

# RSA

Crazy\_13754

2024 年 2 月 20 日

## 摘要

写了些关于 rsa 的东西。

## 目录

<b>1 说在前面</b>	<b>1</b>
<b>2 引言</b>	<b>2</b>
<b>3 RSA 加密与解密过程</b>	<b>2</b>
<b>4 RSA 数学基础与证明</b>	<b>2</b>
4.1 同余	3
4.2 欧几里得算法	3
4.3 裴蜀定理	3
4.4 拓展欧几里德算法	4
4.5 模的逆	4
4.6 欧拉函数	5
4.6.1 使用容斥原理证明	5
4.6.2 使用中国剩余定理证明	5
4.7 欧拉定理	7
<b>5 RSA 数字签名与数字证书</b>	<b>7</b>
5.1 散列函数	7
5.2 数字签名与数字证书	7
<b>6 让你的 RSA 更安全</b>	<b>8</b>

## 1 说在前面

虽然本文花费大力气写了数学种种理论，笔者却深感这并无必要，因为无论作者抑或读者所关心的其实是技术，数学理论不过是些模糊空气——固然是十分重要的支撑，写起来也不容忽视，但也不妨碍人们对其漠不关心。所以，读者大可以略看繁复的证明，注意于各类技术的细节。

## 2 引言

众所周知，密码学的提出是为了保证传输信息的安全。在古代，人们使用的加密方式可能是一张字符映射表，将信息中的字符逐个替换来让信件难以解密，只要这张表格不被敌人接获，信息就将保持安全。换句话说，只要加密的算法是安全的，信息就将是安全的。而现代的加密技术则选择公开加密算法，使用密钥加密。这被称为柯克霍夫原则 (Kerckhoffs's principle)。其中原因也不难想到：使用特定算法加密一旦算法泄露，加密的内容会易遇破解，而使用密钥则是“更新”了这个问题，因而可以长期反复使用。敌人如果买通工作人员，源码的泄露也难以影响其安全性。当加密用软件或硬件长期使用的时候，敌人可能通过各种方式分析其原理。公开的算法经过专家验证，人们对广泛流传的加密技术有信心。

更进一步，加密算法可以粗略分为两种：对称加密与非对称加密。对称加密技术，例如 AES，使用相同密钥加密与解密，速度较快，但密钥本身通常需要非对称加密技术加密并传输。非对称加密技术，例如 RSA，有公钥私钥之分，但速度较慢，一次加密的信息较少。此外，公钥密码能够解决消息鉴别问题，就是指用来检验消息来自于声称的来源并且没有被修改。

公钥体制的基础是陷门（单向函数），即某种实际处理过程的不可逆性。目前的公钥思想基于两种：一是依赖于大数的因数分解的困难性；二是依赖于求模  $p$  离散对数的困难性。RSA 密码算法就是基于大数的因数分解的困难性 [2]。

RSA 加密算法是一种非对称加密算法，在公开密钥加密和电子商业中被广泛使用。RSA 是由罗纳德·李维斯特 (Ron Rivest)、阿迪·萨莫尔 (Adi Shamir) 和伦纳德·阿德曼 (Leonard Adleman) 在 1977 年一起提出的。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。1973 年，在英国政府通讯总部工作的数学家克利福德·柯克斯 (Clifford Cocks) 在一个内部文件中提出了一个与之等效的算法，但该算法被列入机密，直到 1997 年才得到公开。

这篇文章介绍了因子分解的相关算法，然而，这其中的很多方法对 RSA 来说并不适用。对于 RSA 的“暴力”攻击来说，普通数域筛法 (GNFS) 应该是最优的。如果我能看懂这个算法的话，会加上对它的介绍的……

## 3 RSA 加密与解密过程

小发（消息发送者）想要给小收（消息接受者）发一条需要保密的信息。

- 小收取两个非常大的素数  $p$  和  $q$ ，并令  $N = p \cdot q$ ， $\varphi(N) = (p-1) \cdot (q-1)$ 。
- 找到一个素数  $e < \varphi(N)$ ，且要求  $e$  与  $\varphi(N)$  互素，即有  $\gcd(e, \varphi(N)) = 1$ 。
- 计算  $e$  在模  $\varphi(N)$  上的逆元  $d$ ，即求  $d$  满足  $e \cdot d \bmod \varphi(N) = 1$ 。
- 小收将  $(N, e)$  作为公钥 (pk, Public key) 发给对方， $(N, d)$  作为私钥 (pk, Private key) 保存。
- 小发接受公钥后，将原文 (pt, Plaintext) 通过预先设定好的方式转换成数字，记为  $pt$ ，则密文 (ct, Ciphertext) 满足  $ct = p^e \bmod N$ （也就是  $ct \equiv pt^e \pmod{N}$ ），对方将密文发回。
- 小收接收密文，并使用私钥解密： $pt = ct^d \bmod N$ ，也就是  $pt \equiv ct^d \pmod{N}$ 。

