**Lenstra, Hendrik W.**, Universiteit van Amsterdam, Netherlands. *Efficient algorithms in number theory.*

**1. Introduction.**

One of the recent developments in algorithmic number theory is the use of *elliptic curves*. In this lecture it is shown how elliptic curves can be used to find the *prime factor decomposition* of large integers. To do this, one must first be able to recognize whether a number is prime (*primality testing*), and next, if it is not, find a non-trivial divisor (*factorization*). Elliptic curves can be applied both to primality testing and to factorization.

**2. Multiplicative methods.**

For older algorithms to do primality testing and factorization, see [4, 6]. Only two of these will be discussed here, in their most rudimentary form, because they are helpful in motivating and understanding the new methods. The two methods that we describe depend on properties of the *multiplicative group*, in particular on the fact that the order of the multiplicative group modulo a prime number $p$ is $p-1$.

*Primality testing.* If an integer $n>1$ is composite then there are many *pseudoprime tests* that $n$ fails to pass, so that the compositeness of $n$ is usually easy to prove. But if $n$ is prime then it passes all pseudoprime tests that it is subjected to. The problem then becomes how to *prove* that $n$ is prime. If one knows a sufficiently large completely factored divisor $s$ of $n-1$ the following classical result can be used.

**Theorem 1.** *Let $n$ be an integer, $n>1$, and $s$ a divisor of $n-1$. Suppose that there is an integer $a$ satisfying*

$$a^{n-1}\equiv 1 \bmod n,$$
$$\gcd(a^{(n-1)/q}-1,\ n)=1 \quad \textit{for each prime divisor } q \textit{ of } s.$$

*Then every positive divisor $p$ of $n$ is $1 \bmod s$, and if $s\geq\sqrt{n}$ then $n$ is prime.*

To prove this one may assume that $p$ is *prime*. The element $a^{(n-1)/s}$ has order $s$ in the multiplicative group mod $p$. By Lagrange's theorem in group theory this implies that $s$ divides the order of the group, which is $p-1$. The theorem follows.

The basic shortcoming of the primality test based on Theorem 1 is that it can only prove the primality of prime numbers $n$ for which $n-1$ has a large divisor that one knows to factor completely. This is the case, for example, if $n-1$ has many small prime factors, and sometimes also if $n-1$ is the product of a small number and a large prime number $q$; in the latter case one can attempt to prove the primality of $q$ recursively.

*Factorization.* The *Pollard $p-1$-method* attempts to find a non-trivial divisor of an integer $n>1$ in the following way. Pick $a\in\mathbb{Z}/n\mathbb{Z}$ at random, and calculate, by repeated squarings and multiplications mod $n$, integers $a_k$ that are congruent to $a^{k!}$ mod $n$, for $k=1,2,\ldots$ . In addition, calculate $\gcd(a_k-1,n)$ for each $k$, using Euclid's algorithm, and stop if this gcd is a non-trivial divisor of $n$.

The reason that one expects this to work sometimes is as follows. Suppose that $n$ has a prime divisor $p$ for which $p-1$ is built up from small prime factors only. Then $p-1$ divides $k!$ for a relatively small value of $k$. If now $p$ does not divide $a$, then again by Lagrange's theorem the order of $a$ in the multiplicative group mod $p$ divides $k!$. Therefore $p$ divides $a_k-1$, so it divides $\gcd(a_k-1,n)$ as well. Hence if this gcd is different from $n$ it is a non-trivial divisor of $n$.

Along these lines it can be proved that the Pollard $p-1$-method is good in discovering prime divisors $p$ of $n$ for which $p-1$ has no large prime factors. It can also be proved that if $n$ has no such prime divisor $p$ then the method is unlikely to work within a reasonable amount of time.

**3. Elliptic curves.**

Let $n$ be a positive integer. Consider the set of all triples $(x,y,z)\in(\mathbb{Z}/n\mathbb{Z})^3$ for which $x,y,z$ generate the unit ideal of $\mathbb{Z}/n\mathbb{Z}$. The group of units $(\mathbb{Z}/n\mathbb{Z})^*$ acts on this set by $u(x,y,z)=(ux,uy,uz)$. The orbits under this action are the *points of the projective plane over* $\mathbb{Z}/n\mathbb{Z}$. The orbit of $(x,y,z)$ is denoted by $(x{:}y{:}z)$.

