wctf2020——writeup（Crypto）

原创

Kihyun_ 于 2020-03-30 18:56:39 发布 602 收藏

文章标签： 密码学

本文链接：https://blog.csdn.net/Kihyun_/article/details/105185038

大数运算

Author: 52HeRtz

flag等于 wctf2020{Part1-Part2-Part3-Part4} 每一Part都为数的十六进制形式（不需要0x），并用 '-' 连接

Part1 = 2020×2019×2018× ... ×3×2×1 的前8位

Part2 = 520^1314 + 2333^666 的前8位

Part3 = 宇宙终极问题的答案 x, y, z绝对值和的前8位

Part4 = 见图片附件，计算结果乘上1314

$$\int_0^{22} 2x\,dx + 36$$

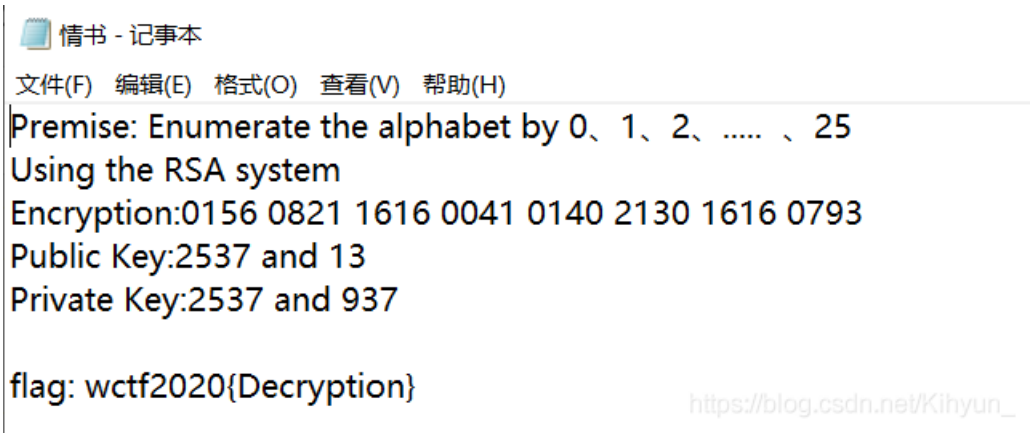
用python直接算就好了

宇宙终极问题的数直接百度上面搜

```
a = 1
for value in range(1,2021):
    a = a * value
print "a="
print a

b = pow(520,1314) + pow(2333,666)
print "b="
print hex(b)

c = 80538738812075974 + 80435758145817515 + 12602123297335631
print "c="
print (c)

d = pow(22,2) + 36
e = d * 1314
print hex(e)
```

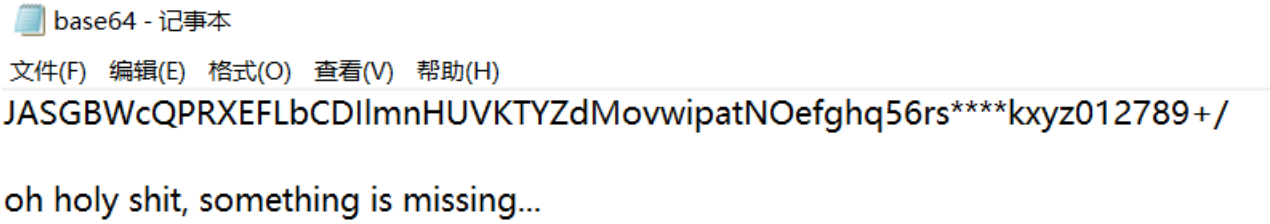
被坑了一下，就是体感说的前八位，是十进制的前八位再转十六进制，并不是直接十六进制数的前八位。

情书



这题目有点意思，一看出题人就是个被甜蜜蜜浸泡着的人
题干说了是用rsa加密的，自然就是用rsa来解密
私钥：n=2537，e=13
公钥：n=2537，d=937
$m=c^d(\mod n)$
一个数一个数算出来对应字母表得到gjmtcwms
一点都不情书，估计要凯撒一下，凯撒爆破得到【iloveyou】
非常情书。

