RSA(2019/1/23)作业writeup

RSA实践

> *在一次RSA密钥对生成中，假设p=473398607161，q=4511491，e=17*
> *求解出d 将得到的d提交*

```
#RSA实践
#三种解法
'''
第一种gmpy2解法
mpz:Multiple-precision Integers（多精度型整数）
gmpy2.invert():求模逆
'''
import gmpy2
p =gmpy2.mpz(473398607161)
q =gmpy2.mpz(4511491)
e =gmpy2.mpz(17)
fn= (p - 1) * (q - 1)
d = gmpy2.invert(e, fn)
print("d is:")
print (d)

'''
直接用扩展欧几里得法
'''
def computeD(fn, e):
    (x, y, r) = ext_euclid(fn, e)
    #y maybe < 0, so convert it
    if y < 0:
        return fn + y
    return y

def ext_euclid(a, b):
    if b == 0:
        return (1, 0, a)
    else:
        x, y, q = ext_euclid(b, a % b)
        # q = gcd(a, b) = gcd(b, a%b)
        x, y = y, (x - (a // b) * y)  # '//'代表整除，可以用python演示一下.
        return (x, y, q)

p = 473398607161
```

```
q = 4511491
e = 17

n = p * q
print("n:  "+str(n))
fn = (p - 1) * (q - 1)
print("f(n):"+str(fn))
d = computeD(fn, e)
print(d)

'''
用工具解密——RSAtools
ps:用这个工具记得将e转化为hex形式
'''
```

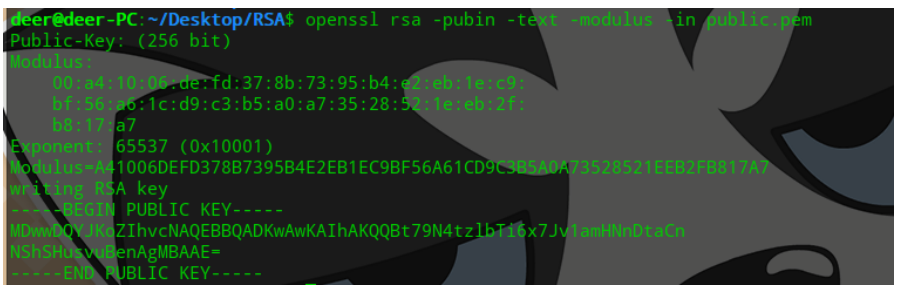
RSA

点击下载所需文件，解压得到

方法一：openssl应用

根据之前讲的，在命令行里输入：`openssl rsa –pubin –text –modulus –in public.pem`

得到公钥里面信息：



注意，这里的modulus和RSAtools一样是n,即

`n =0xA41006DEFD378B7395B4E2EB1EC9BF56A61CD9C3B5A0A73528521EEB2FB817A7`

p = 258631601377848992211685134376492365269

q = 286924040788547268861394901519826758027

e = 65537

通过我们今天讲的也就是RSA实践中给出的三种方法（任选一种）解出：

d = 230717693751110404252872446253287976152957728141801093667842499764982154 94337

用RSAtool生成秘钥：

python rsatool.py -f PEM -o private.pem -n 742076241429452422630570352871109839676460200573078287095879696467013617642 63 -d 230717693751110404252872446253287976152957728141801093667842499764982154 94337

得到秘钥文件,private.pem

得到flag：ISG{256bit_is_weak}

方法二：rsa模块的应用（就是小写的rsa)

获得n,p,q,e,d的方法如上：

```
n = 0xA41006DEFD378B7395B4E2EB1EC9BF56A61CD9C3B5A0A73528521EEB2FB817A7
p = 258631601377848992211685134376492365269
q = 286924040788547268861394901519826758027
e = 65537
d = 230717693751110404252872446253287976152957728141801093667842499764982154 94337
```

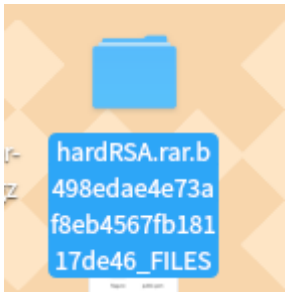
rsa模块里面可以利用PrivateKey函数构建出私钥，同时用decrypt进行解密

```
#-*-coding:utf-8-*-
```

```
import rsa
'''
rsa是一个python模板，python2.7-python3.6均支持
'''
n=74207624142945242263057035287110983967646020057307828709587969
646701361764263
e=65537
d=23071769375111040425287244625328797615295772814180109366784249
976498215494337
p=25863160137784899221168513437649236526 9
q=28692404078854726886139490151982675802 7

prikey = rsa.PrivateKey(n,e,d,p,q)

#print(prikey)

t =
rsa.decrypt(b'\x49\xB9\x6E\xDB\xE3\x96\x1F\x58\xD5\x29\x07\x4B\xD8\x9
3\xD6\xE0\x36\xCE\xAF\x2B\x6D\x21\x4B\x47\x0F\xDC\x0D\x48\x72\x3D\
x6A\x40', prikey)
print(t)
```

Fake_rsa

> *这题的原题的HardRSA*



下载文件本来是这个样子，在linux里面可以直接解压，但是在windows里面不行，可以将.rar后面的后缀删掉，就可以解压

得到以上文件。

用 `openssl rsa -pubin -text -modulus -in pubkey.pem`

```
e = 2
n = 879243482641324068752761405144999371450508936656025929924181716470424916584 61
```

还是用yafu进行分解:

```
p = 319576316814478949870590164193048041239
q = 2751278603513489281732851743815811522 99
```

由于e = 2，所以很容易想到rabin算法

```
#!usr/env/python
#-*-coding:utf-8-*-
import gmpy
import libnum
n=879243482641324068752761405144999371450508936656025929924181716470424916584 61
p=319576316814478949870590164193048041239
q=2751278603513489281732851743815811522 99
e=2
c=0x39de036de3132757e819f769ead64bb487ee3f47e67843afb73748fd9e979be0
mp=pow(c,(p+1)/4,p)
mq=pow(c,(q+1)/4,q)
yp=gmpy.invert(p,q)
yq=gmpy.invert(q,p)
r=(yp*p*mq+yq*q*mp)%n
rr=n-r
s=(yp*p*mq-yq*q*mp)%n
ss=n-s
print (libnum.n2s(r))
print (libnum.n2s(rr))
print (libnum.n2s(s))
print (libnum.n2s(ss))
```