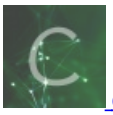
第一届“长城杯”网络安全大赛 Crypto 复现

原创

路由(_)生 于 2021-09-20 16:25:09 发布 105 收藏

分类专栏： crypto 文章标签： python

本文链接：https://blog.csdn.net/qq_52193383/article/details/120392517



baby_rsa

题目代码：

```python
#!/usr/bin/env python3

from Crypto.Util.number import *
from secret import flag, v1, v2, m1, m2


def enc_1(val):
    p, q = pow(v1, (m1+1))-pow((v1+1), m1), pow(v2, (m2+1))-pow((v2+1), m2)
    assert isPrime(p) and isPrime(q) and (p*q).bit_length() == 2048 and q < p < q << 3
    return pow(val, 0x10001, p*q)


def enc_2(val):
    assert val.bit_length() < 512
    while True:
        fac = [getPrime(512) for i in range(3)]
        if isPrime(((fac[0]+fac[1]+fac[2]) << 1) - 1):
            n = fac[0]*fac[1]*fac[2]*(((fac[0]+fac[1]+fac[2]) << 1) - 1)
            break
    c = pow(val, 0x10001, n)
    return (c, n, ((fac[0]+fac[1]+fac[2]) << 1) - 1)


if __name__ == "__main__":
    assert flag[:5] == b'flag{'
    plain1 = bytes_to_long(flag[:21])
    plain2 = bytes_to_long(flag[21:])
    print(enc_1(plain1))
    print(enc_2(plain2))

'''
158087739211657463782246495540327740951985317824559041695522233035139409682928968141592884174992207398758337545739436070478552567399761615985999039329811699795098715919999648568069295978059041340999018268583677783863423767685080315548022490750723667100388893062688067441790866486847380230734589829060669723404143989284111479705939352440779254487327724736197830793283515222691708798070641113188710742910735813430393895611753910397669363762678751845816433359160494617847533411152275151635457094547462725148270006018537353565514956852299956374835067354489006568853653534343086394120350031195166933033770815769755409483110(406259810172502629452305484507389517255665202521634101245656221267547396936812716491271041090381648527877672964036974624754596705408458221503976399230132231029126747484024275015880188664908783946784820615615212533655500290755655079882327290320552989927927125745697048460755146248246541276917439441120757038140436225995304961007133786967618799825426799176315704510721078933487928173216525934717949742271834767329806238354839910670803451849784821913424306274903985169127144519841529603488995895327519192725830987641181610560785367813417501425531970829250707301780925613144005181510199551049897909114603578483660162630830, 430017260469550789813440169817904459801990720660193233820682441428889315396028123180230952564749396972578026461503485467796475451522881586075552393028876891376457486284212476852254633461180812387180497013207262954353767332156814157742552584194186614660104039285912429614341787308465374712361426835173991094664297763773601183551734310161075439772413580640931027418196261634671398333524540944722293495984793583672034524526068337964831118920763437459583949321321994427180487206335563104670192224346937854239966563066122627146090761196348147834381118437736495191011693260727935960275940579883651330370411335661468978682690, 397962725923318964006267849517132395268572731687321330466675723996226603305878815793193140945570115548518730683890166290859630861361164253525359025983787739)
'''
```

比赛时思路：

由于enc_1已经被队长解完了，剩下的enc_2就留给我们写。故这里只写我当时解enc_2的思路。

从代码中可以看出n是由三个未知素数构成的，解密的话我们只需要求出φ(n)就可了。

令fac[0] = f0 , fac[1] = f1 , fac[2] = f2 , f3 = 2 * (f0 + f1 + f2) - 1 ，

则n = f0 * f1 * f2 * f3，φ(n) = (f0 - 1) * (f1 - 1) * (f2 - 1) * (f3 - 1) (f0,f1,f2,f3均为素数)

根据给出的数据，我们可以求出a = f0 * f1 * f2 , b = f0 + f1 + f2 (奇怪的是n = f0 * f1 * f2 * f3，但却不能被f3整除，当时也确实有疑惑，但就姑且这么算吧)

