Crypto

RSA Attack

使用factordb网站对n进行因式分解，得到pq。

代入代码：

```
"""
import gmpy2
def Decrypt(c,e,p,q):
    L=(p-1)*(q-1)
    d=gmpy2.invert(e,L)
    n=p*q
    print(n)
    m=gmpy2.powmod(c,d,n)
    flag=str(m)
    print("flag{"+flag+"}")
if __name__ == '__main__':
    p = 715800347513314032483037
    q = 978782023871716954857211
    e = 65537
    c = 122622425510870177715177368049049966519567512708
    Decrypt(c,e,p,q)
```

libnum.n2s(flag)，得到flag

Chinese Character Encryption

根据题意，不同行同列汉字表示同一个字符，因此使用栅栏将同列汉字拼在一起寻找规律，根据第一个字符与第二个字符为h、g找出规律为（所有字符相加后+48）%128，以下为结果与代码：

```
hhhhhhhhhhhhhhhhgggggggggggggggggaaaaaaaaaaaaaaaammmmmmmmmmmmmmmmeeeeeeeeeeeeeeee{{{{{{{{{
{{{{{{IIIIIIIIIIIIIIIItttttttttttttttt*****Z****ZZ****ssssssssssssssssEEuuEEEEEEEEuuuEEEEE
EEEEEEuEEEEuMMMMMMMMMMMMMMMMSSSSSSSSSSSSSSSS||||||||||||||||
tttttttttttttttthhhhhhhhhhhhhhhhaaaaaaaaaaaaaaaaTTTTTTTTTTTTTTTT~~~~~~~~~~~~~~~~YYYYYYYYYY
YYYYYYOOOOOOOO OOOOOOOuuuuuuuuuuuuuuuu================LLLLLLLLLLLLLLLLEEEuEEEEEEEEEEEEAAAA
AAAAAAAAAAAArrrrrrrrrrrrrrrrnnnnnnnnnnnnnnnn@@@p@@@@@@@@@@@@PPPPPPPPPPPPPPPPiiiiiiiiiiiiii
iinnnnnnnnnnnnnnnnYYYYYYYYYYYYYYYYiiiiiiiiiiiiiiiiNNNNNNNNNNNNNNNN^^^^^^^^^^^^^^^^VVVVVVVV
VVVVVVVVeeeeeeeeeeeeeeeerrrrrrrrrrrrrrrrYYYYYYYYYYYYYYYY]]]]]]]]]]]]]]]]WWWWWWWWWWWWWWWWee
eeeeeeeeeeeeeel1111111 11111111LLLLLLLLLLLLLLLL}}}}}}}}}}}}}}}}
```

```python
"""

t="[illegible]
import pypinyin
a=pypinyin.slug(t,style=pypinyin.TONE)
b=pypinyin.slug(t)
print(a)
p=""
mark=0
word=0
q=""
for i in range(len(a)):
    if a[i]=='ā' or a[i]=='ō'or a[i]=='ē'or a[i]=='ī'or a[i]=='ū'or a[i]=='ǖ':
        flag=1
    if a[i]=='á' or a[i]=='ó'or a[i]=='é'or a[i]=='í'or a[i]=='ú'or a[i]=='ǘ':
        flag=2
    if a[i]=='ǎ' or a[i]=='ǒ'or a[i]=='ě'or a[i]=='ǐ'or a[i]=='ǔ'or a[i]=='ǚ':
        flag=3
    if a[i]=='à' or a[i]=='ò'or a[i]=='è'or a[i]=='ì'or a[i]=='ù'or a[i]=='ǜ':
        flag=4
    if a[i]!='-'and b[i]>='a'and b[i]<='z':
        word += ord(b[i])
    if a[i]=='-':
        q+=(chr((flag+word+48)%128))
        word=0
        flag=0
q+=(chr((flag+word+48)%128))
print(q)
```

输入字符占比大的字符为flag报错，可以发现结果中仍有部分字符未保持一致，且ascii字符相差均为48。寻找真正规律未果，现已确定英文字符的正确性，因此对每个非英文字符进行ascii查表，最终发现']'字符与'-'字符相差48。尝试替换，成功得到flag。

The Password Plus Pro Max Ultra

第一个想法是通过PuLP库进行64个01变量的整数规划，不过因为不知道如何表示取模约束而告终。

根据题意查找去年hgame的WP，发现Crypto有一道名为“The Password”的题目，用同样的方法写出。

以下为过程中改进的异或方程组的代码

import libnum

n = 64

a = []

answer = 0

```
k = 0
B = ""
p = [ ]
time =
for i in range(n):
  b=[]
  for j in range(n):
    flag=0
    for k in range(time):
      if (j == i or j == ((i + p[k])%64)) and flag == 0:
        b.append(1)
        flag = 1
    if flag == 0:
      b.append(0)
  if B[i] == "1":
    b.append(1)
  else:
    b.append(0)
  a.append(b)
print(a)


def guass():
  global n
  r = 0
  for c in range(n):
    t = r
    # 首先找到当前列中的1
    for i in range(r, n):
      if a[i][c] == 1:
        t = i
        break
     if a[t][c] == 0: continue
    # 交换
    a[r], a[t] = a[t], a[r]
```

```
    # 将这一列为1的与第r行异或
    for i in range(r+1, n):
      if a[i][c] == 1:
        for j in range(c, n+1):
          a[i][j] ^= a[r][j]
    r += 1


  if r < n:
    for i in range(r, n):
      if a[i][n] == 1: return 1 # 无解
    return 2 # 无穷解


  for i in range(n-1, -1, -1):
    for j in range(i):
      if a[j][i] == 1:
        a[j][n] ^= a[i][n]
  return 0
ret = guass()
if ret == 1: print('No solution')
elif ret == 2: print('Multiple sets of solutions')
else:
  for i in range(n):
    answer *= 2
    answer += a[i][n]
print(libnum.n2s(answer))
```

