RSA 加密算法的分析与改进

褚有刚[*][†]

2019 年 6 月 3 日

目录

摘要

RSA 加密算法是目前应用最广泛的非对称加密算法，本文主要介绍了 RSA 加密算法的数学基础，讨论了可能的破解方法及其需要的时间，并对其安全性进行定量分析；除此之外，由于 RSA 加密算法在寻找大素数需要大量的计算，从而导致 RSA 算法的效率不高，本文给出了一些加速寻找大素数的算法，以提高 RSA 算法的适用性。

1 引言

随着计算机网络的高速发展，我们的生活已经越来越离不开计算机网络，现在人们普遍将其作为信息传输的媒介，但是在互联网上传输信息存在许多不安全的因素，为了防止信息假冒，篡改和泄露，目前广泛采用的举措是对信息进行加密，通信双方使用密文进行消息传递，只要攻击者无法

[*]学号：171860526

[†]邮箱：ajeo0526@gmail.com

在要求的时间内破译密文，数据就可以认为是安全的，目前的加密机制主要有对称密钥机制和非对称密钥机制，非对称密钥机制在加密以及数字签名之中都有不可替代的优越性，RSA 算法是其中的典型代表。

2 数学基础

RSA 算法是一个基于初等数论定理的公钥密码体制算法，所以在正式给出 RSA 算法需要介绍一些数论基础知识.

2.1 素数

> **素数**（Prime number），又称**质数**，指在大于 1 的自然数中，除了 1 和该数自身外，无法被其他自然数整除的数（也可定义为只有 1 与该数本身两个正因数的数）。大于 1 的自然数若不是素数，则称之为合数（也称为合成数）。

需要注意的是，1 不是素数，欧几里得[1]在《几何原本》[2]中提出了算术基本定理:

> **算术基本定理**，又称为**正整数的唯一分解定理**，即：每个大于 1 的自然数，要么本身就是质数，要么可以写为 2 个或以上的质数的积，而且这些质因子按大小排列之后，写法仅有一种方式。

为了确保该定理的唯一性，1 被定义不是素数，因为在因式分解中可以有任意多个 1（如 3、1×3、1×1×3 等都是 3 的有效约数分解）。

算术基本定理确立了素数在自然数中独一无二的地位，欧几里得认为素数是数的基础，同时期还有一位哲学家德谟克里特[3]提出了原子论，他认为每一种事物都是由原子所组成的，整个世界的本质只是原子和虚空。原子不可分割，并不完全一样。素数就像是自然数的原子一样，无法进行进一步的分解。

素数有许多很好的性质，其中很多性质都是 RSA 算法的基础，下面做简要介绍：

2.1.1 素数的数目

早在公元前 5 世纪毕达哥拉斯学派就给出了素数的数目有无穷多个证明，欧几里得将该证明收录在他的著作《几何原本》，被后世称为欧几里得定理，证明的大意如下：

证明.

1. 对任何有限的素数集合 $\{p_1, p_2, \ldots, p_n\}$，证明至少存在一个素数 q 不属于这个集合;
2. 令 $P = p_1 \cdot p_2 \cdots p_n, q = P + 1$
   - 如果 q 是素数，则至少存在一个素数 q 满足 $q \notin \{p_1, p_2, \ldots, p_n\}$;
   - 如果 q 不是素数，则至少存在一个素数 $p \in \{p_1, p_2, \ldots, p_n\}$ 满足 $p|q$，由于 $p|P$，则 $p|(q-P)$，即 $p|1$，由于没有素数能够整除 1，所以 $p \notin \{p_1, p_2, \ldots, p_n\}$ 中;
3. 综上，对于任何一个有限的素数集，都至少有一个素数不属于这个集合，因此素数有无穷多个.

□

[1]欧几里得（前 325 年—前 265 年），希腊划时代的数学家，被称为“几何学之父”

[2]《几何原本》（Stoicheia）是古希腊数学家欧几里得所著的一部数学著作，共 13 卷。

[3]德谟克利特（前 460 年—前 370 年或前 356 年）(英语：Democritus) 来自古希腊爱琴海北部海岸的自然派哲学家

证明素数有无穷多个的方式有许多种，上面是最早记录在史料中的证明方式.

