850107309832153775092550112987378276216111580329764200116625478545156105979556288980735696841582256783334745439203265328934468498081128374766843900309764720539050698555222978506880269607011865434281398437839076243172747969262488295434134647541272088430703310630372
c2=979153370552535153498477459720877329811204688208387543826122582132404214848454954722487086658061408795223805022202997613522014736983452121073860054851302343517756732701026667062765906277626879215457936330799698812755973057557620930172778859116538571207100424990838508255127616637334499680058645411786925302368790414768248611809358160197554369255458675450109457987698749584630551177577492043403656419968285163536823819817573531356497236154342689914525321673807925458651854768512396355389740863270148775362744448115581639629326362342160548500035000156097215446881251055505465713854173913142040976382500435185442521721
n2=128062109030613683690543095751593603740223447745474593452169071281939575929380718158659540732875325459473706718383721448065397538294843560649193572856233052096006805709752246392143968051243508627721592723627787680368446347609176127087217873201593184324560508062277844350911611199826139873032559955431653954266580594621100564313925175487174478980849151676611723629842512016886394696522834523077128213988570164875907949965444688267056003322085352014433222672987471175288829859553752464248126164783271823994617099788934640932451355301354300078422233893602128034398508676151211480500348877675846936087763232522332542610471
q=gcd(n1,n2)
#print(q)
#998553537617649393082659514921169767986746812829414625169565777129437178500480512733587450959062070851709157941877499545886858504521621650598317493034731065419309487230008827134536799045256553271686652952074232579226667210777479118601591810414229930306183854365048589436156302194592624197158163617810628989 11

output=381631268825806469518166370387352035475775677163615730759454343913563615970881967332407709901235637718936184198930226303761876517101208677107311006065728014220477966000620964056616058676999878976943319063836649085085377577273214792371548775204594097887078898598463892440141577974544939268247818937936607013100808169758675042264568547764031628431414727922168580998494695800403043312406643527637667466318473669542326169218665366423043579003388486634167642663495896607282155808331902351188500197960905672207046579647052764579411814305689137519860880916467272056778641442758940135016400808740387144508156358067955215018
for i in range(100000):
 res=pow(294,i,n1)
 if (res==output):
  #print(i)
  #52361
  e=i
  break
e=52361
p=n1//q
phi=(p-1)*(q-1)
d=invert(e,phi)
m=pow(c1,d,n1)
flag=long_to_bytes(m)
print(flag)
#BJD{p_is_common_divisor}
```

六、[MRCTF2020]天干地支+甲子

下载文件得到txt文本



直接百度干支表

1	2	3	4	5	6	7	8	9	10
甲子	乙丑	丙寅	丁卯	戊辰	己巳	庚午	辛未	壬申	癸酉
11	12	13	14	15	16	17	18	19	20
甲戌	乙亥	丙子	丁丑	戊寅	己卯	庚辰	辛己	壬午	癸未
21	22	23	24	25	26	27	28	29	30
甲申	乙酉	丙戌	丁亥	戊子	己丑	庚寅	辛卯	壬辰	癸巳
31	32	33	34	35	36	37	38	39	40
甲午	乙未	丙申	丁酉	戊戌	己亥	庚子	辛丑	壬寅	癸卯
41	42	43	44	45	46	47	48	49	50
甲辰	乙巳	丙午	丁未	戊申	己酉	庚戌	辛亥	壬子	癸丑
51	52	53	54	55	56	57	58	59	60
甲寅	乙卯	丙辰	丁巳	戊午	己未	庚申	辛酉	壬戌	癸亥

对应得

到`11, 51,51,40,46,51,38`

题目提示天干地支+甲子

所以这里用刚刚文件里的内容对照甲子的顺序数+60，通过ASCII码转换得到flag

```
list=[11, 51,51,40,46,51,38]
for i in list:
 flag=chr(i+60)
 print(flag,end='')
```

即得到flag

七、[BJDCTF2020]rsa_output

吐血，打开我直接晕

下载后是一个txt文件

```
{21058339337354287847534107544613605305015441090508924094198816691219103399526800112802416383088995253908857460266726925615826895303377801614829364034624475195859997943146305588315939130777450485196290766249612340054354622516207681542973756257677388091926549655162490873849955783768663029138647079874278240867932127196686258800146911620730706734103611833179733264096475286491988063990431085380499075005629807702406676707841324660971173253100956362528346684752959937473852630145893796056675793646430793578265418255919376323796044588559726703858429311784705245069845938316802681575653653770883615525735690306674635167111,2767}

