安全-easy RSA（i春秋）

原创

小狐狸FM 于 2020-09-01 20:51:35 发布 695 收藏 4

分类专栏： 安全 # CTF夺旗 文章标签： python rsa ctf

本文链接：https://blog.csdn.net/smallfox233/article/details/108350376

安全 同时被 2 个专栏收录

91 篇文章 8 订阅

订阅专栏

CTF夺旗

38 篇文章 0 订阅

订阅专栏

文章目录

前言

> 解决这个题目需要使用到 `gmpy2` 和 `libnum(可选)` 库，后者直接使用 `pip install libnum` 下载即可，前者可参考下面的教程。

相关介绍

python中安装gmpy2

CTF笔记-RSA算法基础

一、题目

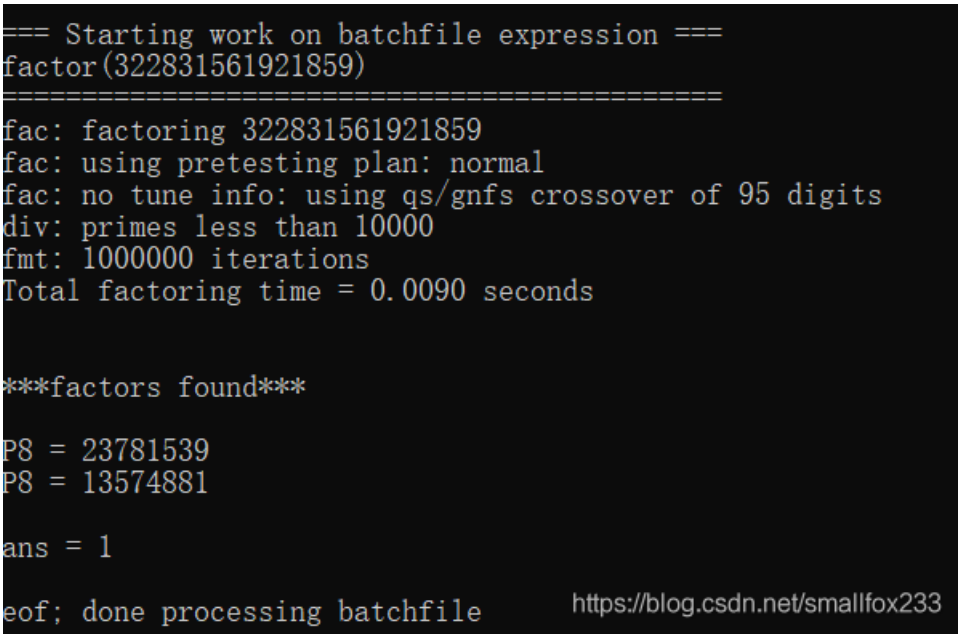
已知一段加密的信息为：0xdc2eeeb2782c，且已知加密所用的公钥：(N=322831561921859 e = 23)

请解密出明文，提交时请将数字转化成ascii码提交，比如你解出的明文是0x6162，请提交字符串ab

提交格式:PCTF{明文字符串}

二、解题方法

> 目前已知的是 `n、e和密文c` ，可以看出 `n` 的位数比较小。
> 所以我们可以先通过 `yafu` 将 `n` 进行分解，分解成 `p和q`



如果你没有 `yafu` 工具的话，也可以使用下面这个在线分解素数的网站
factordb

322831561921859 Factorize! (?)

Result:		
us (?)	digits	number
	15 (show)	322831561921859<15> = 13574881 · 23781539

下面是有关`RSA`算法的公式，已知`n、e、p、q和c`时我们就能先计算出`中间量d`，然后算出`明文m`

$$n = p*q$$

$$\varphi(n) = (p-1)*(q-1)$$

$$c = (m^e) \bmod n$$

$$d = 1 \bmod \varphi(n) / e$$

$$m = (c^d) \bmod n$$

然后用`python`来进行计算，因为代码比较简洁，就不进行解释了

```
import gmpy2
import libnum

n = 322831561921859

c = 0xdc2eeeb2782c

e = 23

p = 23781539

q = 13574881

d = int(gmpy2.invert(e , (p-1) * (q-1))) #计算中间量d

m = libnum.n2s(pow(c, d, n)) #求得明文m

print(m)
```