文章编号: 1673 - 0062(2006)02 - 0096 - 04

大整数素性的计算机测试和软件实现

谢文平,陈大钊

(邵阳学院 数学系,湖南 邵阳 422000)

摘 要:探索和研究了素数的寻找及其素性测试的理论方法,给出了由 Atkin和 Morain提出的确定性素性测试方法及其软件实现,即椭圆曲线素性测试方法 (ECPP). 最后通过与另一确定性测试方法 Jacobi Sum测试方法进行比较,取得了比较理想的结果.

关键词:素性测试; Jacobi Sums测试; ECPP测试

中图分类号: O29 **文献标识码**: B

The Machine Testing and Software Realizing of Primality of Large Number

XIE Wen-ping, CHEN Da-zhao

(Department of Mathematics, Shaoyang College, Shaoyang, Hunan 422000, China)

Abstract: In this paper, we have researched some theories of finding primes and proving primality. And we introduced Elliptic Curve Primality Proving (ECPP) put forward by A. O. L. Atkin and F. Morain, and programmed the ECPP software. At last after we compared with Jacobi Sum Proving Method, the results we got show that ECPP really prove the primality good.

Key words: proving primality; Jacobi Sums Proving; ECPP

0 引言

随着公钥密码体制的发展,大整数 (例如伪随机素数 $2^{20996011} - 1$)的分解困难的特性已经在一些安全领域,诸如数字签名、密钥管理、身份认证等方面得到了应用. 至今为止,最常用的公钥密码体制是 Rivest, Shamir和 Adlema于 1978发明的 RSA密码体制,它的安全性便基于大整数的分解. 还有许多基于有限域上离散对数问题的密码体制,也是基于大整数难分解的[1].

如何确定一个大数 (大于 10 000 000 000),特别是伪素数是否为真的素数已经成为安全领域的一个重要的课题. 最简单的素数测试方法是试除法,该方法是判断素数的充要条件,对于小整数 (例如不大于 10 000 000 000)可以很快的判断,但随着大整数的位数的增大,其判断将变得现实不可行,且当数超过 100位以后,计算时间将随着位数成指数增长[2]. 自然人们会寻找其它理论方

收稿日期: 2006 - 03 - 01

作者简介:谢文平 (1968 -),男,湖南洞口人,邵阳学院数学系讲师. 主要研究方向:计算机网络和算法.

法,1983年 Adleman等人利用分圆域中的分圆整数的 Jacobi和来对大数进行素性判别,1986年,A. O. L. Atkin提出了 ECPP方法,等等[3-4]. 目前关于素性测试的方法,大致可以分为概率性测试和确定性测试,概率测试的理论依据主要是基于 Fermat定理[5]及其变形;若测试建立在素数判断的充要条件基础上,就称为确定性素性测试. 这些理论中,有些已经在计算机上实现了软件编程,然而关于 ECPP的软件实现在国内还没有见到相关的计算机测试方案. 基于该点,本文简介了 ECPP理论思想,给出了 ECPP测试的软件实现,并对 ECPP方法的计算机测试实现了仿真,给出了仿真结果. 希望这些研究工作有利于:一方面能够加强基于大数分解的密码系统的安全性,另一方面也提供对这类密码系统攻击的一些特性研究.

1　ECPP测试算法

1.1　ECPP原理

目前比较流行的 ECPP测试算法是由 Atkin和 Morain提出的,其思想是利用椭圆曲线上的点所构成的 Abel群来替换常用的模 n的剩余类所构成的乘法群[4]考虑 Z/NZ上的椭圆曲线 Weierstraβ方程

$$y^2 = x^3 + ax + b,\ a, b \in Z/NZ;$$
$$(4a^3 + 27b^2) \in (Z/NZ)^*$$

其中 $N > 1, (N, 6) = 1$

假设 N为素数,我们在曲线上增加一些点. 如果 N为素数,则乘法群 (N/NZ)上只有加法、减法、乘法和除法 4种运算;如果 N不是素数,那么某些元素之间就不能进行除法运算. 利用这个现象,便可以对大整数进行素性测试.