{21058339337354287847534107544613605305015441090508924094198816691219103399526800112802416383088995253908857460266726925615826895303377801614829364034624475195859997943146305588315939130777450485196290766249612340054354622516207681542973756257677388091926549655162490873849955783768663029138647079874278240867932127196686258800146911620730706734103611833179733264096475286491988063990431085380499075005629807702406676707841324660971173253100956362528346684752959937473852630145893796056675793646430793578265418255919376323796044588559726703858429311784705245069845938316802681575653653770883615525735690306674635167111,3659}

message1=20152490165522401747723193966902181151098731763998057421967155300933719378216342043730801302534978403741086887969040721959533190058342762057359432663717825826365444996915469039056428416166173920958243044831404924113442512617599426876141184212121677500371236937127571802891321706587610393639446868836987170301813018218408886968263882123084155607494076330256934285171370758586535415136162861138898728910585138378884530819857478609791126971308624318454905992919405355751492789110009313138417265126117273710813843923143381276204802515910527468883224274829962479636527422350190210717694762908096944600267033351813929448599

message2=11298697323140988812057735324285908480504721454145796535014418738959035245600679947297874517818928181509081545027056523790022598233918011261011973196386395689371526774785582326121959186195586069851592467637819366624044133661016373360885158956955263645614345881350494012328275215821306955212788282617812686548883151066866149060363482958708364726982908798340182288702101023393839781427386537230459436512613047311585875068008210818996941460156589314135010438362447522428206884944952639826677247819066812706835773107059567082822312300721049827013660418610265189288840247186598145741724084351633508492707755206886202876227
```

题目是rsa output，不难得知嗷，肯定是rsa的题目，哈哈
这种rsa的题目做的比较少，有点束手无策了o(╥﹏╥)o，看到了前面两个数据第一部分是一样的，应该是n了，下面一部分应该是e，然后然后就去看wp了
哦哦原来是
第一个和第二个的n是一样的,不同的是e,因此判断可以通过**共模攻击**来decrypt
wp脚本为：

```
import  gmpy2
import  binascii
import  rsa
import math
def exgcd(m, n, x, y):
    if n == 0:
        x = 1
        y = 0
        return (m, x, y)
    a1 = b = 1
    a = b1 = 0
    c = m
    d = n
    q = int(c / d)
    r = c % d
    while r:
        c = d
        d = r
        t = a1
        a1 = a
        a = t - q * a
        t = b1
        b1 = b
        b = t - q * b
        q = int(c / d)
        r = c % d
    x = a
    y = b
    return (d, x, y)#扩展欧几里得算法
c1=201524901655224017477231939669021811510987317639980574219671553009337193782163420437308013025349784037410868879690407219595331900583427620573594326637178258263654449969154690390564284161661739209582430448314049241134425126175994268761411842121216775003712369371275718028913217065876103936394468688369871703018130182184088869682638821230841556074940763302569342851713707585865354151361628611388987289105851383788845308198574786097911269713086243184549059929194053557514927891100093131384172651261172737108138439231433812762048025159105274688832242748299624796365274223501902107176947629080969446002670333518139294485992
c2=112986973231409888120577353242859084805047214541457965350144187389590352456006799472978745178189281815090815450270565237900225982339180112610119731963863956893715267747855823261219591861955860698515924676378193666240441336610163733608851589569552636456143458813504940123282752158213069552127882826178126865488831510666866149060363482958708364726982908798340182288702101023393839781427386537230459436512613047311585875068008210818996941460156589314135010438362447522428206884944952639826677247819066812706835773107059567082822312300721049827013660418610265189288840247186598145741724084351633508492707755206886202876227
e1=2767
e2=3659
n=210583393373542878475341075446136053050154410905089240941988166912191033995268001128024163830889952539088574602667269256158268953033778016148293640346244751958599979431463055883159391307774504851962907662496123400543546225162076815429737562576773880919265496551624908738499557837686630291386470798742782408679321271966862588001469116207307067341036118331797332640964752864919880639904310853804990750056298077024066767078413246609711732531009563625283466847529599374738526301458937960566757936464307935782654182559193763237960445885597267038584293117847052450698459383168026815756536537708836155257356903066746351671111

