# RSA 加密中选用一个大的素数 $p$ 作为模数的后果

金柏琲 2022213779

2023 年 10 月 11 日

RSA 如果选用一个大素数作为模数，即公钥和中解密时用的模数不再是原先素数 $p$ 和素数 $q$ 的乘积，而是素数 $p$，会有什么后果呢?

## 1 RSA 加密的基本原理和方法:

### 1.1 生成钥匙

RSA 加密是要生成一个公钥和一个私钥的过程如下:

1. 找两个大素数 $p$ 和 $q$，并且得到他们的乘积 $N$:

$$N = p \times q \tag{1}$$

2. 通过欧拉函数，求 $r$:

$$r = \varphi(N) = \varphi(p) \times \varphi(q) = (p-1) \times (q-1) \tag{2}$$

3. 选择一个小于 $r$ 的整数 $e$，使得 $e$ 和 $r$ 互质。并解出 $e$ 关于 $r$ 的模反元素，命名为 $d$:

$$ed \equiv 1 \pmod{r} \tag{3}$$

4. 将 $p$ 和 $q$ 直接销毁，得到公钥和私钥 ($N$,$e$)，($N$,$d$)。

### 1.2 加密和解密

在获得公钥和私钥之后，把私钥给解密者，公钥给加密者。

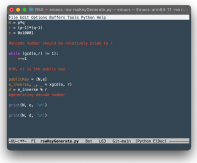
1. 加密过程（加密者通过公钥把信息 n 加密成信息 c):

$$c = n^e \mod N \tag{4}$$

2. 解密过程（加密者把信息 c 送到解密者手中之后的过程):

$$n = c^d \mod N \tag{5}$$

解码的正确性是由欧拉定理和费马小定理支撑的。如下是使用 python 实现 RSA 加密的钥匙生成过程和信息检验过程。



(a) 生成公钥和私钥

(b) 使用生成的钥匙加密 '1234'

(c) 检验正确性

## 2 讨论 RSA 的安全性

### 2.1 为何要选择一个大素数 $N$ 作为最后的模数?

对于一个想要破解私钥的人，手中知道的仅仅是公钥和 RSA 私钥的计算方法，公钥由 $N$ 和 $e$ 组成，为了破解私钥，他必须要将 N 分解成两个素数

的乘积才能得到 $p$ 和 $q$ 从而进一步通过 RSA 的已知算法来得到私钥中的 $d$。

而 RSA 加密方法的安全性就是建立在对大数 $N$ 进行素数分解的困难性之上的，而且 $N$ 越大分解的难度会越难，在破解者没有极强的计算设备的情况下，即便现在已经有了很多优秀的算法，但只要 $N$ 的大小足够大，就基本可以保证对于他 RSA 加密是十分安全的。

## 2.2 不应该使用单独素数 $p$ 作为最后模数的原因：

在钥匙生成的过程中步骤不变，仅仅把最后公布的钥匙中的 $N$ 替换成素数 $p$，首先这样做并不会改变加密解密的正确性，以下进行数学的证明：

新的加密过程：

$$n = c^d \mod p \tag{6}$$

以上是模数是 $p$ 加密的过程，从而可以得到如下的同余方程:

$$c^d \equiv n^{ed} \pmod p \tag{7}$$

再通过已知的 RSA 计算方法中 $ed$ 的同余方程，可以作出以下的推导：

$$ed \equiv 1 \pmod r \tag{8}$$

$$ed = 1 + h\varphi(N) \tag{9}$$

$$n^{ed} = n^{1+h\varphi(N)} = nn^{h\varphi(N)} = n(n^{\varphi(N)})^h \tag{10}$$

若 $n$ 与 $N$ 互质，$N$ 又一定是 $p$ 的整数倍，则由欧拉定理得：

$$n^{ed} \equiv n(n^{\varphi(N)})^h \equiv n(1)^h \equiv n \pmod p \tag{11}$$

若 $n$ 与 $N$ 不互质，则来考虑 $n$=$ph$，以及 $ed$-1=$k$($q$-1)，可以得到：

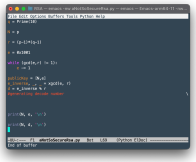
$$n^e d = (ph)^{ed} \equiv 0 \equiv ph \equiv n \pmod p \tag{12}$$

（实际上略作修改也就完成了对原来以 $N$ 为模数的 RSA 加密方法中解密正确性的数学证明）

可见，改变这一点并不会影响正确性，但是却会大大影响安全性。不同于采用大 $N$，对于破解者来说，这种方法之下，他没有必要对一个大数进行质因数分解了，他可以直接通过 $p$ 和 $e$ 这个数来找到满足条件的 $r$，这大大减少了破解者的计算量，$r$ 应该满足几个数学上显然可以得到的结论：

1. 它一定是 $p$-1 的整数倍
2. 一定有 gcd($r$,$e$)==1 成立

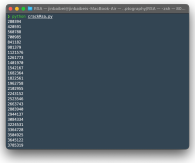
以下是使用 python 暴力破解的实践实验：



(d) 以 $p$ 为模数的生成算法

(e) 生成结果

(f) 暴力破解算法



(g) 破解结果

最终的结果中有满足条件的 $r$，实践证明了改变模数为 $p$ 的不安全性。

# 3 总结

通过数学理论推导和 python 编程实验，足够验证在 RSA 加密过程中，如果把 $N$ 换成 $p$，虽然不会影响加密解密的正确性，但确实会影响其安全性，因此在生成钥匙时选取 $p$ 作为模数是不合适的。（注：实验代码网址是 https://github.com/Random9-BB/cryptography/tree/main/RSA）