第一页

一眼就解密
网站base64
结果
MD5
网站MD5
结果
Url编码
Url
结果
看我回旋踢
凯撒
结果
摩丝
摩斯
password
题目

```
姓名：张三
生日：19900315
key格式为key{xxxxxxxxxx}
```

由格式提示可知flag有10位，生日有八位，剩下两位应是姓名缩写
flag: `zs19900315`
变异凯撒
网站:字符串转ASCII码
中间结果

```
ch = [97,102,90,95,114,57,86,89,102,83,99,79,101,79,95,85,76,94,82,87,85,99]
j=5
for i in range(22):
  print(chr(ch[i]+j),end='')
  j+=1
print()
```

结果
Quoted-printable

Quoted-printable或QP encoding，没有规范的中文译名，可译为可打印字符引用编码或使用可打印字符的编码。Quoted-printable是使用可打印的ASCII字符（如字母、数字与“=”）表示各种编码格式下的字符，以便能在7-bit数据通路上传输8-bit数据, 或者更一般地说在非8-bit clean媒体上正确处理数据。

Quoted-printable

结果

篱笆墙的影子

栅栏

结果

Rabbit

Rabbit

结果

RSA

```
from gmpy2 import *
from Crypto.Util.number import *
p=473398607161
q=4511491
e=17
phi=(p-1)*(q-1)
d=invert(e,phi)
print(d)
```

结果: `125631357777427553`

丢失的MD5

代码

结果

Alice与Bob

分解

新MD5

结果

大帝的密码武器

加密向量

结果

rsrasa

```
from gmpy2 import *
from Crypto.Util.number import *
p=96484230290105156765905517400104265349457376392357398006439893520398525072984913995610350091634270503701075707336333509116912802977771602006252816653784836
q=11874843837980297032092405848653656852760910154543380907650040190704283358909208578251063047732443992230647903887510065547947313543299303261986053486569406
```

```
7
e=65537
c=832082989951746041747735902982036393605400248712561268928896613457424033149298619391004926666056473166465764865262174570063768422808697285817267464015837058999417682141387422596893348407356335530538876418476511737762518202930872128856701803674068074067659236389731613758173927377478327627516901044238690190 34
phi=(p-1)*(q-1)
n=p*q
d=invert(e,phi)
m=pow(c,d,n)
print(m)
```

结果

Windows系统密码

hash

结果

flag: `good-luck`

信息化时代的步伐

中文电报

结果

凯撒?替换?呵呵!

暴力破解

结果

`flag{substitutioncipherdecryptionisalwayseasyjustlikeapieceofcake}`

https://www.dafont.com/illuminati-masonic-cipher.font

萌萌哒的八戒

猪圈密码

结果

flag: `whenthepigwangtoeat`

权限获得第一步

hash

结果

flag: `3617656`

RSA1

dp&&dq泄露(抄自知识库)

原理

```
from Crypto.Util.number import *
from gmpy2 import *
from math import gcd
p =
86376337672570085670996534865410911713204915094336154475391624379112441758856 6
```

```
780639841179052408355344515811350222774520620532769093950403299469990205322 9
q =
126406749739964727691760479371708834209270508214800105815931371353724738805956
137373376306297525773461470392840300825934907766305725849599542053368802284 69
dp =
650079570221683462110904235119326153065004384105625293093094966335862501688183
284072806602615026469307610935487409984138045488171609777830726811691058292 9
dq =
783472263673553449019532580386470672380574033551303889137911760438881683674556
098098256795673512201963002175438762767516968043599582527539160811120550041
c =
247223054038873820735673164676490806626315529059602293990791079956021544181760
563358006388875276141640735304376570850796761573502053519452229893513160764865
735995760419783398722659250627643185360890073102702785261596789374319038628924
007479155251189839599706079341429747366757843259934459420313721073421038 52
n=p*q#计算模数
#计算私钥
d=invert(dp,p-1)
d=invert(d,dq-1)
#CRT
mp=pow(c,dp,p)
mq=pow(c,dq,q)
invp=invert(p,q)
invq=invert(q,p)
m=(mp*q*invq+mq*p*invp)%n
print(long_to_bytes(m))
```