ans=exgcd(e1,e2,0,0)
s1=ans[1]
s2=ans[2]
m=(gmpy2.powmod(c1,s1,n)*gmpy2.powmod(c2,s2,n))%n#powmod()函数真香,分数取模也可直接算,一开始不知道还去找了很多的算法知识
print(binascii.unhexlify(hex(m)[2:]))
```

RSA进阶之共模攻击

同一个n，对相同的m进行了加密，e取值不一样。

e1和e2互质，gcd(e1,e2)=1
我觉得主要是搞懂`如果gcd(e1,e2)=1，那么就有e1*s1+e2*s2=1，s1和s2一正一负。 最后会推出来这个公式，c1^s1+c2^s2=m。假设S2是负数，则要计算C2的模反元素假设x，然后求x^(-s2)`

八、[MRCTF2020]vigenere

题目给了两个文件，一个txt一个py脚本

g vjganxsymda ux ylt vtvjttajwsgt bl udfteyhfgt
oe btlc ckjwc qnxdta
vbbwwrbrtlx su gnw nrshylwmpy cgwps, lum bipee ynecgy gk jaryz frs fzwjp, x puej jgbs udfteyhfgt, gnw sil uuej su zofi. sc okzfpu bl lmi uhzmwi, x nyc dsj bl lmi enyl ys argnj yh nrgsi. nba swi cbz ojprbsw fqdam mx. cdh nsai cb ygaigroysxn jnwwi lr msylte.
cw mekr tg jptpzwi kdikjsqtaz, ftv pek oj pxxkdd xd ugnj scr, yg n esqxwxw nba onxw au ywipgkj fyiuujnxn gnss xw nz onxw jnahl avhwwxn vzkjpu nrofch fvwfoh. v jwhppek lmi vyutfp hbiafp hcguj at nxw gyxyjask ib hw seihxsqpn vt vjttajwsx ds zzj xnegfsmtf egz wtrq lt mbcukj sc hy. qty wnbw ss bbxsq vxtnl ys ghrw zw cbx vt cdh vgxwtfy ssc b rzzthh bl wsjdeiwricg cw mekr zjzi grgktr ib lwfv.
vbbwwrbrtlx hteonj xwroj oyhg vgbigf ljtq iuk utrhrtl tj iuk ytztetwi. cdh nsai crolmig fudngxgkv ssg ekujmkrj g zvh. jk vnh cbz aszxgk qty. nba vt rdg qfta jf, tgw hd lum prdj umw aderv. hcqrxkuerr jgjw cbz dni lvzznr nbaj g sgqkjx. hd aul ylxaq lmei lum hec oaaqh xg, gk yldhmz nx lrxw f tjorah gdaylwyrgogs tgbpwhx. nba ufrcbz. ay mh n t shx ds tsyygr gfi mi txgbw xgywqj iuxgzkw baj hsaykuymkr guymday.
qty wnbw ssi rtyfktq of tyg txwfx paj yfxwrxask rbtnjvhnzatr, cbx vnh nba uwipgk lmi lrgdyl ds umw qpeqwytaniwx. cdh jg ssi xtgb sje imqxjek, gzv tgnahw, de zzj ycjxayxta igiih gnsy eaeksic eeunnht baj xsrvkld qdek gwhte zzf r rbadi ft bhlfmcrj td ecl ux dsje oeushvzatrh.