莱昂哈德 · 欧拉在 1735 年解决巴塞尔问题的同时对该问题进行了发散性的思考，并借此提出了全新的一种证明素数有无穷多个的方法，考虑如下的无穷级数:

$$\zeta(s)=\sum_{n=1}^{\infty}\frac{1}{n^s}=\frac{1}{1^s}+\frac{1}{2^s}+\frac{1}{3^s}+\cdots \tag{1}$$

两边同时乘以 $\frac{1}{2^s}$，得到下式：

$$\frac{1}{2^s}\zeta(s)=\frac{1}{2^s}\sum_{n=1}^{\infty}\frac{1}{n^s}=\frac{1}{2^s}+\frac{1}{4^s}+\frac{1}{6^s}+\cdots \tag{2}$$

用 (1) - (2)，得到下式：

$$(1-\frac{1}{2^s})\zeta(s)=(1-\frac{1}{2^s})\sum_{n=1}^{\infty}\frac{1}{n^s}=\frac{1}{1^s}+\frac{1}{3^s}+\frac{1}{5^s}+\cdots \tag{3}$$

$$\frac{1}{3^s}(1-\frac{1}{2^s})\zeta(s)=\frac{1}{3^s}(1-\frac{1}{2^s})\sum_{n=1}^{\infty}\frac{1}{n^s}=\frac{1}{3^s}+\frac{1}{9^s}+\frac{1}{15^s}+\cdots \tag{4}$$

用 (3) - (4), 得到下式：

$$(1-\frac{1}{3^s})(1-\frac{1}{2^s})\zeta(s)=(1-\frac{1}{3^s})(1-\frac{1}{2^s})\sum_{n=1}^{\infty}\frac{1}{n^s}=\frac{1}{1^s}+\frac{1}{5^s}+\frac{1}{7^s}+\cdots \tag{5}$$

重复上述步骤，最终得到下式：

$$\Pi_{p\in P^4}(1-\frac{1}{p^s})\zeta(s)=1 \tag{6}$$

(6) 被称为欧拉乘积式.

$$\zeta(s)=\Pi_{p\in P}(1-\frac{1}{p^s})^{-1} \tag{7}$$

当 $s=1$ 时，(7) 变为：

$$\Pi_{p\in P}(1-\frac{1}{p})^{-1}=\zeta(1)=1+\frac{1}{2}+\frac{1}{3}+\cdots \tag{8}$$

(8) 的右边为调和级数[5]，由于调和级数发散，所以素数有无穷多个.

□

2.1.2 素数测试

在 RSA 算法中需要使用到随机的大素数，这就涉及到了如何确定一个数是否为素数，常用的方法为试除法，但是效率很低，没有什么实际用处，另有一类更有效率的测试方法，但是只能适用于特定的数字之上。

- 试除法
  测试 n 是否为素数的最基本的方法为试除法，该算法将 n 依次除以每一个大于 1 且小于等于 n 的平方根的自然数 m，如果存在一个自然数 m 满足 $m|n$，则 n 为合数，否则 n 为素数，实际上，若 n 为合数，不妨设 n 可以写作 $n=a\cdot b$，则 a 与 b 中必然至少有一个数字不大于 $\sqrt{n}$，否则假设 $a>\sqrt{n}$ 且 $b>\sqrt{n}$，则 $a\cdot b>n$ 与 $a\cdot b=n$ 矛盾.

[4] P 是所有素数的集合

[5] 调和级数（英语：Harmonic series）是一个发散的无穷级数。

- 筛法

  埃拉托斯特尼[6]**筛法**是列出所有小素数最有效的方法之一，其名字来自于古希腊数学家埃拉托斯特尼，所使用的原理是从 2 开始，将每个素数的各个倍数，标记成合数。一个素数的各个倍数，是一个差为此素数本身的等差数列。此为这个筛法和试除法不同的关键之处，后者是以素数来测试每个待测数能否被整除.