之后就不断尝试将φ(n)化为只含a,b的式子。

然后，然后就被队友写出来了，然后我就躺好了，队友带飞。

正确的解题思路：

- enc_1

参考：[ONE LINE CRYPTO](CryptoCTF 2020 | CryptoHack Blog)

有以下式子和条件：

{

```
import gmpy2
from Crypto.Util.number import *
#enc_1
c1 = 158087739211657463782246495540327740951985317824559041695522233035139409682928968141592884174992207398758337
545739436070478552567399761615985999039329811699795098715919999648568069295978059041340999018268583677783863423767
676850803155480224907507236671003888930626880674417908664868473802307345898290606697234041439892841114797059393524
407792544873277247361978307932835152226917087980706411131887107429107358134303938956117539103976693637626787518
458164333591604946178475334111522751516354570945474627251482700060185373535655149568522999563748350673544890065
688536535343430863941203500311951669330337708157697554094831 1
e1 = 65537

primes = []
for x in range(500):
    for y in range(500):
        num = pow(x, (y+1))-pow((x+1), y)
        if len(bin(num)[2:]) < 2048:
            if isPrime(num):
                primes.append(num)

for p in primes:
    for q in primes:
        n1 = p*q
        if len(bin(n1)[2:]) == 2048:
            try:
                d1 = gmpy2.invert(e1,(p-1)*(q-1))
                flag = long_to_bytes(pow(c1,d1,n1))
                if b'flag' in flag or b'ctf' in flag:
                    print(flag)
            except:
                pass
#b'flag{8102c552-3d78-4a'
```

enc_2

看了队友的wp之后恍然大悟！！！

令a = f0 * f1 * f2 , b = 2 * (f0 + f1 + f2) - 1，n = a * b

$c \equiv m_i \pmod{n}$

```
#enc_2
c2 = 406259810172502629452305484507389517255665202521634101245656221267547396936812716491271041090381648527877672
964036974624754596705408458221503976399230132231029126747484024275015880188664908783946784820615615212533655500 2
907556550798823272903205529899279271257456970484607551462482465412769174394411207570381404362259953049610071337 8
696761879982542679917631570451072107893348792817321652593471794974227183476732980623835483991067080345184978482 1
913424306274903985169127144519841529603488995895327519192725830987641181610560785367813417501425531970829250707 3
017809256131440051815101995510498979091146035784836601626308 3
n2 = 430017260469550789813440169817904459801990720660193233820682441428889315396028123180230952564749396972578026
461503485467796475451522881586075552393028876891376457486284212476852254633461180812387180497013207262954353767 3
321568141577425525841941866146601040392859124296143417873084653747123614268351739910946642977637736011835517343 1
016107543977241358064093102741819626163467139833352454094472229349598479358367203452452606833796483111892076343 7
459583949321321994427180487206335563104670192224346937854239966563066122627146090761196348147834381118437736495 1
910116932607279359602759405798836513303704113356614689786826 9
#n = 191*193*627383*185935563905698933884772988621508432001355056043924161005737146444366816645366450086517424766 0
#20344620813162196668438718912147293226345015542898206139511104467995075381641428294263601989775020994993771788 14
#13823952091423640659878296010412658344713439244770180457400325541493186854703180053691731596197238201363111053 52
#85012451603572537743754115268578278428250198592918006225382007248578357398887537320128561126351840603314417475 42
#16784310597632784542393540282286489459684674546578134256526442793095456260961132793907776956827087675433526964 02
#77726607195276868129620895284648898858733965899439317638568413061
e2 = 65537
b = 39796272592331896400626784951713239526857273168732133046667572399622660330587881579319314094557011554851873 06
838901662908596308613611642535253590259837873 9
#pd = 191*193*627383*172075473847731712775868228546503193989105983587397515755503132707011112362878983329943354966
#96193251606797193553381878777583114857851974927104 91
#a = n2/b#不是整数！
x1 = 191
x2 = 193
x3 = 627383
x4 = 172075473847731712775868228546503193989105983587397515755503132707011112362878983329943354966961932516067971
935533818787775831148578519749271049 1
phi2 = (x1-1)*(x2-1)*(x3-1)*(x4-1)
d2 = gmpy2.invert(e2,phi2)
m2 = pow(c2,d2,b)
print(long_to_bytes(m2))
#b'42-b659-0c96ef827f05}'
```