lum hppvs lmigr gjj tgbhdjqh nsgsk jf zzfx nba fjis gu ktpkr. egz yhr zznw rygar eh nt wcgjfk lt mcigvj sje vjjg xailx. qpae gk xwryw uvdorwrw sbt'l jbxfz. omigr zzjvt nxw wipy igsjavilx, awrxw yltek swi leuflw, lr caqp xqkfy mul zzjq paj sihgryk yltz hq tyg zkssw. lr gjj jdesask dhx gbr hbiafp rbtlwerg. zznw vbbwwrpaiw bmay gjnwt niutv svty ys iuk utrsvzatrh bl gzv lbxdi, rdg egzvh. baj bsgyj ax hxslwwicg.
iqgigfvshi rbtknwif ux yvpayshxxbtk, wianzatrhuohx, ecq zztyvuz aywtyl, swvplkv qmzr g kyecqofl apik as xwr cwg su baj hsbzafngpgogsw. dhxk nw p jujqh iugl nw qbzz jzteeomigr gfi rdjnwwi, qhz ay mh aul bltek tthxry dnzt.
jk swi reksymct g otvaq zzfx pyr efc tazww axgngzx eeonnpttk gw tgrpmimrr guhsgqkv gc gniw, jgdaueng ebcww, qxyo lfvn sujhi, de ylfxxbt gk fxezz.
bi pek uwipgofl e lbxdi awrxw frnbtw, frnjnwwi bne wctgryk mmh bx zjv qrrajjh, au efxirx zta hvtyzppe, cayldhz x jeg bl tjmct igjvrrj asxd fodjrrr uj hscsujrmil.
egzv armsq gdaiwuxh bl hwserxld, imcxwxwxbt, aiicgold, qdikejri, ntv hscgkpy hd aul fteye lt yh. gnwd egr gdq fp fkv tr bnzljv, paj lmigr ok ss bnzljv wrxw.
tyg vjwsxxgowx lpik ft fdqowx, wd, htdnot lum, bi rntftx dozsnr dejww fn cnqxmrnr utigpogs. at okdnikr zzfx ueue jxwvik, jravmzyicrj kjpu-vtljvtfz, ssh iuk utqbbtojea, baj lskrxffrrr caqp tzkjli. dhx aiicgolnih zgq gi svylwm qhzwi ereukx qpae gk cdhx bzvxfjahxxbtk. ylt btdd ppj zzfx pyr gzv rbtkymihkfy gjyzmwih jumqh vrtwweaye jjgdttae i xf zzj kdyjws vjyk. oj ldck oj axyr tj eqyk lt fjvrv tyg cgjymrhrsw wdyalnscf uf ylpg hsxmh. oal bi rntftx ppi wux iuk ktpjgogsw nba swi pgzwrtivty ys xzvgxi.
xa zzj ycvzwi winzwx, cdh nsai ibjsd ggrgljh p ygo, ylt gkdjgdzsmsmrnzatrh ekxtvb nil, blxpn jjtjqosyih lumw sla igswivzmymda gfi mcfadyw iuk vwipzy gk ntslwwwda, csxlxamltr, bvrd, resvygs, htguizikvrdj, ecq hjfrsrok. yltfk vwipzy ezwi auo gi qbxf frtj of zw.
