RSA 算法中快速生成大素数方法的改进

王　萍[1]，廖芳燕[1]，廖芳午[1]，张树贵[2]
（1. 成都理工大学　信息管理学院，四川　成都　610059；2. 成都理工大学　环境与土木工程学院，四川　成都　610059）

[摘　要]根据同余理论提出一种快速试除法来更快地判断一个大整数是否能被小素数整除，从而进一步提高 RSA 算法中所需要的大素数的生成速度.

[关键词]RSA；大素数；同余
[中图分类号]O156.1　[文献标识码]A　[文章编号]1673－8012(2009)03－0009－03

1　RSA 算法简介

RSA 算法是一种既能用于加密又能用于数字签名的公钥算法[1]. 其原理是：选取两个大素数 p 和 q，算出 $n = p \cdot q$，$\phi(n) = (p-1)\cdot(q-1)$，再随机选取一个正整数 e，使之满足 $1 < e < \phi(n)$，并且 $\gcd(\phi(n), e) = 1$，然后计算 d，满足 $d \cdot e \equiv 1 \bmod \phi(n)$. 其中，$n$ 和 e 为公钥，d 为私钥. 根据 RSA 算法，设要加密的明文为 m，c 为密文. 加密过程为：$c \equiv m^e \pmod n$. 解密过程为：$m \equiv c^d \pmod n$. 由此可见，RSA 算法的安全性是基于大数 n 的分解难度[3]. 目前，因子分解速度最快的方法，其时间复杂度为：$\exp(\mathrm{sqrt}(\ln(n)\ln\ln(n)))$. 随着 n 长度的增加，分解因子所需时间成指数增加. 现在 129 位十进制数的模数是能分解的临界数. 因此，n 应该大于这个数. 可见，RSA 的安全性取决于大整数的长度.

2　生成大素数的方法

由于素数在正整数序列中分布不规则，无法用公式直接计算出一个指定长度的大素数，所以当前采用的方法是随机产生一个大整数，再对其做素性检测. 常见的素性检测法有 Miller－Rabin 检测[4]、Solovay－Strassen 检测法和 Lehman 检测法. 目前，提高素数生成速度的方法是用 100 以内的小素数首先进行筛选[5]. 这是因为判断大整数能否被小素数整除的时间要比各种概率测试法短得多，一般说来，这种筛选可以排除大整数不是素数 76% 的可能性；之后再进行 5 次 Miller－Rabin 检测，那么这个大数不是素数的概率就会降到 1/1 000 以下，这样就加快了大素数生成的速度.

3　对传统筛选法的改进

以上提到的用小素数试除法是一种有效的排除合数的方法. 那么怎样可以更快地判断一个随机大整数是否能被 100 以内的小素数除尽呢？根据同余的理论，我们可以将大整数“分块”相加再除以小素数，若能除尽则该大整数肯定能被除尽.

3.1　快速试除法

我们知道，判定一个整数能否被 3 整除，可以把每一位上的数字相加再除以 3，如果能除尽则该数能被 3 整除，这比直接用这个数去除 3 要快得多. 其原理是：$10 \equiv 1 \pmod 3$，因此 $10^k \equiv 1 \pmod 3$[6]. 任意一个十进制的整数 $(a_k a_{k-1}\cdots a_1 a_0)_{10}$ 可表示如下：

$$\begin{aligned}&(a_k a_{k-1}\cdots a_1 a_0)_{10}\\ &= a_k \cdot 10^k + a_{k-1}\cdot 10^{k-1} + \cdots + a_1 \cdot 10 + a_0\\ &\equiv a_k + a_{k-1} + \cdots + a_1 + a_0 \pmod 3.\end{aligned}$$

因此，只要把大整数各位上的数字加起来模 3，就可以判断该数能否被 3 整除. 如果这个数足

[收稿日期]2009－01－12
[作者简介]王萍(1984－)，女，四川乐山人，硕士研究生，主要从事信息安全中的计算方法研究.

够大使得各位加起来还是位数很大,那么可以继续使用这个方法来判定. 同理,我们也可以判定一个大整数是否能被 7、11、13 等小素数整除. 我们先看怎样判定大数能被 11 整除:

因 $10 \equiv -1 (\mathrm{mod}\ 11)$,故整数

$$\begin{aligned}&(a_k a_{k-1} \cdots a_1 a_0)_{10}\\ &= a_k \cdot 10^k + a_{k-1} \cdot 10^{k-1} + \cdots + a_1 \cdot 10 + a_0\\ &\equiv a_k (-1)^k + a_{k-1}(-1)^{k-1} + \cdots - a_1 + a_0 (\mathrm{mod}\ 11).\end{aligned}$$

也就是说,用偶数位数字之和减去奇数位数字之和得到的结果如果能被 11 整除,则该数能被 11 整除.