## 4 RSA 数学基础与证明

在无特殊说明的情况下，本章中所有的字母均指代正整数， $p, p_1, \dots, p_r$  为素数（注意 1 不是素数）。

## 4.1 同余

同余指的是, 对于  $a, b \in \mathbb{Z}, n \in \mathbb{Z}^*$ , 若  $\exists k \in \mathbb{Z}$ , 满足  $a - b = k \cdot n$ , 则称  $a, b$  模  $n$  同余, 记作  $a \equiv b \pmod{n}$ 。你想问我为什么要用乘法定义? 噢据王鲲鹏老师说, 这样定义有助于我们“操作”。

## 4.2 欧几里得算法

欧几里得算法 (或辗转相除法) 是得到两数最大公因数的算法, 其代码如下:

Listing 1: gcd

```
1 int gcd(int a, int b)
2 {
3     if (b == 0)
4         return a;
5     else
6         return gcd(b, a % b);
7 }
```

下证明算法正确性。设  $a > b, c_1 = \gcd(a, b), c_2 = \gcd(b, a \bmod b)$ 。此时,  $c_2 \mid b$  且  $c_2 \mid (a \bmod b)$ , 又因为  $a = k_1 b + (a \bmod b)$ , 则  $c_2 \mid a$ , 因此  $c_2$  为  $a, b$  的因数,  $c_2$  因而为  $c_1$  的因数。同理,  $c_1$  为  $c_2$  的因数, 即有  $c_1 = c_2$ 。不断进行此操作, 即可得到两数的最大公因数。

若  $a < b$ , 则  $\gcd(a, b) = \gcd(b, a)$ , 同上。

## 4.3 裴蜀定理

裴蜀定理可表述为: 若  $a, b$  不全为零, 则  $\forall x, y \in \mathbb{Z}$ , 有  $\gcd(a, b) \mid ax + by$ , 且  $\exists x, y$ , 使  $ax + by = \gcd(a, b)$ 。

因为  $\gcd(a, b) \mid a, b$ , 所以  $\forall x, y \in \mathbb{Z}$ ,  $\gcd(a, b) \mid ax + by$  一定成立。

因为  $\frac{a}{\gcd(a, b)}$  与  $\frac{b}{\gcd(a, b)}$  均为整数且互素, 因而  $ax + by = \gcd(a, b)$  有解可转为  $\frac{a}{\gcd(a, b)}x + \frac{b}{\gcd(a, b)}y = 1$ , 即证明  $a_1, b_1$  互素时,  $a_1 x + b_1 y = 1$  恒有解。

展开欧几里得算法可得:

$$\begin{aligned}a_1 &= q_1 b_1 + r_1 \\b_1 &= q_2 r_1 + r_2 \\&\dots \\r_{n-2} &= q_n r_{n-1} + r_n \\r_{n-1} &= q_{n+1} r_n\end{aligned}$$

且  $r_n = \gcd(a_1, b_1) = 1$ , 则有:

$$\begin{aligned}
1 &= r_n \\
&= r_{n-2} - q_n r_{n-1} \\
&= r_{n-2} - q_n(r_{n-3} - q_{n-1} r_{n-2}) \\
&\dots
\end{aligned} \tag{1}$$

用同样方法不断消去  $r_{n-2}, r_{n-3}, \dots, r_1$  可得  $a_1 x + b_1 y = 1$  有解。

#### 4.4 拓展欧几里德算法

拓展欧几里德算法就是求  $ax + by = c$  的一组解的方法。由 4.3 节,  $c$  一定是  $\gcd(a, b)$  的倍数。此时, 重复方程 1 的过程即可求  $ax + by = c$  的一组解。算法过程如下:

Listing 2: exgcd

```

1  int exgcd(int a, int b, int &x, int &y)
2  {
3      if (b == 0)
4      {
5          x = 1;
6          y = 0;
7          return a;
8      }
9      int d = exgcd(b, a % b, y, x);
10     y -= (a / b) * x;
11     return d;
12 }
```

