RSA是现代网络应用中最重要的加密算法，运用非对称加密的方式解决了密钥配送问题。在通信过程中，比如A生成一对密钥对，包含公钥和密钥，而用公钥加密的密文，只能由密钥解开，所以要跟A秘密通话，就用A给的公钥来加密，这个消息也只有A才能知道了。至于怎么确定公钥是来自A的，那就涉及到数字签名技术，本文不做探讨。

这对公钥密钥看起来是个很神奇的东西，其实加解密很简洁，简洁到让人感叹数学之美。

用公钥(n, e)加密：**明文e≡ 密文 (mod n)**

用私钥(n, d)解密：**密文d≡ 明文 (mod n)**

**加密 a ,解密b**

**加密 b ,解密a, 加密解密不对称**

A a 客户端 B b服务器

TCP 3次

Ab

cd

A a B a对称，冒充，

上述表达式是同余式，也就是“≡”两边mod n是相等的。mod运算就是取被除数 / 除数得到的余数，运算符是%。比如5%3=2。所以上式也可表达成

用公钥(n, e)加密：**密文 = 明文e% n**

用私钥(n, d)解密：**明文 = 密文d% n**

这样更简洁了吧，嘿嘿

可是为什么这么神奇呢？非常好奇，好好研究了一把，发现智商捉急啊，所以这种东西我还是选择记录下来，过后肯定忘了。

我尝试用推导的方式去描述RSA的神奇，在此之前，我们需要了解点必备的知识，分别是**中国剩余定理**，**模运算规则**，**费马小定理**。

**一、中国剩余定理**

我们先来看看例子

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| a % 3 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 |
| a % 5 | 0 | 1 | 2 | 3 | 4 | 0 | 1 | 2 | 3 | 4 | 0 | 1 | 2 | 3 | 4 |

从这个表格，很容易发现一个规律：

(a % 3)是以3为周期在循环，(a % 5)是以5为周期在循环，而且数对(a % 3, a % 5)在这个[0,14]之间并没有出现重复，也就是说由数对(a % 3, a % 5)可以确定a的值。

如果a继续往上增加，就开始跟表格产生重复，也就是以3\*5=15为周期的重复，这时数对(a % 3, a % 5)不能确定唯一的a值，而是形如a+15n的数列，每个周期都存在唯一a。之所以会这样，是因为3与5的互质关系。**如果两个数的最大公约数是1，则称这两个数互质**。也可以发现它们的最小公倍数为两者乘积，这个乘积形成了数对(a % 3, a % 5)的周期。这就是中国剩余定理的内容。概括起来说就是：**如果n个数互质，且乘积为P，有一个未知数M，已知M分别除以n个数的余数，那个在0<M<P的范围内，可以确定唯一的M值**。 2\*3\*5\*7=210 0-210

M=2\*x+1=3\*y+1=2\*z+3=7\*m+6

//13 1 1 3 6

如果不两两互质呢？假如表格的两个数是3与6，那么我们会发现数对(a % 3, a % 6)在以3\*6=18为周期之前，还有一个小周期6，所以在[0,18)范围内，给出数对(a % 3, a % 5)并不能确定唯一的a值。

中国剩余定理有个很实际的应用，叫做Mignotte门限方案，就是分别给5个特工一组密文，要解开此密文，必须有3名特工聚一起才行，能防止一个特工拥有密文的全部，或者某个特工被抓而解不开密文。具体不做介绍了。

**二、模运算规则**

从RSA的加解密公式就看到使用了模运算，那么关于模运算的一些特性是必须要了解的。

100\*13=1300

300\*77= 21000+2100=23100

250\*13 3250

250\*77=19250

阐述之前，先来看看我们以前玩过的数学小魔术。你心想三位数，乘以13，告诉我乘积的后三位数，我就能知道你想的是哪个数！假如你想的是233，233\*13=3029，所以你告诉我的是029，我只要把29\*77=2233，积的后三位数233就是你心想的数。这里就有点非对称的影子了，(1000, 13)就是公钥，(1000, 77)就是私钥。

其实这里神奇的地方是运用了模运算的一些规律：

1）**若a ≡ b (mod p)，则对于任意的c，都有(a \* c) ≡ (b \* c) (mod p)**

2）**(a \* b \* c) % p ＝ ((a \* b) % p \* c) % p**

3\*N %2 =5\*N %2

我们来分解下。

3%2=1 15%2=1

5%2=1

3\*n%2=1

13 \* 77 = 1001 => 1001 % 1000 = 1

=> 1001 ≡ 1 (mod 1000)

=> a \* 1001 ≡ a (mod 1000)

就是说任何数乘以1001除以1000得到余数是本身。因此：

(a \* 1001) % 1000 = (a \* 13 \* 77) % 1000

= ((a \* 13) % 1000) \* 77) % 1000

理解一下就能发现，我们并不需要知道a \* 13的积是多少，而是知道它除以1000的余数就行了，因为反正最后都是要模1000，那事先拿一部分来模1000并不影响结果。跟除法类似，两个数乘积除以另一个数，那事先用一个乘数除以除数，得到的值再乘以另一个乘数再除，并不影响结果。这个思想，也是RSA用来当被模数过大时优化计算力的算法。