Assume now for simplicity that $\gcd(n,6)=1$. An *elliptic curve* over $\mathbb{Z}/n\mathbb{Z}$ is a plane cubic curve $E$ over $\mathbb{Z}/n\mathbb{Z}$ defined by a polynomial of the form $f=Y^2Z-X^3-aXZ^2-bZ^3$, where $a,b\in\mathbb{Z}/n\mathbb{Z}$ are such that $4a^3+27b^2\in(\mathbb{Z}/n\mathbb{Z})^*$. A *point* on $E$ over $\mathbb{Z}/n\mathbb{Z}$ is a point $(x{:}y{:}z)$ of the projective plane for which $f(x,y,z)=0$. Let the set of these points be denoted by $E(\mathbb{Z}/n\mathbb{Z})$.

The set of points on an elliptic curve $E$ over $\mathbb{Z}/n\mathbb{Z}$ can in a natural way be made into an additively written *abelian group*. The zero element is $O=(0{:}1{:}0)$, and if $P=(x{:}y{:}z)$ then $-P=(x{:}-y{:}z)$. If $n$ is *prime*, so that $\mathbb{Z}/n\mathbb{Z}$ is a *field*, one can add two points $P$ and $Q$ as follows (see [8]). Consider the line through $P$ and $Q$ (the tangent line to the curve if $P=Q$) and let $R$ be the third intersection point of the line with the curve. Then $P+Q=-R$. For general $n$ the addition operation is somewhat more complicated to describe (cf. [1]). In the applications to prime factor decomposition one can simply attempt to use the formulae that are valid in the case that $n$ is prime. This fails if division is required by a non-zero element of $\mathbb{Z}/n\mathbb{Z}$ that is no unit. But then a gcd-calculation leads to a non-trivial divisor of $n$, which is exactly what one is looking for.

If $n=p$ is a prime number, then by a theorem of Hasse (1934) one can write $\#E(\mathbb{Z}/p\mathbb{Z})=p+1-t$ with $t\in\mathbb{Z}$, $|t|<2\sqrt{p}$. Schoof [7] gave an algorithm to calculate $t$ that is based on the interpretation of $t$ as the "trace of Frobenius". His algorithm runs in time $O((\log p)^9)$, and it is not clear whether it is useful in practice.

For general $n$ no good algorithm is known to calculate the order of the group $E(\mathbb{Z}/n\mathbb{Z})$ of points on an elliptic curve $E$. As for the multiplicative group, one has the formula

$$\#E(\mathbb{Z}/n\mathbb{Z})=n\cdot\prod_{p|n,\ p\ \text{prime}}(\#E(\mathbb{Z}/p\mathbb{Z})/p),$$

but it requires knowledge of the prime factorization of $n$. One can of course attempt to use Schoof's algorithm, but if $n$ is not prime it is not likely to give an answer; and even if it does then this answer has no obvious interpretation - in particular it need not give the order of $E(\mathbb{Z}/n\mathbb{Z})$.

Let again $n=p$ be a prime number. The strength of the methods to be discussed in the next section, when compared to the multiplicative methods of section 2, is due to the fact that there are *many* elliptic curves over $\mathbb{Z}/p\mathbb{Z}$ and that, imprecisely speaking, for a randomly chosen $E$ the order $\#E(\mathbb{Z}/p\mathbb{Z})$ is a random number near $p$. More accurately, one has the following proposition, the proof of which depends on results of Deuring (1941).

**Proposition 2.** *There are positive effectively computable constants $c_1$ and $c_2$ such that for any prime number $p>3$ and any set $S$ of integers $m$ for which $|m-(p+1)|<\sqrt{p}$ one has*

$$\frac{\#S-2}{2[\sqrt{p}]+1}\cdot c_1(\log p)^{-1}\le\frac{N}{p^2}\le\frac{\#S}{2[\sqrt{p}]+1}\cdot c_2(\log p)\cdot(\log\log p)^2,$$

*where $N$ denotes the number of pairs $(a,b)\in(\mathbb{Z}/p\mathbb{Z})^2$ for which $f=Y^2Z-X^3-aXZ^2-bZ^3$ defines an elliptic curve $E$ over $\mathbb{Z}/p\mathbb{Z}$ with $\#E(\mathbb{Z}/p\mathbb{Z})\in S$.*

Note that $N/p^2$ is the probability that a random pair $(a,b)$ has the stated property. The proposition asserts that, apart from a logarithmic factor, this probability is essentially equal to the probability that a random number near $p$ is in $S$.

