xctf攻防世界 CRYPTO高手进阶区 RSA_gcd

原创

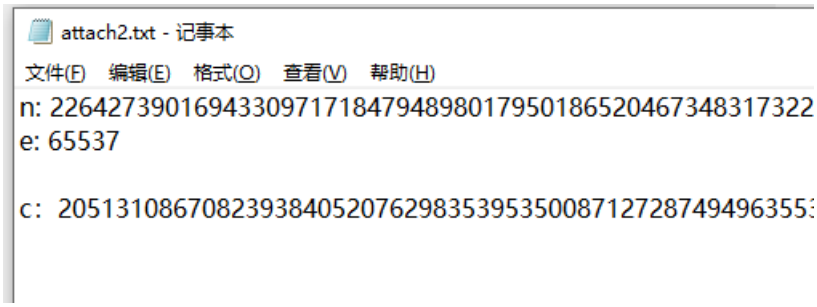
l8947943 已于 2022-02-17 11:33:15 修改 59 收藏 1

分类专栏： 攻防世界crypto之路 文章标签： crypto

于 2022-02-17 11:29:42 首次发布

0x01. 进入环境，下载附件

题目给的是两个txt文件，每个文件包含三个字符：n、e、c，如图：



经典的RSA问题

0x02. 问题分析

0x02_1. 解题思路

题目给的按时是RSA_gcd，猜想可能是素数分解问题，从而分部分进行加解密。

引用大佬的一个总结：https://blog.csdn.net/bunner_/article/details/109198214

> **给定两个不同的n的时候一定要看看n1，n2有没有最大公约数(素数)，如果有,那么该最大公约数就是两者共同的p**
>
> **给定两个相同的n的时候，且n非常小，那就要考虑共模攻击了**

那么本题的出发点则是，

- 利用两个文件的 n 和 n 求共同的最大公约数
- 如果是素数，那么它就是公共的 p
- 根据 RSA 解题过程即可完成

0x02_2. 解题代码

```python
import gmpy2, libnum
from Crypto.Util.number import long_to_bytes


# 用于求解a和b的最大公约数
def get_p(a, b):
    p = gmpy2.gcd(a, b)
    return p


if __name__ == '__main__':
    n1 = 23220619839642624127208804329329079289273497927351564011985292026254914394833691542552890810511751239656361686073628273309390314881604580204429708461587512500636158161303419916259271078173864800267063540526943181173708108324471815782985626723198144643256432774984884880698594364583949485749575467318173034467846143380574145455195152793742611717169602237969286580028662721065495380192815175057945420182742366791661416822623915523865907103876359351798762751470563960185272604884593330511327205589531429840386352237939926516377081504949647854750654045688440399833814039093413020987735333250809100578455738989843142460896c1 = 97006147484135032912609662318635621175020962846162167074452763552748690866197965276184732134225099968434302965265941135726758405593450773444190989008187095776423249004055824996836047869811440998780217845675406540408339120631417099136534163948887662814652006828523787944788013292512248010068209258585072731305042365638221208385207462702807311214428394122583971919630360405535396978465350388415412090505030610010709097258065742300902460418914865069809392942455372526109447995739208442352210969563910957161116299985940757625073454309455234927759157908280780045370532078348674473499421302847644692281587005331197384496c2 = 22642739016943309717184794898017950186520467348317322177556419830195164079827782890660385734113396507640392461790899249329899658620250506845740531699023854206947331021605746078358967885852989786535093914459120629747240179425838485974008209140597947135295304382318570454491064938082423309363452665886141604328435366646426917928023608108047038219675329265682851368156207746884610512281208476525779907075440563814950810746323363335046213875175891303637316966882888821332342965634481201448096291608869591017776383939395473073231222410071843114613354889703106055400559293034722652656193992266085504702658129257148796092991c2 = 20513108670823938405207629835395350087127287494963553421797351726233221750526355985253069487753150978011340115173042210284965521215128799369083065796356395285905154260709263197195828765397189267866348946188652752076472172155755940282615212228370367042435203584159326078238921502151083768908742480756781277358357734545694917591921150127540286087770229112383605858821811640935475859936319249757754722093551370392083736485637225052738864742947137890363135709796410008845576985297696922681043649916650599349320818901512835007050425460872675857974069927846620905981374869166202896905600343223640296138423898703137236463508e = 65537

# 拿到共同的p，再分别求出各自的素数q
p = get_p(n1, n2)
q1 = n1 // p
q2 = n2 // p

# 求私钥d，另外gmpy2.invert()函数和libnum.invmod()函数等价
d1 = libnum.invmod(e, (p - 1) * (q1 - 1))
d2 = libnum.invmod(e, (p - 1) * (q2 - 1))

# 将密文c用私钥d解密成明文
m1 = gmpy2.powmod(c1, d1, n1)
m2 = gmpy2.powmod(c2, d2, n2)

# 将明文转字符
message = long_to_bytes(m1) + long_to_bytes(m2)
print(message.decode())
```

最终的答案为：`flag{336BB5172ADE227FE68BAA44FDA73F3B}`