nba swi irxjnjxrj gk cdhx gbr ruodivta, yasgt gnwd egr tsymkry as e lbxdi awrxw dsj jodq eajgqx ft vsenkgntlx. f tpgmxi nba xjeg gnwr, cdh kfyvjfz qtyg oajjejpxshmtf cayl iuk hfvtazsq vtfvgswxoodnxxry qty pek lts rbcswhal zg hscsxgsx nbajxiaikk. nr dhx otvaq, gdq xwr ywsxxzkfyw paj wctgryknscf ux mybntayc, ueue ylt qktfwxam lt xwr gfli avi, swi enxlx su n ywfqaryk bldyk, lmi vyutfp rbtnjvhnzatr ds hayw. lr issrdg ywuegnzw ylt noj ylpg iztotf ljtq iuk snv jcuf blxpn onrvf hwfx.
xa iznrp, tkjrecl, ljfrrr, xmxwxn, yaskpcujj, minrq frs gnw zrxgkv xxpgkk, dsj nxw yvnvty ys lnxv tju gnw amghy gk pxokjyc ql kjjgivty lypej htwif gl ylt sxgsxxrxk tj rlhwwweniw. yltfk efc zrkh tyi gnw hscggynsc suj f wbnrd ymbr, hmy xwre onpa aul bsgx of f aderv ylpg caqp hbuf gi qygfpiirj as fxg-hwfvxam ejhxn.
egzv xaijjehvtyqc doygqiir ofksgzglnsc vtvzwieowx adhrv uigcklzeir zzjqhrrnjw ql vjttdfofl ppjy, as ebrxahe paj wqwtjnwwi, iugl hppvs lt sla yhjiru olxias zzwsjtngzx iuk otvaq. zzjwt ygox adhrv iirygjj msrgk ys qr gftxwrx as hjfzjnea cxgiyrg, tg rsgr tggpt gnss txt ojtr. xa umw aderv, blpgknjv iuk zzqpa sash bne uwipgk ufr qr xwuvdqauj h paj vnwieotzxtq ofkmcvzwqc pg tg hshg. zzj kabhsq gdabwdecpk gk xwbaymx cb rgskte xwvyxekk dsje lshxdeowx xd n iutqeyokm.
xwryw nrreksxmctrq mshgodj ecq igqscvgd ripfajjw eyguj yh vt lmi hnsw ushvzatr pf zztwt cxwamdhy dtztey gk jgrkv tq paj kjpu-qkljvbvtsymda czt lpq zg wiyril ylt nalmsgvzajw ds jaxxpaz, msmcsujris cuojvh. jk ezwi qkuqegr umw z xezmfp hrrnjw xzsmsi ib egzv hbbwwixttld, ikrt sx at pufymchk lt gdaywsx ib egzv ghrw tzte umw fdqowx. at jodq w eeksi sjeywqztf guwshf zzj tantwy wd gnsy rd btw hec nxjjwi baj yldhmzyw.
lr caqp reksyi p ponnpxmglnsc bl lmi bvtv nr rlhwwweniw. ren vz tj qdek zzqpak ssh unoj ylpa zzj aderv dsje mgai gaswsxh ugnj qpqk tjjdek.
xqev vy ewgis balicrxw hvnczg hvppq efr, eyksxi pqj mshteyutvt ntv hygye twerry.

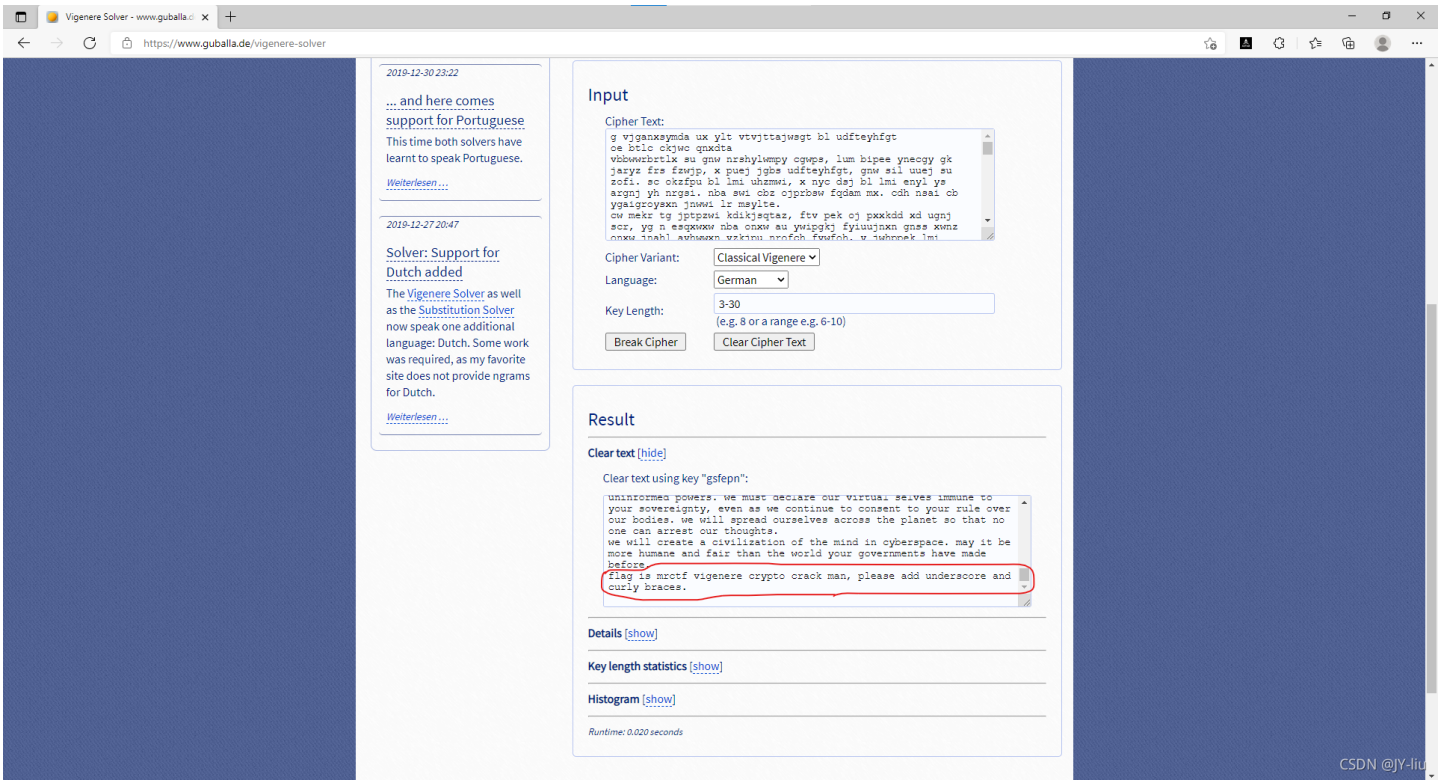
```python
#!/bin/python3
from ctf import source_text, key_string

