在网络的世界，RSA算法可以说是最重要的算法了。

一切要从加密算法说起，传统的加密方式都是对称加密的。所谓对称，指的是加密用的密钥和解密用的密钥是同一个。对称加密需要双方持有相同的密钥，而如何安全的传输密钥是必须要解决的问题。

非对称加密的思想是，加解密使用不同的密钥或者说策略，而这两个密钥存在一定的对应关系。这样就避免了解密密钥的传输，从而更加安全。详细的说，非对称加密的主要包括以下过程：

> 1、乙方生成两把密钥（公钥和私钥）。公钥是公开的，私钥是保密的
>
> 2、 甲方获取乙方的公钥，然后使用其对信息加密
>
> 3、 乙方获得加密后的信息后，用私钥解密

因此，只要能够保证公钥加密的内容只能被私钥揭开，而且无法通过公钥推断出私钥，则这种方式是安全的。

在讲解RSA算法之前，我们需要复习一些数学知识。

## 互质关系

如果两个正整数，除了1以外没有其他的公因子，则两个数互质。比如5和7互，13和12互质。关于互质关系我们有许多推论，而和我们RSA算法相关的主要有这样几点：

- 如果$a$为质数，$b$为小于$a$的数，则$a$与$b$互质
- 如果$a$为质数，而$b$为大于$a$的数，且$b$不能被$a$整除，则$b$与$a$互质
- 如果$a$是大于1的整数，则$a$与$a-1$互质。

这几点都比较好理解。

## 欧拉函数

对于给定正整数$n$, 欧拉函数的值为小于等于$n$的正整数中，与$n$互质的数的个数，表示为$\phi(n)$;

对于质数$p$，与所有小于它的正整数都互质，所以$\phi(p)=p-1$。

对于数$n$，如果$n=pq$, 而$p$和$q$都为质数，则$\phi(n)=\phi(p)\phi(q)$，这个等式的证明可以学习数论。

## 欧拉定理

如果两个正整数$a$和$n$互质，则有以下等式成立

$a^{\phi(n)} \equiv 1(mod\ n)$

等式的意思是，$a$的$n$的欧拉函数次方除以$n$余1

欧拉函数可以大大简化某些运算，比如7和10互质，根据欧拉定理，

$7^{\phi(10)} \equiv 1(mod\, 10)$

已知$\phi(10)$ 值为4，所以可以有以有下式

$7^{4k} \equiv 1(mod\, 10)$

也就是说，7的4倍次方的个位数肯定为1.

### 费马小定理

如果$a$与质数$p$互质，则

$a^{p-1} \equiv 1(mod\, p)$

可以发现，费马小定理是欧拉定理的特例。因为当$p$为质数时，其欧拉函数的值为$p-1$。

### 模反元素

如果两个正整数$a$和$n$互质，那么一定可以找到数$b$，使得以下等式成立

$ab \equiv 1(mod\, n)$

这可以通过欧拉定理得出，即$b = a^{\phi(n)-1}$

OK，以上是RSA需要的数学原理，下面我们介绍算法的过程，并证明算法是正确的，而且是安全的。

### RSA算法步骤

> - 随机选择两个不相等的质数$p$和$q$
> - 计算$p$和$q$的乘积$n$
> - 计算$n$的欧拉函数$\phi(n) = (p-1)(q-1)$
> - 随机选择一个整数$e$，但是要保证$1 < e < \phi(n)$ 且$e$与$\phi(n)$互质
> - 计算$e$对于$\phi(n)$ 的模反元素$d$
> - 将$n$和$e$封装成公钥，$n$和$d$封装成私钥

