CTFRSA加解密

原创

1ens 已于 2022-01-30 20:11:46 修改 192 收藏
分类专栏： 密码学 文章标签： 密码学
于 2021-10-06 00:06:35 首次发布

最近做题遇到好几次RSA解密，最近密码学课也学到了 RSA密码算法，写一篇文章促进对RSA的理解。

一：RSA的基本原理的理解：

1.学习RSA加密得知道欧拉函数，想理解欧拉函数又得弄明白同余类与剩余系的知识，这些是基础。

2.RSA的密钥生成原理：

第一步：选择两个较大素数P，Q.

第二步：计算n=pq，z=（p-1）（q-1）。

第三步：随机选取e（其中e<n），e与z没有公因数（e，z互为质数）

第四步：选取d使得ed-1能够被z完全整除。ed mod z=1

第五部：公钥是（n，e） 私钥是（n，d）

加密/解密算法：

如上所述给出 (n,e) 和 (n,d)。

加密：由 $c = m^e \bmod n$ 将明文 m 转变为密文 c（即：当 m^e 除以 n 所得的余数）。

注意：$m < n$ （如果需要，则分块）

解密：$m = c^d \bmod n$（即：c^d 除以 n 所得的余数）。

核心思想：$m = (\underbrace{m^e \bmod n}_{c})^d \bmod n$

二 下面以SUCTF 2019 signin记录一下RSA的应用与解密方法

IDA View-A | Pseudocode-C | Pseudocode-B | Pseudocode-A | Hex View-1 | Struct

```
__int64 __fastcall main(int a1, char **a2, char **a3)
{
  char v4[16]; // [rsp+0h] [rbp-4A0h] BYREF
  char v5[16]; // [rsp+10h] [rbp-490h] BYREF
  char v6[16]; // [rsp+20h] [rbp-480h] BYREF
  char v7[16]; // [rsp+30h] [rbp-470h] BYREF
  char v8[112]; // [rsp+40h] [rbp-460h] BYREF
  char v9[1000]; // [rsp+B0h] [rbp-3F0h] BYREF
  unsigned __int64 v10; // [rsp+498h] [rbp-8h]

  v10 = __readfsqword(0x28u);
  puts("[sign in]");
  printf("[input your flag]: ");
  __isoc99_scanf("%99s", v8);
  sub_96A(v8, v9);
  __gmpz_init_set_str(v7, "ad939ff59f6e70bcbfad406f2494993757eee98b91bc244184a377520d06fc35", 16LL);
  __gmpz_init_set_str(v6, v9, 16LL);
  __gmpz_init_set_str(v4, "10346103590081691412139010129904904441395040517371217043416168653987816098454 9", 10LL);
  __gmpz_init_set_str(v5, "65537", 10LL);
  __gmpz_powm(v6, v6, v5, v4);
  if ( (unsigned int)__gmpz_cmp(v6, v7) )
    puts("GG!");
  else
    puts("TTTTTTTTTql!");
  return 0LL;
}
```

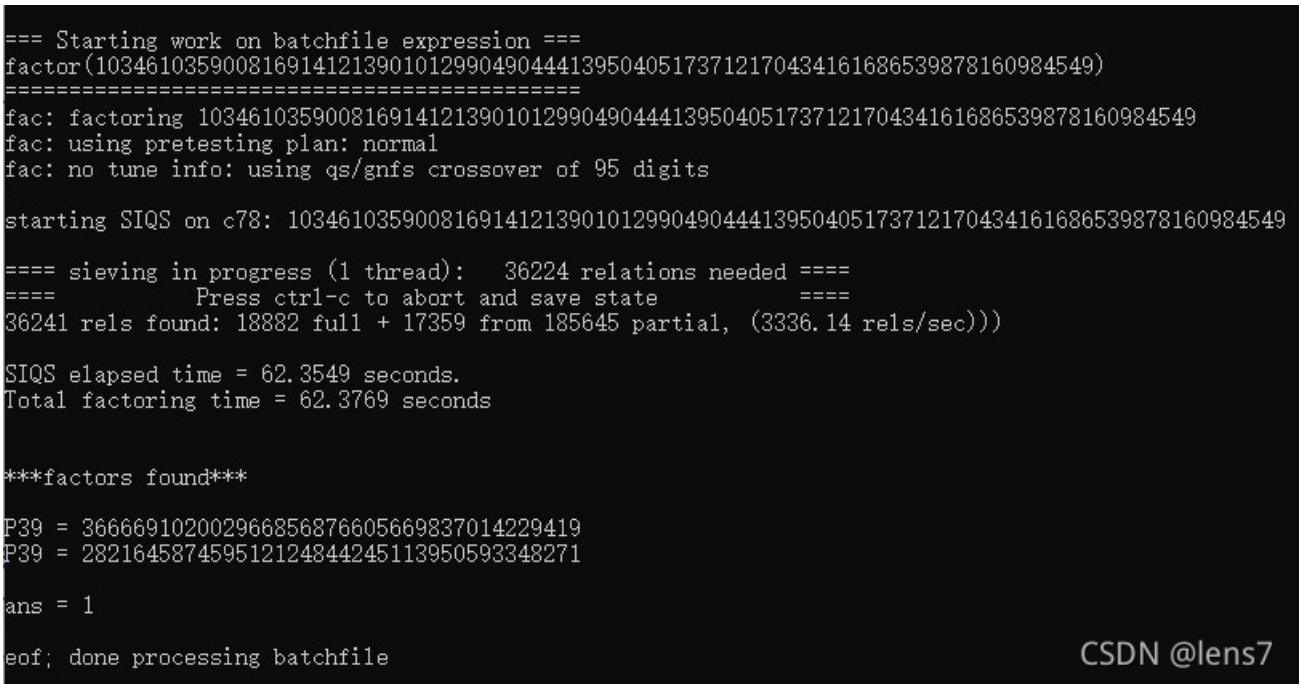

这里函数分析得到

N=10346103590081691412139010129904904441395040517371217043416168653987816098454 9

c = 0xad939ff59f6e70bcbfad406f2494993757eee98b91bc244184a377520d06fc35

e = 65537

第一步我们要求取的是pq两个素数的值 从其他大佬那里偷来神器RSAyafu 他可以讲给定的数分成两个相乘的素数



直呼nb

第二种使用方法：

如果因数过长，将 因数 用文本文件存放在 yafu 目录下，例如：data.txt 。文件最后一行一定要换行，否则 eof; done processing batchfile。

```
.\yafu-x64.exe "factor(@)" -batchfile data.txt
```

p = 366669102002966856876605669837014229419

q = 282164587459512124844245113950593348271

下面就是使用python解密了

（先得安装gmpy2库）

学习一下gmpy2库的使用

gmpy2.invert（a，c） 对a，求b，使得a*b=1（mod c）

gmpy2.powmod(a,n,p) 对于给定的整数p,n,a,计算a^n mod p

补充一下binascii.unhexlify函数

unhexlify(hexstr)将十六进制转换为二进制形式。

hex（）函数记得加[2:]去掉开头的0x。

求d脚本

```
import gmpy2
from Crypto.Util import number
e=65537
n=879243482641324068752761405144999371450508936656025929924181716470424916584 61
p=27512786035134892817328517438158115229 9
q=31957631681447894987059016419304804123 9
d=gmpy2.invert(e,(p-1)*(q-1))
print(d)
```

密文拿到手之后，需要密文的数字形式，用winhex打开查看十六进制形式

最后得到的密文的十六进制形式数字个数需要是偶数被，不是则尝试加0或者删除