下面我们假设 E为 (N/NZ)上的(离散)椭圆曲线集合,记为 $E(Z/NZ)$.

首先需要确定离散曲线 E上的点的个数,即 E的势,我们记之为 m. 为了求 m,需要构造一个复数乘法的二次域 $K = Q(\sqrt{D})$,这里 D为负数,且模 4为 0或 1,称之为判别系数,N可以表示为域 K中的两个元素的乘积. ECPP关键在于判别系数 D的寻找,且使 N能够在 $K = Q(\sqrt{D})$空间中分解为两个元素(复数)之积.

为了提高测试速度,通常需要预先建立一张判别系数表,ECPP在执行过程中将去搜索适当的 D值. 使用 Comacchia算法[2]可以取得这样的 D值.

得到 D值后,那么方程 $x^2 + dy^2 = 4p$,其中 $d = -D, d > 0$的两个根可以表示为:$\frac{x \pm y\sqrt{D}}{2}$,记 $\pi = \frac{x + y\sqrt{D}}{2}$,$\bar{\pi}$为 π的共轭. 显然有 $\pi\bar{\pi} = p$,由文献[2]中的定理 7.2.15我们可以有结论:如果 N为素数,那么 $m = N + 1 - \pi - \bar{\pi} = N + 1 - x$

记 $O_D = Z + \left(\frac{D + \sqrt{D}}{2}\right) Z$为整数环,$\bar{\omega}(D)$为 O_D的单位根个数,因此有:

$$\bar{\omega}(D) = \begin{cases} 2, & if \quad D < -4 \\ 4, & if \quad D = -4 \\ 6, & if \quad D = -3 \end{cases}$$

在 O_D中,N也可以被分解为 2个数的乘积,因此 $\omega(D) \mid N - 1$,则存在 $(N-1)/2$(当 $D = -3$时,该值为 $(N-1)/3$)个值 $g \in Z/NZ$使得对每个素数 $p, p \mid \omega(D)$,满足 $g^{(N-1)/p} \neq 1$. 任意选择一个值 g,如果 $D = -4$,则有 4个与 $E(Z/NZ)$同构的椭圆曲线集合,它们对应的椭圆曲线方程为

$$y^2 = x^3 - g^k x,\ k = 0, 1, 2, 3$$

(注:对于 $D = -3$有 6个这样的同构,它们的方程为 $y^2 = x^3 - g^k$, $k = 0, 1, 2, 3, 4, 5$)

如果 $D < -4$,令 $c = j/(j - 1728)$,这里 $j = \left\{j \frac{D + \sqrt{D}}{2}\right\}$为 O_D的 j-不变量(参考文献[2]的引理 7.2.1),此时有 2个同构,它们对应的方程为:

$$y^2 = x^3 - 3cg^{2k}x + 2cg^{3k},\ k = 0, 1$$

得到椭圆曲线后,就可以通过寻找曲线上的点,利用 Pocklington定理[2]来验证 N是否为素数.

最后值得一提的是,对某些值 eps > 0, ECPP的时间复杂度甚至可以达到 $O((\log n)^{4 + \text{eps}})$[6],几乎为多项式时间. 这也是 ECPP成为目前主流测试方法的原因之一.

1.2　算法描述

设 N为一个大于 1,且与 6互素的正奇数. 假设已有一系列负判别系数 $\{D_n\}$, $n \geq 1$表,下面给出测试过程:

Step 1　过滤

采用 Rabin-Miller进行概率测试,如果测试为合数,则 N为合数,软件返回,否则 N为强伪随机数,进入下一步.

Step 2　初始化

令 $i = 0, n = 0,$,并记 $N_i = N$

Step 3　试除法

如果 $N_i < 2^{30}$,用试除法判断 N_i,如果不是素数,则转到 Step 15

Step 4 选择下一个判别系数

令 $n++, D = D_n$,如果 Jacobi值 $(D/N) \neq 1$,则继续在表中寻找判别系数 D,否则使用 Cornacchia's Algorithm直到找到一个新的判别系数为止.