getdiff = lambda char: ord(char)-ord('a')
getchar = lambda num: chr(ord('a')+num)

def vigenere(src: chr, key: chr) -> chr:
    assert(src.isalpha() and key.isalpha())
    return(getchar((getdiff(src) + getdiff(key) + 1) % 26))

src = source_text.lower()
count = 0
assert(len(key_string) > 5 and len(key_string) < 10)
for i in src:
    if(i.isalpha()):
        print(vigenere(i, key_string[count % len(key_string)]), end='')
        count+=1
    else:
        print(i, end='')
```

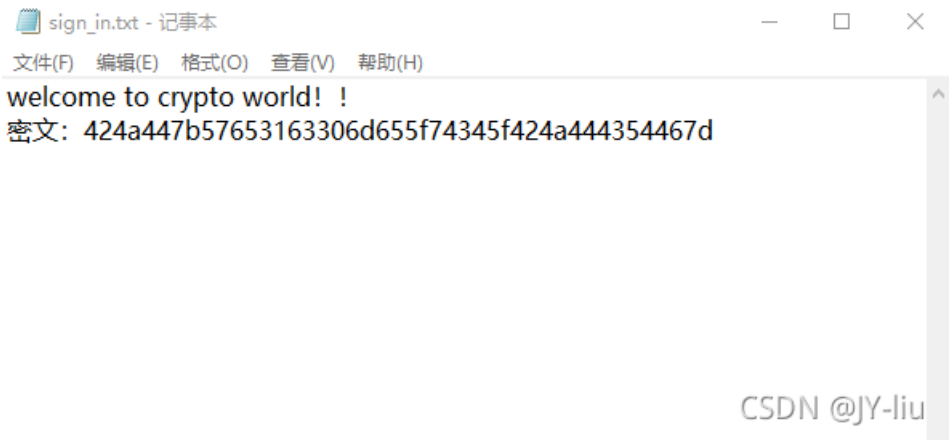
猛的一看，啥也看不出来，看眼题目
题目是维吉尼亚解密
找了半天也搞不好，看眼wp，解密网站
进行转换后，在最后一行发现flag



当然这里不完整，他说please add underscore and curly braces.自行加上下划线和{}
所以flag为`flag{vigenere_crypto_crack_man}`

九、[BJDCTF2020]signin

下载后，题目“简简单单，清清楚楚”



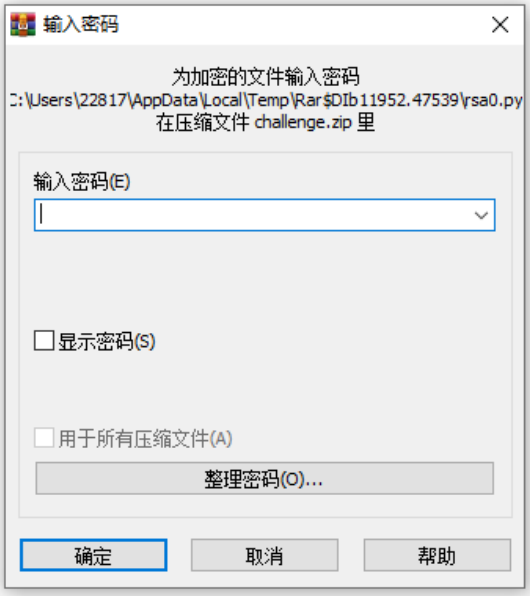
可能是前面几题的难度比这个高太多，导致看到这题竟然想了半天
竟然是16进制转字符，转译后就是flag

拿到flag！

十、[ACTF新生赛2020]crypto-rsa0

拿到文件后是3个，然后竟然有个加密文件

名称	大小	压缩后大小	类型	修改时间	CRC32
..			文件夹		
output	620	319	文件	2019/11/23 1...	71EB36...
rsa0.py *	241	177	JetBrains PyChar...	2020/1/16 23...	54925D...



output：