结果

flag: `W31c0m3_70_Ch1n470wn`

传统知识+古典密码

六十甲子顺序表

初始值: `28,30,23,08,17,10,16,30`

+甲子(60): `88,90,83,68,77,70,76,90`

ASCALL码转字符串:

```
ch=[88, 90, 83, 68, 77, 70, 76, 90]
for i in range(8):
  print(chr(ch[i]),end='')
print()
```

结果: `XZSDMFLZ`

工具:uTools

栅栏+凯撒

flag: `SHUANGYU`

世上无难事

密文:VIZZB IFIUOJBWO NVXAP OBC XZZ UKHVN IFIUOJBWO HB XVIXW XAW VXFI X QIXN VBD KQ IFIUOJBWO WBKAH NBWXO VBD XJBCN NKG QLKEIU DI XUI VIUI DKNV QNCWIANQ XN DXPIMKIZW VKHV QEVBBZ KA XUZKAHNBA FKUHKAKX XAW DI VXFI HBN QNCWIANQ NCAKAH KA MUBG XZZ XEUBQQ XGIUKEX MUBG PKAWIUHXUNIA NVUBCHV 12NV HUXWI XAW DI XUI SCQN QB HZXW NVXN XZZ EBCZW SBKA CQ NBWXO XAW DI DXAN NB NVXAP DXPIMKIZW MBU JIKAH QCEV XA BCNQNXAWKAH VBQN HKFI OBCUQIZFIQ X JKH UBCAW BM XLLZXCQI XAW NVI PIO KQ 640I11012805M211J0XJ24MM02X1IW09

密文末尾连续字符数刚好是答案32位，距离最近的三个连续字符PID就应是关键词key,

用quipqiup破解

结果

答案包含小写字母，大写字母全部转为小写。

flag: `640e11012805f211b0ab24ff02a1ed09`

Unencode

UUencode

结果

old-fashion

工具:uTools

字符置换

[AFCTF2018]Morse

摩斯+十六进制解码

结果

flag: `1s't_s0_345y`

RSA3

共模攻击(抄自知识库)

原理

欧几里得扩展算法(Extended Euclidean Algorithm)用于计算两个整数的最大公约数（GCD），同时还可以找到使得 `ax+by=gcd(a,b)` 成立的整数 x 和 y。

```
from sympy import gcdex
from gmpy2 import *
from Crypto.Util.number import *
c1=22322035275663237041646893770451933509324701913484303338076210603542612758956262869640822486470121149424485571361007421293675516338822195280313794991136048140918842471219840263536338886250492682739436410013436651161720725855484866690084788721349555662019879081501113222996123305533009325964377798892703161521852805956811219563883312896330156298621674684353919547558127920925706842808914762199011054955816534977675267395009575347820387073483928425066536361482774892370969520740304287456555508933372782327506569010772537497541764311429052216291198932092617792645253901478910801592878203564861118912045464959832566051361
n=22708078815885011462462049064339185898712439277226831073457888403129378547354
```

```
029242026701655181905243077900475584664904400102414148528328648313070261605727
469847361114950879886970634750193158311763271070078722801648012767739364992953
041659868602735421642256593445901516192761360790283154285797785961259628235367
932777330372700440726219723158632459918198357262240459035408454178806226216451
014060586812241038809017442014775240855412978976090230089804627390900785281847
403077069964764736301510211895673767394135421769269604496969530850643657314256
557348758350703735694484803986438233921626667067356748887150892531115480 1
e1=11187289
c2=187020100451870155565486916423949828356692621472302127313099386752264585552
104259724294184492734105353879859310367118542656239050668056657518032691068807
467690034789007910995902395139254497488140759040174715855728484735564905654500
626647064491284158347879619472662597897859629222387011340797204142284140661930
714953046123410529874556159300235368238014992697733571860874527475008406404193
650115544211830375056534612867327409837027408226711480456194976671845861236572
856040618756539095678223289140653377977334446403515187754876498199782623636172
657979828431796308887294072384966509877204287082171152579890078673316983971
e2=9647291
s=gcdex(e1,e2)#欧几里得扩展算法
s1=s[0]
s2=s[1]
#如果s1或s2为负,则取其绝对值,并计算相应密文的模逆
if s1<0:
    s1=-s1
    c1=invert(c1,n)
elif s2<0:
    s2=-s2
    c2=invert(c2,n)
m = pow(int(c1),int(s1),n)*pow(int(c2),int(s2),n) % n
print(long_to_bytes(m))
```