Step 5 分解 m,并判断 m 是否符号要求

分别对 $m = N + 1 + x, m = N + 1 - x$进行因子分解(一般先用试除法初判,然后用 pollard ρ和 $p-1$方法分解[2]).如果 $\exists q \mid m$,且 $q > (\sqrt[4]{N_i} + 1)^2$,那么转到 step 6,否则返回 step 4.这里的 q为素数或伪素数(至少要通过 Rabin-Miller测试).

Step 6 计算椭圆曲线

$if(D == -4)\{a = -1; b = 0\}$,

$if(D == -3)\{a = 0; b = -1\}$

对于其它 D值,令 $c = j/(j - 1728) \bmod N_i$以及 $a = -3c \bmod N_i, b = 2c \bmod N_i$

Step 7 寻找值 g

g为模 N_i的的二次非剩余值,此外当 $D = -3$ 时, $g^{(N_i-1)/3} \neq 1 \pmod{N_i}$

Step 8 寻找椭圆曲线上的点 P

随机选取 $x \in Z/N_iZ$,使得 x满足 Legendre-Jacob符号[3] $((x^3 + ax + b)/N_i)$的值为 0或 1.然后使用文献[2]中的算法 1.5.1计算 $y \in Z/N_iZ$,使之满足 $y^2 = x^3 + ax + b$.如果找不到 y,则转到 Step 14.最后置 $k = 0$.

Step 9 寻找正确的曲线

在椭圆曲线 $y^2 = x^3 + ax + b$上计算 $P_2 = (m/q) \cdot P, P_1 = (q) \cdot P$.如果在计算过程中某些除法运算不可能实现,则转到 Step 14;如果 P_1 为零点,则转到 Step12

Step 10 设置变量

令 $k = k + 1$,如果 $k \geq \omega(D)$则转到 Step 14;否则如果 $D < -4$,则令 $a = a \cdot g^2, b = b \cdot g^3$,如果 $D = -4$,令 $a = a \cdot g$,如果 $D = -3$,令 $b = b \cdot g$,转到 Step 8

Step 11 寻找新的椭圆点 P

随机选取 $x \in Z/N_iZ$,使得 x满足 Legendre-Jacob符号[3] $((x^3 + ax + b)/N_i)$的值为 0或 1.然后使用文献[2]中的算法 1.5.1计算 $y \in Z/N_iZ$,使之满足 $y^2 = x^3 + ax + b$.如果找不到 y,则转到 Step 14;如果 P_1 不为零点,转到 Step 10

Step 12 检查 P

如果 P_2 为零点,转到 Step 11

Step 13 递归

令 $i = i + 1, N_i = q$,转到 Step 3

Step 14 结果处理

如果 $i = 0$,则 N不是素数,终止算法.否则置 $i = i - 1$,转到 Step 4

2 计算机仿真测试及结果分析

我们算法是在 VC++6.00环境下实现的,计算机主频 2.0G、内存 512M.在算法实现过程中,有以下几个需要关注的问题:

1)大整数的素性证书链表的建立

每个大整数,通过素性测试算法后可以标志是否为素数.我们需要建立一张递增的素数表,使得后一个大整数的素性证书包含前一个大整数的素性证书.该表的建立可以大大提高 ECPP的速度.在 Jacobi Sum测试方法中,需要花大量的时间来建立和处理素性证书的问题,而在 ECPP算法中,建立和处理几乎是多项式时间.

2)大整数的四则运算实现软件

我们采用了大整数的运算库 BigNum,BigMod两个类,通过它们可以操作各种运算.

3)寻找椭圆曲线上的点

能否快速找到这些点也是 ECPP运算速度的决定因素,由于是随机选取的,所以在不同的执行过程中,可能会消耗不同的时间.无论如何,ECPP算法能够测试很多伪素数的素性,这些是 Dubois-Selfridge-Miller-Rabin等算法办不到的.下面我们给出 ECPP的仿真测试结果.

为便于比较,我们对比于另一种常用的确定行素性测试方法 Jacobi Sum测试方法[3],相互间做了个比较.单机上的运算结果见表 1.