还有一个规律需要推导下：

**aed % p ＝ (ae % p)d % p**。

把ae看成一个整体，也就是求d个ae相乘再模p，从上面知道，事先用乘数模p并不影响结果，也就是每个ae先模p，一共有d个，相乘，再模p跟原来是一样的。

**三、费马小定理**

费马小定理的内容是这样的：**如果n是一个质数，对于任意的 a 有 an- a = n的倍数**。关于费马小定理的证明，有一种很具象的证明，即“循环移位”的方法。后面这段简单说下，也可直接跳过。

以27为例，想象成一个二进制就好了，这个二进制由7个bit组成，当然就有27种排列组合，所谓循环移位是指，取其中一组排列，像跑马灯那样，每次往右移一位，尾部的数往头部塞，一直移移移，到最后发现会重复，是以7为周期。

我们再想想，不是所有排列都能以7为周期。

1）首先这组排列的数字都相同就不行，也就是全是1或全是0，无论怎么移都是一样的。

2）如果n不是7，而是素数。比如6，那么取一个排列，使其按6的其中一个约数为周期重复排列，那么周期就不是6了，比如101101，此时周期为3。

而其它情况，都能呈现以7为周期。这个“简单说”好像有点长，如果还是不理解呢可以谷歌或者看看matrix67的博文（文章最后有二维码）。

费马小定理还有一种说法：**对任意a，随着 i 的增加，ai% n 呈现出长度为n-1的周期性，n为质数**。可以由上一种说法证明得到：

an- a ＝ n的倍数 => an- a ≡ 0 (mod n)

=> an% n = a % n

也就是说 a1%n，a2%n，...，到an%n，余数又回到原点。而从上节说到的模运算可以知道：(a \* a) % n ＝ ((a % n) \* a) % n，就是说 ai% n 是由 ai-1% n 决定，这样环环相扣还能回到原点，就是说明要循环啦，以n-1为周期在循环。**这个规律正是RSA的基础**。

上面说到n为质数，如果n不是质数呢？假设n＝p\*q，p、q都是质数，对任意a，随着 i 的增加，ai % n会呈现怎样的周期性？根据费马小定理，ai % p的循环周期为p-1，ai % q的循环周期为q-1。再回顾下上面的中国剩余定理，pq互质，在一个循环周期内，ai 可以由数对(ai % p, ai % q)确定，也就是数对(ai % p, ai % q)存在周期为(p-1)(q-1)的循环，可得出 ai 也存在周期为(p-1)(q-1)的循环，必然ai % n的循环周期也是(p-1)(q-1)。

总结一下，若n为质数，则ai % n的循环周期是n-1；若n是两质数p,q的乘积，则ai % n的循环周期是(p-1)(q-1)。

**四、推导RSA**

根据上面每一节最终推导的结果，再看看下RSA公式，应该能感觉出什么东西了。

把加密公式代入解密公式：明文 ＝ (明文e % n)d % n，即明文 ＝ 明文ed % n，这很有可能是明文的 i 次方模n的余数在以ed-1为周期循环，或者有比ed-1更小的周期，顺着这个想法，我们看看怎么构造e、d、n使其成立。

由上文用费马小定理和中国剩余定理给过证明的：给出两质数p、q，其乘积为n，那么随着 i 的增加，明文i % n呈现周期为(p-1)(q-1)的循环。

我们最终是要得到明文，所以取明文的1次方，那么有：

明文 % n ＝ 明文h(p-1)(q-1)+1 % n。

我们不能暴露pq，不然密钥就没有意义了，所以这里尝试构造出ed ＝ h(p-1)(q-1) + 1，也就是ed ≡ 1 (mod m), m = (p-1)(q-1)。

其实至此，RSA的实际意义已表明，就是**模运算规则有非对称性，以及余数有周期性**，这是我要的答案，不知道描述得准不准确。

至于怎么找到这对ed呢？直接上定理简单说，这里的术语是称作d是e的模反元素，由欧拉定理：aφ(n)≡ 1 (mod n)（正整数a，n互质）可以证明。其中欧拉函数φ(n)是求1到n的范围内，有多少个数与n互质，这个值也是ai % n的余数序列有多长的周期的解，这两者有啥关系我也不清楚。所以从小于m的数里找到一个与m互质的，作为e，e是肯定能找到的。那么有e \* eφ(n)-1 ≡ 1 (mod n)，也就是d必然存在。

这里可能有点乱，我们回过头来看看密钥对的生成步骤做个对比。

1、随机选择两个不相等的质数p和q。

2、计算p和q的乘积n。

3、计算p-1和q-1的乘积m。

4、随机选个整数e，e与m要互质，且0<e<m。

5、计算e的模反元素d。

6、n，e组成公钥，n，d组成私钥。

本文只是拜读了matrix67大神的《跨越千年的RSA算法》和阮一峰的《RSA算法原理》以及了解了关于模运算知识后的产物，有兴趣的可以去看这两篇，甚至RSA论文。

自己是因为好奇而试图去明白RSA的神奇之处，作些笔记，当然会缺乏很多严谨性，只在了解层面上，作为一个数学渣的智商也写不出更严谨的东西了，只要大概明白为啥RSA可以密文d取余能回到明文，可以非对称就足矣。

数学里很多高深莫测的东西往往都是异想不到的十分的简洁（这是不是个通用规律？现实中的很多成功产品也是这样），在此推荐吴军的《数学之美》，还有matrix67的《数学之美番外篇》