## 4. Elliptic curve methods.

*Primality testing.* The following theorem is analogous to Theorem 1.

**Theorem 3.** *Let $n$ be an integer, $n>1$, with $\gcd(n,6)=1$. Let $E$ be an elliptic curve over $\mathbb{Z}/n\mathbb{Z}$, and $m$, $s$ positive integers with $s$ dividing $m$. Suppose that there is a point $P\in E(\mathbb{Z}/n\mathbb{Z})$ satisfying*

$$m\cdot P=O,$$

$$\gcd(z_q,n)=1 \text{ for each prime divisor } q \text{ of } s, \text{ where } (m/q)\cdot P=(x_q:y_q:z_q).$$

*Then $\#E(\mathbb{Z}/p\mathbb{Z})\equiv 0 \bmod s$ for every prime divisor $p$ of $n$, and if $s>(n^{1/4}+1)^2$ then $n$ is prime.*

The proof is analogous to the proof of the Theorem 1.

To use Theorem 3 to prove the primality of a number $n$ that one suspects to be prime one can proceed as follows. Choose a random elliptic curve $E$ over $\mathbb{Z}/n\mathbb{Z}$, and determine a number $m$ such that *if* $n$ is prime *then* $\#E(\mathbb{Z}/n\mathbb{Z})=m$; this can be done with Schoof's algorithm (if Schoof's algorithm fails then $n$ is not prime). Next let $s$ be the largest divisor of $m$ that one is able to factor completely. If $s>(n^{1/4}+1)^2$ one now looks for a point $P\in E(\mathbb{Z}/n\mathbb{Z})$ as in Theorem 3, and applies the theorem to prove that $n$ is prime. If $s$ is smaller one can either use refinements of Theorem 3 that are analogous to existing refinements of Theorem 1, or start all over again with a different elliptic curve. One can keep changing the elliptic curve until the number $s$ appearing in the algorithm is sufficiently large. This alternative has no analogue for the multiplicative method from section 2.

In the primality test of Goldwasser and Kilian [3] one changes curves until the conjectural order $m$ of $E(\mathbb{Z}/n\mathbb{Z})$ is of the form $m=2q$, where $q$ is a number that is very likely to be prime in the sense that it passes certain pseudoprime tests. With the help of Theorem 3, with $s=m=2q$, one can then prove the primality of $n$ provided that one knows that $q$ is prime. To prove the primality of $q$ one proceeds recursively, replacing $n$ by $q$.

See [1, 2] for a primality test depending on elliptic curves with "complex multiplication".

*Factorization.* The analogue of the Pollard $p$-1-method is as follows. Let $n$ be the composite integer that one wishes to factor, and assume that $n>1$, $\gcd(n,6)=1$. Pick a random pair $(E,P)$, where $E$ is an elliptic curve over $\mathbb{Z}/n\mathbb{Z}$ and $P\in E(\mathbb{Z}/n\mathbb{Z})$. This can be done by choosing $a,x,y\in\mathbb{Z}/n\mathbb{Z}$ at random, putting $P=(x:y:1)$, and letting $E$ be defined by $f=Y^2Z-X^3-aXZ^2-bZ^3$, where $b$ is chosen such that $P\in E(\mathbb{Z}/n\mathbb{Z})$; so $b=y^2-x^3-ax$. Next calculate, by repeated duplications and additions, the points $P_k=k!\cdot P\in E(\mathbb{Z}/n\mathbb{Z})$, for $k=1,2,\ldots$ . In addition, if $P_k=(x_k:y_k:z_k)$, calculate $\gcd(z_k,n)$ for each $k$, and stop if this gcd is a non-trivial divisor of $n$. If $k$ reaches a certain upper bound that one fixes beforehand, and no non-trivial divisor of $n$ has been found, then one changes the pair $(E,P)$ and starts all over again.

As for the Pollard $p$-1-method, one can show that a given pair $(E,P)$ is likely to be successful in this algorithm is $n$ has a prime divisor $p$ for which $\#E(\mathbb{Z}/p\mathbb{Z})$ is built up from small primes only. The probability for this to happen increases with the number of pairs $(E,P)$ that one tries. This has no analogue for the Pollard $p$-1-method.