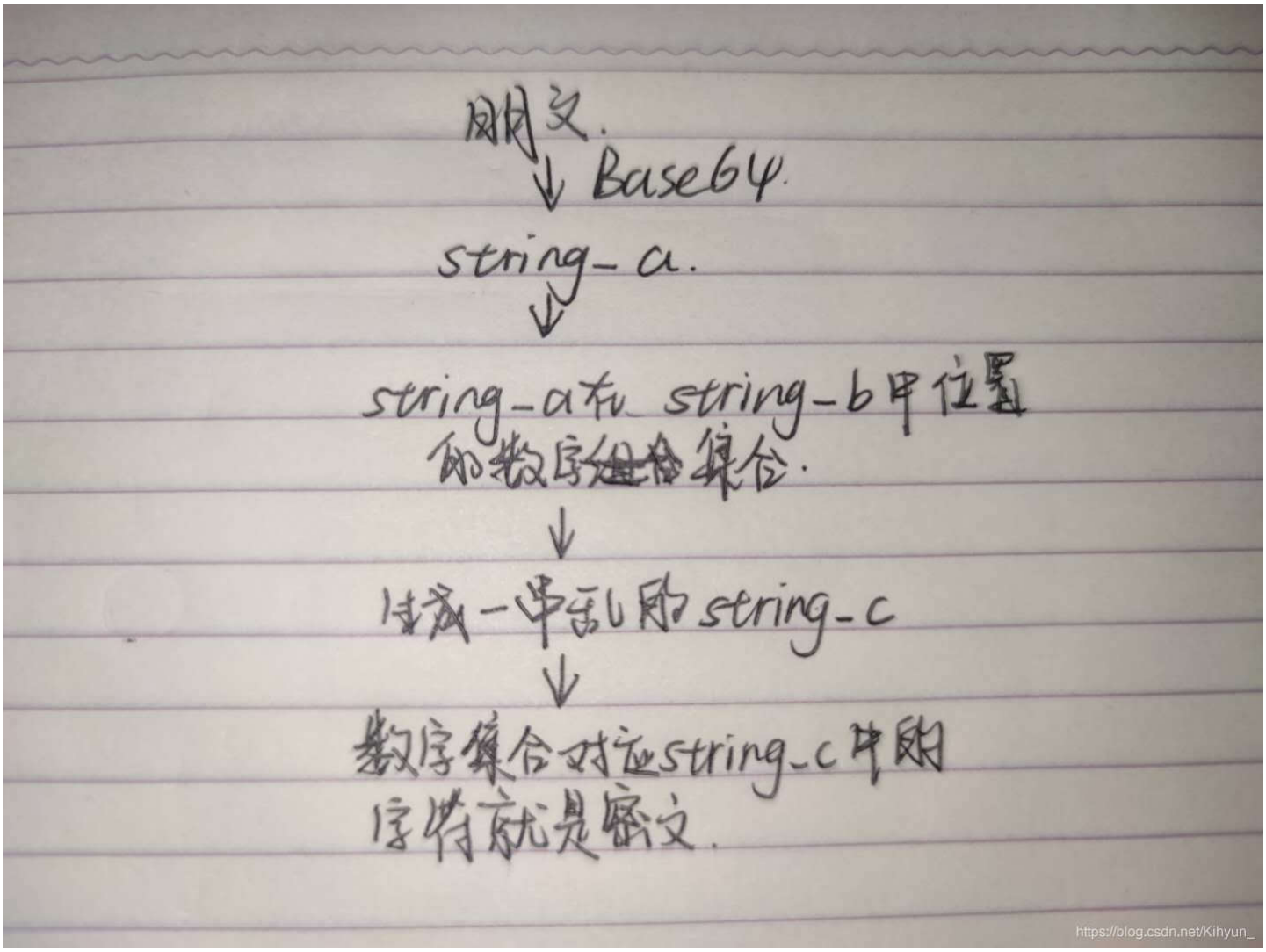
B@se



根据题目直接想到用base64解码，把题目里面那一串扔去解码，得到的是乱码，经过队友翻看大佬的博客发现新的base64的玩法，就是base里面的每一个字符都在string1中找到对应的位置，再在string2中找到该位置对应的字符，再进行base64解密，相当于是经过了三层加密。

加密过程：将明文先用base64加密成string_a，而后将base64中可加密的字符按照string_b=“ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/” 排列，提取string_a中每个字符在string_b中的位置(maybe数组内的数字），然后生成一串string_b乱排列的字符串string_c，最终的密文就是以上提取的数字集合在string_c中的对应的字符。

