

RSA算法原理（二）

作者： 阮一峰

日期： 2013年7月 4日

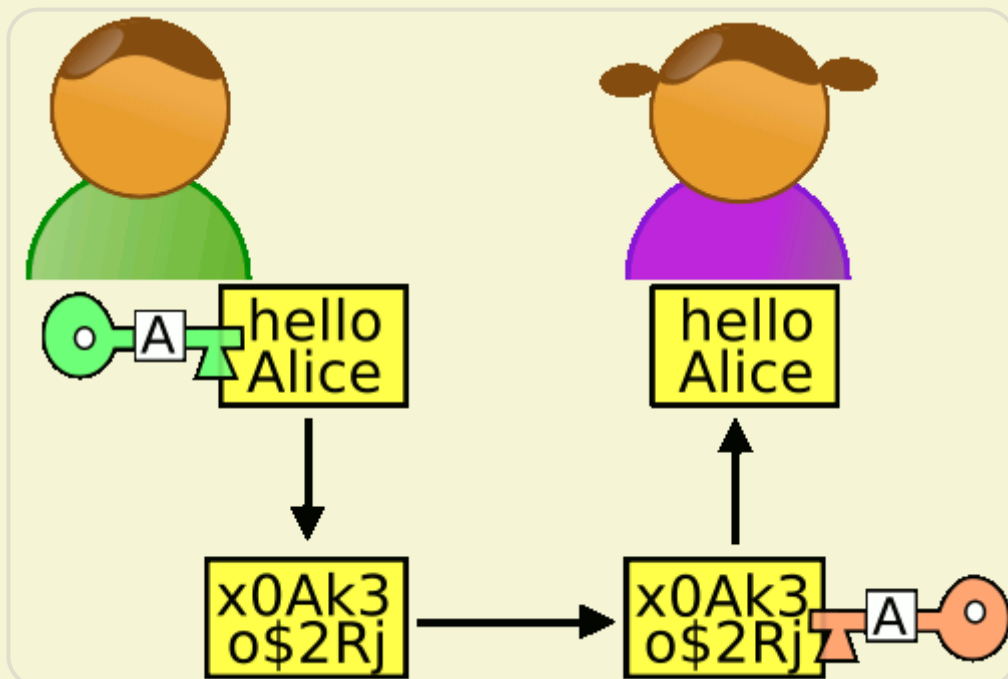
上一次，我介绍了一些[数论知识](#)。

有了这些知识，我们就可以看懂[RSA算法](#)。这是目前地球上最重要的加密算法。



六、密钥生成的步骤

我们通过一个例子，来理解RSA算法。假设[爱丽丝](#)要与鲍勃进行加密通信，她该怎么生成公钥和私钥呢？



第一步，随机选择两个不相等的质数p和q。

爱丽丝选择了61和53。（实际应用中，这两个质数越大，就越难破解。）

第二步，计算p和q的乘积n。

爱丽丝就把61和53相乘。

$$n = 61 \times 53 = 3233$$

n的长度就是密钥长度。3233写成二进制是110010100001，一共有12位，所以这个密钥就是12位。实际应用中，RSA密钥一般是1024位，重要场合则为2048位。

第三步，计算n的欧拉函数 $\phi(n)$ 。

根据公式：

$$\phi(n) = (p-1)(q-1)$$

爱丽丝算出 $\phi(3233)$ 等于 60×52 ，即3120。

第四步，随机选择一个整数e，条件是 $1 < e < \phi(n)$ ，且e与 $\phi(n)$ 互质。

爱丽丝就在1到3120之间，随机选择了17。（实际应用中，常常选择65537。）

第五步，计算e对于 $\phi(n)$ 的模反元素d。

所谓"模反元素"就是指有一个整数d，可以使得ed被 $\phi(n)$ 除的余数为1。

$$ed \equiv 1 \pmod{\phi(n)}$$

这个式子等价于

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

于是，找到模反元素d，实质上就是对下面这个二元一次方程求解。

$$ex + \phi(n)y = 1$$

已知 $e=17$, $\phi(n)=3120$,

$$17x + 3120y = 1$$

这个方程可以用"[扩展欧几里得算法](#)"求解，此处省略具体过程。总之，爱丽丝算出一组整数解为 $(x,y)=(2753,-15)$ ，即 $d=2753$ 。

至此所有计算完成。

第六步，将n和e封装成公钥，n和d封装成私钥。

在爱丽丝的例子中， $n=3233$ ， $e=17$ ， $d=2753$ ，所以公钥就是 $(3233,17)$ ，私钥就是 $(3233, 2753)$ 。