第四步随机数的选择其实很简单，只要选取一个质数，而这个质数又不是$\phi(n)$的因数即可。

第五步，计算模反元素d的过程如下，对于

$ed \equiv 1(mod\, \phi(n))$

其等价于

$ed - 1 = k\phi(n)$

其中$d$与$k$为未知数，可以在这条直线上任意选取一个点，作为方程的解。

### RSA 算法可靠性

我们知道，$n$和$d$封装为私钥，而$n$是公钥的一部分，所以如果能够根据$n$和$e$推导出$d$，则私钥泄漏，算法不安全。

我们看怎么才能推导出$d$

由$ed \equiv 1(mod\ \phi(n))$ 可知，只有知道$e$和$\phi(n)$，才能算出$d$。

因为$e$是已知的，所以关键点是算出$n$的欧拉函数$\phi(n)$。

除了暴力破解$\phi(n)$, 我们还有一个捷径， 即$\phi(n) =（p-1）(q-1)$;

因为 $n=pq$, 我们可以通过因式分解数$n$得到$pq$，进而得到结果

然而，对于极大的数做因式分解是一件非常困难的事情。维基百科写到

> 對极大整数做因数分解的难度决定了RSA算法的可靠性。換言之，對一极大整数做因数分解愈困难，RSA算法愈可靠。假如有人找到一种快速因数分解的算法的话，那么用RSA加密的信息的可靠性就肯定会极度下降。但找到这样的算法的可能性是非常小的。今天只有短的RSA钥匙才可能被强力方式解破。到目前为止，世界上还没有任何可靠的攻击RSA算法的方式。只要其钥匙的长度足够长，用RSA加密的信息实际上是不能被解破的

## 加密和解密

加密： 对于给定的数值$m$，我们保证$m<n$($n$的值一般很大，而我们的$m$的值可以取两个字节作为数值)所谓加密就时计算出下式中的$c$

$$m^e \equiv c(mod\ n)$$

也即是对$m$求$e$次方，除以$n$得到余数$c$。

解密： 解密过程为计算下式的m

$$c^d \equiv m(mod\ n)$$

也就是说拿着密文$c$，取$d$次方，除以$n$得到余数$m$

## 算法证明

证明过程即通过加密的等式，能够推导出解密等式成立。

对于

$$c^d \equiv m(mod\ n)$$

我们已知

$$m^e \equiv c(mod\ n)$$

我们可以得到

$$c = m^e - kn$$

将此带入第一个式，我们有

$(m^e - kn)^d \equiv m(mod\, n)$

对等式右边展开后除了第一项，其他项都存在n，则我们有

$m^e d \equiv m(mod\, n)$

因此我们就是要证明上式成立。

由RSA算法步骤一节我们有，

$ed - 1 = k\phi(n)$

上式可以写成

$ed - 1 = h（p - 1）= j(q - 1)$

我们要证明

$m^{ed} \equiv m(mod\, pq)$

我们可以分别证明

$m^{ed} \equiv m(mod\, q)$与$m^{ed} \equiv m(mod\, p)$

（这里是中国余数定理的一部分，但也可以这样思考，$m^e d - m$ 是$q$的倍数，也是$p$的倍数，则其必然是$pq$的倍数）

对于$m^{ed} \equiv m(mod\, q)$，证明过程如下

> 1. 如果$m \equiv 0(mod\, q)$，则m为q的倍数，则$m^{ed} \equiv m(mod\, q)$也成立
> 2. 如果$m \not\equiv 0(mod\, q)$，
>
>    $m^{ed} = m^{ed-1}m = m^{j(q-1)}m = (m^{q-1})^j m$
>
>    如果$m < q$，则因$q$为质数，则$mq$互质，如果$m > q$，则因$q$是质数，而$m$不能被$q$整除，则$mq$互质，所以我们总是有
>
>    $m^{q-1} \equiv 1(mod\, q)$
>
>    也即
>
>    $(m^{q-1})^j \equiv 1(mod\, q)$
>
>    而后两边同时乘以$m$
>
>    则有
>
>    $m^{ed} \equiv m(mod\, q)$

对于另一半的证明也同理

最终等式得证