week1-V

Crypto

InfantRSA

题目介绍:

真*签到题

p = 681782737450022065655472455411;

q = 675274897132088253519831953441;

e = 13;

c = pow(m,e,p*q) = 275698465082361070145173688411496311542172902608559859019841

解题思路:

知道所有的条件了,我们只需要算出私钥d就可以解密了,私钥d是公钥在模 $\phi(n)$,即是这里的 $\phi(p*q)=(p-1)*(q-1)$

直接用gmpy2模块下的invert可以求出私钥来, d=gmpy2.invert(e,(p-1)*(q-1))

然后m (即flag) 为: m=pow(c,d,p*q)

这里解密方程的证明:

```
欧拉定理: 若a,n为正整数,且a,n互素(即\gcd(a,n)=1),则a^{\varphi(n)}=1 \mod n。
尼知1 = ed \mod \varphi(n), \ m < n, \ c = m^e \mod n, \ \gcd(m,n) = 1, \ 求证 m = c^d \mod n_*
证明:
等式左边为m
等式右边为c^d \mod n
                                        c = m^e \mod n
                      \therefore c^d \bmod n = (m^e \bmod n)^d \bmod n = m^{ed} \bmod n
                                     \therefore 1 = ed \mod \varphi(n)
                                       \therefore ed = k\varphi(n) + 1
            \therefore c^d \bmod n = m^{ed} \bmod n = m^{k\varphi(n)+1} \bmod n = m(m^{\varphi(n)})^k \bmod n
                                        gcd(m,n) = 1
                                       \therefore m^{\varphi(n)} = 1 \mod n
 \therefore c^d \bmod n = m(m^{\varphi(n)})^k \bmod n = m(1 \bmod n)^k \bmod n = m(1)^k \bmod n = m \bmod n
                                            m < n
                                        m \mod n = m
                                 c c^d \mod n = m \mod n = m
```

```
import gmpy2
p = 681782737450022065655472455411;
q = 675274897132088253519831953441;
e = 13;
c = 275698465082361070145173688411496311542172902608559859019841
d = gmpy2.invert(e,(p-1)*(q-1))
m = pow(c,int(d),p*q)
print hex(m)[2:-1].decode('hex')
```

flag:hgame{t3Xt6O0k_R5A!!!}

Affine

题目介绍:

Some basic modular arithmetic...

解题思路:

加密代码:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import gmpy2
from secret import A, B, flag
assert flag.startswith('hgame{') and flag.endswith('}')
TABLE = 'zxcvbnmasdfqhjklqwertyuiop1234567890QWERTYUIOPASDFGHJKLZXCVBNM'
MOD = len(TABLE)
cipher = ''
for b in flag:
   i = TABLE.find(b)
   if i == -1:
       cipher += b
    else:
        ii = (A*i + B) \% MOD
        cipher += TABLE[ii]
print(cipher)
# A8I5z{xr1A_J7ha_vG_TpH410}
```

是拓展了字符表的仿射加密

我们可以知道放射加密的公式是: c=Ax+B (mod m) 其中, x为明文, c为密文, m为模数, 这里 m=len(TABLE)

所以对应的解密公式即为: $x\equiv (c-B)*A^{-1} \pmod m$, 这里, A^{-1} 是A在模m下的逆元, (如rsa中d是e在模 $\phi(n)$ 下的逆元一般),

题目中,我不知道A,B,所以我选择爆破A,B的方法,然后在所有答案中找包含字符'hgame'的结果。 (因为知道flag中前五个字符为hgame,所以其实也可以解方程,但是因为TABLE比较小,为了抢个一血,就直接爆了,甚至脚本也是在原基础上直接改的)

exp