*Efficiency.* With the help of Proposition 2 one can estimate the running time of the above algorithms, provided that one knows how certain sets of integers are distributed in short intervals. The Goldwasser-Kilian primality test can be proved to run in *expected polynomial time* (i.e., bounded by a power of $\log n$), if one assumes the truth of a standard conjecture about the number of primes in an interval of the form $(x,x+\sqrt{x})$. The factorization method can be proved to be successful within expected time $\exp((1+o(1))\sqrt{2(\log p)(\log\log p)})\cdot(\log n)^2$, where $p$ is the least prime factor of $n$ and the $o(1)$ is for $p\to\infty$, provided that one makes a reasonable assumption about the number of integers in the interval $(x,x+\sqrt{x})$ that are built up from prime factors $\leq y$.

The practical merits of the Goldwasser-Kilian primality test are not yet clear, since it depends on Schoof's algorithm. The factorization method depending on elliptic curves has proved to be of great practical value, see [5].

**References.**


1. W. Bosma, *Primality testing using elliptic curves*, report 85-12, Mathematisch Instituut, Universiteit van Amsterdam 1985.
2. D.V. Chudnovsky, G.V. Chudnovsky, *Sequences of numbers generated by addition in formal groups and new primality and factorization tests*, research report RC 11262 (#50739), IBM Thomas J. Watson Research Center, Yorktown Heights 1985.
3. S. Goldwasser, J. Kilian, *A provably correct and probably fast primality test*, preprint, M.I.T. 1985; Proc. 18th Annual ACM Symp. on the Theory of Computing (STOC), Berkeley, May 28-30, 1986.
4. H.W. Lenstra, Jr., R. Tijdeman (eds), *Computational methods in number theory*, Math. Centre Tracts **154/155**, Mathematisch Centrum, Amsterdam 1982.
5. P.L. Montgomery, *Speeding the Pollard methods of factorization*, preprint, 1986.
6. H. Riesel, *Prime numbers and computer methods for factorization*, Progress in Math. **57**, Birkhäuser, Boston 1985.
7. R.J. Schoof, *Elliptic curves over finite fields and the computation of square roots mod p*, Math. Comp. **44** (1985), 483-494.
8. J.H. Silverman, *The arithmetic of elliptic curves*, Graduate Texts in Mathematics **106**, Springer-Verlag, New York 1986.


**Schoen, Richard M.**, University of California at San Diego, La Jolla, U.S.A. *New developments in the theory of geometric partial differential equations. (No abstract)*

**Schönhage, Arnold**, Universität Tübingen, Fed. Rep. of Germany. *Equation solving in terms of computational complexity.*

<u>1. Introduction.</u> Computational complexity is a new principle in Mathematics, also intimately related to Logic and Foundations and to Numerical Methods, of growing interest in other fields like Geometry, Number Theory, Algebra. Here we consider algebraic equations under deterministic, sequential models of computation - no probabilistic methods; for a recent survey on parallel complexity see [1]. The prime fields GF(p) and $\mathbb{Q}$ together with their finite extensions admit exact arithmetic; elements are encoded as tuples of binary integers. Computations with real or complex numbers use dynamic precision in the sense of recursive dependance. Inputs $\alpha \in \mathbb{R}$ are potentially given at any precision: When called with a specified parameter value N, some <u>oracle</u> will deliver some (binary) rational a with $|a-\alpha| < 2^{-N}$, without extra cost. In this setting, equality is undecidable.
Time complexity is discussed with regard to multitape Turing machines, also for pointer machines, cf.[8]; the latter are real-time equivalent to simple random access machines.- Algebraic complexity theory (we recommend the survey [9]) uses the concept of straight-line programs (computation trees with branching), where "time" is the (maximal) number of arithmetic operations. "Nonscalar complexity" refers to counting * and / only.

The complexity of algebraic equation solving in full generality seems to be discouragingly high, but considerable progress has been made in factoring polynomials and on related topics (see refs. in [4]). Factorization of univariate polynomials over $\mathbb{Q}$ is possible in polynomial time [6], also testing for solvability by radicals.- Here we focus on the complexity of very basic equation solving: first ax=b, then P(z)=0 over $\mathbb{C}$, finally solving systems of linear equations and characteristic equations.