觉得这个加密方式挺有趣的，手写了个图，应该直观一点



因为题干隐了四个字符，还是要找出来的，现学现用，用了find()函数，找到了这四个字符其实还要确定它在string1的排列顺序，应该要试出来？一共24种可能，说起来还挺方，但是幸好第二次就解出一个非常可观的flag了，确定是34uj。**但是如果是要试全部可能的话怎么写python的代码呢？**

（看了官方writeup用的是**itertools.permutations(,)**）

```python
import base64


base = 'MyLkTaP3FaA7KOWjTmKkVjWjVzKjdeNvTnAjoH9iZOIvTeHbvD=='
string1 = 'JASGBWcQPRXEFLbCDIlmnHUVKTYZdMovwipatNOefghq56rs    kxyz012789+/'
string2 = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
flag = ''
missing = ''

for a in string2:
    string3 = string1.find(a)
    if(string3==-1):
        missing = missing + a
print(missing)  # missing=34ju

string1 = 'JASGBWcQPRXEFLbCDIlmnHUVKTYZdMovwipatNOefghq56rs34ujkxyz012789+/'

for i in base:
    if (i) != '=':
        index = string1.find(i)     # 在string1里面找base
        flag = flag + string2[index]

    else:
        flag = flag + '='

print (flag)
print (base64.b64decode(flag))
```

babyrsa

---

babyrsa - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

c = 28767758880940662779934612526152562406674613203406706867456395986985664083182
n = 73069886771625642807435783661014062604264768481735145873508846925735521695159
e = 65537

一题常规的RSA题目

首先分解n，网址：**http://factordb.com/**

解出p和q

p=189239861511125143212536989589123569301

q=38612312537192365119121986981129358645 9

然后上代码跑出d和m

```python
import math

# 求欧拉函数
def getEuler(prime1,prime2):
    return (prime1-1)*(prime2-1)

# 求私钥
def getDkey(a,Eulervalue):
    k = 1
    while True:
        if (((Eulervalue * k) + 1) % e) ==0:
            (d,a) = divmod(Eulervalue * k + 1,e)
            return d
        k +=1

# 求明文
def Ming(c,d,n):
    return pow(c,d,n)

if __name__=='__main__':
    p = 189239861511125143212536989589123569301
    q = 386123125371923651191219869811293586459
    e =  65537
    n = q*p
    d = getDkey(e,getEuler(p,q))
    c = 28767758880940662779934612526152562406674613203406706867456395986985664083182
    m =Ming(c,d,n)
    print("d=")
    print (d)
    print ("m=")
    print (m)
    print("m的十六进制为: " + hex(m))
```

d=30854876581442056228588093398155288897790570329196285069001545119486056472273

m=8239891082549742021051617585664977861000486186198588435001607556627956 45
m=0x77637466323032307b6a7573745f405f70696563655f30665f63616b337d

最后m的十六进制可以直接去hex解码就可以解出flag了：wctf2020{just_@_piece_0f_cak3}
贴一个hex解码的网址：http://stool.chinaz.com/hex

佛说：只能四天

密文：尊即寂修我劫修如婆愍闍嚤婆莊愍耨羅嚴是唸婆斯吶眾唸修迦慧迦嚼唸斯願嚤摩隸所迦摩吪即塞願修咒莊波斯訶喃壽祇僧若即亦嗲蜜迦須色唸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吪念彌諸喟嚴諦咒陀叻咤叻諦鉢隸祇婆諦嚼阿兜宣囉吪色鉢吶諸劫婆咤咤唸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嚤亦唸寂僧迦色莊壽吪哆尊僧唸喃壽喟兜我空所吶般所即諸吪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嚤薩壽修闍夷闍是壽僧劫祇蜜嚴嚼我若空伏諦念降若心吪咤隸喟耨鉢伏吪色寂喃唸吪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

