第四届美团网络安全高校挑战赛_hamburgerRSA

原创

M3ng@L 于 2021-12-13 15:22:37 发布 112 收藏 1
文章标签： 密码学 python

本文链接：https://blog.csdn.net/qq_51999772/article/details/121906311

hamburgerRSA

题目描述

```
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)
```

程序分析

p,q是随机选取64位二进制的素数；

关键是生成的公私钥使用的是PP,QQ；

而PP,QQ是通过简单的字符串p,q连接的形式组成

本题的关键就是如何去表示PP,QQ

由于是采用字符串相连接的形式，我们可以使用$10 * p+$ 这样的形式来表示

那么

设$x=len(str(p))$；$y=len(str(q))$

$PP=10^{x+y}*p+10^{y}*p+q$

$QQ=10^{x+y}*q+10^{y}*q+p$

两者相乘表示为n

$n=PP*QQ$

中间的省略号部分是因为与之后的推导没有关系（也可以说是由于加法前后两项重合了数位无法简单表示），所以省略不考虑了

通过观察，已知的n实际上是由$p*$ 的10的k次方组成的

到这里我们发现n的某一部分数位其实就是
$p*$ 的数位（比如15600000，它的数位是6-8），但是由于加法前后两项之间有重合，解释一下

> 比如10 $*$ 和 假设 的最高位为
> 那么前者的数位覆盖了117-157；后者的数位覆盖了97-137
> 这里就有重叠的地方了，由于重叠，加法运算之后，我们就没有办法直接用n的这一部分直接等于$p*$ 的某一部分了

那么为了进一步表示n，我们需要知道x,y的确切大小

---

x,y的确切大小

通过n的表达式，我们可以知道n的最高位（10进制）等于10 $*$ $p*$ 的位数（10进制）

这里我做了一下测试：

```
while True:
    p = getPrime(64)
    q = getPrime(64)
    temp = p*q
    print(len(str(p)),end=" ")
    print(len(str(q)),end=" ")
    print(len(str(temp)))
```

可以观察到p,q的十进制位数是19或者20，也就是说

而$p*$ 的十进制位数为39或者38

n的十进制位数为156

剩下的就是一个加法运算，如果

那么3*

显然不满足条件；

如果 ：

那么3*

也不满足条件

如果 （这里 的大小可以互调的）

那么3*

综上，

$p*q$的确切大小

根据之前重叠的说法，我们只需要看在n中没有重叠的部分，没有重叠的部分是直接等于对应的$p*$ 的对应数位的；

n的高位：

与10 *~. ，10 * . 10 * 有关

数位分别覆盖117-156；97-136；98-137

那么没有重叠的部分只有前19位的高位（实际上我认为还需要考虑加法之后的进位问题，也就是说可能只有18位的高位与$n*$ 相同，但是幸运的是在这道题关键位数没有进位，之后就可以少爆破两位）

n的低位：

与10 * . 10 有关

数位分别覆盖19-58；20-59；0-39

没有覆盖的部分只有后18位的低位

总结起来：

已知$n*$ 的高19位和低18位

而$n*$ 的总位数为39

剩下两位需要爆破一下

由于$n*$ 不大，可以直接用sagemath的$factor()$分解，判断条件是

```
if len(factor(pq)) == 2 and factor(pq)[0][0].nbits() == 64:
```

得到p和q之后计算得到PP,QQ就进行正常的RSA求密

代码实现

```
# sage
nbit = 64
n = 17726912575650865254624232606513840297154275111242332603388086286882216423445228073817024558979847403304746092055255001896857126797828375674272223192245119 3
high = str(n)[:19]
low = str(n)[-18:]
for i in range(10):
    for j in range(10):
        pq = int(high + str(i) + str(j) + low)
        f = factor(pq)
        if len(f) == 2 and f[0][0].nbits() == 64:
            p = f[0][0]
            q = f[1][0]
            print(p,q)

from Crypto.Util.number import *
import gmpy2
p,q = 9788542938580474429,18109858317913867117
c = 47718022601324543399078395957095083753201631332808949406927091589044837556469300807728484035581447960954603540348152501053100067139486887367207461593404096
e = 65537
PP = int(str(p) + str(p) + str(q) + str(q))
QQ = int(str(q) + str(q) + str(p) + str(p))
fai_n = (PP-1)*(QQ-1)
d = gmpy2.invert(e,fai_n)
m = pow(c,d,PP*QQ)
print(long_to_bytes(m))
```

参考文章（wp）

第四届美团网络安全高校挑战赛Write up (qq.com)

然后这道是类似的题目：

CTFtime.org / Crypto CTF 2021 / Hamul / Writeup