考虑通解的情况。若已知解为  $x_1, y_1$  且  $x_2, y_2$  也为一组解, 那么两方程相减可得:

$$\begin{aligned}
(x_1 - x_2)a + (y_1 - y_2)b &= 0 \\
(x_1 - x_2)a &= (y_2 - y_1)b \\
(x_1 - x_2)\frac{a}{\gcd(a, b)} &= (y_2 - y_1)\frac{b}{\gcd(a, b)}
\end{aligned}$$

由于  $\frac{a}{\gcd(a, b)}$  与  $\frac{b}{\gcd(a, b)}$  互素, 则有  $\frac{a}{\gcd(a, b)} | (y_2 - y_1)$ ,  $\frac{b}{\gcd(a, b)} | (x_1 - x_2)$ , 即通解可以写为:

$$\begin{cases} x_k = x_1 + k \frac{b}{\gcd(a, b)} \\ y_k = y_1 - k \frac{a}{\gcd(a, b)} \end{cases} \tag{2}$$

#### 4.5 模的逆

若  $a \cdot x \equiv 1 \pmod{b}$ , 则  $x$  为  $a$  模  $b$  意义下的逆元。变换形式, 可得  $a \cdot x + b \cdot y = 1$ 。由 4.3 节, 当  $\gcd(a, b) = 1$ , 也就是  $a, b$  互素时逆元一定存在。

## 4.6 欧拉函数

欧拉函数  $\varphi(n)$  表示小于等于  $n$  的正整数中与其互质的数字个数。其可表示为：

$$\varphi(n) = \begin{cases} 1, & n = 1 \\ n(1 - \frac{1}{p_1})(1 - \frac{1}{p_2}) \cdots (1 - \frac{1}{p_r}), & n = p_1^{k_1} p_2^{k_2} \cdots p_r^{k_r} \end{cases}$$

$n$  一定属于以上某条件，可用归纳法证明：

- $n = 1, 2$  满足以上条件。
- $n \geq 3$ , 若  $n$  满足以上条件, 若  $n+1 = p_1^{k_1} p_2^{k_2} \cdots p_r^{k_r}$  则也满足以上条件, 若  $n+1 \neq p_1^{k_1} p_2^{k_2} \cdots p_r^{k_r}$ , 那它一定是个合数, 可以分解为两个或多个小于  $n$  的数字乘积。

下面考虑  $\varphi(n)$ ：

当  $n = p^k$  时, 不包含  $p$  的数才能与其互质。而包含  $p$  的数字有  $p, 2 \cdot p, \dots, p^{k-1} \cdot p$  共  $p^{k-1}$  个, 因此此时  $\varphi(n) = p^k - p^{k-1} = p^k(1 - \frac{1}{p})$ 。

当  $n = p_1^{k_1} p_2^{k_2} \cdots p_r^{k_r}$  时, 可以通过以下两种方式证明该式。

### 4.6.1 使用容斥原理证明

先考虑  $n = p_1^{k_1} p_2^{k_2}$  的情况。在计数  $\varphi(n)$  的过程中不计数  $p_1, 2p_1, \dots, \lfloor \frac{n}{p_1} \rfloor p_1$ , 同理对  $p_2, 2p_2, \dots, \lfloor \frac{n}{p_2} \rfloor p_2$  也不应该计数, 并根据容斥原理 (计数时重复计数的部分要扣除) 还需要在结果中加上  $p_1 p_2, 2p_1 p_2, \dots, \lfloor \frac{n}{p_1 p_2} \rfloor p_1 p_2$  的数量。这证明了  $\varphi(n) = n(1 - \frac{1}{p_1})(1 - \frac{1}{p_2})$ 。

同理, 有  $\varphi(mn) = mn(1 - \frac{1}{p_{m1}})(1 - \frac{1}{p_{m2}}) \cdots (1 - \frac{1}{p_{mr}})(1 - \frac{1}{p_{n1}})(1 - \frac{1}{p_{n2}}) \cdots (1 - \frac{1}{p_{nk}}) = \varphi(m)\varphi(n)$ , 这就证明了  $n = p_1^{k_1} p_2^{k_2} \cdots p_r^{k_r}$  时,  $\varphi(n) = n(1 - \frac{1}{p_1})(1 - \frac{1}{p_2}) \cdots (1 - \frac{1}{p_r})$ 。