实际应用中，公钥和私钥的数据都采用[ASN.1](#)格式表达（[实例](#)）。

七、RSA算法的可靠性

回顾上面的密钥生成步骤，一共出现六个数字：

p
q
n
 $\varphi(n)$
e
d

这六个数字之中，公钥用到了两个（n和e），其余四个数字都是不公开的。其中最关键的是d，因为n和d组成了私钥，一旦d泄漏，就等于私钥泄漏。

那么，有无可能在已知n和e的情况下，推导出d？

(1) $ed \equiv 1 \pmod{\varphi(n)}$ 。只有知道e和 $\varphi(n)$ ，才能算出d。

(2) $\varphi(n)=(p-1)(q-1)$ 。只有知道p和q，才能算出 $\varphi(n)$ 。

(3) $n=pq$ 。只有将n因数分解，才能算出p和q。

结论：如果n可以被因数分解，d就可以算出，也就意味着私钥被破解。

可是，大整数的因数分解，是一件非常困难的事情。目前，除了暴力破解，还没有发现别的有效方法。维基百科这样写道：

"对极大整数做因数分解的难度决定了RSA算法的可靠性。换言之，对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，那么RSA的可靠性就会极度下降。但找到这样的算法的可能性是非常小的。今天只有短的RSA密钥才可能被暴力破解。到2008年为止，世界上还没有任何可靠的攻击RSA算法的方式。

只要密钥长度足够长，用RSA加密的信息实际上是不能被解破的。"

举例来说，你可以对3233进行因数分解（ 61×53 ），但是你没法对下面这个整数进行因数分解。

```
12301866845301177551304949
58384962720772853569595334
79219732245215172640050726
36575187452021997864693899
56474942774063845925192557
32630345373154826850791702
61221429134616704292143116
02221240479274737794080665
351419597459856902143413
```

它等于这样两个质数的乘积：

```
33478071698956898786044169
84821269081770479498371376
85689124313889828837938780
02287614711652531743087737
814467999489
    ×
36746043666799590428244633
79962795263227915816434308
76426760322838157396665112
79233373417143396810270092
798736308917
```

事实上，这大概是人类已经分解的最大整数（232个十进制位，768个二进制位）。比它更大的因数分解，还没有被报道过，因此目前被破解的最长RSA密钥就是768位。

八、加密和解密

有了公钥和密钥，就能进行加密和解密了。

(1) 加密要用公钥 (n,e)

假设鲍勃要向爱丽丝发送加密信息m，他就要用爱丽丝的公钥 (n,e) 对m进行加密。这里需要注意，m必须是整数（字符串可以取ascii值或unicode值），且m必须小于n。

所谓"加密"，就是算出下式的c：

$$m^e \equiv c \pmod{n}$$

爱丽丝的公钥是 (3233, 17)，鲍勃的m假设是65，那么可以算出下面的等式：

$$65^{17} \equiv 2790 \pmod{3233}$$

于是，c等于2790，鲍勃就把2790发给了爱丽丝。

(2) 解密要用私钥(n,d)

爱丽丝拿到鲍勃发来的2790以后，就用自己的私钥(3233, 2753) 进行解密。可以证明，下面的等式一定成立：

$$c^d \equiv m \pmod{n}$$

也就是说，c的d次方除以n的余数为m。现在，c等于2790，私钥是(3233, 2753)，那么，爱丽丝算出

$$2790^{2753} \equiv 65 \pmod{3233}$$

因此，爱丽丝知道了鲍勃加密前的原文就是65。

至此，"加密--解密"的整个过程全部完成。

我们可以看到，如果不知道d，就没有办法从c求出m。而前面已经说过，要知道d就必须分解n，这是极难做到的，所以RSA算法保证了通信安全。

你可能会问，公钥(n,e) 只能加密小于n的整数m，那么如果要加密大于n的整数，该怎么办？有两种解决方法：一种是把长信息分割成若干段短消息，每段分别加密；另一种是先选择一

种"对称性加密算法"（比如[DES](#)），用这种算法的密钥加密信息，再用RSA公钥加密DES密钥。

九、私钥解密的证明

最后，我们来证明，为什么用私钥解密，一定可以正确地得到m。也就是证明下面这个式子：

$$c^d \equiv m \pmod{n}$$

因为，根据加密规则

$$m^e \equiv c \pmod{n}$$

于是，c可以写成下面的形式：

$$c = m^e - kn$$

将c代入要我们要证明的那个解密规则：

$$(m^e - kn)^d \equiv m \pmod{n}$$

它等同于求证

$$m^{ed} \equiv m \pmod{n}$$

由于

$$ed \equiv 1 \pmod{\varphi(n)}$$

所以

$$ed = h\varphi(n) + 1$$

将ed代入：

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

接下来，分成两种情况证明上面这个式子。

(1) m与n互质。

根据欧拉定理，此时

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

得到

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

原式得到证明。

(2) m与n不是互质关系。

此时，由于n等于质数p和q的乘积，所以m必然等于kp或kq。

以 $m = kp$ 为例，考虑到这时k与q必然互质，则根据欧拉定理，下面的式子成立：

$$(kp)^{q-1} \equiv 1 \pmod{q}$$

进一步得到

$$[(kp)^{q-1}]^{h(p-1)} \times kp \equiv kp \pmod{q}$$

即

$$(kp)^{ed} \equiv kp \pmod{q}$$

将它改写成下面的等式

$$(kp)^{ed} = tq + kp$$

这时t必然能被p整除，即 $t = t'p$

$$(kp)^{ed} = t'pq + kp$$

因为 $m=kp$, $n=pq$, 所以

$$m^{ed} \equiv m \pmod{n}$$

原式得到证明。

(完)