结果

flag: `49d91077a1abcb14f1a9d546c80be9ef`

RSA2

dp泄露(e较小)

原理

```
from gmpy2 import *
from Crypto.Util.number import *
e = 65537
n =
248254007851526241177721526698901802985832766176221609612258877371620580060433
101538328030305219918697643619814200930679612109885533801335348445023751670478
437073055544724280684733298051599167660303645183146161497485358633681492129668
802402065797789905550489547645118787266601929429724133167768465309665906113
dp =
```

```
905074498052346904643025132879518330691925174573054004621877253318682675055421
970943552016695528560364834446303196939207056642927148093290374440210503657

c =
140423670976252696807533673586209400575664282100684119784203527124521188996403
826597436883766041879067494280957410201958935737360380801845453829293997433414
188838725751796261702622028587211560353362847191060306578510511380965162133472
698713063592621028959167072781482562673683090590521214218071160287665180751
#使用给定的dp和e计算d
for k in range(1,e):
  #计算p
  if (e*dp-1)%k==0:
    p=(e*dp-1)//k+1
    #如果n能被p整除，找到p和q
    if n%p==0:
      q=n//p
      phi=(p-1)*(q-1)#欧拉函数
      d=invert(e,phi)#模逆元
      m=pow(c,d,n)
      break
print(long_to_bytes(m))
```

结果

flag: `wow_leaking_dp_breaks_rsa?_98924743502`

还原大师

```
# -*- coding: utf-8 -*-                # 指定编码为UTF-8
#!/usr/bin/env python                   # 指定解释器为Python
import hashlib                          # 导入 hashlib 模块，用于生成哈希

# 定义要还原的明文字符串，其中包含问号（?）需要被替换
k = 'TASC?O3RJMV?WDJKX?ZM'

# 开始遍历字母A到Z（ASCII码65到90）
for i in range(26):
    # 将第一个问号替换为当前的字母
    temp1 = k.replace('?', str(chr(65 + i)), 1)

    for j in range(26):  # 遍历字母A到Z
        # 将第二个问号替换为当前的字母
        temp2 = temp1.replace('?', chr(65 + j), 1)

        for n in range(26):  # 再次遍历字母A到Z
            # 将第三个问号替换为当前的字母
            temp3 = temp2.replace('?', chr(65 + n), 1)
```

```
            # 计算temp3的MD5哈希值，并转换为大写
            s = hashlib.md5(temp3.encode('utf8')).hexdigest().upper()  # 注意大
小写

            # 检查生成的哈希值的前四位是否为'E903'
            if s[:4] == 'E903':
                print(s)  # 输出符合条件的密文
```

结果

`flag{E9032994DABAC08080091151380478A2}`

异性相吸

将两个文件内容转为二进制，然后进行异或，再转为字符串

RSA

打开pub文件，将两行字符串转成数字，分别是n和e，分解n，得到p，q，密文在enc文件中，解密

RSAROLL

打开data文件，{n，e}，分解n，得到p，q，转移到脚本中，删除{}

```
from gmpy2 import *
from Crypto.Util.number import *
p,q,n,e=49891,18443,920139713,19
m=[]
phi=(p-1)*(q-1)
d=invert(e,phi)
with open(r"D:\CTF\RsaRoll\data.txt","r") as c:
    for line in c.readlines():
        line=line.strip('\n')
        m.append(pow(int(line),d,n))
for i in m:
    print(chr(i),end='')
print()
```

结果

`flag{13212je2ue28fy71w8u87y31r78eu1e2}`