```
TABLE = 'zxcvbnmasdfghjklqwertyuiop1234567890QWERTYUIOPASDFGHJKLZXCVBNM'
MOD = 62
for A in range(MOD):
    for B in range(MOD):
        flag=""
        cipher = 'A8I5z{xr1A_J7ha_vG_TpH410}'
        for b in cipher:
            i = TABLE.find(b)
            if i == -1:
                flag += b
            else:
                try:
                    ii = (i-B)*invert(A,MOD)%MOD
                    flag += TABLE[ii]
                except:
                    break
        if 'hgame' in flag:
            print flag
            exit(0)
            # A8I5z{xr1A_J7ha_vG_TpH410}
```

flag: hgame{M4th_u5Ed_iN_cRYpt0}

not_One_time

题目介绍:

In cryptography, the one-time pad (OTP) is an encryption technique that cannot be cracked, but... Just XOR;P

nc 47.98.192.231 25001 hint: reduced key space

解题思路:

一次性密码本本身无解,程序漏洞在于key的空间被缩减了,可以多次连接获取密文文本,此时一直不变的flag反而可以看作是用来加密的key了。然后爆破flag,使得flag的每一位能解密 **密文文本** 的**每一项**的对应位,解密成功的判断条件是恢复出的明文在给出的key的空间中。

举个栗子:

flag的第一个字符肯定是h吧,

比如我们连接20次,获得20组密文,

h和每一组的密文的第一个字符异或,肯定也是一个可见字符, 并且20组异或下来,获得的可见字符肯定也是20个,不会少一个。

所以我们只需要遍历所有的密钥空间内的元素,找到满足条件(即,和20组密文的对应位异或下来能生成20个可见字符)的字符,组合起来即为flag

exp

```
from pwn import *
import string, binascii, base64

def getex():
    example=[]
```

```
for _ in range(50):
       p=remote('47.98.192.231',25001)
       example.append(base64.b64decode(p.recv()))
       p.close()
   return example
ex=getex()
            #获取50组密文
tablekey=string.ascii_letters+string.digits
tableflag=tablekey+'~!@#$%^&*()_+-=`{}' #flag的可能的字符
flag=[]
for index in range(len(ex[0])):
   flag_may=''
   for i in tableflag:
       possible=[]
       for j in ex:
          a = chr(ord(i)^ord(j[index]))
          if a in tablekey:
                             #如果异或结果是在密钥空间里的话,将结果放入数组
              possible.append(a)
       if len(possi) == len(ex): #如果数组长度等于密钥的组数(这里的50)
          flag_may+=i
                       #该字符符合条件,放入该位候选(因为有可能一个位置有多个字符满
足条件,可能需要二次排除)
   flag.append(flag_may)
print flag
              #看每一组是否只有一个字符,如果不是的话,多跑几遍脚本,取个交集就可。
flag_string=""
for i in flag:
   flag_string+=i
print flag_string
```

```
| Section | Sect
```

可以看到第二次每一组都只剩一个候选了, 那么直接得到flag

hgame{r3us1nG+M3\$5age-&&~rEduC3d_k3Y-5P4Ce}

Recoder

题目介绍:

We found a secret oracle and it looks like it will encrypt your input... nc 47.98.192.231 25002

解题思路:

根据题目名字, 再经过多次尝试可以发现, 服务器会将你的输入进行简单的置换,

然后在交互十次之后会给出置换后的flag,长度为32位

```
[root@iZbp15axph2ymoey4m1kisZ ~]# nc 47.98.192.231 25002
> ~!@#$%^&*() +`1234567890-=
~2)^`_%!@$*+&(#13 9 8457- 0=6
> abcdefghijklmnopgrsatxyz
apkgnlfbceimhjdoq y xrst z a
> qwertyuiopasdfghjklzxcvbnm
qhaufsywetodiprgj v cklxn bmz
> qwertyuiopasdfghjklzxcvbnmQWERTY
qhaufsywetodiprgjYQvRWcklxnEbmzT
> 1
1
> 2
2
> 3
3
 4
4
> 5
5
> 6
Rua!!!
hL+jm5{gae$IUtmp3}iu!0m_PRAnTTe!
```

那么我们只需要输入32位字符,观察他置换后结果,得到映射关系,然后我们根据这个映射关系,将最后给出的置换过后的flag置换回来即可。(注意,为了让我们的映射关系为——映射,即映射关系为双射关系,我们的输入应该是**不重复**的字符)

exp

```
cin='qhaufsywetodiprgjYQvRWcklxnEbmzT'
cout='qwertyuiopasdfghjklzxcvbnmQWERTY'
flagout='hL+jm5{gae$IUtmp3}iu!0m_PRAnTTe!'
index=[]
for i in cout :
    index.append(cin.find(i))
flag=''
for i in range(32):
    flag+=flagout[index[i]]
print flag
```

flag:hgame{jU\$t+5ImpL3_PeRmuTATi0n!!}