- 费马[7]小定理和费马素性测试

  费马小定理是这样的一个定理：假如 a 是一个整数，p 是一个素数，那么 $a^p - a$ 是 p 的倍数，可以表示为：

  $$a^p \equiv a \pmod p$$

  如果 a 不是 p 的倍数，这个定理也可以写作：

  $$a^{p-1} \equiv 1 \pmod p$$

  下面使用二项式定理证明费马小定理：

  证明. 考虑二项式系数 $\binom{p}{a} = \frac{p!}{a!(p-a)!}$ ，其中 p 为素数，如果 a 不为 0 且不为 p，则 $p|\binom{p}{a}$ ，因此有：

  $$(x+1)^p \equiv x^p + 1 (mod\ p) \tag{9}$$

  下证 $a^p \equiv a(mod\ p)$, p 为素数

  使用数学归纳法对 a 进行归纳

  Base Case 当 $a = 1$ 时，$a^p \equiv a \equiv 1(mod\ p)$

  Induction Princple 当 $a < k$ 时 $a^p \equiv a(mod\ p)$

  Induction Step 当 $a = k$ 时，由 (9) ，令 $x = k$ 有 $(k+1)^p \equiv k^p + 1(mod\ p)$，由 I.H，知 $k^p \equiv k(mod\ p)$，两式相加，得到:

  $$(k+1)^p + k^p \equiv k^p + k + 1(mod\ p)$$

  即 $(k+1)^p \equiv k + 1(mod\ p)$

  □

  根据费马小定理，若想要测试一个数字 p 是否为素数，则可随机选择 n 来检验 $n^p - n$ 能否被 p 整除即可。这个测试的缺点在于有些合数（卡迈克尔数[8]）即使不是素数，也会符合费马恒等式，因此这个测试无法辨别素数与卡迈克尔数，最小的三个卡迈克尔数为 561，1105，1729。卡迈克尔数比素数还少上许多，所以这个测试在实际应用上还是有用的。

---

[6]埃拉托斯特尼（英语：Eratosthenes，前 276 年一前 194 年，出生于昔兰尼，即现利比亚的夏哈特；逝世于托勒密王朝的亚历山大港），古希腊数学家、地理学家、历史学家、诗人、天文学家。埃拉托斯特尼最重要的贡献是设计出经纬度系统，计算出地球的直径

[7]皮埃尔・德・费马（姓氏依发音亦作费尔玛。Pierre de Fermat，1601 年 8 月 17 日－ 1665 年 1 月 12 日），法国律师、业余数学家（也被称为数学大师、业余数学家之王）。他在数学上的成就不低于职业数学家，似乎对数论最有兴趣，亦对现代微积分的建立有所贡献。

[8]在数论上，卡迈克尔数是正合成数 n，且使得对于所有跟 n 互素的整数 b，$b^{n-1} \equiv 1 \pmod n$

2.2 欧拉定理

2.2.1 欧拉函数

$\phi(n)$ 称为欧拉函数[9]，表示的是自然数中不大于 n 的与 n 互质的自然数的个数.

欧拉函数具有以下性质：

1. 如果 $n = p^k$(p 为一个素数)，则:

$$\phi(n) = \phi(p^k) = p^k - p^{k-1} = p^k(1 - \frac{1}{p})$$

根据算数基本定理，由于 n 的唯一的质因子分解形式为 p^k，对于其他的自然数 $m(< n)$，如果 $gcd(m, n) \neq 1$，则 m 的质因子分解中必然包含 p，即 m 必然为 p 的倍数形式，这样的数字共有 p^{k-1} 个，即 $1 \cdot p, 2 \cdot p, \cdots, p^{k-1} \cdot p$，其他的数字都和 p^k 互素，因此根据 ϕ 函数的定义有 $\phi(p^k) = p^k - p^{k-1}$.

2. 欧拉函数是积性函数
如果 $gcd(m, n) = 1$，则 $\phi(mn) = \phi(m) \cdot \phi(n)$. 这条定理在证明的过程中使用到了**中国剩余定理**，证明较为繁杂，不赘述.

