美团杯MEITUAN网络安全大赛CTF2021-humburgerRSA题解

原创

rickliuxiao 于 2021-12-12 01:05:01 发布 199 收藏 1
分类专栏： Crypto 文章标签： web安全 安全

题目如下：

output.txt

```
n = 177269125756508652546242326065138402971542751112423326033880862868822164234452280738170245589798474033O
c = 477180226013245433990783959570950837532016313328089494069270915890448375564693008077284840355814479609S
```

task.py

```
from Crypto.Util.number import *

flag = open('flag.txt').read()
nbit = 64

while True:
    p, q = getPrime(nbit), getPrime(nbit)
    PP = int(str(p) + str(p) + str(q) + str(q))
    QQ = int(str(q) + str(q) + str(p) + str(p))
    if isPrime(PP) and isPrime(QQ):
        break

n = PP * QQ
m = bytes_to_long(flag.encode())
c = pow(m, 65537, n)
print('n =', n)
print('c =', c)
```

先看一下PP和QQ的组成：

```
PP = int(str(p) + str(p) + str(q) + str(q))
```

即字符串相连，再转换成int数字。那么，X = int(str(p) + str(p)) 就是 ：

```
X == 10^(p的位数)*p + p
```

推理分析思路如下：

```
令：
x, y = len(str(p)), len(str(q))
P = str(p) + str(p)
Q = str(q) + str(q)

P = 10**(x)*p + p    # (1)
Q = 10**(y)*q + q    # (2)

再令：
xx, yy = len(str(P)), len(str(Q))
PP = P+Q = 10**(xx)*P + Q
QQ = Q+P = 10**(yy)*Q + P

将(1)、(2)加入进上面的方程中，得到：
N = PP*QQ = (10**(xx)*P + Q)*(10**(yy)*Q + P)
  = 10**(xx+yy)*P*Q + 10**(xx)*P*P + 10**(yy)*Q*Q +P*Q
  = 10**(xx+yy)*(10**(x)*p + p)*(10**(y)*q + q)
   + 10**(xx)*(10**(x)*p + p)*(10**(x)*p + p)
   + 10**(yy)*(10**(y)*q + q)*(10**(y)*q + q)
   + (10**(x)*p + p)*(10**(y)*q + q)
  = 10**(xx+yy+x+y)*p*q + ... + p*q

由于 xx+yy+x+y 足够大，而 str(N)[:?] 就是 str(p*q)[:?]，并且两者相等，而且 str(N)[?:] 与 str(p*q)[?:] 两者相等。
```

我们自己用题目给的脚本，跑一遍，得到一组 `p*q`和 `n=PP*QQ` 的数据。

```
from Crypto.Util.number import *

nbit = 64

for i in range(10):
    while(True):
        p, q = getPrime(nbit), getPrime(nbit)
        P = int(str(p) + str(p))
        Q = int(str(q) + str(q))
        PP = int(str(P) + str(Q))
        QQ = int(str(Q) + str(P))
        if isPrime(PP) and isPrime(QQ):
            n = PP * QQ
            print(n,p*q)
# 172552610852624337784035949632908517355734035684070753814679795210135425973527923366032328492820431356488
```

从 n=172552610852624337784035949632908517355734035684070753814679795210135425973527923366032328492820431356488, p*q=172552610852624337765055162439119840201中，可以发现

```
str(N)[:19]==str(p*q)[:19]
str(N)[-19:]==str(p*q)[-19:]
且len(p*q)=39，为39位。
```

那么，已知前19位和后19位，总共位数是39，只需要爆破1位即可。

sage 脚本 exp:

```
from Crypto.Util.number import *
from tqdm import tqdm

# 常规解题RSA
def decrypt_RSA(c, e, p, q):
    phi = (p-1) * (q-1)
    d = inverse(e, phi)
    m = pow(c, d, p*q)
    print(long_to_bytes(m))

# 题目中的数据
n = 177269125756508652546242326065138402971542751112423326033880862868822164234452280738170245589798474033
c = 477180226013245433990783959570950837532016313328089494069270915890448375564693008077284840355814479609


low = str(n)[-19:] # 低19位
high = str(n)[:19] # 高19位
pq_prob = []
p = 0
q = 0

# 爆破1位，得到p*q
for i in range(10):
    pq_prob.append(int(high + str(i) + low))

for x in tqdm(pq_prob): # 对p*q的值进行因子分解
    f = factor(x)
    if (len(f) == 2 and f[0][0].nbits() == 64): # 分解成功
        p, q = f[0][0], f[1][0]
        print(p,q)
        break

print(p,q)
P = int(str(p) + str(p))
Q = int(str(q) + str(q))
PP = int(str(P) + str(Q))
QQ = int(str(Q) + str(P))
N = PP * QQ
print(N == n)
decrypt_RSA(c, 65537, PP, QQ)
```