RSA Attack 2

part1

寻找n1、n2的公因数q，即可求解m

part2

小指数公钥攻击，七次开方求解m

```
import gmpy2
flag=0
c=10262871020519116406312674685238364023536657841034751572844570983750295909
n=14157878492255346300993349653813018105991884577529909522555551468374307942
k = 0
while True:
    print (k)
    a, b = gmpy2.iroot(c + k*n, 7)
    if b:
        m = a
        break
    k += 1
print (m)
```

part3

共模攻击

```
"""

from gmpy2 import *
import libnum

n=188195091881062303634448133504681620561644346427294046329830825182253880695447773745441
e1=2519901323
e2=3676335737
s = gcdext(e1, e2)
s1 = s[1]
s2 = -s[2]

c1=32307797262255448725314411690093070720737545787618883879834032063645484514967365139054
c2=94081859562227916143983671964170784679029465088879982233500738585416673645928312943476
c2 = invert(c2, n)
m = (pow(c1,s1,n) * pow(c2 , s2 , n)) % n
print (m)
```

IoT

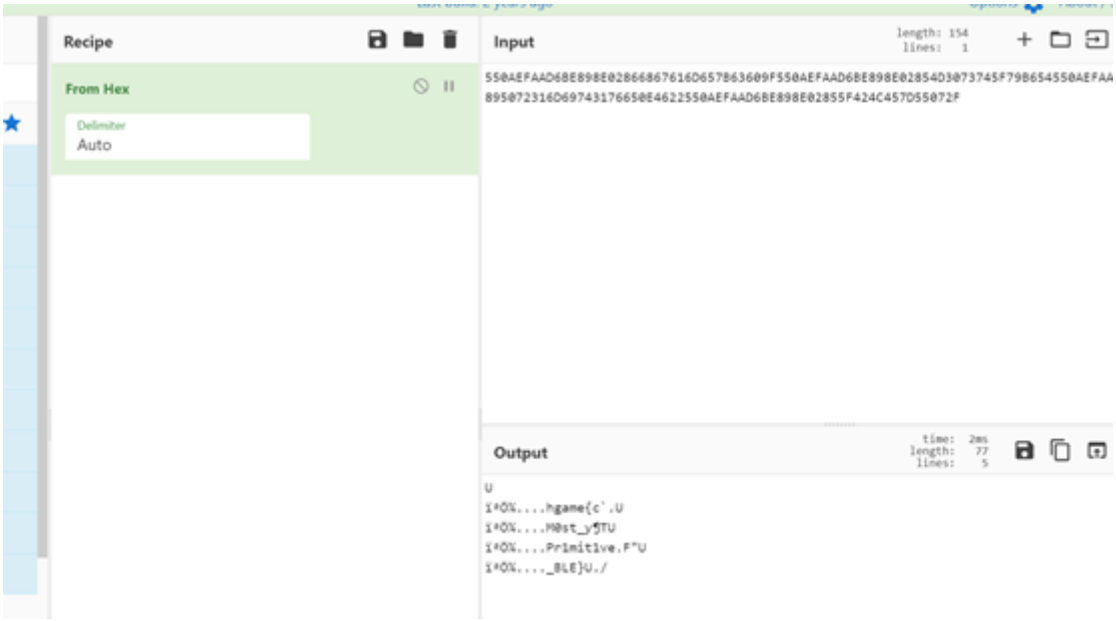
空气中的信号

将无线信号以字节为单位翻转，并转化为16进制。

```
"""

b="0123456789ABCDEF"
a="10101010010100001111011101010101011010110111111011001000101110001010000000110000100010
c=""
k=""
f=""
for i in range(0,len(a),8):
    for j in range(8):
        c+=a[i+7-j]
for i in range(0,len(c),4):
    p=0
    for j in range(4):
        p*=2
        p+=int(c[i+j])
    k+=b[p]
print(k)
```

放入CyberChef



掐头去尾，得到flag

Misc

奇妙小游戏

sha256暴力开门，进入小游戏

通过不断失败找到规律：从下方题目所给数字代表的道路出发，途中若碰到可以左右拐弯的道路强制转弯，答案为最终到达的终点数字。

玩小游戏得到flag。

RE

xD MAZE

拖入ida。分析代码，发现正确输入为hgame{二十八个数字}，‘0’，’1‘，’2‘，’3‘有特殊含义、使对应数组a的下标i移动，并且需要保证每次移动均不会使a[i]值等于32。

了解这些前提条件后，对应数组的下标值写出flag。