3. 根据算数基本定理，每个数字都能够唯一地分解为一系列质因子的乘积

$$n = p_1^{k_1} p_2^{k_2} \cdots p_r^{k_r}$$

这里 $p_1 < p_2 < \cdots < p_r$

根据性质 2

$$\begin{aligned} \phi(n) &= \phi(p_1^{k_1} p_2^{k_2} \cdots p_r^{k_r}) \\ &= \phi(p_1^{k1})\phi(p_2^{k_2}) \cdots \phi(p_r^{k_r}) \end{aligned}$$

再根据性质 1

$$\phi(n) = p_1^{k_1} p_2^{k_2} \cdots p_r^{k_r}(1 - \frac{1}{p_1})(1 - \frac{1}{p_2}) \cdots (1 - \frac{1}{p_r})$$

因此对于一个整数 n，如果我们能够得到其质因子分解形式，那么就可以得到 $\phi(n)$ 的值. 该性质以下两种形式比较重要:

- 当 n 为素数的时候，$\phi(n) = n(1 - \frac{1}{n}) = n - 1$,
- 当 n 为两个素数乘积 $(n = pq)$ 的时候，$\phi(n) = pq(1 - \frac{1}{p})(1 - \frac{1}{q}) = (p - 1)(q - 1)$.

2.2.2 欧拉定理

在数论中，欧拉定理（也称费马-欧拉定理或欧拉 ϕ 函数定理）是一个关于同余的性质，欧拉定理表明，若 n,a 为正整数，且 n 和 a 互素 $(gcd(n, a) = 1)$)，则：

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

即 $a^{\phi(n)}$ 与 1 在模 n 下同余

欧拉定理的证明过于复杂，这里不赘述.

[9]莱昂哈德·欧拉（德语：Leonhard Euler，台湾旧译尤拉，1707 年 4 月 15 日— 1783 年 9 月 18 日）是一位瑞士数学家和物理学家，近代数学先驱之一，他一生大部分时间在俄国和普鲁士度过。

当 n 为素数的时候，欧拉定理的形式为

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

即费马小定理.

3 RSA 算法

3.1 密钥生成

首先我们需要明确，无论什么报文最终在计算机中都是表示为二进制的形式，所以我们所谓的加密信息就是加密一串数字，解密信息也就是解密一串数字.

RSA 加密算法的具体算法流程如下：

1. 随机生成两个大素数 p 和 q，RSA 实验室推荐，公司使用时，p 和 q 的乘积为 1024 比特的数量级；
2. 计算 n 和 $\phi(n)$，其中 $n = p \cdot q, \phi(n) = \phi(pq) = (p-1)(q-1)$；
3. 求出一个数 d，使得
$$ed \equiv 1 \pmod{\phi(n)}$$
即 d 为 e 关于 $\phi(n)$ 的模逆元；
4. 生成的公钥为 (n, e)，私钥为 (n, d).

3.2 通信流程

假设 Alice 想要和 Bob 通信，Alice 生成的公钥为 (n, e)，私钥为 (n, d) Bob 需要给 Alice 发送报文 m，使用 $< n, e >$ 进行加密，密文 $c = m^e\ mod\ n$，Alice 使用 $< n, d >$ 进行解密，明文 $m = c^d\ mod\ n$.

RSA 的工作原理：

$$\begin{aligned}
&\because c = m^e\ mod\ n \\
&\therefore c = m^e + kn && \text{(k is a constant)} \\
&\therefore c^d = (m^e + kn)^d = m^{ed} + k'n && \text{(k' is a constant)} \\
&\because ed \equiv 1 \pmod{\phi(n)} \\
&\therefore ed = h\phi(n) + 1 && \text{(h is a constant)} \\
&\therefore c^d = m^{h\phi(n)+1} + k'n \\
&\therefore c^d = m \cdot (m^{\phi(n)})^h + k'n \\
&\because gcd(n, m) = 1 \\
&\therefore m^{\phi(n)} \equiv 1 \pmod{n} && \text{(Euler Theroem)} \\
&\therefore m^{\phi(n)} = 1 + ln && \text{(l is a constant)} \\
&\therefore c^d = m \cdot (1 + ln)^h + k'n \\
&\therefore c^d = m \cdot (1 + k''n) + k'n && \text{(k” is a constant)}
\end{aligned}$$

$$\therefore c^d = m + k'''n \qquad \text{(k"' is a constant)}$$

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

□

一个使用 RSA 算法加密的例子：

取两个素数 $p = 11, q = 13$，p 和 q 的乘积为 $n = p \times q = 143$，算出 $\phi(n) = (p-1) \times (q-1) = 120$；再选取一个与 $\phi(n)$ 互质的数 $e = 7$，则公钥为 <n, e> = <143, 7>;