从两种算法测试结果来看,Jacobi Sum 测试效果要好,ECPP在时间上略有优势.产生两种确定性算法的测试结果差异的主要原因在于 ECPP算法的椭圆曲线上的点寻找算法不够精确.从概率上说,为了安全起见,ECPP测试结果要比 Jacobi Sums测试更稳妥.

由于算法针对性很强,表中给出的结果不代表实际运算效果.确定性算法尽管速度慢,但其基本能给出正确的测试结果,对于不大于 1000位以上的数,单机计算时间也是能够承受得起的.

3 结束语

随着计算机的发展,运算速度的提高,寻找素

表 1　两种算法的比较及测试结果

Table 1　Compare of two algorithms and testing result

统计量 100个	Jacobi Sum Test		ECPP Test	
均为 10进制的伪素数	测试为合数的个数	平均测试时间	测试为合数的个数	平均测试时间
50位以下	3	-	3	-
50	0	0.729	0	0.73 s
60	0	1.2 s	0	1.08 s
70	0	2.28 s	0	2.07 s
80	0	3.04 s	0	2.19 s
90	0	5.92 s	1	3.27 s
100	0	7.44 s	1	4.89 s
150	0	36.9 s	2	22.47 s
200	0	111.9 s	6	136.05 s

说明:这些数由伪随机素数生成器生成.生成的数为概率素数. 50位以下的 100个伪素数[7]中,两种算法均找到了相同的 3个复合数,它们是: 118 670 087 467 = 688 969 ×172 243　11 377 272 352 951 = 1 686 511 ×6 746 041　21 569 059 132 741 = 3 283 981 ×6 567 961

数和素性测试的速度会越来越快,测试的位数也将随之扩大,目前利用互联网的联机系统,来适当弥补确定性算法的亚指数及时间的缺点,可以测试 200 000多位的大整数.越来越多的素性将会发现,这是人类进步的标志.

参考文献:

[1] 曾泳泓,成礼智,周敏.数字信号处理的并行算法[M].长沙:国防科技大学出版社, 1999.

[2] Henri Cohen. A Course in Computational Algebraic Number Theory[J]. Springer - Verlag, 1996, 50: 467 - 473.

[3] Cohen H, Lenstra Jr. Primality Testing and Jacobi Sums [J]. Math. Comp., 1984, 42: 297 - 330.

[4] Atkin AOL, Morain F. Elliptic Curves and Primality Proving[J]. Math. Comp., 1993, 61(7): 29 - 68.

[5] A Wiles. Modular Elliptic Curves and Fermat's Last Theorem, [J]. Ann. Math., 1995, 141(3): 443 - 551.

[6] Lenstra Jr. Algorithms in Number Theory[A]. In: Handbook of Theoretical Computer Science, Vol A: Algorithms and Complexity[C]. Amsterdam and New York: The MIT Press, 1990: 673 - 715.

[7] Adleman L M, Pomerance C, Rumely R S. On Distinguishing Prime Numbers from Composite Numbers[J]. Ann. Math., 1983, 17(1): 173 - 206.

(上接第 87页)

作用及其对 PAI-1和 tPA活性的影响[J].天然产物研究与开发, 2003, 15(5): 441 - 445.

[58] 崔彦军,高文军,乔汉臣,等.苦味叶下珠对慢性乙型病毒性肝炎的疗效及其对 T细胞亚群的影响[J].中国医药学报, 1998, 13(5): 74.

[59] 张均倡,张　云,罗上武,等.肝康对小鼠免疫功能的影响[J].中西医结合肝病杂志, 1998, 8(2): 98 - 99.

[60] 万金志,罗上武,张均倡,等.肝丹对小鼠免疫功能影响的实验研究[J].中草药 1998, 29(9): 614 - 615.

[61] 姜素椿,吕占秀.传染病基础与临床[M].北京:军事医学科学出版社, 1999.

[62] 刘锡光.病毒性肝炎实验诊断学[M].北京:人民卫生出版社, 1999.

[63] 牛晓峰,吕居娴,贺浪　,等.陕西叶下珠药用开发研究 Ⅱ.叶下珠人工栽培[J].西北药学杂志, 1995, 10(6): 250 - 252.