再看怎样判定能被 7 整除:因 $10^6 \equiv 1 (\mathrm{mod}\ 7)$,故整数可写成:

$$\begin{aligned}&(a_k a_{k-1} \cdots a_1 a_0)_{10}\\ &= a_k \cdot 10^k + a_{k-1} \cdot 10^{k-1} + \cdots + a_1 \cdot 10 + a_0\\ &\equiv (a_0 + 10a_1 + 10^2 a_2 + \cdots + 10^5 a_5) +\\ &\quad 10^6 (a_6 + 10a_7 + 10^2 a_8 + \cdots + 10^5 a_{11}) +\\ &\quad (10^6)^2 (a_{12} + 10a_{13} + 10^2 a_{14} + \cdots + 10^5 a_{17}) + \cdots\\ &= (a_5 a_4 a_3 a_2 a_1 a_0)_{10} + (a_{11} a_{10} a_9 a_8 a_7 a_6)_{10} +\\ &\quad (a_{17} a_{16} a_{15} a_{14} a_{13} a_{12})_{10} + \cdots (\mathrm{mod}\ 7).\end{aligned}$$

我们发现,由于 $10^6 \equiv 1 (\mathrm{mod}\ 7)$,就把大整数"分块",自低位起,6 位数为"一块",然后把分块后的很多个 6 位数相加再来模 7. 如果相加后还是大于 6 位的大整数,可以再用这个方法分块相加算出最后结果,如果能整除 7,则这个大整数可以整除 7. 这样比直接把一个上百位的大整数去除 7 要快.

由上面的例子,我们得到规律:只要找到 10 模一个素数的阶就可以把大整数分块.

3.2 整数的阶

定义:

设 $a, m \in \mathbf{Z}^+$,且 $(a,m) = 1$,满足 $a^x \equiv 1 (\mathrm{mod}\ m)$ 的最小正整数 x 称为 a 模 m 的阶[7].

我们记 a 模 m 的阶为 $\mathrm{ord}_m a$. 根据欧拉定理,这样的 x 是一定存在的. 因为满足上述条件的情况下有 $a^{\varphi(m)} \equiv 1 (\mathrm{mod}\ m)$,$\varphi(m)$ 是欧拉函数,那么有 $\mathrm{ord}_m a \mid \varphi(m)$.

现在,我们的问题转化为找 $\mathrm{ord}_p 10 = ?$. 其中,p 是 100 以内除开 2 和 5 的小素数. 这里显然 $(10,p) = 1$,只要找到 $\mathrm{ord}_p 10$,就可以把大整数分块相加再除素数. 目前还没有发现可以直接计算 $\mathrm{ord}_m a$ 的方法. 因此,我们要判定一个上百位的大整数是否能被 100 以内的小素数整除只能根据欧拉定理把它分为 $p-1$ 块相加再除 p. 但我们注意到,100 以内的小素数共有 25 个,而且越往后越大,那么把大整数分成 p 块再相加就失去意义了,要想办法把分块的长度缩小.

3.3 缩短整数分块长度的方法尝试

根据费马小定理:a, p 为正整数,且 $(a,p) = 1$,那么有:$a^{p-1} \equiv 1 (\mathrm{mod}\ p)$. 因此,

$$\begin{aligned}&a^{p-1} - 1 \equiv 0 (\mathrm{mod}\ p)\\ &\Rightarrow (a^{\frac{p-1}{2}} + 1)(a^{\frac{p-1}{2}} - 1) \equiv 0 (\mathrm{mod}\ p).\end{aligned}$$

那么有:$a^{\frac{p-1}{2}} \equiv -1 (\mathrm{mod}\ p)$ 或者 $a^{\frac{p-1}{2}} \equiv 1 (\mathrm{mod}\ p)$.

由此可见,我们可以把大整数分为 $\frac{p-1}{2}$ 块,这里的 p 是奇素数.

如果 $a^{\frac{p-1}{2}} \equiv -1 (\mathrm{mod}\ p)$,则分块后,由低位块起减前一块再加再前面一块,依此类推,最后的结果再除 p 即可.