对于 e 值，使用欧几里得扩展算法可以算出其逆：$d = 103$，验证 $(e \times d) \mod \phi(n) = (7 \times 103) \mod 120 = 1$ 成立，则私钥为 <n, d> = <143, 103>;

假设发送方要发送的明文 $m = 85$，发送方已经得到了接收方的公钥为 <143, 7>，于是发送方算出加密后的密文 $c = m^e \mod n = 85^7 \mod 143 = 123$ 并发送给接收方;

接收方收到报文之后使用私钥进行解密：$c^d \mod n = 123^{103} \mod 143 = 85$，所以接收方可以得到发送方发送给他的真实信息 m，实现了安全通信。

在实际使用的过程中 m 值的长度一般要远大于 n 的长度，因此实际加密 m 时，需要首先把 m 分成比 n 小的数据分组，再对每组单独加密和解密。

3.3 RSA 算法的加密强度

目前密码的破译主要有两种方法，方法之一是密钥穷举法，方法之二是密码分析法. 由于 RSA 加密和解密过程都是用指数计算，其计算工作量巨大，以现代计算机的算力，可以认为使用密钥穷举法破解是根本不可能的，因此要对 RSA 算法加密后的信息进行破译只能采用密码分析法，常用的一种途径是想办法计算出 d，下面证明计算出 d 和对 n 进行质因数分解具有相同的难度：

证明. ”⇒”: 如果能够通过不对 n 进行大数分解确定 d，那么分解对 n 进行质因数分解就变得容易起来. 如果我们知道 $\phi(n)$，根据 $\phi(n) = (p-1)(q-1)$ 和 $n = p \times q$，就可以得到 $p+q = n - \phi(n) - 1$，进而通过求解一个一元二次方程就可以得到 p 和 q，分解 n 完成.

G.L.Miler 在 1975 年指出，利用 $\phi(n)$ 的任何倍数都可以轻松分解出 n 的因子. 因此根据 $ed \equiv 1 \pmod{\phi(n)}$，可以得到一个 $\phi(n)$ 的倍数 $ed - 1$，也就是说只要计算出 d 就能够完成对 n 的质因数分解.

”⇐”: 如果能够对 n 进行质因数分解得到 p 和 q，那么根据 $\phi(n) = p \times q$ 既可以得到 $\phi(n)$，进而通过 $ed \equiv 1 \pmod{\phi(n)}$ 得到 d，也就是说破解 RSA 密码不会比对 n 进行质因数分解更困难. □

综上所述，破解 RSA 密码和质因数分解同样困难，如果参数 p, q, e 选取恰当的话，RSA 的加密强度就取决于抗因子分解强度.

3.4 大数分解的难度

大数分解是一件非常困难的事情，著名数学家费马和勒让德都曾经研究过质因数分解的算法，现代的质因数算法一般都是勒让德方法的扩展，其中，R.Schroeppel 算法是比较好的算法，但是用此法对 n 进行质因数分解仍然需要大约 $e^{\sqrt{\ln n \ln\ln n}}$ 次运算，可见分解 n 的运算次数随着 n 的位数的增加会指数倍增加，对于不同长度的二进制数 n，Schroeppel 算法分解 n 时所需的运算次数以及使用一台 1s 运算 1 亿次的计算机破解所需要的时间如表 1 所示：

RSA 算法的可靠性依赖于大数分解的难度，换言之，对一个极大整数做质因数分解愈困呐，RSA 算法愈可靠，如果能够找到一个快速质因数分解的算法，那么 RSA 算法将不再可靠，但是质

表 1: 使用 *Schroeppel* 算法分解因子需要的运算次数

n 的二进制位数	256	512	1024	2048
运算次数	1.4627×10^{13}	6.6869×10^{19}	4.4237×10^{29}	1.2113×10^{44}
破解时间/年	4.6382×10^{-3}	2.1204×10^{4}	1.4028×10^{14}	3.8411×10^{28}

因数分解一般被认为是 NPC 问题，尽管还并未再理论上论证，但是经过千百年来的众多学者研究，迄今为止还没有找到一个有效的算法，目前破解 RSA 密码只能依赖于现代的计算机技术，随着近年来计算机运算速度以及并行计算技术的发展，破解低位的 RSA 密码已经称为可能，但是只要 n 的长度达到一定要求，RSA 加密仍然是相当安全的.

