

2022年HGAME中CRYPTO的Easy RSA

原创

沐一·林 于 2022-04-04 16:21:50 发布 20 收藏

分类专栏: [CTF 密码学](#) 文章标签: [ctf](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/xiao__1bai/article/details/122766733

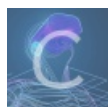
版权



[CTF 同时被 2 个专栏收录](#)

167 篇文章 6 订阅

订阅专栏



[密码学](#)

51 篇文章 1 订阅

订阅专栏

2022年HGAME中CRYPTO的Easy RSA

Easy RSA[已完成]

描述

这 RSA 不是有手就行?!

(100分的题能拿125分, 这不血赚)

题目地址 <https://cmfj-1308188104.cos.ap-shanghai.myqcloud.com/Week1/Easy%20RSA.zip>

基准分数 125

当前分数 125

完成人数 417

CSDN @沐一·林

照例下载附件：

```
from math import gcd
from random import randint
from gmpy2 import next_prime
from Crypto.Util.number import getPrime
from secret import flag

def encrypt(c):
    p = getPrime(8)
    q = getPrime(8)
    e = randint(0, p * q)
    while gcd(e, (p - 1) * (q - 1)) != 1:
        e = int(next_prime(e))
    return e, p, q, pow(ord(c), e, p * q)

if __name__ == '__main__':
    print(list(map(encrypt, flag)))
```

生成一个8位的随机质数
在0~65535中生成一个随机整数
这里用循环知道获取到的e与d=((p-1)*(q-1))互质数为止
很常规的RSA加密算法
列表遍历flag，每次加密一个flag字符。

既然是常规的RSA加密，那用常规的RSA解密即可，每次解密一个字符，那连起来就是一串了。

CSDN @沐一·林

结合遍历列表中元祖，直接常规RSA解密即可

```
import libnum
from Crypto.Util.number import long_to_bytes
list1=[(12433, 149, 197, 104), (8147, 131, 167, 6633), (10687, 211, 197, 35594), (19681, 131, 211, 15710), (33577, 251, 211, 38798), (30241, 157, 251, 35973), (293, 211, 157, 31548), (26459, 179, 149, 4778), (27479, 149, 223, 32728), (9029, 223, 137, 20696), (4649, 149, 151, 13418), (11783, 223, 251, 14239), (13537, 179, 137, 11702), (3835, 167, 139, 20051), (30983, 149, 227, 23928), (17581, 157, 131, 5855), (35381, 223, 179, 37774), (2357, 151, 223, 1849), (22649, 211, 229, 7348), (1151, 179, 223, 17982), (8431, 251, 163, 30226), (38501, 193, 211, 30559), (14549, 211, 151, 21143), (24781, 239, 241, 45604), (8051, 179, 131, 7994), (863, 181, 131, 11493), (1117, 239, 157, 12579), (7561, 149, 199, 8960), (19813, 239, 229, 53463), (4943, 131, 157, 14606), (29077, 191, 181, 33446), (18583, 211, 163, 31800), (30643, 173, 191, 27293), (11617, 223, 251, 13448), (19051, 191, 151, 21676), (18367, 179, 157, 14139), (18861, 149, 191, 5139), (9581, 211, 193, 25595)]
flag=""
for i in list1:
    e=i[0]
    q=i[1]
    p=i[2]
    c=i[3]
    n=p*q
    d = libnum.invmod(e, (p - 1) * (q - 1)) #invmod(a, n) - 求a对于n的模逆,这里逆向加密过程中计算ψ(n)=(p-1)(q-1),对ψ(n)解密,也就是对应根据ed=1modψ(n),求出d
    m = pow(c, d, n) # pow(x, y[, z])--函数是计算x的y次方,如果z在存在,则再对结果进行取模,其结果等效于pow(x,y)%z,对应前面解密算法中M=D(C)=C^d(mod n)
    #print(m) #明文的十进制格式
    string = long_to_bytes(m) # m明文,用长字节划范围
    flag+=string.decode()
print(flag)
```

结果如下:

```
└─$ python 10.py  
hgame{L00ks_l1ke_y0u've_mastered_RS4!}
```

解毕!

敬礼!



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)