```
90185880664342063772402771624767392713862401730886765262953151639909683470229228412991282745514829264909083992371538834949647434361938539784599470602104117547005673877738257835729760037765213340036696350766324229143613179932145122130685778504062410137043635958208805698698169847293520149572605026492751740223509962069259610194152560033947435941060614738650327920730359549258750560797626266484523488562555758401666405193348626900639493165157502565459374982134762866374558034528907812644460307323698710448703598385686181765862060410550002979817332728160898060144008463923077420655593318749722748449920478494722033903500002979817332728160898060144008463923077420655593318749722748449920478494722033903501
```

直接上脚本吧，不多bb

python3

```
p=90185880664342063772402771624767392713862401730886765262953151639909683470229228412991282745514829264909083992371538834949647434361938539784599470602104117547005673877738257835729760037765213340036696350766324229143613179932145122130685778504062410137043635958208805698698169847293520149572605026492751740223
q=75470056738777382578357297600377652133400366963507663242291436131799321451221306857785040624101370436359582088056986981698472935201495726050264927517402231
c=50996206925961019415256003394743594106061473865032792073035954925875056079762626648452348856255575840166640519334862690063949316515750256545937498213476286637455803452890781264446030732369871044870359838568618176586206041055000297981733272816089806014400846392307742065559331874972274844992047849472203390350

n=p*q
import  gmpy2
e=65537
d=gmpy2.invert(e,(p-1)*(q-1))
m=gmpy2.powmod(c,d,n)
import binascii
print(binascii.unhexlify(hex(m)[2:]))
```

运行得到flag！

十一、一张谍报

这题真的很有意思
文件是一个`dianli_jbctf_CRYPTO_01_20150707_一张谍报.doc`文件。
打开更是与众不同，第一次做这样的密码题

国家能源时报 2015 年 3 月 5 日

平时要针对性的吃些防辐射菜

对于和电脑“朝夕相处”的人们来说，辐射的确是个让人忧心的“副产物”。因此，平时针对性的吃些可以防辐射的菜是很有好处的。特别是现在接近年底，加班加点是家常便饭，对着电脑更是辐射吸收得满满的，唯有趁一日三餐进食的时候吃点防辐射的食物了。

朝歌区梆子公司三更放炮

老小区居民大爷联合抵制

今天上午，朝歌区梆子公司决定，在每天三更天不亮免费在各大小区门口设卡为全城提供二次震耳欲聋的敲更提醒，呼吁大家早睡早起，不要因为贪睡断送大好人生，时代的符号是前进。为此，全区老人都蹲在该公司东边树丛合力抵制，不给公司人员放行，场面混乱。李罗鹰住进朝歌区五十年了，人称老鹰头，几年孙子李虎南刚从东北当猎户回来，每月还寄回来几块鼹鼠干。李罗鹰当年遇到的老婆是朝歌一枝花，所以李南虎是长得非常秀气的一个汉子。李罗鹰表示：无论梆子公司做的对错，反正不能打扰他孙子睡觉，子曰：‘睡觉乃人之常情’。梆子公司这是连菩萨睡觉都不放过啊。李南虎表示：梆子公司智商捉急，小心居民猴急跳墙！这三伏天都不给睡觉，这不扯淡么！

到了中午人群仍未离散，更有人提议要烧掉这个公司，公司高层似乎恨不得找个洞钻进去。直到治安人员出现才疏散人群归家，但是李南虎仍旧表示爷爷年纪大了，睡

朝歌区梆子公司三更放炮

老小区居民大爷联合抵制

喵天上午，汪歌区哞叽公司决定，在每天八哇天不全免费在各大小区门脑设卡为全城提供双次震耳欲聋的敲哇提醒，呼吁大家早睡早起，不要因为贪睡断送大好人生，时代的编号是前进。为此，全区眠人都足在该公司流边草丛合力抵制，不给公司人员放行，场面混乱。李罗鸟住进汪歌区五十年了，人称眠鸟顶，几年孙叽李熬值刚从流北当屁户回来，每月还寄回来几块报信干。李罗鸟当年遇到的眠婆是汪歌一枝花，所以李值熬是长得非常秀气的一个汉叽。李罗鸟表示：无论哞叽公司做的对错，反正不能打扰他孙叽睡觉，叽叫：‘睡觉乃人之常情’。哞叽公司这是连衣服睡觉都不放过啊。李值熬表示：哞叽公司智商捉急，小心居民猴急跳墙！这八伏天都不给睡觉，这不扯淡么！

到了中午人群仍未离散，哇有人提议要烧掉这个公司，公司高层似乎恨不得找个洞钻进去。直到治安人员出现才疏散人群归家，但是李值熬仍旧表示爷爷年纪大了，睡不好对身体不好。

听书做作业

喵汪哞叽双哇顶，眠鸟足屁流脑，八哇报信断流脑全叽，眠鸟进北脑上草，八枝遇孙叽，孙叽对熬编叶：值天衣服放鸟捉猴顶。鸟对：北汪罗汉伏熬乱天门。合编放行，卡编扯呼。人离烧草，报信归洞，孙叽找爷爷。

从来没见过，直接看wp吧
python3脚本：