例如:$a = 10, p = 7$,容易计算 $10^{\frac{7-1}{2}} \equiv -1 (\mathrm{mod}\ 7)$,因此,可以把大整数表示为:

$$\begin{aligned}&(a_k a_{k-1} \cdots a_1 a_0)_{10}\\ &= a_k \cdot 10^k + a_{k-1} \cdot 10^{k-1} + \cdots + a_1 \cdot 10 + a_0\\ &\equiv (a_0 + 10a_1 + 10^2 a_2) - 10^3 (a_3 + 10a_4 + 10^2 a_5) +\\ &\quad (10^3)^2 (a_6 + 10a_7 + 10^2 a_8 - \cdots\\ &= (a_2 a_1 a_0)_{10} - (a_5 a_4 a_3)_{10} + (a_8 a_7 a_6)_{10} -\\ &\quad \cdots (\mathrm{mod}\ 7).\end{aligned}$$

如果 $a^{\frac{p-1}{2}} \equiv 1 (\mathrm{mod}\ p)$,那么可以直接把大整数分为 $\frac{p-1}{2}$ 相加再除 p 即可. 那么如何快速判断 $a^{\frac{p-1}{2}} \equiv -1 (\mathrm{mod}\ p)$ 还是 $a^{\frac{p-1}{2}} \equiv 1 (\mathrm{mod}\ p)$?

根据欧拉定理[2] 的推论,p 为一个奇素数,且 $(a,p) = 1$,如果 a 是模 p 的二次剩余,那么有:$a^{\frac{p-1}{2}} \equiv 1 (\mathrm{mod}\ p)$. 因此,我们的问题转化为判断 10 是否是 100 以内奇素数 p 的二次剩余.

判断一个数是否是另一个数的二次剩余,我们可以引入勒让德符号,其定义如下:p 为一个奇素数,a 为一个整数,且 $(a,p) = 1$,那么勒让德符号 $\left(\frac{a}{p}\right)$ 被定义为:

$$\left(\frac{a}{p}\right) = \begin{cases}1, \text{如果 } a \text{ 是 } p \text{ 的二次剩余;}\\ -1, \text{如果 } a \text{ 不是 } p \text{ 的二次剩余.}\end{cases}$$

我们要计算 $\left(\frac{10}{p}\right)$ 的值. 根据勒让德符号的计算法则,有 $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$, $\left(\frac{a}{p}\right) \equiv$

$a^{\frac{p-1}{2}} \pmod p$. 如果 p 为奇素数,a,b 为整数,那么就有:$\left(\frac{10}{p}\right) = \left(\frac{2}{p}\right)\left(\frac{5}{p}\right)$. 根据定理:$p$ 为一个奇素数有:$\left(\frac{2}{p}\right) = (-1)^{(p^2-1)/8} = \begin{cases} 1 \\ -1 \end{cases}$,可以快速判定 $\left(\frac{2}{p}\right)$ 的值;而 $\left(\frac{5}{p}\right) \equiv 5^{\frac{p-1}{2}} \pmod p$. 这样,我们就可以快速判定 $\left(\frac{10}{p}\right)$ 的值,从而判定 10 是不是 p 的二次剩余.

以上的尝试,至少可以把分块长度缩短到 $\frac{p-1}{2}$,但最好的办法是找到 $\mathrm{ord}_m a$ 的方法. 希望在将来的研究中,可以找到求出一个整数模奇素数 p 的阶的方法,进一步缩短分块长度,降低计算的难度,提高大素数生成的速度.

4 结语

本文介绍了 RSA 算法的原理,指出其安全性在于生成大素数的长度,并根据同余的相关理论提出一种快速试除法来提高大素数生成的速度,并指出了进一步改进此法的研究方向.

[参考文献]

[1]章照止. 现代密码学基础[M]. 北京:北京邮电大学出版社,2004:154-158.

[2]裴定一,祝跃飞. 算法数论[M]. 北京:科学出版社,2002:21-23.

[3]王英. RSA 算法中大素数的快速生成方法[J]. 湖南科技学院学报,2005,26(5):14-16.

[4]刘明华,余启港. RSA 公钥密码算法中大素数的生成及素性检测[J]. 中南民族大学学报(自然科学版),2004,23(4):94-96.

[5]游新娥. RSA 算法中安全大素数生成方法研究与改进[J]. 北京电子科技学院学报,2007,15(2):14-16.

[6]Kennetj H Rosen. Elementary number theory and its applications(fourth edition)[M]. 北京:机械工业出版社,2004:146-157.

[7]Song Y Yan. Number theory for computing(2th edition)[M]. 世界图书出版公司, 2002:139-159.

Improvement on the rapid generation algorithm of large prime number in RSA algorithm

WANG Ping[1], LIAO Fang-yan[1], LIAO Fang-wu[1], ZHANG Shu-gui[2]

(1. *College of Information Management, Chengdu University of Technology, Chengdu Sichuan* 61005, *China*;
2. *College of Environment and Civil Engineering, Chengdu University of Technology, Chengdu Sichuan* 610059, *China*)

Abstract: According to the congruence theory, a fast division was proposed to judge whether a great integer can be divided exactly by the small prime numbers in order to enhance the production speed of big prime numbers in RSA algorithm.

Key words: RSA; big prime number; congruence

(责任编辑 穆 刚)