3.5 改进的 RSA

3.5.1 RSA 算法存在的问题

RSA 算法虽然安全性好，但是它也存在一些不足之处：

1. 需要经过大量的计算来计算密钥，而且在加密和解密的过程中，都需要计算某一个整数的模 n 的整数次幂，需要较高的计算开销；
2. 目前尚没有较好的方法生成两个大素数 p 和 q，一般使用随机生成和素性检测的方法生成大素数，因此生成 p 和 q 的效率较低；
3. 为了保证 RSA 算法的安全性，一般使用大密钥空间，这将会进一步降低 RSA 算法的执行效率。

3.5.2 大素数生成方法

素数生成的核心问题是判断一个数是否是一个素数。目前，生成素数的方法主要有确定性素数生成和概率性素数生成，确定性素数生成的优点是能够保证生成的数一定是素数，但是这种方法生成的素数具有一定的规律性，可能会被攻击者破解；而概率性素数没有规律可循，而且速度较快，但是这样生成的数一般都是伪素数，还需要对生成数的素性进行检验。

在构造 RSA 算法中的大素数是，一般首先利用概率性素数生成方法生成伪素数，然后再利用素性检验的办法进行检验，素性检验的步骤如下：

1. **预处理** 由于进行素性检验会比较耗时，所以在进行素性检验之前，首先使用前文提及的埃拉托斯特尼筛法过滤掉一部分合数；
2. **素性检测** 素性检测就是判断一个整数是否为素数，前文已经提及若干办法，在实际中应用的一般是 Miller Rabin 算法[10]，Miller Rabin 算法本质上是费马素性检验的一个变体，为了介绍 Miller Rabin 算法，首先介绍一下二次探测定理：

   若 p 为素数，$a^2 \equiv 1 \pmod p$，则 $a \equiv 1 \pmod p$ 或者 $a \equiv p-1 \pmod p$.

   假设要判定的数是 n，Miller Rabin 算法的流程如下：

   (a) 把 $n-1$ 分解为 $2^r m$ 的形式;

[10]米勒-拉宾素性检验是一种素数判定法则，利用随机化算法判断一个数是合数还是可能是素数。

(b) 随机选择一个数 $a(1<a<n-1)$，计算 $x=a^m \mod n$;

(c) 重复执行 $x=x^2$ r 次，同时结合二次探测定理进行判断：如果自乘之后的数满足 $\mod n=1$，但是之前的数不满足 $\mod n=1$ 且不满足 $\mod n=p-1$，则说明 n 是合数；

(d) 执行 r 次之后如果 $a^{r-1} \mod n \neq 1$，则同样说明 n 是合数.

Miller Rabin 算法并不能够保证通过测试的数一定是素数，但是通过多次检测就能够使得错误率足够低，据统计，使 n 通过以 $a(1<a<n)$ 的 Miller Rabin 算法测试的概率约为 $\frac{1}{4}$，选取 k 个小于 n 的正整数进行测试，k 次测试全部通过的概率仅为 $\frac{1}{4^k}$，当 k 足够大的时候，可以认为该事件发生的概率足够小。

3. **素数验证** 采用 Pocklington 定理[11]对通过前面计算得到的伪素数进行验证.

在使用上述算法生成大素数的过程中，首先使用了埃拉托斯特尼筛法排除掉绝大多数合数，再使用多次 Miller Rabin 算法进行素性检测，如果测试次数足够多，基本上可以判定得到的数就是素数，同时配合 Pocklington 定理，可以进一步提高素数的可靠性。

参考文献

[1] 向进. Rsa 加密算法的安全性分析. *吉首大学学报: 自然科学版*, 1(32):42–43, 2011.

[2] 张宏, 刘晓霞, and 张若岩. *RSA 公钥密码体制中安全大素数的生成*. PhD thesis, 2008.

[3] 石井, 吴哲, 谭璐, 王昊鹏, and 王娜. Rsa 数据加密算法的分析与改进. *济南大学学报: 自然科学版*, 27(3):283–286, 2013.

[11] 普罗斯定理是数论的一个定理，可以判断普罗斯数 (普罗斯数，也就是满足 $k2^n+1$ 形式的数) 是否是质数