网上找的**新佛曰解码**都挂了，第一步都解不出来也是非常绝望了，先放在这里，题型扩充吧。

来填坑了。
大佬帮忙找了一个没挂的网址，翻译了一下，如下：
平等文明自由友善公正自由诚信富强自由自由平等民主平等自由自由友善敬业平等公正平等富强平等自由平等民主和谐公正自由诚信平等和谐公正公正自由法治平等法治法治法治和谐和谐平等自由和谐自由自由和谐公正自由敬业自由文明和谐平等自由文明和谐平等和谐文明自由和谐自由和谐和谐平等和谐法治公正诚信平等公正诚信民主自由和谐公正民主平等平等平等平等自由和谐和谐和谐平等和谐自由诚信平等和谐自由自由友善敬业平等和谐自由友善敬业平等法治自由法治和谐和谐自由友善公正法治敬业公正友善爱国公正民主法治文明自由民主平等公正自由法治平等文明平等友善自由平等和谐自由友善自由平等文明自由民主自由平等平等敬业自由平等平等诚信富强平等友善敬业公正诚信平等公正友善敬业公正平等平等诚信平等公正自由公正诚信平等法治敬业公正诚信平等法治平等公正友善平等公正诚信自由公正友善敬业法治法治公正公正公正平等公正诚信自由公正和谐公正平等

社会主义核心价值观，再翻译一波：
RLJDQTOVPTQ6O6duws5CD6IB5B52CC57okCaUUC3SO4OSOWG3LynarAVGRZSJRAEYEZ_ooe_doyouknowfence

看结尾是栅栏密码，题目关键字【四】，栅栏4位：
R5UALCUVJDCGD63RQISZTBOSO54JVBORP5SAT2OEQCWY6CGEO53Z67L_doyouknowCaesar_

看结尾是凯撒，选择爆破：O5RXIZRSGAZDA63ONFPWQYLPL54GSYLOM5PXQ2LBNZTV6ZDBL53W67I
这是倒数第四个移位，看全是大写字母，数字在2-7之间，想到Base32，解码
wctf2020{ni_hao_xiang_xiang_da_wo}

不禁鼓掌，真的是把【四】好好地利用了。

leak

e = 65537

n =
15680834359857877495737569681518898068216674060930283109969649206824633719879251089881849623916633901520730510210143163428316854449298458656679999647115025238214414825723670724726750616567087750637025312769531416398708407646256009545663583365072060633785219936236212080870792591389795652778093042357434328784 7

c =
10854207880905777466674806623547329249534375379044396602063606080741839373725869635256934562148895809485630586560310088583867259176407215718333613924358843558310442326892143947311324449382169256096044368804899455746352609998530366724362371145484157392223305128956186559972200410713430207030123734540035425786 9

dp =
73476313991883702727476568040454685135335695288543966398718100438260165838631735387749912227668615050915122154624975037386502448565234971942718278027582 5

是dp泄露的题目。

找了别人的总结看了。https://www.jianshu.com/p/74270dc7a14b

dp = d mod(p-1)

ed = dp * e - k1(p-1) = 1 mod f(n)

k2(p-1)*(q-1) +1 = dp * e - k1(p-1)

(p-1) * [k2(q-1) + k1] + 1= dp * e

因为dp < (p-1) （因为dp＝d mod（p-1））
所以e > [k2(q-1) + k1]

假设 x = k2*(q-1) + k1
x∈（0,e)
x * (p-1) +1 = dp * e

在0到e中肯定有一个x可以被n整除，求出p和q
（据说如果不知道n的话还可以去判断p是否为素数**gmpy2.isprime()**）

```
import gmpy2
import libnum
import binascii
e = 65537
n = 156808343598578774957375696815188980682166740609302831099696492068246337198792510898818496239166339015207305
102101431634283168544492984586566799996471150252382144148257236707247267506165670877506370253127695314163987084076462560095456635833650720606337852199362362120808707925913897956527780930423574343287847
dp = 734763139918837027274765680404546851353356952885439663987181004382601658386317353877499122276686150509151221546249750373865024485652349719427182780275825
c = 108542078809057774666748066235473292495343753790443966020636060807418393737258696352569345621488958094856305865603100885838672591764072157183336139243588435583104423268921439473113244493821692560960443688048994557463526099985303667243623711454841573922233051289561865599722004107134302070301237345400354257869

for i in range(1,e):
    a = dp*e -1
    if a % i == 0:
        p = ((dp*e-1)//i)+1
        if n % p ==0:
            q = n//p
            # print (p)
            # print (q)
            phi = (p-1)*(q-1)
            d = gmpy2.invert(e,phi)
            # print (d)
            m= pow(c,d,n)
            # print binascii.a2b_hex(m)
            print (libnum.n2s(m))
            # print(hex(m))
            # print m
```