### 4.6.2 使用中国剩余定理证明

先介绍中国剩余定理。它旨在解决形如下式的问题：

$$\begin{cases} x \equiv a_1 \pmod{n_1} \\ x \equiv a_2 \pmod{n_2} \\ \dots \\ x \equiv a_r \pmod{n_r} \end{cases} \quad (\forall i, j \leq r, \gcd(n_i, n_j) = 1) \quad (3)$$

这个定理推导基于以下性质：

1. 若  $a \bmod n_1, n_2, \dots, n_r = 0$ , 则  $a$  为  $n_1, n_2, \dots, n_r$  公因子之积的倍数。
2. 若  $a \bmod n = 1, k < n$ , 则  $k \cdot a \bmod n = k$ 。
3. 由  $a + k \cdot n \equiv a \pmod{n}$ , 可拓展得若  $a_1, a_2, \dots, a_r \bmod n = 0$ , 则  $a + k_1 a_1 + k_2 a_2 + \dots + k_r a_r \equiv a \pmod{n}$

由性质 3 得我们可以把该问题拆分, 形如下式。

$$\begin{cases} x_1 \equiv a_1 \pmod{n_1}, x_1 \equiv 0 \pmod{n_2}, x_1 \equiv 0 \pmod{n_3}, \dots, x_1 \equiv 0 \pmod{n_r} \\ x_2 \equiv 0 \pmod{n_1}, x_2 \equiv a_2 \pmod{n_2}, x_2 \equiv 0 \pmod{n_3}, \dots, x_2 \equiv 0 \pmod{n_r} \\ \vdots \\ x_r \equiv 0 \pmod{n_1}, x_r \equiv 0 \pmod{n_2}, \dots, x_r \equiv a_r \pmod{n_r} \\ x = \sum_{i=1}^r x_i \end{cases}$$

此时, 利用性质 2, 可以将  $x_i \equiv a_i \pmod{n_i}$  简化为求  $x'_i \equiv 1 \pmod{n_i}$  且  $x_i = a_i \cdot x'_i$ 。令  $N = n_1 \cdot n_2 \cdots n_r$ , 则根据性质 1 则有  $x'_i$  满足  $x_i = \frac{N}{n_i} \cdot q_i$ , 且  $\frac{N}{n_i} \cdot q_i \equiv 1 \pmod{n_i}$ , 即  $q_i$  为求  $\frac{N}{n_i}$  模  $n_i$  的逆, 记为  $\text{invert}(\frac{N}{n_i}, n_i)$  [1]。此时, 方程已经被我们化为如下形式:

$$\begin{cases} x_1 = a_1 \frac{N}{n_1} \cdot \text{invert}(\frac{N}{n_1}, n_1) \\ x_2 = a_2 \frac{N}{n_2} \cdot \text{invert}(\frac{N}{n_2}, n_2) \\ \vdots \\ x_i = a_i \frac{N}{n_i} \cdot \text{invert}(\frac{N}{n_i}, n_i) \\ \vdots \\ x_r = a_r \frac{N}{n_r} \cdot \text{invert}(\frac{N}{n_r}, n_r) \\ x \equiv \sum_{i=1}^r x_i \pmod{n} \end{cases}$$

这就给出了方程 3 的解。若  $\exists i, j$ , 使  $\gcd(n_i, n_j) > 1$ , 则可通过除其公因子“合并”方程。关于逆元的求法可参考[这个链接](#)。

下面将用中国剩余定理证明欧拉公式。

设  $A = \{x | \gcd(x, m) = 1, x \leq m\}, B = \{y | \gcd(y, n) = 1, y \leq n\}, C = \{z | \gcd(z, mn) = 1, z \leq mn\}$ , 则  $\varphi(m) = |A|, \varphi(n) = |B|, \varphi(mn) = |C|$ 。

$$\because \gcd(m, n) = 1$$

$$\therefore A \cap B = \{1\}$$

若  $mn \bmod N = 0$ , 即  $N \in C$ , 令  $N = k_1 m + p = k_2 n + q$ 。

$$\because \gcd(N, mn) = 1$$

$$\therefore \gcd(N, m) = \gcd(N, n) = 1$$

$$\therefore \gcd(k_1 m + p, m) = \gcd(p, m) = 1, \gcd(k_2 n + q, n) = \gcd(q, n) = 1,$$

$$\therefore \gcd(p, m) = \gcd(q, n) = 1$$

由中国剩余定理, 方程组

$$\begin{cases} N \equiv p \pmod{m} \\ N \equiv q \pmod{n} \end{cases}$$

有通解  $N = kmn + pnt_1 + qnt_2$ , 其中  $nt_1 \equiv 1 \pmod{m}, nt_2 \equiv 1 \pmod{n}$ 。