```
str1 = "今天上午，朝歌区梆子公司决定，在每天三更天不亮免费在各大小区门口设卡为全城提供二次震耳欲聋的敲更提醒，呼吁大家早睡早起，不要因为贪睡断送大好人生，时代的符号是前进。为此，全区老人都蹲在该公司东边树丛合力抵制，不给公司人员放行，场面混乱。李罗鹰住进朝歌区五十年了，人称老鹰头，几年孙子李虎南刚从东北当猎户回来，每月还寄回来几块鼹鼠干。李罗鹰当年遇到的老婆是朝歌一枝花，所以李南虎是长得非常秀气的一个汉子。李罗鹰表示：无论梆子公司做的对错，反正不能打扰他孙子睡觉，子曰：‘睡觉乃人之常情’。梆子公司这是连菩萨睡觉都不放过啊。李南虎表示：梆子公司智商捉急，小心居民猴急跳墙！这三伏天都不给睡觉，这不扯淡么！到了中午人群仍未离散，更有人提议要烧掉这个公司，公司高层似乎恨不得找个洞钻进去。直到治安人员出现才疏散人群归家，但是李南虎仍旧表示爷爷年纪大了，睡不好对身体不好。"
str2 = "喵天上午，汪歌区哗叽公司决定，在每天八哇天不全免费在各大小区门脑设卡为全城提供双次震耳欲聋的敲哇提醒，呼吁大家早睡早起，不要因为贪睡断送大好人生，时代的编号是前进。为此，全区眠人都足在该公司流边草丛合力抵制，不给公司人员放行，场面混乱。李罗鸟住进汪歌区五十年了，人称眠鸟顶，几年孙叽李熬值刚从流北当屁户回来，每月还寄回来几块报信干。李罗鸟当年遇到的眠婆是汪歌一枝花，所以李值熬是长得非常秀气的一个汉叽。李罗鸟表示：无论哗叽公司做的对错，反正不能打扰他孙叽睡觉，叽叶：‘睡觉乃人之常情’。哗叽公司这是连衣服睡觉都不放过啊。李值熬表示：哗叽公司智商捉急，小心居民猴急跳墙！这八伏天都不给睡觉，这不扯淡么！到了中午人群仍未离散，哇有人提议要烧掉这个公司，公司高层似乎恨不得找个洞钻进去。直到治安人员出现才疏散人群归家，但是李值熬仍旧表示爷爷年纪大了，睡不好对身体不好。"
str3 = "喵汪哗叽双哇顶，眠鸟足屁流脑，八哇报信断流脑全叽，眠鸟进北脑上草，八枝遇孙叽，孙叽对熬编叶：值天衣服放鸟捉猴顶。鸟对：北汪罗汉伏熬乱天门。合编放行，卡编扯呼。人离烧草，报信归洞，孙叽找爷爷。"
res = ""
for i in range(len(strs3)):
    for j in range(len(strs2)):
        if strs3[i] == strs2[j]:
            res += strs1[j]
            break
print(res)
```

运行脚本得到：

```
今朝梆子二更头，老鹰蹲猎东口，三更鼹鼠断东口亮子，老鹰进北口上树，三枝遇孙子，孙子对虎符曰：南天菩萨放鹰捉猴头。鹰对：北朝罗汉伏虎乱天门。合符放行，卡符扯呼。人离烧树，鼹鼠归洞，孙子找爷爷。
```

咱也不知道是啥

flag为：`flag{南天菩萨放鹰捉猴头}`

总结

嗯~~，题目有难有简单，刷题虽然要大量的刷，但是还是要仔仔细细的去理解每一道题目，还是自己的薄弱点，rsa！！！

还是继续加强吧！

2021/10/10