RSA 加密算法
对信息加密的方式有两种：对称加密和非对称加密

直到上世纪 70 年代人们还在使用对称加密算法，也就是说人们通过事先商定好的密钥对数据经行加密和解密，但这种加密方式有很多缺陷：
1.当用户过多时人们往往需要记住很多密钥，负担过重。
2.人们只能通过线下交换密钥以确保安全性，成本过高。

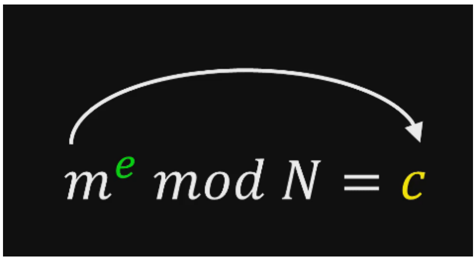
非对称加密很好地解决了上面的问题，其中最有影响力的即为 RSA 非对称加密算法。

RSA 是 1977 年由罗纳德·李维斯特（Ron **R**ivest）、阿迪·萨莫尔（Adi **S**hamir）和伦纳德·阿德曼（Leonard **A**dleman）一起提出的，当时他们三人都在麻省理工学院工作。RSA 就是他们三人姓氏开头字母拼在一起组成的。

RSA 算法会生成两套密钥，一套公钥，一套私钥，两者具有数学关联，其中公钥是对所有人公开的信息，用于对信息进行加密，对应的私钥仅接收者持有，用于对信息进行解密。
由于公钥是公开的，为防止被加密的信息被他人轻易地反推出来，RSA 算法采用了单项函数——模运算（Modular Arithmetic）用于满足正向加密容易，逆向解密难的需求。

为什么模运算逆向解密困难呢?
例如：求$3^3 \bmod 7$ 很容易，答案是 6，但如何计算 3 的多少次方对 7 取余等于 6 呢?
由于求余运算并不可逆，所以只能一个一个地去尝试，但当底数足够大时那么一个一个地去尝试就很不现实了，RSA 加密正是利用了这个特性。

RSA 加密过程：
假设需要加密的原始数据为 m,我们对它求 e 次幂：



这里的{e,n}就组成了公钥，c 就是加密后的密文

$$m^{e} \bmod N = c$$

由于采用了模运算，如上文所述逆向反推出原始数据很困难。

RSA 解密过程：

$$c^{d} \bmod N = m$$

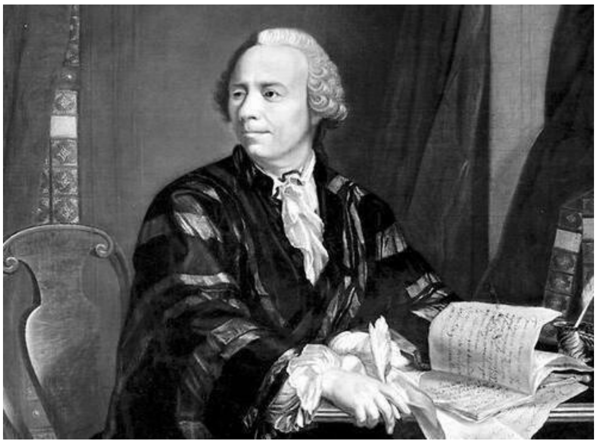
这里的{d,n}即组成了私钥，通过私钥我们可以将密文解密，得到原始数据。

为了方便理解，我们将上面两个公式进行整理得到如下公式：

$$m^{\boxed{ed}} \bmod N = m$$

由上式可知，如何选择 e,d 便成为了公钥加密的关键问题。

由此我们不得不提到欧拉在 1763 年的一个重要发现——欧拉定理



欧拉定理
Euler's Theorem

$$m^{\varphi(n)} \equiv 1 \pmod{n}$$

该定理表示任意一个与 n 互质的正整数 m，取它的ϕ（n）次方，并除以 n 取余数，其结果永远等于 1。这里的ϕ（n）就是欧拉函数。它代表在小于或等于 n 的正整数中有多少个与 n1 互质的数。

RSA 非对称加密算法是目前最有效的安全算法之一，其安全性依赖于大数分解，即利用了数论领域的一个事实，那就是虽然把两个大质数相乘生成一个合数是一件十分容易的事情，但要把一个合数分解为两个质数却十分困难。合数分解问题目前仍然是数学领域尚未解决的一大难题，至今没有任何高效的分解方法。所以，只要 RSA 采用足够大的整数，因子分解越困难，密码就越难以破译，加密强度就越高。

RSA 加密算法的工作原理如下。

① 任意选择两个不同的大质数 p、g，计算 N=p*g（N 称为 RSA 算法中的模数）。

② 计算 N 的欧拉函数 φ（N）=（p-1）（g-1），φ（N）定义为小于 N 并与 N 互质的数的个数。

③ 从【0，（N）-1】中选择一个与 φ（N）互质的数 e 作为公开的加密指数。

④计算解密指数 d，使 ed=1modφ（V。其中，公钥 PK={，M}，私钥 SK={dM。

⑤公开 e、N，但对 d 保密。

⑥将明文 X（假设 X 是一个小于 N 的整数）加密为密文 Y，计算方法为

Y=X″ mod N

⑦将密文 γ（假设 γ 也是一个小于 N 的整数）解密为明文 X，计算方法为

X=F mod N 227

值得注意的是：e、d、X 满足一定的关系，但破译者只根据 e 和 N（不是 p 和 q）计算出 d

数字签名者：YH
日期：2021.06.15 12:01:36 +08'00'