因此, 对于每一个二元组  $(p, q)$ , 都有唯一的  $N$  与之对应  $\Rightarrow A \times B$  与  $C$  构成双射。

根据乘法原理, 二元组  $(p, q)$  的数量为  $\varphi(n)\varphi(m)$ , 而与  $mn$  互质的数  $N$  的数量为  $\varphi(mn)$ 。

$$\therefore \varphi(mn) = \varphi(m)\varphi(n)$$

$$\therefore \forall n \in \mathbb{Z}^*, n = p_1^{k_1} p_2^{k_2} \cdots p_n^{k_n}, \varphi(n) = p_1^{k_1} (1 - \frac{1}{p_1}) p_2^{k_2} (1 - \frac{1}{p_2}) \cdots p_n^{k_n} (1 - \frac{1}{p_n}) = p_1^{k_1} p_2^{k_2} \cdots p_n^{k_n} (1 - \frac{1}{p_1}) (1 - \frac{1}{p_2}) \cdots (1 - \frac{1}{p_n})。$$

## 4.7 欧拉定理

欧拉定理指的是，如果两个正整数  $a$  和  $n$  满足  $\gcd(a, n) = 1$ ，则有：

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

证明……先等等吧……

## 5 RSA 数字签名与数字证书

现在设想这样一个情形：小收公布公钥之后，收到了一条信息要求他给小发转账。于是小收转了 50，但第二天，小收却声明他从未发过类似于“vivo50”这样的信息。原来，小攻，作为一名黑客，冒充小发发了一条信息。小收发现了这个公钥体制的问题——没有办法证明信息的发送者。

同一时间，另一头的小攻也在为另一个问题苦恼——他发送的信息明明是“vivo500”，可是他仅仅收到了 50！原来，信息在传输的过程中被他的对手小黑（另一名黑客），改成了 vivo50！他咬牙切齿，因为小收的消息系统太烂了——根本没办法知道信息发送过程中有没有被篡改。

RSA 数字签名与数字证书可证明发信人是其自身，以及消息的完整性。

### 5.1 散列函数

验证信息没有被篡改很简单：使用散列函数，对原文进行“摘要”。散列函数应当满足这样的性质：当输入发生微小的变化的时候结果变化很大，并且难以找到有相同输出的输入，这样，当消息发生变化，用相同算法生成的“摘要”也就就会发生变化。

### 5.2 数字签名与数字证书

RSA 算法原理保证了其有这样的性质：即使使用私钥加密，用公钥也可以解密。这是由第 4 章中介绍的数学原理决定的。因此，在上文的例子中，小收也可以制作一份密钥，并将公钥公布。在发送信息的时候，用自己的私钥对摘要进行加密。如果小攻想要修改信息，那么他必须知道小收使用的私钥——不然小发得到的消息和摘要将会不一致。

然而，小攻十分滴聪明狡猾。他用小收的密钥加密了“vivo500”的假消息，然后自己制作了一份公钥发到网上，用私钥对消息的摘要加了密。小收又上了当——这份消息看起来同样是小发的消息，也有“小发的签名”。

这时候需要由数字证书解决这个问题。证书授权 (CA, Certificate Authority) 由相应的 CA 机构颁发给小收（应经过线下验证），包括了小收的公钥、小收的身份信息、CA 的签名。这样，小发就可以找到真正的小收公钥与对应消息。

验证证书的合法性是套娃的过程。小收的公钥与小收的身份信息对应的摘要，应该与 CA 的解密后签名一致。CA 根证书是安装系统内置在系统或浏览器中的，这样如果还想信息造假，需要修改用户的系统或浏览器文件，而这通常是不可能的。

现在如果小攻有能力发送信息、劫持他人发送的信息，且没有能力分解公钥，小攻将无法完成欺骗，除非他可以劫持网络到修改他们发送的信息，或是可线下潜入某位仁兄的房间，修改他们的电脑数据——但如果这样，他直接用他们的账户给自己转账似乎更合理些。

## 6 让你的 RSA 更安全

### 参考文献

- [1] Bat 特白. 中国剩余定理 (crt). [EB/OL]. <https://zhuanlan.zhihu.com/p/44591114> 发布于 2018-10-16 12:05.
- [2] 陈传波 and 祝中涛. Rsa 算法应用及实现细节. 计算机工程与科学, 28(9):13–14, 2006.